Machine Learning for the Design of Screening Tests:

General Principles and Applications in Criminology and Digital Medicine

by

Chelsea Krantsevich

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2023 by the
Graduate Supervisory Committee:

P. Richard Hahn, Chair
Visar Berisha
Hedibert Lopes
Rosemary Renaut
Yi Zheng

ARIZONA STATE UNIVERSITY

May 2023

ABSTRACT

This dissertation explores applications of machine learning methods in service of the design of screening tests, which are ubiquitous in applications from social work, to criminology, to healthcare. In the first part, a novel Bayesian decision theory framework is presented for designing tree-based adaptive tests. On an application to youth delinquency in Honduras, the method produces a 15-item instrument that is almost as accurate as a full-length 150+ item test. The framework includes specific considerations for the context in which the test will be administered, and provides uncertainty quantification around the trade-offs of shortening lengthy tests.

In the second part, classification complexity is explored via theoretical and empirical results from statistical learning theory, information theory, and empirical data complexity measures. A simulation study that explicitly controls two key aspects of classification complexity is performed to relate the theoretical and empirical approaches. Throughout, a unified language and notation that formalizes classification complexity is developed; this same notation is used in subsequent chapters to discuss classification complexity in the context of a speech-based screening test.

In the final part, the relative merits of task and feature engineering when designing a speech-based cognitive screening test are explored. Through an extensive classification analysis on a clinical speech dataset from patients with normal cognition and Alzheimer's disease, the speech elicitation task is shown to have a large impact on test accuracy; carefully performed task and feature engineering are required for best results. A new framework for objectively quantifying speech elicitation tasks is introduced, and two methods are proposed for automatically extracting insights into the aspects of the speech elicitation task that are driving classification performance. The dissertation closes with recommendations for how to evaluate the obtained insights and use them to guide future design of speech-based screening tests.

ACKNOWLEDGMENTS

First of all, I would like to express my appreciation and gratitude to my family members: Mom, Dad, Camille, Cozy, Rachel, Christopher, Buffy and Arya, and my in-laws Artem, Tatiana and Mikhail. Your cheerleading and care during this program have been a huge source of comfort and support.

To my friends in the SoMSS graduate program, thank you for the long days of studying together, the game nights, the sushi meals, and the research brainstorming. Special thanks to Drew Herren, Demetri Papakostas, and Camille Moyer for your support in the spring of 2023. I will be eternally grateful for the SoMSS grad students' impact on my decision to stay at ASU in 2018.

I want to express my heartfelt gratitude to: Dr. Dieter Armbruster for ongoing support and mentorship; Dr. Rosie Renaut for organizing the professional development program; Dr. Douglas Cochran for facilitating RTG and NSF INTERN (NSF-DMS Award No. 150264); Dr. Visar Berisha, Dr. Yi Zheng, Dr. Rosie Renaut, and Dr. Hedibert Lopes, for helpful guidance and discussion as members of my dissertation committee; and Dr. P. Richard Hahn, for training and guidance throughout my academic journey these past five years.

Thanks also to my past and current colleagues at Aural Analytics for your support and patience while I was finishing this dissertation, and for creating a special environment of friendship at the company; special thanks to Dr. Gabriela Stegmann, Kan Kawabata, and Dr. Shira Hahn in this regard.

Finally, I want to say thank you from the bottom of my heart to my smart, funny, caring, and inspiring husband Nikolay. Every day I learn a new lesson from you about what it means to love and be loved. You've shown me how to face challenges with courage and how to grow as a person through any experience that life brings your way. It has been such a privilege being on this journey together.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

This dissertation centers on applying techniques in machine learning, decision theory, Bayesian inference, and neuropsychological battery design to the task of creating a screening test. Screening tests are in widespread use in social and behavioral sciences, and are ubiquitous in medical settings. They are used for applications ranging from identifying youth at risk of joining a gang (Hennigan *et al.* (2014)), to measuring quality of life (Michel *et al.* (2018)), assessing risk of suicide (Delgado-Gomez *et al.* (2016)), and checking for early warning signs of cognitive decline (Nasreddine *et al.* (2005)), to name but a few.

A screening test can be naturally cast as a classification problem, in which the goal is to separate between two groups: people who meet the condition being screened (e.g., at risk of joining a gang or committing suicide) and people who do not. While relying on intuition from domain experts is a standard approach to screening test design, and is often necessary for initial versions, a data-driven approach to creating the test is a viable alternative, in particular for test refinement. By using a data-driven approach, decades of machine learning research can immediately be applied to obtain classification models that function as screening tests, given appropriate training data. These models make predictions on whether new participants belong to one group or the other by learning patterns from data on past screened participants.

One of the goals of the present work is to emphasize the importance of carefully designing the data collection protocol that will be used to acquire data for creating the screening test; this is the same protocol under which new patients will provide data when taking the screening test. All too often, machine-learning based screening

tests are instead created based on whatever data is readily available for the task at hand. We posit that targeted design of the data collection context can produce substantial gains in terms of the ability of the screening test to discriminate between groups, in particular for screening tests based on high dimensional data present in digital health applications. We furthermore produce recommendations for how such targeted design can be performed, with an example using speech data for a cognitive screening test.

While performance (e.g., sensitivity and specificity in separating between groups) is a critical aspect of test design, real world constraints on the context of screening test administration bring additional considerations. When the results of the screening test are used to determine the allocation of scarce or expensive resources, such as community support counselors or extensive neuropsychological testing, for example, the problem becomes much more challenging. Tests must be designed to account for these limitations, along with achieving good performance. Other considerations to be taken into account may include, but are not limited to reliability, brevity, and correlation with current gold standards for assessing the same condition.

In light of these considerations, another goal of the present work is to disentangle these distinct, and often opposing, requirements during screening test design. We provide methods that can be used to explicitly account for these opposing aims, and quantitatively measure trade-offs between favoring one over the other.

To summarize, this dissertation encapsulates three aims.

The first aim is to highlight the importance of both identifying and prioritizing the opposing requirements that arise while designing a particular screening test for a particular application, and to provide a method for explicitly balancing the resulting trade-offs. Chapter 2 encapsulates this aim.

The second aim is to raise awareness of the importance of the data collection

context when designing screening tests, particularly digital health screening tests, and to provide recommendations for how to design the data collection protocol using machine-learning based data-driven approaches. Chapter 4 covers this aim.

The third aim is to provide sufficient theoretical background in which both of the previous two aims can be situated. For the first aim, the theoretical background is provided in the first part of Chapter 2. For the second aim, we split the relevant theoretical background into its own chapter, Chapter 3, due to the extensive nature of the material and the fact that a deep exploration of the relevant topics is interesting in its own right.

We now provide a brief look into specifics of each of the remaining chapters of this dissertation.

In Chapter 2, we propose a method for creating a short, tree-based adaptive test using questions from a large pool from an existing, lengthy screening test. On an application to screening for youth delinquency in Honduras, we produce a 15-question tree-based adaptive screening test, and show that it is almost as accurate as a full traditional screening instrument consisting of over 200 questions. We furthermore provide a Bayesian decision theory framework for quantifying the uncertainty around the accuracy lost by shortening the lengthy test. The proposed method can be used to explicitly choose a screening test length satisfying the desired trade-off between accuracy and brevity.

In Chapter 3, we delve into theoretical and empirical measures of classification complexity, and assess the measures on a simulated data example that explicitly tweaks two levers of classification complexity: class overlap and decision boundary complexity. This chapter serves as a theoretical foundation for the classification complexity analysis in the following chapter, which is substantially more involved than the classification problem presented in Chapter 2. This special prelude chapter

is made necessary by the complex nature of a digital screening test, and the lack of theoretical underpinnings for much of speech-based machine learning literature.

Finally, Chapter 4 is dedicated to speech-based screening tests for detecting cognitive impairment. First, we perform a large-scale analysis comparing classification complexity of 5 speech feature sets, calculated on 13 different speech elicitation tasks. The classification problem of interest is separating between cognitively normal participants and participants diagnosed with Alzheimer's disease. Following the large scale analysis, we propose two tree-based methods that can be used to guide the design of future speech elicitation tasks, with the goal of reduced complexity of the resulting speech features.

Chapter 2

ADAPTIVE SCREENING TESTS

2.1 Overview

Screening tests, or *instruments*, serve as an important aid to decision making in many fields, for example education, mental health, or social work[1]. These tests work by sorting the population of test takers into groups which then receive different follow-up services, such as specific curricula, therapies, medical testing, or support resources. Designing an effective screening instrument, in other words, developing the test items and determining the order and mode in which they are administered, presents many challenges. In this chapter, we look specifically at the trade-off between two competing goals when designing a screening instrument: brevity and accuracy.

Lengthy instruments can cause exam fatigue for participants as well as administrators, potentially limiting the number of individuals that can be screened at all. On the other hand, a conveniently brief assessment with poor accuracy is equally unacceptable on a screening test whose results determine the allocation of costly follow-up resources.

The problem considered here is: starting with a screening test comprised of many questions (a large item bank), can a shorter screening test be derived without sacrificing accuracy relative to the full test? The basic statistical challenge in navigating this trade-off is that the accuracy of both the full-length test and the abridged test must be estimated from data. A secondary challenge is that computational search for

[1]The material from this chapter is modified from Krantsevich *et al.* (2023). Each of the co-authors have given their written permission for the use of this material; see appendix A.

subsets of items which preserve accuracy can be computationally infeasible.

In this chapter, we propose solutions to both of these issues. We address statistical uncertainty in the accuracy of the screening test by framing the design of the abridged instrument as a problem in Bayesian decision theory. We address the computational challenge by constructing the adaptive screening test as a decision tree, allowing us to adapt existing algorithms for the purpose of designing short screening tests.

In Section 2.2, we review previous research in two related areas: adaptive testing and posterior model summarization. In Section 2.3, we present our Bayesian decision theory framework, and describe how it can be applied to obtain a screening test for a particular screening context. Section 2.4 includes a review of prior work in youth delinquency assessments, a description of our data and model specifications, and results. We end the chapter with a discussion in Section 2.5, which centers on evaluating the ethical implications of our proposed method for designing tree-based adaptive screening tests.

2.2 Previous Work

The work encapsulated in this chapter brings together (at least) two distinct strands of research. First, we build on recent work using classification trees to design abridged screening tests. To this literature we add a principled approach to constructing the tree and determining its maximum depth, using ideas from Bayesian decision theory to evaluate the trade-offs between screening tests of different lengths. We also introduce a novel method for fitting a classification tree, which makes use of its application in this specific context as a screening test. Second, this Bayesian decision theoretic approach is a natural extension of ideas developed in recent work on utility-based posterior summarization. Here, we apply these ideas in the novel context of adaptive screening tests.

2.2.1 Background on Adaptive Testing

In this section, we first introduce the concept of adaptive screening tests, then review relevant literature from both Item Response Theory and tree-based adaptive tests built using the CART algorithm (Breiman *et al.* (1984)).

As described in Krohn and Thornberry (2008), Hennigan *et al.* (2014) and Hare *et al.* (2018), reducing the instrument length is an important step in screening tool design, since long instruments can cause fatigue and frustration for both participants and test administrators. Additionally, lengthy instruments limit the number of subjects that can be screened due to the time cost of administering a 150+ question interview. Thus, developing a rapid and accurate screening tool is of the utmost importance.

An *adaptive test* is one where the next question a subject is administered depends on his or her answer to the previous set of questions. Adaptive tests are a powerful approach for developing shortened screening tests, because while any given test taker may only see a small number of questions, the wide variety of available questions allows different subjects to be classified more accurately than if every subject was administered the same small number of questions.

Traditionally, adaptive tests have been constructed using item response theory (IRT), and we begin the next part of this section with a review of fundamental concepts from IRT. Item response theory requires estimating the latent constructs of each test taker at the time of testing (Wainer (2000)). Examples of IRT-based adaptive testing in the academic or personnel selection setting include the Graduate Management Admission Test (Rudner (2010)), the Graduate Record Examination (Almond and Mislevy (1998)), and the Armed Services Vocational Aptitude Battery (Sands *et al.* (1997)).

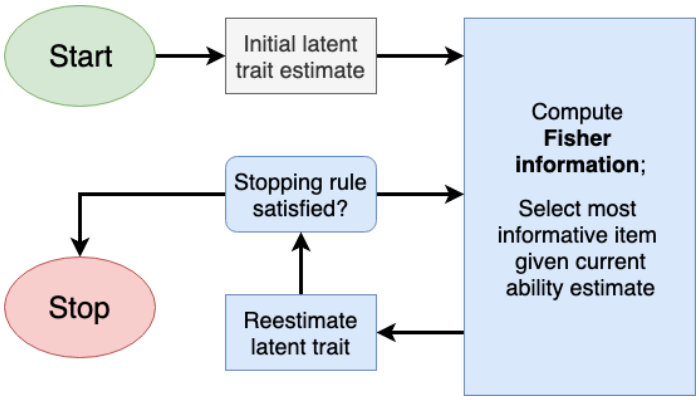

Figure 2.1: Progression of an IRT-based adaptive test. Setting up the test requires choosing an item selection criterion, a method for estimating the latent trait, and a stopping criterion.

Below we give a brief overview of IRT at a high level. For a comprehensive overview on IRT, see van der Linden and Hambleton (1997). We also refer interested readers to Bock (1997) for a full treatise on the history of IRT, Lord *et al.* (1968) and Lord (1980) for foundational texts on IRT, and Hambleton *et al.* (1991), Embretson and Reise (2000), and de Ayala (2009) for more accessible and recent treatments of the subject.

An adaptive test designed based on IRT proceeds by starting with an initial risk level, administering the most informative item (based on the participant's risk estimate and each item's item response function), updating the risk estimate based on their response, and iterating until a stopping criterion is satisfied. This process is shown in Figure 2.1.

Deploying an IRT-based adaptive test requires several specifications: the item response function (IRF) family (and calibrating individual item parameters), the algorithm for estimating the latent trait, the criterion for selecting successive items, and the stopping criterion (Chang (2004), Chang (2015), van der Linden (2008), Wainer (2000)). In the next few paragraphs we describe each of these steps and provide relevant references.

The item response function is the cornerstone of IRT. It is a mathematical relationship describing, for each item in the item bank, how the examinee's response to that item depends on the latent trait being measured. Each item has item-specific parameters, which determine how the IRF changes over the range of examinee ability.

In order for an IRT-based adaptive test to be effective, item parameters of the IRF must be carefully calibrated. Bock and Aitkin (1981), Bock *et al.* (1988), Muraki and Carlson (1995), Gibbons and Hedeker (1992), and Gibbons *et al.* (2007) worked on Expectation-Maximization (EM) algorithms for estimating parameters. Cai (010a) and Cai (010b) vastly improve computational speed for parameter calibration in multidimensional IRT with a Metropolis-Hastings Robbins-Monro algorithm.

Beyond choosing an IRF family and calibrating item parameters, IRT setup also implies specifying methods for estimating the latent trait at each step, selecting successive questions, and terminating the exam. Magis and Raîche (2012), Magis and Barrada (2017) and Chalmers (2016) provide examples of each of these choices in their R packages `catR` and `mirtCAT`.

There are two major downsides to IRT-based adaptive tests in our application. One, the real-time estimation of the latent trait parameter requires intense computational resources, necessitating test administration via a laptop or computer and making the screening process more challenging. Two, as noted by Gibbons *et al.* (2016) and Zheng *et al.* (2020), many constructs in practical screening or diagnostic contexts are multidimensional, and IRT tests traditionally only measured a single latent construct of interest determining the condition being screened. While multidimensional extensions of IRT-based adaptive tests to a handful of dimensions exist (Haley *et al.* (2006), Frey and Seitz (2009), Gibbons *et al.* (2016), Paap *et al.* (2017), Wang and Chang (2011), Wang *et al.* (2012), Yao *et al.* (2014)Dirven *et al.* (2017)), this is likely unsuitable for capturing the relationship between the 38 different scales

that are represented in the risk assessment for youth delinquency, which we use for a data application (see Section 2.4.2).

Here, we focus instead on the recent use of classification trees for the purpose of constructing adaptive tests. The following part of this section reviews relevant work in tree-based adaptive screening tests.

Tree-based adaptive tests, constructed entirely beforehand using classification trees, are a recently explored alternative to IRT-based tests, and have already been used for measuring a variety of medical and behavioral screening test settings. Zheng *et al.* (2020) is the first tree-based adaptive test to our knowledge to be used for assessing youth risk of delinquency, which is our demonstrated application as well. The authors utilized item-response data from crime prevention programs in Honduras, comparing the performance of several tree-based adaptive tests fit using the CART algorithm (Breiman *et al.* (1984)), including one fit to synthetic data generated using the Synthetic Minority Over-sampling Technique (SMOTE, Chawla *et al.* (2002)).

The tree-based approach to adaptive testing involves collecting responses to a large number of items, as well as a true outcome measurement; a classification tree is then fit to this data to maximize predictive accuracy. Specifically, the Classification And Regression Trees (CART) algorithm, introduced by Breiman *et al.* (1984), is applied to the item response-outcome data. Here we review the basics for reference. A modern survey of CART and other tree-growing methods can be found in Loh (2011).

A classification or regression tree T partitions a covariate space $\mathcal{X}$ into k disjoint hypercubes, A_1, A_2, ..., A_k, by repeatedly splitting $\mathcal{X}$ one variable at a time. Each internal node of a final fitted tree contains a splitting variable and an associated cutpoint, $x_i \leq b$. The number of leaf nodes k corresponds to the size of the partition, and the data stored in each leaf node of the fitted tree represents the output of the tree

function. In a regression tree, values $\mu_1, \ldots, \mu_k \in \mathbb{R}$ are associated to the k leaf nodes, so that $\mathrm{x} \in A_j$ implies $T(\mathrm{x}) = \mu_j$, $1 \leq j \leq k$. In a classification tree with c classes, the j^{th} leaf node contains a probability distribution $\{p_{1j}, p_{2j}, \ldots, p_{cj}\}$ over the classes, and for $\mathrm{x} \in A_j$, $T(\mathrm{x})$ is the class with the highest probability: $T(\mathrm{x}) = \mathrm{argmax}_{i \in \{1,\ldots,c\}} p_{ij}$.

In the classic CART algorithm, the tree is fit to data with the goal of minimizing node impurity according to a given criterion (e.g. mean squared error for regression and Gini index for classification). The algorithm proceeds by first growing a very deep tree, then pruning back to the final tree. In the growing step, all data begins in the root node; the variable and cutpoint defining right and left groups with the smallest combined node impurity is selected as the first split, and the process is recursively repeated until a stopping criterion is reached. The pruning step takes in a complexity parameter α and returns the subtree that minimizes $R(T) + \alpha \cdot |T|$, where $R(T)$ is the risk (total node impurity) of the tree and $|T|$ is the number of leaves. This is the final fitted tree. For more details on the CART growing and pruning algorithms, see Breiman *et al.* (1984).

To use a tree as an adaptive screening test, items are used as splitting variables and item responses as cutpoints. After fitting the classification tree to item response–outcome data, a new subject takes the tree-based adaptive test by first answering the root node item, then moving right or left according to their response and the cutpoint. Subsequent items are administered based the pattern of item responses. The assessment ends when the subject lands in a terminal or "leaf" node, with their predicted outcome class being the one assigned to that leaf node. Alternatively, one can use the probability of having a positive screening outcome, stored in the leaf node as the tree output, and assign a "positive" result $Y = 1$ to subjects having probability above a certain threshold. The cutoff for this threshold is determined separately. See Figure 2.2 for an example of the latter approach, with two items.

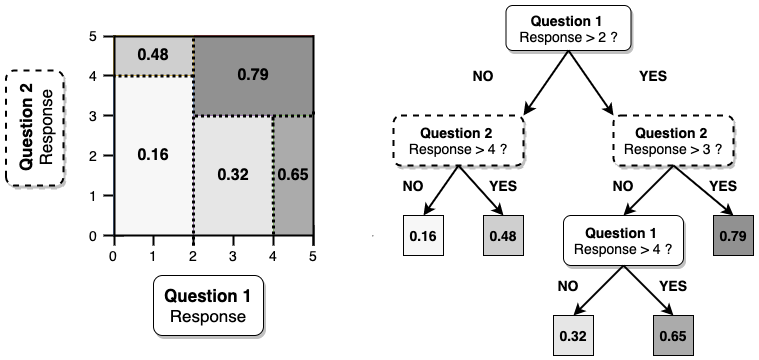

Figure 2.2: A tree-based adaptive test. The splitting variables are $\mathcal{X} = \{\text{Question 1, Question 2}\}$. A subject with responses of 3 and 4, respectively, would land in the right-most leaf node and have "positive" probability 79%.

In the past several years, multiple groups have experimented with tree-based adaptive tests in various settings, including measuring quality of life in Multiple Sclerosis patients (Michel *et al.* (2018)), predicting risk of suicide attempt (Delgado-Gomez *et al.* (2016)) and reproducing a clinician's diagnosis of depression (Gibbons *et al.* (2013)).

Gibbons *et al.* (2013) depart from traditional tree-growing approaches by fitting the tree to a large amount of artificial data, generated as follows (Gibbons and Wang (2019)): first, item response vectors are created via local perturbations[2]. Next, a Random Forest model (Breiman (2001)) is fit to the original data, and used to predict artificial outcome classes for the artificial item response vectors. A single classification tree is then fit to this large artificial dataset and the fitted tree is used as the adaptive test. Gibbons *et al.* (2013) and Gibbons *et al.* (2016) claim that the use of artificial data increases stability of the final classification tree, although they do not discuss details. Our method, while also utilizing artificially generated data, does so for fundamentally different reasons rooted in past work on posterior summarization

[2]By "local perturbation" we mean that item response vectors were selected uniformly at random from vectors that are in a neighborhood of the observed item response vectors in terms of the L_1 distance.

for model selection (see Section 2.2.2). Synthetic data in our case is used to approximate a posterior distribution of a utility function, and we select the optimal decision tree according to this utility. Further details are discussed in Section 2.3.2.

While tree-based adaptive tests have several advantages over IRT, including ease of deployment and fewer modeling assumptions, there is no clear standard to determine how deep to grow the tree, or in other words, when to terminate the test; instead, this choice is made by the default regularization parameters in the tree-growing software. The exam length has important implications; in particular, shortening the exam too much can lead to unacceptable levels for instrument sensitivity and specificity. However, to the best of our knowledge there is no standard stopping criterion for a tree-based adaptive test to ensure a certain sensitivity and specificity.

2.2.2 Background on Utility-Based Posterior Summarization

A recent line of research has recast the problem of variable or model selection as one of posterior summarization. The idea is to find a single model-summarizing "action" that minimizes a penalized loss function which favors simple models. In the present context, the idea is to find a shortened screening test that is suitably accurate relative to the non-shortened instrument. This line of work began with Hahn and Carvalho (2015) for linear regression models and has subsequently been expanded in various directions (Bashir *et al.*, 2019; Puelz *et al.*, 2017; Woody *et al.*, 2019).

The technique explored in these papers is a two stage process: first, a highly flexible and accurate model is fit; then, draws from the posterior distribution are projected onto simpler structures, producing low-dimensional model summaries. In this way, an analyst may visualize how much accuracy (however that is defined) is lost relative to an "ideal" non-simplified model.

In the screening test setting studied here, the "ideal" non-simplified model is a

non-shortened screening instrument that incorporates responses to every item in the item bank in order to predict the probability of a positive test result. In this step, we may use any state-of-the-art predictive algorithm to obtain an probability estimate of having a positive outcome. Then, we consider the trade-off in model accuracy that is made by administering a greatly-shortened adaptive test, in which each subject sees only a small number out of the many items available. We use a Bayesian decision theory framework to formalize these trade-offs.

2.3 A Decision Theory Framework for Adaptive Screening

In the following sections, we first review general elements of Bayesian decision theory, including definitions that will be used throughout the chapter. Following this review, we explain its particular application in producing a shortened screening test, and measuring the trade-offs of test shortening for a particular population.

Throughout the chapter, we use calligraphy $\mathcal{X}$ and $\mathcal{Y}$ to denote the support of item response vectors and outcome classes, upper-case X and Y to denote a random vector/variable representing an item response vector or outcome class, and lower-case x and y to denote a single instantiation of the random vector/variable. As is standard in Bayesian statistics, we treat model parameters as random variables, rather than fixed parameters of the data generating process of X and Y; the random vector of all unknown model parameters is represented by θ, with Θ being its support and $\theta^{(j)}$ a single instantiation. When referring to observed data, we use subscripts $\mathrm{x}_{1:n}$ and $y_{1:n}$; synthetic data are denoted by $\tilde{\mathrm{x}}$ and $\tilde{y}$.

2.3.1 Review of Bayesian Decision Theory

In this section we provide an introduction to Bayesian decision theory using the terminology of Parmigiani and Inoue (2010), according to which an analyst chooses

from among a *set of actions*, Γ. Each action $\gamma : \mathcal{X} \to \{0, 1\}$ has consequences that depend on an unknown *state of the world*, $y \in \mathcal{Y}$. In order to evaluate the merits of possible actions, a quantitative value is assigned to each possible (action, state) pair, either a *utility* value $U(\gamma(\mathrm{x}), y)$ or a *loss* value $L(\gamma(\mathrm{x}), y)$. With the *utility function* framework, which we employ, the analyst chooses the action that maximizes (in some sense) a utility.

We adopt the *expected utility principle*, which implies the chosen action maximizes expected utility over a target population with density $f(\mathrm{x}, y)$. This expected utility is

$$\mathbb{E}U(\gamma) := \mathbb{E}[U(\gamma(\mathrm{X}), Y))] = \int_{\mathcal{X}} \int_{\mathcal{Y}} U(\gamma(\mathrm{x}), y) f(\mathrm{x}, y)\, dy\, d\mathrm{x}, \tag{2.1}$$

and the optimal action is

$$\gamma^* = \mathrm{argmax}_{\gamma \in \Gamma} \mathbb{E}U(\gamma).$$

To summarize, our decision theory formulation consists of:

(1) A *utility function* U.

(2) A *target population* defined by a distribution function $F_{\mathrm{X},Y}$.

(3) A *set of actions*, denoted Γ.

These three elements come together in defining our expected (integrated) utility $\mathbb{E}U(\gamma) = \mathbb{E}[U(\gamma(\mathrm{X}), Y))]$, where $\gamma \in \Gamma$ and $\mathbb{E}(\cdot)$ denotes expectation with respect to $F_{\mathrm{X},Y}$.

In our application to screening tests for detecting a particular condition, an action γ is a tree-based adaptive screening test, which takes the subject's item responses $\mathrm{x} \in \mathcal{X}$, and assigns an outcome of either "positive" ($\gamma(\mathrm{x}) = 1$) or "negative" ($\gamma(\mathrm{x}) = 0$). Subjects with a positive outcome are then subsequently presented with follow-up care, such as further intensive testing or enrollment into a community counseling program. We apply the preceding framework to the screening test problem as follows:

(1) Our *utility function*, U, is a weighted average of sensitivity and specificity.

(2) Our *target population* is the group of subjects to be screened for risk of a particular condition, such as joining a gang, attempting to commit suicide, or having cognitive impairment. We let $f(\mathrm{x}, y)$ denote the joint density function of item responses and true outcome values for subjects in the target population.

(3) Our *set of actions*, Γ, is a collection of candidate screening tests of varying lengths. (This action space will be populated using a tree growing algorithm, detailed later.)

Section 2.3.2 describes these three steps in greater detail.

In practice, the density function $f(\mathrm{x}, y)$ is unknown and must be estimated from available data. To do so, we will parametrize f by a vector θ, which we will estimate via Bayesian inference. We choose a prior $\pi(\theta)$ and, after conditioning on data $(\mathrm{x}_{1:n}, y_{1:n})$, arrive at a posterior $\pi(\theta \mid \mathrm{x}_{1:n}, y_{1:n})$. Rather than integrating over the estimation uncertainty in θ as would be done in traditional Bayesian decision theory, we will instead consider posterior uncertainty of the utility $\mathbb{E}U(\gamma, \theta)$, defined as

$$\mathbb{E}U(\gamma, \theta) := \int_{\mathcal{X}} \int_{\mathcal{Y}} U(\gamma(\mathrm{x}), y) f(\mathrm{x}, y \mid \theta) \, dy \, d\mathrm{x}. \tag{2.2}$$

As a function of θ, $\mathbb{E}U(\gamma, \theta)$ is itself a random variable, which we denote $\mathbb{E}U_\theta(\gamma)$ for notational convenience. In this work, we will be interested in the posterior distribution of $\mathbb{E}U_\theta(\gamma)$ induced by the posterior distribution over θ.

By integrating over the posterior, we can also obtain an overall expected utility for a particular screening test γ:

$$\begin{aligned}\mathbb{E}U(\gamma) &= \int_\Theta \left[\int_{\tilde{\mathcal{X}}} \int_{\tilde{\mathcal{Y}}} U(\gamma(\tilde{\mathrm{x}}), \tilde{y}) f(\tilde{\mathrm{x}}, \tilde{y} \mid \theta) \, d\tilde{y} \, d\tilde{\mathrm{x}} \right] \pi(\theta \mid \mathrm{x}_{1:n}, y_{1:n}) \, d\theta, \\ &= \int_\Theta U(\gamma, \theta) \pi(\theta \mid \mathrm{x}_{1:n}, y_{1:n}) \, d\theta.\end{aligned} \tag{2.3}$$

It is this quantity that is optimized in traditional Bayesian decision theory; we will instead mainly be interested in posterior exploration of the random variable $\mathbb{E}U(\gamma, \theta)$.

2.3.2 Applying the Framework to Create an Adaptive Screening Test

Here we describe how the three steps of the Bayesian decision theory framework are applied to adaptive screening tests. Recall that our set of actions Γ is comprised of adaptive screening tests γ for assessing the probability of having the condition being screened. Each test γ consists of two parts:

(1) A binary tree $T : \mathcal{X} \to (0, 1)$ representing the screening test (see Figure 2.2, right). T predicts the probability $T(\mathrm{x})$ of a positive test result, given item responses $\mathrm{x} \in \mathcal{X}$.

(2) A threshold function $\mathrm{Thr}_C : (0, 1) \to \{0, 1\}$ that maps the probability $T(\mathrm{x})$ to an outcome class prediction via a cutoff $C \in [0, 1]$:

$$\mathrm{Thr}_C(T(\mathrm{x})) \quad = \begin{cases} 0, \text{“negative”} & \text{if } T(\mathrm{x}) < C \\ 1, \text{“positive”} & \text{if } T(\mathrm{x}) \geq C \end{cases}$$

Put together, the adpative test is $\gamma(\cdot) = \mathrm{Thr}_C(T(\cdot))$, where $\gamma(\mathrm{x}) \in \{0, 1\}$ for any given set of item responses x. The framework described in the next three sections provides a way to compare different screening tests of this form.

Step 1 of the framework is specifying a utility function U. The adaptive test γ should maximize $\mathbb{E}U_\theta(\gamma)$, the expectation of U with respect to the density $f(\mathrm{x}, y)$ (which is parameterized by θ) over our target population.

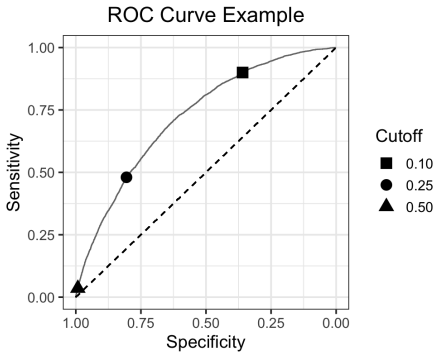

Figure 2.3: Each cutoff C results in a particular (Specificity, Sensitivity) point on the ROC curve. Lowering the cutoff increases sensitivity and decreases specificity. Choosing a cutoff means choosing an acceptable (Specificity, Sensitivity) combination, or point on the ROC curve.

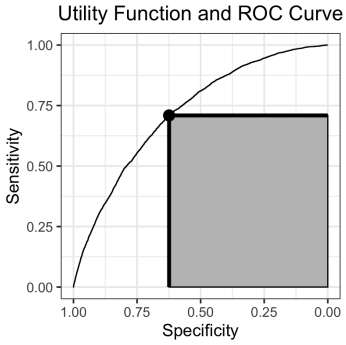

Figure 2.4: The cutoff that maximizes the utility function (2.4) for $w = 0.5$ is the point on the ROC curve that maximizes the perimeter of the shaded rectangle. The height and width of the rectangle are sensitivity and specificity, respectively, for that cutoff.

In our application, we want the utility function to carry practical significance for the adaptive test. Two important quantities are sensitivity and specificity, which measure the true positive rate and true negative rate, respectively:

$$\text{Sensitivity} = \Pr(\gamma(\mathrm{X}) = 1 \mid Y = 1), \;\; \text{Specificity} = \Pr(\gamma(\mathrm{X}) = 0 \mid Y = 0).$$

As a reminder, γ is an adaptive test mapping item responses X to an outcome class Y, which represents the screening test result. We can generically label the test result as either "positive" or "negative" ("positive" means $Y = 1$, "negative" means $Y = 0$).

Ideally, sensitivity and specificity would both be 1. In practice, there is a trade-off between these two quantities, based on the cutoff C. A high cutoff means that many predicted probabilities will be below the threshold and consequently labeled "negative", leading to high specificity and low sensitivity. A low cutoff leads to more "positive" class predictions, increasing sensitivity and reducing specificity. This trade-off can be visualized in a Receiver Operating Characteristic (ROC) curve, shown in Figure 2.3.

To incorporate the importance of both sensitivity and specificity, our expected

utility $\mathbb{E}U_\theta(\gamma)$ is equal to a weighted average of the two, for a user selected weight $w \in (0,1)$:

$$\mathbb{E}U_\theta(\gamma) = w \cdot \text{Sensitivity}(\gamma) + (1-w) \cdot \text{Specificity}(\gamma). \tag{2.4}$$

For $w = 0.5$, this utility function can be directly visualized within a ROC curve as shown in Figure 2.4.

Since the final adaptive tree and the associated sensitivity and specificity highly depend on w, we recommend carefully selecting the value of the weight in conjunction with stakeholders who understand the implications of favoring sensitivity or specificity for the population where the test will ultimately be deployed. Multiple values of w can and should be examined via the methods presented in Section 2.4.4.

For completeness, we introduce the point-wise (individual) specification of the utility function U which induces this expected (population level) utility. Formally, we define our utility function U as

$$U(\gamma(\mathrm{x}), y) \quad = \begin{cases} U_0 & \text{if } y = 0, \gamma(\mathrm{x}) = 0 \\ U_1 & \text{if } y = 1, \gamma(\mathrm{x}) = 1 \\ 0 & \text{otherwise} \end{cases}, \tag{2.5}$$

where

$$U_0 = \frac{1-w}{\Pr(Y=0)}, \qquad U_1 = \frac{w}{\Pr(Y=1)}.$$

Our expected utility over the target population is

$$\begin{aligned} \mathbb{E}U(\gamma) &= \mathbb{E}[U(\gamma(\mathrm{X}), Y)] \\ &= \mathbb{E}\left[\frac{w}{\Pr(Y=1)} \cdot \mathbb{1}(\gamma(\mathrm{X})=1, Y=1) + \frac{1-w}{\Pr(Y=0)} \cdot \mathbb{1}(\gamma(\mathrm{X})=0, Y=0)\right] \\ &= w \cdot \frac{\Pr(\gamma(\mathrm{X})=1, Y=1)}{\Pr(Y=1)} + (1-w) \cdot \frac{\Pr(\gamma(\mathrm{X})=0, Y=0)}{\Pr(Y=0)} \\ &= w \cdot \text{Sensitivity}(\gamma) + (1-w) \cdot \text{Specificity}(\gamma), \end{aligned}$$

as desired.

With the utility function defined in 2.4 and a value of w specified, the optimal action γ^* is the tree-based adaptive test (i.e., "positive" probability prediction and associated cutoff) that maximizes this weighted average. Formally,

$$\gamma* = \text{argmax } \mathbb{E}U(\gamma) = \text{argmax } \left\{w \cdot \text{Sensitivity}(\gamma) + (1-w) \cdot \text{Specificity}(\gamma)\right\}.$$

Since the expected utility of a given action γ is a simple expression at the population level, we can evaluate $\mathbb{E}U_\theta(\gamma)$ over a sample from the target population by directly computing sensitivity and specificity of γ for a particular set of item responses and true outcome classes. To be more specific, after drawing a sample $\{\tilde{\mathrm{x}}_{ij}, \tilde{y}_{ij} \mid \theta^{(j)}\}_{i=1}^N$ from the target population (where $\tilde{\mathrm{x}}_{ij}$ is an item response vector, $\tilde{y}_{ij}$ is the outcome class, and $\theta^{(j)}$ is a single fixed draw from the posterior $\pi(\theta \mid \mathrm{x}_{1:n}, y_{1:n})$—see Section 2.3.2), we compute a draw $\mathbb{E}U_{\theta^{(j)}}(\gamma)$ as

$$\mathbb{E}U_{\theta^{(j)}}(\gamma) = w \cdot \frac{\sum_{i=1}^N \mathbb{1}(\gamma(\tilde{\mathrm{x}}_{ij}) = 1, \tilde{y}_{ij} = 1)}{\sum_{i=1}^N \mathbb{1}(\tilde{y}_{ij} = 1)} + (1-w) \cdot \frac{\sum_{i=1}^N \mathbb{1}(\gamma(\tilde{\mathrm{x}}_{ij}) = 0, \tilde{y}_{ij} = 0)}{\sum_{i=1}^N \mathbb{1}(\tilde{y}_{ij} = 0)}. \tag{2.6}$$

In the next section, we describe how to sample from the target population in order to obtain draws of $\mathbb{E}U_\theta(\gamma)$ for any given action γ.

Step 2 of the Bayesian decision theory framework is specifying a target population over which we seek to maximize $\mathbb{E}U_\theta(\gamma)$. In our application, that means defining a specific subgroup from the population for whom the screening test will be targeted. For example, if the screening test is being designed to screen people ages 50-59 in Canada for symptoms of Major Depressive Disorder, our target population would be Canadian residents aged 50-59.

After specifying the target population, the optimal action γ^* (the "optimal" adaptive test) would maximize the weighted average of sensitivity and specificity for this group specifically. The target population can mean the entire population of a par-

ticular country, or can be more specific to people of a certain age, zip code, and so forth.

After specifying the target population, we draw synthetic samples from the joint density $f(\mathrm{x}, y)$ of the item responses X and outcome class Y in the target population. We use the composite model specification

$$f(\mathrm{x}, y) = f(\mathrm{x}) f(y \mid \mathrm{x})$$

and specify the random variable θ parameterizing $f(\mathrm{x}, y)$ as

$$\theta = (\theta_{\mathrm{X}}, \theta_Y),$$

with θ_{X} parameterizing $f(\mathrm{x})$ and θ_Y parameterizing $f(y \mid \mathrm{x})$. This specification allows for additional flexibility in modeling the relationship between the item responses X and the probability of a "positive" outcome $Y = 1$. Practically, we draw synthetic data from $f(\mathrm{x}, y)$ as follows:

(1) Fit each component of the composite form using a Bayesian model: one for $f(\mathrm{x})$ with unknown parameters θ_{X}, and one model for $f(y \mid \mathrm{x})$ with unknown parameters θ_Y.

(2) For each posterior draw $\theta^{(j)} = (\theta_{\mathrm{X}}^{(j)}, \theta_Y^{(j)})$, $1 \leq j \leq D$, draw samples

$$\left\{\tilde{\mathrm{x}}_{ij}, \tilde{p}_{ij}, \tilde{y}_{ij} \mid \theta^{(j)}\right\}_{i=1}^{N}$$

from the conditional predictive distribution $f(\tilde{\mathrm{x}}, \tilde{y} \mid \theta^{(j)})$. Here, $\tilde{\mathrm{x}}_{ij}$ are the synthetic item responses, $\tilde{p}_{ij} = \mathbb{E}(\tilde{Y} \mid \tilde{\mathrm{x}}_{ij}, \theta_Y^{(j)})$ is the synthetic probability of belonging to class $Y = 1$, and $\tilde{y}_{ij}$ is the synthetic class status.

Taken together, we will have a sample of size N for each posterior draw $\theta^{(j)}$, $1 \leq j \leq D$, which is $N \cdot D = M$ synthetic data in total; this data is denoted $\{\tilde{\mathrm{x}}_k, \tilde{p}_k, \tilde{y}_k\}$, $1 \leq k \leq M$.

Since we fit two models corresponding to different components of the same composite model specification, we use a single dataset for fitting the models for $f(\mathrm{x})$ and $f(y \mid \mathrm{x})$. Modeling details and specifics on sampling are provided in Section 2.4.3. As a reminder, the "synthetic data" in this setting is merely a computational approach for evaluating the integrals at the heart of the decision theory framework.

Next, we describe Step 3 of the framework, populating the action space Γ. In our application, Γ consists of tree-based adaptive tests; each $\gamma \in \Gamma$ is of the form form $\gamma(\cdot) = \mathrm{Thr}_{C_T}(T(\cdot))$, where T is a binary regression tree and C_T is the cutoff for classification into the "positive" group. The number of possible binary trees is much too large for brute force enumeration[3]; many possible heuristics are available, and different procedures will lead to higher-utility screening instruments than others. Here we focus on one method for populating our action space, motivated by the Bayesian decision theory context.

We first obtain a regression tree T by applying a particular tree growing algorithm (described shortly) to large Monte Carlo samples from the posterior predictive distribution $f(\tilde{\mathrm{x}}, \tilde{y} \mid \mathrm{x}_{1:n}, y_{1:n})$. We then choose the cutoff C_T that optimizes the expected utility (2.4) relative to T over these samples.

Our proposed heuristic for obtaining the regression tree T relies on a novel stopping criterion we call *maxIPP*, for "**max**imum **I**tems **P**er **P**ath." The maxIPP criterion denotes the maximum number of *unique* items in each root-to-leaf path of the decision tree defining the adaptive test, and consequently, the number of items each individual will be administered during their screening test. The tree is grown using a variation of the CART algorithm; it achieves the maxIPP constraint by restricting the items available for splitting in a given path after m unique items have been used.

Presuming that item responses can be stored for future splits, the maxIPP of a

[3]In our application our item bank consists of 173 items, each with up to 6 possible cutpoints.

tree-based adaptive test is precisely the maximum number of questions any participant will answer. The maxIPP characteristic is similar to maximum depth; to see the distinction, consider the tree in the right of Figure 2.2, which has a maximum depth of 3, but a maxIPP of 2.

For each value of maxIPP $= m$, we use an adapted version of the CART algorithm to obtain an approximately optimal tree. CART consists of a tree growing phase, followed by a tree pruning phase. Our modification uses the usual greedy algorithm (minimizing sum-of-squares) for the growing phase, with a twist: once m unique variables have been used as splitting variables in any particular path, only these same variables are considered as candidates for future splits down this path. This algorithm is implemented as a modification to the `rpart` package, with `maxvpp` (the application-agnostic term meaning "**max**imum **v**ariables **p**er **p**ath") available as an option for `rpart.control`. For the pruning stage, we start at the root tree T_0 in the list of subtrees returned by `rpart`, and for each next tree in the list, compute the reduction in root mean square error (on a holdout set) relative to the previous tree. If this reduction is not above a given threshold[4] for at least 10 consecutive subtrees in the list, we return to the last subtree that met this threshold and call this tree T_m^*.

Categorizing trees by maxIPP is useful in our context of shortening lengthy instruments. While maximum depth also limits the number of items, maxIPP allows for further splitting on items already administered, without counting them against the tree "cost".

For a given m, we calibrate an approximately optimal tree with maxIPP m (denoted T_m^*) to synthetic data drawn from the posterior of θ_X and the posterior predic-

[4]We found that using 10^{-4} for maxIPP < 5 and 10^{-5} for maxIPP between 5 and 15 works well in practice; we did not consider maxIPP values above 15 as they produced very similar results to those near 15.

tive of $\tilde{\mathrm{X}}$. Specifically, our synthetic data used for calibrating T_m^* are $\{\tilde{\mathrm{x}}_k, \bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\mathrm{x}}_k)\}$, $1 \leq k \leq M$, where the second element is the posterior predictive "positive" probability, given $\tilde{\mathrm{x}}_k$:

$$\bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\mathrm{x}}_k) = \int_{\Theta_Y} \mathbb{E}(\tilde{Y} \mid \tilde{\mathrm{x}}_k, \theta_Y) \pi(\theta_Y \mid \mathrm{x}_{1:n}, y_{1:n}) \, d\theta_Y. \tag{2.7}$$

As a reminder, $\pi(\theta \mid \mathrm{x}_{1:n}, y_{1:n})$ is the posterior density of θ, having observed data $(\mathrm{x}_{1:n}, y_{1:n})$. We use the term "calibrate" rather than "fit" for the process of applying the maxIPP algorithm to synthetic data, in order to reserve the term "fit" for the context of fitting the Bayesian models to real data.

Having obtained T_m^*, the cutoff $C_{T_m^*}$ is then optimized relative to the (unconditional) posterior predictive expected utility:

$$\begin{aligned} C_{T_m^*} &= \underset{C \in [0,1]}{\operatorname{argmax}} \, \mathbb{E}U(\mathrm{Thr}_C(T_m^*)) \\ &= \underset{C \in [0,1]}{\operatorname{argmax}} [w \cdot \mathrm{Sensitivity}(\mathrm{Thr}_C(T_m^*)) + (1-w) \cdot \mathrm{Sensitivity}(\mathrm{Thr}_C(T_m^*))], \end{aligned}$$

where the inner expression on the right-hand side (i.e., the weighted average of sensitivity and specificity of $\mathrm{Thr}_C(T_m^*)$) is approximated using

$$w \cdot \frac{\sum_{k=1}^M \mathbb{1}(\mathrm{Thr}_C(T_m^*(\tilde{\mathrm{x}}_k)) = 1, \tilde{y}_k = 1)}{\sum_{k=1}^M \mathbb{1}(\tilde{y}_k = 1)} + (1-w) \cdot \frac{\sum_{k=1}^M \mathbb{1}(\mathrm{Thr}_C(T(\tilde{\mathrm{x}}_k)) = 0, \tilde{y}_k = 0)}{\sum_{k=1}^M \mathbb{1}(\tilde{y}_k = 0)}. \tag{2.8}$$

In summary, T_m^* is our final regression tree with maxIPP m that predicts the probability of being "positive" given a set of item responses, and $\mathrm{Thr}_{C_{T_m^*}}$ maps these probabilities to a predicted class status 0 or 1 (0 = "negative", 1 = "positive"). The threshold is chosen relative to the specific regression tree T_m^*, to optimize the utility function for the target population. We use $\gamma_m^* = \mathrm{Thr}_{C_{T_m^*}}(T_m^*)$ to denote our approximately optimal tree-based adaptive test of length m.

Our action space Γ consists of one adaptive test γ_m^* for each value of m under consideration for a given application. We emphasize this is just one proposed heuristic

for obtaining an adaptive screening test that optimizes Equation (2.4), while administering at most m items. One can obtain adaptive tests with m items using other tree growing methods calibrated with other synthetic or real data. Each of these can be compared using the criteria described in the following paragraphs before choosing a final adaptive test; see appendix B for comparisons of several methods.

Once we have (at least) one action γ_m^* for each test length m, we need to choose the value of m for the final adaptive screening test; although this is not a separate step of the framework, it is the final step of using our method to design an adaptive tree-based screening test. In general, shorter screening tests can only degrade accuracy (utility), so the relevant questions are "by how much?" and "with what statistical uncertainty"?

To address these questions we define a random variable (with respect to the posterior distribution) $\Delta_{\theta,m}$ that characterizes the utility loss due to shortening to m questions. That is, we are interested in the difference in expected utility between that of the shortened exam $\mathbb{E}U_\theta(\gamma_m^*)$ and that of the full, non-shortened, exam $\mathbb{E}U_\theta(\gamma^*)$. Here, the optimal non-shortened action is $\gamma^*(\cdot) = \text{Thr}_{C^*}(\bar{\mathbb{E}}(\tilde{Y} \mid \cdot))$, where $\bar{\mathbb{E}}(\tilde{Y} \mid \cdot)$ is as in (2.7), and Thr_{C^*} is optimized relative to the posterior predictive expected utility; specifically, Thr_{C^*} is optimized using Equation (2.8), but with $\bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\text{x}})$ in place of $T_m^*(\tilde{\text{x}})$. We denote this difference as

$$\Delta_{\theta,m} = \mathbb{E}U_\theta(\gamma_m^*) - \mathbb{E}U_\theta(\gamma^*). \tag{2.9}$$

To obtain Monte Carlo samples of $\Delta_{\theta,m}$, for each posterior draw $\theta^{(j)}$ compute

$$\Delta_{\theta^{(j)},m} = \mathbb{E}U_{\theta^{(j)}}(\gamma_m^*) - \mathbb{E}U_{\theta^{(j)}}(\gamma^*),$$

where $\mathbb{E}U_{\theta^{(j)}}(\gamma)$ is computed using (2.6).

Boxplots may then be plotted for each value of m. See Figure 2.5 for an example with boxplots of $\Delta_{\theta,m}$ varying the number of items m and the weight w that defines

the utility function U. These utility difference plots visually represent our statistical uncertainty of the trade-offs between assessment sensitivity/specificity and length.

2.3.3 Comparison to Existing Methods for Designing Adaptive Tests

In Section 2.3.2 we proposed a novel algorithm for obtaining adaptive tests of different lengths to populate the action space, our second main contribution. Here we compare to existing work on tree-based adaptive tests, as an IRT-based test is not appropriate for our application (see Section 2.2.1). For comparisons between tree-based adaptive tests and IRT, see Gibbons *et al.* (2016) and Zheng *et al.* (2020).

As far as we know, current tree-based adaptive tests are fit using existing algorithms; built-in hyperparameters decide test length, and the optimization criteria (typically Gini index) is not specific to the adaptive testing context. Typically, the decision tree is fit to item response–outcome data. Two exceptions are Gibbons *et al.* (2016), who fit the tree to locally perturbed artificial data for increased model stability, and Zheng *et al.* (2020), who utilized SMOTE to help with class imbalance.

The purpose of synthetic data in our application is to provide an MCMC approximation of the expected utility integral over the target population. We obtain this data by modeling the somewhat high-dimensional joint density of item responses–outcome class via two sophisticated Bayesian models, and use a context-specific utility function (i.e. sensitivity and specificity) for tree optimization, rather than Gini index. Finally, our novel maxIPP stopping criterion is an application-specific design choice, exploiting the fact that items can be reused for splitting.

Note that traditional Bayesian decision theory advocates optimizing γ directly with respect to (2.3). We depart from this tradition in two respects. One, we will restrict the length of the screening test γ and perform several constrained optimizations. Two, we include posterior uncertainty in our optimal utility by examining the

posterior distribution of $\mathbb{E}U_\theta(\gamma)$ via plots, rather that taking the expectation over Θ when computing $\mathbb{E}U(\gamma)$.

2.4 Screening for Youth Delinquency in Honduras

Our central application for the proposed method is the design of a brief screening test to identify youth who are at high risk of falling into delinquent behavior, so that they may receive additional social support intended to mitigate that risk. Specifically, we consider data from Honduras, where decades of political, civil, and economic instability have made gang recruitment and violent crime a major concern (Meyer (2019), UNODC (2018)).

For the past two decades, the "Northern Triangle" nations of El Salvador, Guatemala and Honduras in Central America have faced pressing challenges with political, civil and economic instability. Poverty and unemployment are widespread, and fragile judicial systems weakened by corruption are unable to curb the high levels of crime and violence that threaten many communities (Meyer (2019)). The region is a major trafficking corridor for transporting illegal drugs from South America to the United States (USD (2020)), and Honduras and El Salvador in particular face some of the highest homicide rates in the world, despite declines in recent years (UNODC (2018)). These challenges, along with family systems splintered by crime victimization and migration, result in many youth being susceptible to recruitment by gangs (Meyer (2019)) and other problematic behaviors.

Certain targeted interventions, such as family counseling and community support resources, have demonstrated significant promise in reducing risk factors of criminal behavior for at-risk youth in Honduras (Katz *et al.* (2021)). In order to allocate these limited resources in an effective way, a screening instrument is deployed to identify youth with the highest risk of delinquency. We propose our Bayesian decision theory

framework for designing such a test.

In the following sections, we review past work on screening tests of youth delinquency risk, describe our data on youth delinquency in Honduras, and then demonstrate how the method can be used to create a tree-based adaptive test to screen for risk of violent behavior.

2.4.1 Previous Work on Youth Risk Assessment

Our method adds to a well-established literature on youth risk assessment, specifically their application to crime prevention programs aimed at youth in Central America. In this context, we present a novel contribution by reanalyzing data from Honduras and show that an adaptive screening test consisting of only a handful of items can provide comparably accurate risk assessment to a questionnaire with over a hundred questions.

For a broad overview of the difficulties facing youth in Honduras, please consult Berk-Seligson *et al.* (2014). Here we focus on risk assessment tools used in crime prevention, which has recently gained momentum as an effective alternative to more aggressive suppression strategies.

As a key component of crime prevention, so-called "secondary prevention" programs identify individuals within high risk communities who are at an especially high risk for criminal activity, and provide them with targeted interventions. To effectively execute this secondary prevention strategy, high risk youth must first be identified via a screening tool and are subsequently enrolled in the intervention.

For the model utilized between 2013 and 2015 in Honduras specifically, high risk youth were first identified using a Spanish adaptation of the Youth Services Eligibility Tool (YSET) (Hennigan *et al.* (2014)), and then enrolled in a seven-module family counseling program. This model represented the first time empirical data was uti-

lized for identifying the youth with the highest risk of criminal behaviors. Following initial successes, a more locally focused risk assessment tool was created, incorporating screening tools from around the world. The data for the present work consists of responses to this revised Honduran YSET, and is described more fully in Section 2.4.2.

The risk assessment tools utilized in Honduras are based on a large body of research surrounding the risk factor paradigm. Risk factors are characteristics that increase the likelihood of a given problem behavior, whereas protective factors are ones that reduce this likelihood (Arthur *et al.* (2002)). These factors are typically categorized under domains such as community, family, school, peer, and individual (Howell and Jr. (2005)). Table 2.1 provides a list of risk and protective factors measured by the Instrumento de Medicion de Comportamientos (IMC). Data using this instrument were used for obtaining the results in the following sections.

The risk factor paradigm entered the youth delinquency sphere in 1992 with Hawkins *et al.* (1992), who provided a comprehensive review of the literature on risk and protective factors related to substance abuse in adolescents. In subsequent years, multiple groups developed youth risk assessment tools, including three that were used to expand the item bank for the revised Honduran YSET: the Communities That Care (CTC) Youth Survey (Arthur *et al.* (2002), Arthur *et al.* (2007)), the Eurogang Youth Survey (Weerman *et al.* (2009)), and the Youth Eligibility Services Tool (YSET) (Hennigan *et al.* (2014), Hennigan *et al.* (2015)), a Los Angeles-specific adaptation of the empirically-developed Gang Risk of Entry Factors instrument.

While these instruments have been deployed in countries around the world, they were largely developed for use in the United States and Europe. Research on youth risk assessments for secondary prevention programs within developing countries includes Katz and Fox (2010) and Maguire *et al.* (2011), focusing on the Caribbean

Table 2.1: Risk and protective factors measured by the IMC.

Domain	**Risk Factors**	**Protective Factors**
Community	Transitions and mobility	Rewards for prosocial involvement
	Low neighborhood attachment	Opportunities for prosocial involvement
	Community disorganization	
	Laws and norms favorable to drug use	
	Perceived availability of drugs	
Family	Family history of antisocial behavior	Attachment
	Parental attitudes favorable towards drug use	Opportunities for prosocial involvement
	Poor family management	Rewards for prosocial involvement
	Family conflict	
	Weak parental supervision	
	Family gang influence	
School	Academic failure	Opportunities for prosocial involvement
	Low commitment to school	Rewards for prosocial involvement
Peer/Individual	Rebelliousness	Belief in the moral order
	Rewards for antisocial involvement	Rewards for prosocial involvement
	Favorable attitudes towards drug use	Interaction with prosocial peers
	Favorable attitudes towards antisocial behavior	Social skills
	Perceived risks of drug use	
	Friends' use of drugs	
	Interaction with antisocial peers	
	Intentions to use	
	Antisocial tendencies	
	Critical life events	
	Impulsive risk taking	
	Neutralization of guilt	
	Negative peer influence	
	Peer delinquency	

nation of Trinidad and Tobago, and Webb *et al.* (2016), focusing on the Northern Triangle nation of El Salvador. These works including protective factors in addition to risk factors, which provide an avenue to learn about positive interventions the community can undertake. For more information on risk and protective factors in low- and middle-income countries, we refer the reader to the systematic reviews of Murray *et al.* (2018) on risk and protective factors for antisocial behavior, and Higginson *et al.* (2018) on risk and protective factors related to gang membership.

2.4.2 Data on Youth Delinquency in Honduras

The instrument used to collect data for this project was the Instrumento de Medicion de Comportamientos (IMC), a revised version of the original Honduran YSET, which was itself a Spanish adaptation of the YSET developed by Hennigan *et al.* (2014). Under a collaboration with the Center for Violence Prevention and Community Safety at Arizona State University, the item bank for the Honduran YSET was expanded to include protective factors and increase the number of risk factors measured, drawing on the Communities That Care survey, Eurogang Youth Survey, and others. This revised item bank was further refined to increase predictive power in the local context.

Our data consists of responses to the IMC from 3972 school-attending youth. The IMC covers basic demographics about the youth, along with 173 items measuring 38 risk and protective factors over four domains: community, family, school and peer/individual. The risk and protective factor scales are provided in Table 2.1. Our variable X consists of responses to these 173 items.

Our data also include answers to 18 items that measure seven problem behaviors. Three items measure *violent behavior*, four items measure *property crime*, three items measure *gang involvement*, three items measure *alcohol and drug use*, two items mea-

sure *drug sales*, two items measure *weapons carrying*, and one item measures *truancy*. In what follows, the outcome Y is a binary variable denoting whether or not the youth is at risk of *violent behavior*. The three items related to *violent behavior* are:

(1) In the past 6 months, have you hit someone with the intention of hurting them?

(2) In the past 6 months, have you attacked someone with a weapon?

(3) In the past 6 months, have you used a weapon or force to get money or goods from someone?

In this application, a "positive" test result means that the youth is "at-risk" of engaging in violent behavior. Youth are deemed to be "at-risk" ($Y = 1$) if they answer "yes" to any of the three items above. Items measuring the other six problem behaviors are not utilized for this analysis.

Connection to previous notation. We have responses to the 173 items X and an outcome variable Y denoting whether or not the youth is in the "at-risk" group for violent behavior ("at-risk" = 1,"not-at-risk" = 0). The variable γ denotes an adaptive test which takes the youth's responses to a subset of the 173 items and predicts a risk class. For our purposes, γ is composed of two parts: a binary decision tree T that maps item responses to a risk probability, and a threshold C that determines risk class based on risk probability. We will analyze the quality of a risk assessment γ using an expected utility function $\mathbb{E}U$; $\mathbb{E}U(\gamma)$ is a weighted average of the sensitivity and specificity of the risk assessment γ.

2.4.3 Modeling

We model the data (X, Y) compositionally as $f(\mathrm{x}, y) = f(y \mid \mathrm{x}) f(\mathrm{x})$. We model $f(\mathrm{x})$ as a Gaussian copula factor model and $f(y \mid \mathrm{x})$ as a logistic XBART model using the `bfa` and `xbart` packages, respectively; this model specification is quite flexible.

As with any Bayesian modeling endeavor, we recommend interrogating model quality and adjusting hyperparameters accordingly via standard posterior predictive checks, including plots to avoid model misspecification; see, for example, Gelman *et al.* (1996) and Gabry *et al.* (2019). These model checks should be performed on training data, and not adjusted after obtaining results on hold-out or validation data. This is the approach we used to determine the number of factors in the model for $f(x)$.

The model we use for $f(\mathrm{x})$ is a Gaussian copula factor model (GCFM), proposed by Murray *et al.* (2013) and implemented in the R package `bfa`. Gaussian copula factor models unite Gaussian factor models with the Gaussian copula. The joint distribution of the fitted model assumes the dependence structure of the Gaussian factor model, but with marginal distributions estimated nonparametrically from the data. The joint dependence structure of the Gaussian factor model is reasonable considering the factor-based nature of the latent constructs being measured by adaptive tests. Additionally, the nonparametric estimation of the marginal distributions is an advantage over methods that assume normal marginals.

As described in Carvalho (2006), in a k-dimensional Gaussian factor model, the i^{th} observation of a $p \times 1$ random vector z can be represented as

$$\mathrm{z}_i = \mathbf{\Lambda} \mathrm{f}_i + \nu_i,$$

where $\mathbf{\Lambda}$ is a $p \times k$ matrix of factor loadings, f_i is a $k \times 1$ vector of factor scores with $\mathrm{f}_i \sim N(0, \mathbf{I})$, and $\boldsymbol{\nu}_i$ is a $p \times 1$ noise vector, with $\boldsymbol{\nu}_i \sim N(0, \mathbf{\Psi})$, $\mathbf{\Psi} = \mathrm{diag}(\Psi_1, \ldots, \Psi_p)$.

Under these assumptions, $\mathrm{z} \sim N(0, \mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{\Psi})$.

Intuitively, a copula is a joint distribution function that allows for separation of the marginals from the dependence structure; the copula completely describes all dependencies among variables. A joint distribution F has a Gaussian copula if it can be written as

$$F(X_1, X_2, \ldots, X_p) = \Phi_p(\Phi^{-1}(F_1(X_1)), \Phi^{-1}(F_2(X_2)), \ldots, \Phi^{-1}(F_p(X_p)) \mid \mathcal{C}),$$

where Φ_p is the p-dimensional multivariate Gaussian CDF with correlation matrix $\mathcal{C}$, and Φ^{-1} is the inverse Gaussian CDF.

The Gaussian copula factor model starts by assigning the latent variable z a k-dimensional Gaussian factor model: $\mathrm{f}_i \sim N(0, \mathbf{I})$, $\mathrm{z}_i \mid \mathrm{f}_i \sim N(\mathbf{\Lambda}\mathrm{f}_i, \mathbf{I})$. We then define x as

$$x_{ir} = F_r^{-1}\left(\Phi\left(\frac{z_{ir}}{\sqrt{1 + \sum_{t=1}^{k} \lambda_{rt}^2}}\right)\right),$$

where $F_r^{-1}(t) = \inf\{t : F_r(x) \geq t, x \in \mathbb{R}\}$ is the pseudo-inverse of $F_r, 1 \leq r \leq p$. By making this specification, $F(\mathrm{x})$ has a Gaussian copula with covariance matrix $\mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{I}$, and marginals $F_1, F_2, \ldots, F_p$.

The `bfa` R package presented in Murray *et al.* (2013) fits a Gaussian copula factor model to data using a parameter-expanded Gibbs sampling scheme. Their method allows for inference on joint distributions of mixed continuous and discrete variables, which is necessary for modeling the joint distribution of the item responses and demographic variables from the IMC data. We fit the Gaussian copula factor model to item response and demographic data from the target population in the IMC data, then obtain samples $\{\tilde{\mathrm{x}}_\mathrm{i}\}_{i=1}^{k}$ using the predictive distribution of the fitted model.

Note that the GCFM was fit to an augmented vector including age: $(\mathrm{X}, \mathrm{Age})$. This allows us to condition on age in defining the posterior prediction distribution that represents our target population. While we could have accomplished this by only

fitting the GCFM to data from a particular age group, fitting the model to the entire population and sampling conditionally after the fact allows for borrowing information from the larger population, and deploying it in service of a subpopulation with fewer data. We only use item responses as splitting variables (inputs) for the adaptive tests.

Sensitivity analysis to the number of factors via posterior predictive checks revealed that 3 or more factors yielded similar conclusions; results for the $k = 3$ factor specification in the Gaussian factor copula model are reported here.

We model $f(y \mid \mathrm{x})$ using a log-linear Accelerated Bayesian Additive Regression Trees (XBART) model that builds on the log-linear Bayesian Additive Regression Trees (BART) model for multinomial logistic regression of Murray (2020) with a modification of the "accelerated" model fitting algorithm of He *et al.* (2018) developed by Wang and Hahn (2021).

In the log-linear Accelerated Bayesian Additive Regression Trees (XBART) model, the probability of observing class s given covariate x_i in a setting with c classes follows a logistic specification

$$\pi_s(\mathrm{x}_i) = \frac{h^{(s)}(\mathrm{x}_i)}{\sum_{t=1}^{c} h^{(t)}(\mathrm{x}_i)}.$$

Following Murray (2020), $\log[h^{(s)}(\mathrm{x}_i)] = \sum_{l=1}^{L} g(\mathrm{x}_i, T_l^{(s)}, \mu_l^{(s)})$ is given a sum of trees representation, where $g(\mathrm{x}_i, T_l^{(s)}, \mu_l^{(s)})$ is a tree with splits given by $T_l^{(s)}$ and leaf mean parameter $\mu_l^{(s)}$. This yields

$$\pi_s(\mathrm{x}_i) = \frac{\exp[\sum_{l=1}^{L} g(\mathrm{x}_i, T_l^{(s)}, \mu_l^{(s)})]}{\sum_{t=1}^{c} \exp[\sum_{l=1}^{L} g(\mathrm{x}_i, T_l^{(t)}, \mu_l^{(t)})]}.$$

In our application, we utilize this multinomial logistic XBART model with $c = 2$ classes to predict the probability of being "at-risk", given item responses x_i, as $\bar{\mathbb{E}}(\tilde{Y} \mid \mathrm{x}_i) = \pi_{s=1}(\mathrm{x}_i)$.

The log-linear XBART classification model provides class probability predictions $\bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\mathrm{x}}_k)$, the probability that a youth with item responses $\tilde{\mathrm{x}}_k$ is in the "at-risk"

group. This modeling choice provides the predictive accuracy and Bayesian uncertainty quantification abilities of BART-based models, with the computational speed-up of the XBART family and the classification-specific adaptions implemented by Wang and Hahn (2021). Notably, this approach is substantially less constrained than typical IRT approaches, which require that the risk probability relates to the item response via the same low-dimensional latent factors. Here, while we assume that the item responses have a latent dimension of $k = 3$, the risk probability can depend directly on every single item individually (with no dimension reduction). However, regularization priors in the tree ensemble representation favor trees that utilize far fewer than every available item.

After fitting a Gaussian copula factor model for $f(\mathrm{x})$ and an XBART model for $f(y \mid \mathrm{x})$, we can obtain data $\{\tilde{\mathrm{x}}_{ij}, \tilde{p}_{ij}, \tilde{y}_{ij} \mid \theta^{(j)}\}$ from the conditional predictive distribution $f(\tilde{\mathrm{x}}, \tilde{y} \mid \theta^{(j)})$. The entire process can be summarized as follows:

(1) Fit a Gaussian copula factor model with parameters θ_{X} to item response data $\mathrm{x}_{1:n}$.

(2) Fit a multinomial logistic XBART model with parameters θ_Y to item response/risk status data $(\mathrm{x}, y)_{1:n}$.

(3) Fixing the j^{th} posterior draw of model parameters $\theta_{\mathrm{X}}^{(j)}$, draw N samples $\{\tilde{\mathrm{x}}_{ij}\}_{i=1}^{N}$ from the conditional predictive distribution $f(\tilde{\mathrm{x}} \mid \theta_{\mathrm{X}}^{(j)})$ using the fitted Gaussian copula factor model.

(4) Compute the probability $\tilde{p}_{ij} = \Pr(\tilde{Y} = 1 \mid \tilde{\mathrm{x}}_{ij}, \theta_Y^{(j)})$ using the j^{th} posterior tree ensemble from the fitted multinomial logistic XBART model.

(5) Sample the class label $\tilde{y}_{ij} \sim \text{Bernoulli}(\tilde{p}_{ij})$.

(6) Our dataset conditioned on the j^{th} posterior draw $\theta^{(j)} = \{\theta_{\mathrm{X}}^{(j)}, \theta_Y^{(j)}\}$

is $\{\tilde{\mathrm{x}}_{ij}, \tilde{p}_{ij}, \tilde{y}_{ij} \mid \theta^{(j)}\}_{i=1}^{N}$.

Additionally, during step (4), we compute the posterior predictive mean probability

$$\bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\mathrm{x}}_{ij}) = \frac{1}{D}\sum_{j=1}^{D} \tilde{p}_{ij} \approx \int_{\Theta_Y} \mathbb{E}(\tilde{Y} \mid \tilde{\mathrm{x}}_k, \theta_Y)\pi(\theta_Y \mid \mathrm{x}_{1:n}, y_{1:n})\, d\theta_Y.$$

By repeating this process D times, $1 \leq j \leq D$, we obtain D population-level samples from our target population. In total, the synthetic data is

$$\{\tilde{\mathrm{x}}_k,\ \tilde{p}_k,\ \bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\mathrm{x}}_k),\ \tilde{y}_k\}_{k=1}^{M}, \qquad M = N \cdot D.$$

We use synthetic data $\{\tilde{\mathrm{x}}_k, \bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\mathrm{x}}_k)\}_{k=1}^{M}$ for calibrating the regression tree with m items, T_m^*. We use $\{\tilde{\mathrm{x}}_k, \tilde{\mathrm{y}}_k\}_{k=1}^{M}$ for choosing the optimal cutoff $C_{T_m^*}$.

We also use $\{\tilde{\mathrm{x}}_k, \tilde{\mathrm{y}}_k\}_{k=1}^{M}$ for doing uncertainty quantification plotting, but broken up into D sample populations as $\{\tilde{\mathrm{x}}_{ij}, \tilde{y}_{ij} \mid \theta^{(j)}\}_{i=1}^{N}$, $1 \leq j \leq D$. For each value of j, we compute $\mathbb{E}U_{\theta^{(j)}}(\gamma)$ for both

$$\gamma_m^*(\cdot) = \mathrm{Thr}_{C_{T_m^*}}(T_m^*(\cdot)) \quad \text{and} \quad \gamma^*(\cdot) = \mathrm{Thr}_{C^*}(\bar{\mathbb{E}}(\tilde{Y} \mid \cdot))$$

using Equation (2.6). The draws of the differences $\Delta_{\theta^{(j)},m}$ between these utilities are then used for uncertainty quantification of $\Delta_{\theta,m} = \mathbb{E}U_\theta(\gamma_m^*) - \mathbb{E}U_\theta(\gamma^*)$.

For this analysis, we drew $N = 1000$ Monte Carlo samples of the form

$$\{\tilde{\mathrm{x}}_{ij}, \tilde{p}_{ij}, \bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\mathrm{x}}_{ij}), \tilde{y}_{ij} \mid \theta^{(j)}\}_{i=1}^{N}$$

for each of $D = 1000$ posterior parameter draws $\theta^{(j)}$, $1 \leq j \leq 1000$. We drew another 100,000 synthetic data from the same fitted models for the pruning step from our description of populating the action space in Section 2.3.2.

We take a moment to connect the concepts introduced here to the previous notation from the beginning of this section. Recall that we fit the Gaussian copula factor

model to item responses and age from the IMC data, then obtain synthetic item response data $\{\tilde{\mathrm{x}}_k\}_{k=1}^{M}$ from the target population using the predictive distribution of the fitted model.

The plots in the next section compare the utility of the shortened screening test γ_m^* to the utility of the full-length test γ^*, which uses all 173 items on the IMC. The instrument $\gamma^*(\cdot) = \mathrm{Thr}_{C^*}(\bar{\mathbb{E}}(\tilde{Y} \mid \cdot))$ is composed of a regression function $\bar{\mathbb{E}}(\tilde{Y} \mid \cdot)$ predicting the probability of the Honduran youth being "at-risk", followed by a thresholding function Thr_{C^*} to predict risk class status. We fit the regression function as an XBART model using the IMC data, then obtain predicted "at-risk" probabilities $\bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\mathrm{x}}_k)$ for the synthetic item responses $\tilde{\mathrm{x}}_\mathrm{k}$. The thresholding function is chosen to optimize the utility function for the target population, given predicted "at-risk" probability $\bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\mathrm{x}}_k)$.

The shortened instrument with m items, $\gamma_m^*(\cdot) = \mathrm{Thr}_{C_{T_m^*}}(T_m^*(\cdot))$, is composed of a binary regression tree T_m^* with maxIPP $= m$, and a thresholding function $\mathrm{Thr}_{C_{T_m^*}}$. The adapted test T_m^* is calibrated using synthetic data $\{\tilde{\mathrm{x}}_k, \bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\mathrm{x}}_k)\}$, $1 \leq k \leq M$. The thresholding function for γ_m^* is computed similarly to the one for γ^*, except that it optimizes the cutoff using "at-risk" probabilities $T_m^*(\tilde{\mathrm{x}}_k)$ rather than $\bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\mathrm{x}}_k)$.

Sections 2.4.4, 2.4.5, and 2.4.6 provide a demonstration of the three steps of the method using the data for youth delinquency in Honduras. Section 2.4.7 provides out-of-sample validation of the method on a hold out set, along with a subgroup analysis using the same hold-out set.

Throughout Sections 2.4.4, 2.4.5, and 2.4.6 we demonstrate the method fitting everything in sample. Out of sample validation results are provided in Section 2.4.7.

Recall the three steps in the decision theory framework laid out above: 1) a utility function for measuring the "goodness" of the assessment; 2) a target population; 3) a method for obtaining assessments of different lengths. Sections 2.4.4, 2.4.5, and 2.4.6

demonstrate how the utility difference plots change as we vary these three choices, respectively, when applied to the Honduras youth risk assessment data.

2.4.4 Demonstration of Changing the Utility Function

First, we highlight how the plots change when we vary Step 1, the utility function. Figure 2.5 shows boxplots of the difference in expected utility for three different weights w in the utility function from Equation (2.4).

For each weight w and each value of m, we compute draws of the utility difference $\Delta_{\theta^{(j)},m} = \mathbb{E}U_{\theta^{(j)}}(\gamma_m^*) - \mathbb{E}U_{\theta^{(j)}}(\gamma^*)$ using synthetic data from each posterior draw j; the posterior distribution of $\Delta_{\theta,m}$ is then visualized via a boxplot of the draws $\{\Delta_{\theta^{(j)},m}\}_{j=1}^{D}$. The distribution of $\Delta_{\theta,m}$ can vary depending on our choice of both m and w.

Figure 2.6 provides a visual example of how (Specificity(γ), Sensitivity(γ)) for $\gamma \in \{\gamma^*, \gamma_m^*\}$, in conjunction with w, lead to different draws of $\Delta_{\theta,m}$ for $m = 3$. In particular, as w gets closer to 0 or 1, it is easier for the shortened test γ_m^* to achieve a utility value closer to that of the non-shortened instrument γ^*. Practically, a value of w close to 0 or 1 amounts to strongly favoring either sensitivity or specificity, at the expense of the other; such decisions can have unintended ramifications, which are discussed further in Section 2.5.1.

2.4.5 Demonstration of Changing the Target Population

Next, we vary Step 2, the target population. The boxplots in Figure 2.7 represent the same quantity as Figure 2.5 (namely, the distribution of $\Delta_{\theta,m}$). However, Figure 2.7 shows expected utility differences for adaptive tests calibrated using two target populations: all Honduran youth, and youth ages 15 and older. We chose to target youth ages 15 and older since age 15 marks the transition from middle school to sec-

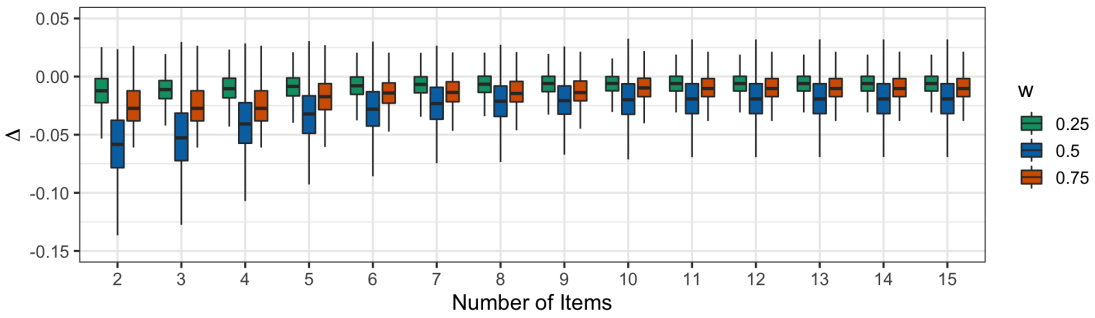

Figure 2.5: Each boxplot represents the posterior distribution of $\Delta_{\theta,m} = \mathbb{E}U_\theta(\gamma^*_m) - \mathbb{E}U_\theta(\gamma^*)$ for a particular number of items m, and a particular value of w in the utility function from Equation 2.4. The samples of $\Delta_{\theta,m}$ that form the boxplot are obtained by computing $\mathbb{E}U_{\theta^{(j)}}(\gamma)$ (via sensitivity and specificity) for γ^*_m and γ^*, on a synthetic data sample $\{\tilde{\mathrm{x}}_{ij}, \tilde{y}_{ij} \mid \theta^{(j)}\}_{i=1}^N$. This sample is obtained from posterior draw $\theta^{(j)} = (\theta_{\mathrm{X}}^{(j)}, \theta_{Y}^{(j)})$ of the two Bayesian models.

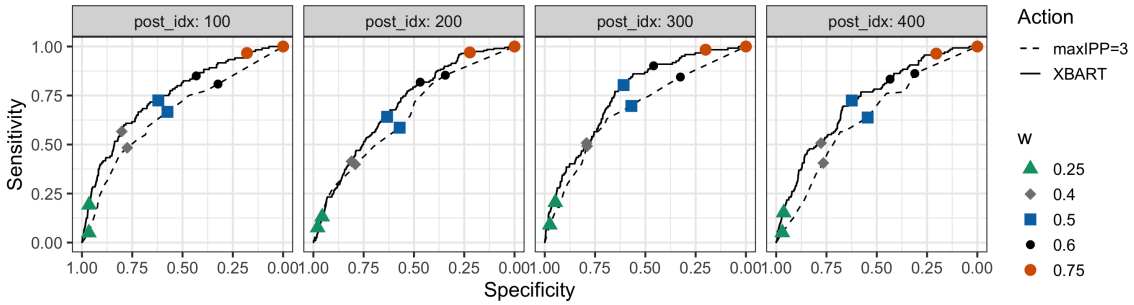

Figure 2.6: Here we show how to obtain posterior draws of $\Delta_{\theta^{(j)},m}$ from posterior draws $\theta^{(j)}$. The four plots show ROC curves obtained from synthetic data samples $\{\tilde{\mathrm{x}}_{ij}, \tilde{y}_{ij} \mid \theta^{(j)}\}_{i=1}^N$ for $j = 100, 200, 300, 400$. The values of j shown here were arbitrarily chosen for demonstration and are not inherently important. The ROC curves are computed based on the predicted "at-risk" probabilities $\bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\mathrm{x}}_{ij})$ and $T^*_m(\tilde{\mathrm{x}}_{ij})$ from the XBART action γ^* and maxIPP $= 3$ action $\gamma^*_{m=3}$ (respectively) for each specific j^{th} population. For each given w, there is exactly one cutoff C which maximizes the utility function $\mathbb{E}U_\theta(\gamma) = w \cdot \text{Sensitivity}(\gamma) + (1-w) \cdot \text{Specificity}(\gamma)$ over all sample populations (all values of j), for $\gamma = \gamma^*_{m=3} = \text{Thr}_{C_{T^*_m}}(T^*_m(\cdot))$. That cutoff C corresponds to a particular (Specificity, Sensitivity) pair for each value of j (for both the XBART and maxIPP$=$ 3 actions), which are visualized as points on the ROC curves from those two actions for the j^{th} synthetic population. Those Sensitivities and Specificities are used to compute the realized utility values $\mathbb{E}U_{\theta^{(j)}}(\gamma^*_m)$ and $\mathbb{E}U_{\theta^{(j)}}(\gamma^*)$, along with their difference, $\Delta_{\theta^{(j)},m}$, which contributes one point to the boxplots in Figure 2.5 for the given values of w and $m = 3$. Notice that values of w closer to 1 lead to differences in sensitivity between γ^* and $\gamma^*_{m=3}$ (distance between points on the Sensitivity axis) being smaller than differences in specificity (distance between points on the Specificity axis). This can be observed in the points corresponding to $w = 0.75$ and, to a lesser extent, $w = 0.6$. The opposite behavior is observed for $w = 0.25$ and $w = 0.4$.

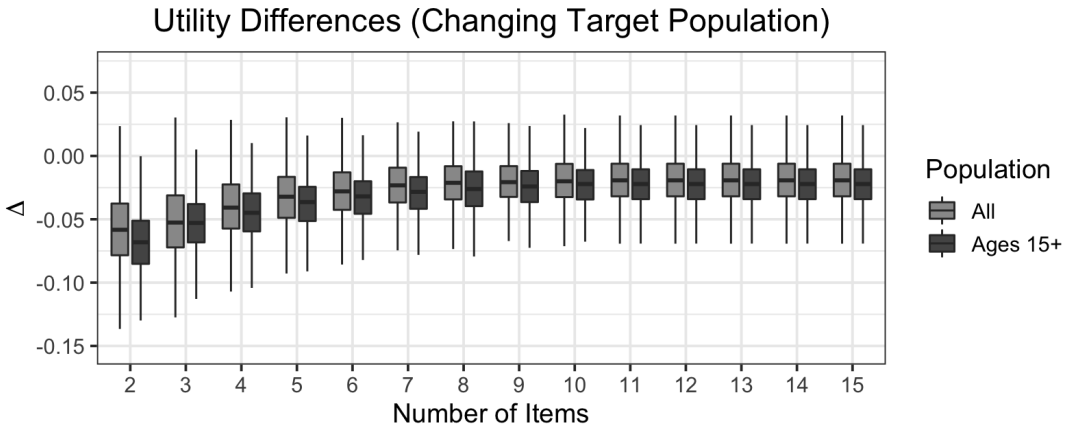

Figure 2.7: Utility difference plots when calibrating the trees to a different target population. Plots here are shown for a target population being every youth in the full IMC data ("All"), and those youth ages 15 and older ("Ages 15+"). The utility plots are quite similar, although exam truncation seems to result in a greater loss of utility (relative to the full screening instrument) for the older group. Calibrating the adaptive test to the subgroup also results in slightly more certainty compared to the full population.

ondary school, as well as the quinceañera ceremony. To change the target population, we used the GCFM fit to the entire dataset, but then drew samples $\{\tilde{\mathrm{x}}_{ij}, \tilde{y}_{ij} \mid \theta^{(j)}\}$ using the conditional predictive distribution, $f(\tilde{\mathrm{x}}, \tilde{y} \mid \mathrm{x}_{1:n}, y_{1:n}, \mathrm{Age} \geq 15)$.

The expected utility plots are similar; however, targeting the subgroup when designing the adaptive test yields slightly less variability in the posterior estimates of the utility difference. Interestingly, these similar results arise based on adaptive tests that use different splitting items and cutpoints. Figures 2.8 and 2.9 show the trees with maxIPP of 3 representing the adaptive tests for these two target populations. The items corresponding to these trees and their response options are listed in Table 2.2. Notice that because of the maxIPP criterion, these trees have a maximum depth of 5, but have only 3 unique items in each root-to-leaf path.

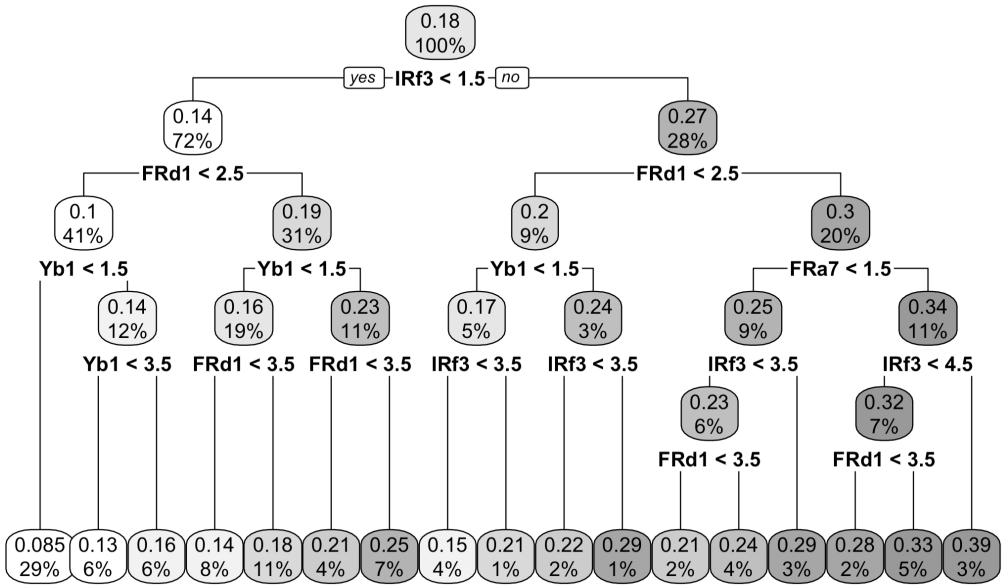

Figure 2.8: Tree representing the adaptive test calibrated using the entire group of Honduran youth as the target population. The items and item responses corresponding to each node label and cutpoint, respectively, are found in Table 2.2. This figure and Figure 2.9 were created using the `rpart.plot` package (Milborrow (2021)).

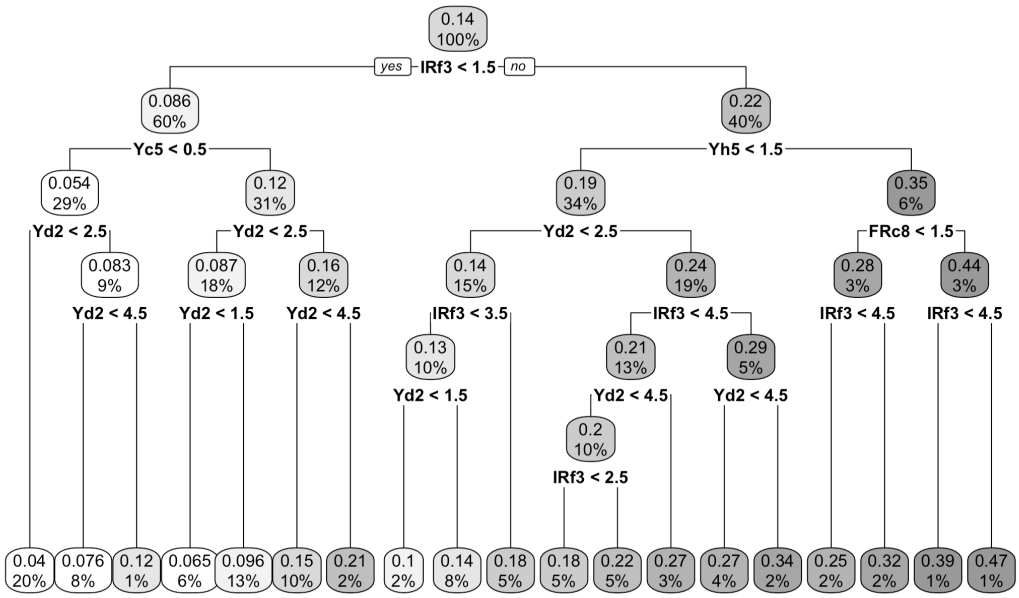

Figure 2.9: Tree representing the adaptive test calibrated using the group of Honduran youth ages 15 and older as the target population. The items and responses corresponding to the node labels and cutpoints are found in Table 2.2.

Table 2.2: Items corresponding to the splitting variables present in the tree-based adaptive tests of Figures 2.8 and 2.9. The right-most column shows the target population for which this variable is included in its adaptive test.

Variable	Item	Response Options	Population(s)
IRf3	In the past 6 months, how many of your best friends have tried beer, wine or hard liquor (for example, vodka, whiskey or gin) when their parents didn't know about it?	1 = None of my friends 2 = 1 of my friends 3 = 2 of my friends 4 = 3 of my friends 5 = 4 of my friends or more	All Youth
Yc5	In the last year, have you fought or had a problem with a friend?	0 = No 1 = Yes	All Youth
FRd3_ctc	People in my family often insult or yell at each other.	1 = No! 2 = no 3 = yes 4 = Yes!	All Youth
FRa6	Has anyone in your family had a severe alcohol or drug problem?	1 = No 2 = Yes	All Youth Age $\geq$ 15
Yh5	During the last six months, how many friends have belonged to or have joined a gang or "mara"?	1 = None 2 = A few 3 = Half 4 = Most 5 = All	All Youth Age $\geq$ 15
Yd2	Sometimes I find it exciting to do things that could get me in trouble.	1 = Strongly disagree 2 = Disagree 3 = Neither agree or disagree 4 = Agree 5 = Strongly agree	Age $\geq$ 15
Ya6	People "blame me" for lying or cheating.	1 = Never 2 = Rarely 3 = Half the time 4 = Often 5 = Always	Age $\geq$ 15

2.4.6 Demonstration of Changing the Action Space

Finally, we can consider different algorithms for populating the action space. In this work we have focused on the composite action $\gamma = \text{Thr}_C(T)$, a regression tree T predicting "at-risk" probability followed by a cutoff C that determines risk status.

Our proposed method for populating the action space is a regression tree obtained by applying the maxIPP growing and pruning method to synthetic data obtained from the posterior predictive distribution, and a threshold optimized to the utility function for the tree T.

Many other methods are possible. For example, one can calibrate the regression tree using a stopping criterion like maximum depth, or apply the algorithm to different synthetic data or to real data; a classification tree can be used as the adaptive test directly instead of a regression tree followed by a cutoff. We explore these possibilities in appendix B. The main takeaway is that tree-based adaptive tests that do not optimize the utility function at all during their design are significantly worse at reproducing the utility of a full-item assessment, relative to adaptive tests that do.

2.4.7 Out-of-Sample Corroboration

The proposed method will be empirically reliable only insofar as the posterior predictive distribution suitably reflects the distribution of future outcomes. To verify that our Gaussian copula factor and XBART models are succeeding in this regard, we perform the following hold-out experiment. Our data was collected in two different time periods, the first wave between September and November of 2017 and the second wave between January and February of 2018. The earlier-collected data is our training set and consists of 2787 youth; the later data is our testing set and consists of 1185 youth. Simply put, this experiment answers the question: how would our approach have performed if we had applied it in 2018, based on the 2017 data?

Figure 2.10 demonstrates the expected utility difference plots we obtained by applying our method on the training data, and the actual expected utility differences on the testing data. To compute the actual expected utility, we used our proposed method on the training data to obtain a tree-based adaptive test for each value of

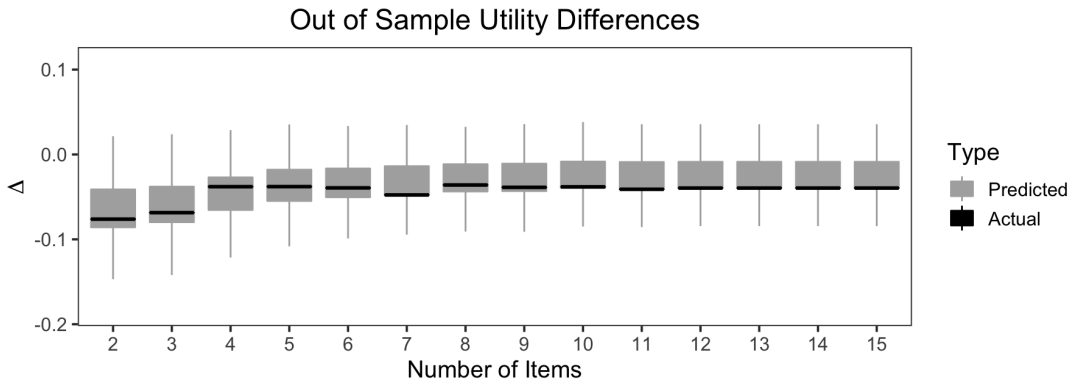

Figure 2.10: Plots of the projected difference in expected utility produced via our method on the training data, and the actual difference in expected utility computed on the test data; these are results for $w = 0.5$.

maxIPP, along with a full-item (non-shortened) test. We also produced the boxplots representing our uncertainty around $\Delta_{\theta,m}$ using the training data alone. We then predicted risk classes on the testing set using both the tree-based adaptive test and the full item test, and computed the difference in empirical utility over the testing set. The empirical utility on the testing set is always within our predicted range, in fact within the 25^{th} and 75^{th} quantiles of the distribution.

Beyond utility differences relative to the full item test, practitioners are interested in the absolute sensitivity and specificity of the instrument. Table 2.3 provides out-of-sample sensitivity and specificity values for the adaptive tests from a subset of maxIPP values shown in Figure 2.10, along with adaptive tests calibrated using utility functions with $w = 0.4$ and $w = 0.6$. Increasing w results in higher sensitivity and lower specificity, as expected. For full results on maxIPP 2 to 15, along with these quantities for other types of adaptive tests, see the tables in appendix B.

Finally, we use the holdout set to show how specifying a particular target population can improve sensitivity, specificity, or overall utility when building adaptive tests. The two target populations under consideration are "All Youth" and "Ages 15+". Table 2.4 shows the number of participants in each of the age groups from our data in

Table 2.3: Sensitivity and specificity on the test data for five adaptive tests, optimized for $w = 0.4$, 0.5, 0.6. The synthetic data for calibrating the adaptive tests was obtained from models fit to the training data only.

Number of Items	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity
	$w = 0.4$		$w = 0.5$		$w = 0.6$	
3	0.396	0.841	0.778	0.522	0.882	0.355
6	0.507	0.816	0.771	0.587	0.931	0.300
9	0.514	0.812	0.750	0.609	0.924	0.354
12	0.528	0.803	0.757	0.600	0.931	0.362
15	0.528	0.803	0.757	0.600	0.931	0.372

Table 2.4: Counts of participants in each age group in the training and testing sets.

Data	Ages 8-14	Ages 15+	Total
Training Set	2297 (82.4%)	490 (17.6%)	2787 (100%)
Testing Set	898 (75.8%)	287 (24.2%)	1185 (100%)

both the training and testing sets. For the adaptive test with target population "All Youth", we fit the Gaussian copula factor model and logistic XBART model to the entire training data and obtained synthetic data using these models, which was then used for calibrating the tree-based adaptive test. For the adaptive test with target population "Ages 15+", we used the same models fit to the entire population, but drew synthetic data from the group of youth ages 15 and older using the conditional predictive distribution $f(\tilde{\mathrm{x}}, \tilde{y} \mid \mathrm{x}_{1:n}, y_{1:n}, \text{Age} \geq 15)$. We then calibrated a tree-based adaptive test to this synthetic data. This process was repeated for maxIPP = 2 to 15, leaving a total of 28 regression trees. We computed the optimal cutoffs that maximized the utility function (2.4) for $w = 0.6$.

After calibrating the trees and computing the optimal cutoffs (using the training data only) to obtain 28 tests, both sets of adaptive tests were then deployed to predict "at-risk" status on youth ages 15 and older in the testing set, and sensitivity, specificity, and utility for this group were computed for each of the 28 tests. We chose a value of 0.6 for this analysis, because we are targeting a group of older youth that

have been shown to receive positive treatment effects from the secondary prevention counseling program (see Katz *et al.* (2021)). For this group it is more important to not miss the youth that are at the highest risk, than to prevent "not-at-risk" youth from mistakenly receiving the intervention.

Figure 2.11 shows the differences in empirical out-of-sample sensitivity, specificity, and overall utility between the adaptive tests calibrated to the two different populations, for each value of maxIPP; the absolute quantities are given in Table 2.5. The adaptive tests optimized for "All Youth" with $w = 0.6$ are not appropriate for this particular subpopulation, because those questions indicate that all of the youth ages 15+ in the test set are "at-risk" (leading to 0 specificity, which clearly is unacceptably low). Trying to increase sensitivity for the entire population results in items that are uninformative for the older youth. When we calibrate the adaptive test specifically to this subgroup, we sacrifice only a small amount of sensitivity for huge gains in specificity.

The improvement by focusing the test to a specific group is an important finding related to *focused deterrence* and *multiple gating*. Focused deterrence implies introducing interventions specific to the group where they will be deployed; multiple gating means targeting youth for secondary prevention programs who are at the highest risk of the delinquent behavior and living within the highest risk neighborhoods (Katz *et al.* (2021)). Both of these methodologies are important aspects of successful community-based crime prevention programs (Abt and Winship (2016), Katz *et al.* (2021)), and using accurate screening instruments for the population where an intervention will be introduced is critical to their successful implementation.

While the lack of specificity on the older group of youth using an adaptive test calibrated to all youth is alarming, this highlights the importance of using adaptive tests designed specifically for the group on which they will be deployed. All adaptive

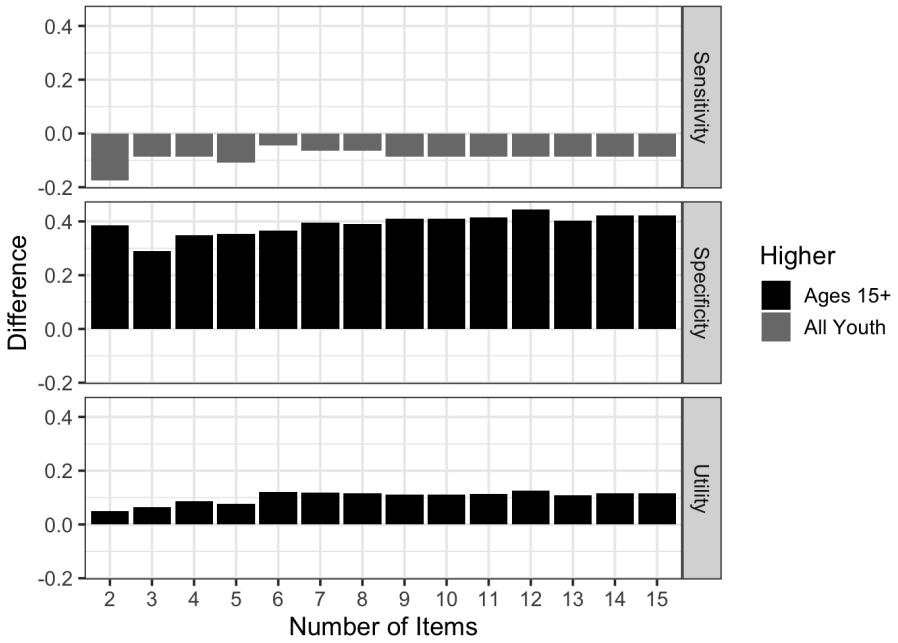

Figure 2.11: Differences in sensitivity, specificity, and utility for youth ages 15+ in the testing data, between two adaptive tests ($\gamma_{\text{All Youth}}$ and $\gamma_{\text{Ages 15+}}$) created using training data. The adaptive test $\gamma_{\text{All Youth}}$ is designed to approximately optimize expected utility for all youth, and $\gamma_{\text{Ages 15+}}$ for youth ages 15+. The bar height in the upper plot is Sensitivity($\gamma_{\text{Ages 15+}}$) − Sensitivity($\gamma_{\text{All Youth}}$) computed on youth ages 15+ in the testing data, and similarly for specificity and utility.

tests that are created using a machine learning (ML) algorithm, such as CART, do so by heuristically optimizing a given criterion over a specific dataset. This may have unintended consequences when the data for which the test was optimized differs in distribution to the specific group on which the screening test will be deployed.

The benefit of our proposed method for obtaining the adaptive test (chosen to optimize the criteria in our Bayesian decision theory evaluation framework), is that these choices are directly placed in front of the screening test designer when the adaptive test is created. One must think critically about the target population for which the test is optimized, and the utility function being optimized–these are decisions that are inherently made in other tree-based adaptive test procedures, but under the hood.

Table 2.5: Specificity, sensitivity, and utility with $w = 0.65$ on youth ages 15 and older from the testing data, for adaptive tests calibrated on two different target populations in the training data: youth ages 15 and older, and all youth. "Target Population" shows the population from which synthetic data were obtained for calibrating the test.

Target Population	**Age ≥ 15**	**All Youth**	**Age ≥ 15**	**All Youth**	**Age ≥ 15**	**All Youth**
maxIPP	Sensitivity		Specificity		Utility	
2	0.826	1.000	0.386	0.000	0.650	0.600
3	0.913	1.000	0.290	0.000	0.664	0.600
4	0.913	1.000	0.349	0.000	0.687	0.600
5	0.891	1.000	0.353	0.000	0.676	0.600
6	0.957	1.000	0.365	0.000	0.720	0.600
7	0.935	1.000	0.394	0.000	0.719	0.600
8	0.935	1.000	0.390	0.000	0.717	0.600
9	0.913	1.000	0.411	0.000	0.712	0.600
10	0.913	1.000	0.411	0.000	0.712	0.600
11	0.913	1.000	0.415	0.000	0.714	0.600
12	0.913	1.000	0.444	0.000	0.725	0.600
13	0.913	1.000	0.402	0.000	0.709	0.600
14	0.913	1.000	0.423	0.000	0.717	0.600
15	0.913	1.000	0.423	0.000	0.717	0.600

A further benefit is that data from a larger population can be used to adapt a screening test to a subpopulation where fewer data are available. We borrow information from the whole population when fitting the GCFM model, but sample from the subpopulation of older youth using the conditional posterior predictive distribution from that fitted model. This conditional sample is then used for calibrating the adaptive test to the subgroup. This is an unusual and exciting example of transfer learning—utilizing the information that an ML algorithm obtains from larger datasets when applying the algorithm in service of a slightly different problem where fewer data are available.

2.5 Discussion

From a practical perspective, the summary of our analysis is highly encouraging: a much shorter assessment can be given that will nearly match the predictive accuracy (as characterized by the weighted sensitivity and specificity) of the much longer original assessment. Specifically, we were able to design adaptive tests of varying lengths for the target population of youth ages 15 and older, living in 5 of the poorest and most violent cities in Honduras. Out-of-sample sensitivity over 0.9 and specificity over 0.4 was achieved for an adaptive test that uses only 9 items. This is an increase in specificity of 0.4 over an adaptive test optimized to youth of all ages together. If a more convenient screening tool leads to more individuals being screened, limited crime mitigation resources can be employed in a more effective manner.

However, precisely because the stakes are so high, it is critical to carefully inspect the algorithms and proposed methods for potential ethical implications. Accordingly, we conclude this chapter with an examination of potential pitfalls of our proposed method. The importance of such considerations have recently been emphasized under the broad heading of "ethical AI" (artificial intelligence) (cf. Johndrow and Lum (2019) and Chouldechova and Lum (2020)). Two main concerns include disparate impacts on particular subpopulations, and the difficulty in interpreting or interrogating automated decisions from sophisticated data-driven algorithms.

2.5.1 Evaluating Disparate Impact

Biased training data can result in risk assessment tools that produce unethical or unfair decisions for particular groups of people, in domains such as criminal justice (Chouldechova and Lum (2020), Chouldechova (2017), Eckhouse *et al.* (2018)) and child welfare (Chouldechova *et al.* (2018)). For example, historical data may unfairly

indicate that a certain racial group is at higher risk of re-arrest, simply due to more aggressive policing in their neighborhoods; a statistical model trained on this type of historically biased data will produce unethical decisions on important questions like pre-trial release.

Similarly, our method is only as unbiased as the data used to train the model. In our particular case, the outcome used in the IMC data is self-reported; unlike in United States recidivism data, for which "re-arrest" is an inaccurate and racially-biased proxy for "re-offense" (see Johndrow and Lum (2019)), the delinquency data on the IMC is based on the individual youth self-reporting whether they engage in the behavior, as opposed to school or law enforcement records that may be biased by historical law enforcement patterns.

While our assessment would disadvantage a group of youth who were systematically dishonest in their self-reported violent behavior, and it is possible that there may be such a group, such patterns in the youth represented in the IMC have thus far not been observed; the scales used in the IMC were chosen for their efficacy, internal validity and reliability (Katz *et al.* (2021)).

The nature of historically advantaged or disadvantaged groups also differs: the youth for whom the current application is intended are fairly homogeneous. These youth are of the same race and ethnicity and experience similar levels of poverty, living in the poorest neighborhoods within the five most dangerous and violent cities in Honduras, which is itself one of the most violent countries in the world. While ethnic minority groups live in parts of rural Honduras, this analysis has been undertaken for the scope of application in five particular urban neighborhoods under consideration.

Although our algorithm is unlikely to result in disparate impacts among racial groups in these neighborhoods (simply due to lack of heterogeneity), there is a possibility for differential impact by age, and possibly other features like gender or religion.

In Katz *et al.* (2021), positive treatment effects from the secondary prevention program were observed for older youth (divided at age 14 and older), whereas mixed treatment effects were observed for the younger group. This highlights the importance of careful selection of the weight w in designing the adaptive test.

As a concrete example, in the randomized controlled trial (RCT) which continued after the initial IMC data collection (Katz *et al.* (2021)), services were given to 994 youth deemed to be "at-risk", out of 4495 screened. Supposing that 994 of the 4495 screened youth were truly "at-risk", a decrease in sensitivity of 5% would result in 50 more "at-risk" youth being denied the intervention, whereas a 5% rise in specificity would result in 175 more "not-at-risk" youth being prevented from incorrectly receiving the intervention. An adaptive test that trades this increased specificity for decreased sensitivity may be acceptable within a younger group, but not for an older one.

Similarly, harmful consequences can arise from a shift in the target population between test creation and deployment. An adaptive test that optimizes utility for youth over a large age range (e.g., 8-17) may not have acceptable accuracy for youth within a more specific age group; indeed, this was the case for youth ages 15+ (see Section 2.4.7).

To summarize, the possibility for disparate impact using our proposed method, as with most automated decision making via ML algorithms, hinges on whether or not particular subpopulations are given due consideration in the test design process. Attention and care must be given to the selection of the target population and the weight w when optimizing the adaptive test, to ensure the best outcomes for the youth being screened for risk of delinquency.

2.5.2 Ability to Scrutinize Automated Decisions

Independent from concerns surrounding flawed training data and the differential impacts it creates, the sheer complexity of a data driven risk assessment invites skepticism. Flaws can be hard to identify when the inputs and outputs are high dimensional numerical vectors (Chouldechova and Lum (2020)). On this point, we consider our method to be a significant advance over existing approaches.

One, our final risk prediction assessment tool is a single decision tree, which can easily be understood and adapted as needed to reduce potential bias or problematic prediction patterns. For example, if a particular item results in lower predicted risk probability based on behavior that is believed to increase it, that item can be excluded from the item pool and the decision tree re-calibrated to the remaining items.

Two, the inputs to our method are transparent – a utility function, a target population, and a set of candidate instruments generated by a heuristic. Sensitivity to these choices can and ought to be investigated; the execution of such comparisons is precisely what our novel decision theory framework facilitates. Although the process is quite involved, its transparency and flexibility should make it *less* prone to unanticipated flaws than ad-hoc methods of abridging screening tests, whether data-driven or human guided.

Most importantly, the application-specific nature of many aspects of this work answer the call for locally-designed screening tests that are suitable for the specific setting in which they will be deployed. In particular, we note the relevance of designing a screening test with a specific population in mind, and of choosing the most important utility function for evaluating the goodness of that screening test.

2.5.3 Summary of Our Contribution

As a final summary, we recap the contributions made in this chapter. In Honduras and other low- and middle-income countries, critical community support resources are in short supply and must be used efficiently and effectively. In order to allocate secondary prevention resources in the best manner possible, an accurate and short screening instrument is needed that will allow administrators to screen as many youth as possible, while not sacrificing too much by way of the specificity and sensitivity of the instrument. While tree-based adaptive tests are a promising avenue for such instruments, a clear method for understanding the losses induced by shortening the instrument has been lacking.

To address this problem, we have presented a novel three-step framework for determining how to shorten screening instruments in a principled way, which consists of choosing a utility function, specifying a target population, and comparing a populated action space of screening tests via expected utility.

To emphasize, while particular choices for each of these steps were presented in our analysis of the Honduras data, many other choices are possible. For example, the specific value of w in the utility function can be chosen based on whether specificity or sensitivity is more important; or, another utility function involving other classification metrics can be chosen. The target population can be specified as youth of a particular age, neighborhood, gender, school, or any other subpopulation for which a specific screening instrument may be useful, as long as some data for this target population are available. And while we have focused on tree-based adaptive tests relying on the CART algorithm in this analysis, one can utilize other tree-growing algorithms for populating the action space, or compare IRT-based adaptive tests as well. The framework itself is generic, in the sense that once a practitioner has chosen a utility

function, a target population, and an algorithm for populating the action space, the same procedure can be applied to understand the trade-offs of shortening the exam to different lengths, or of making a different choice at one of the three steps.

These choices should be made carefully by policy-makers and local stakeholders, aided by researchers who can explain the trade-offs associated with one decision versus another. Researchers can provide insight via the utility plots, or similar plots created for uncertainty quantification of sensitivity or specificity at the relative or absolute level. Local-stakeholders and policy-makers can assess which outcomes are most important for the group being screened in their specific application. These groups working in concert should adjust the assessment to accommodate desired levels of sensitivity and specificity for the particular population in which it will be deployed, as much as possible considering practical limitations (e.g. counselor availability in our application).

Chapter 3

CLASSIFICATION COMPLEXITY

3.1 Overview

In this chapter, we introduce ideas from statistical learning theory, information theory, and empirical measures of data complexity to analyze the complexity of a classification problem. We explore quantification of classification difficulty from both a theoretical and empirical point of view.

Before beginning, we briefly situate this chapter in the context of the preceding and following chapters. Digital screening tests present a significantly more complicated classification problem compared to the last chapter on item-based screening tests. The difficulty arises because data used for digital screening tests are very high dimensional in nature, on the order of millions of samples taken at discrete time points. This data must be transformed to a lower-dimensional set of meaningful features to be used in a classification model that serves as the screening test. Because screening test performance entirely relies on the nature of the underlying classification problem, the impact of this transformation on said difficulty is of immediate interest.

In the next chapter, we will explore the impact of both the data collection protocol (the speech elicitation task) and the subsequent feature extraction step on classification difficulty, and provide recommendations for how to design processes that reduce the classification complexity of the resulting data. To discuss classification difficulty in a meaningful way, the concept of classification difficulty must first be defined. What are the key factors that underpin how likely we are to be successful in solving the classification problem with a particular dataset, and how well will the models

(screening tests) we obtain perform on unseen data in the future? We seek to provide both theoretical and empirical answers to these questions in the body of this chapter, which serve as a backdrop for understanding the results on speech-based screening tests in Chapter 4.

The questions raised above have been the subject of decades of fundamental research in two key areas, statistical learning theory and information-theoretic divergence measures. We review these areas in the first two sections and explicitly discuss their contributions in measuring two key aspects of classification complexity.

More recently, a body of work has emerged on empirical quantities from a particular dataset that inform classification complexity. In the third section, we review such data complexity measures and discuss how they relate to the themes presented from the statistical learning theory and information theory literature.

In the final section, we explore a simulated example in which two aspects of classification complexity (decision boundary complexity and class overlap) are explicitly controlled and co-varied. We show how the statistical learning theory, information theory, and empirical data complexity measures provide information on the underlying complexity of the simulated data.

First, we present definitions to be used throughout. In a generic binary classification problem, we assume there is an underlying data generation process in which a supervisor (or oracle), given a set of inputs X, assigns a true outcome y according to $P(Y \mid \mathrm{X})$. The inferential goal is to learn, from a finite set of independent and identically distributed (i.i.d) data $\{(\mathrm{x}_1, y_1), \cdots, (\mathrm{x}_n, y_n)\}$ drawn according to $P(\mathrm{X}, Y)$, some model of the data generation process, which allows for prediction of of class label $\tilde{y}$ for a new datapoint $\tilde{\mathrm{x}}$ drawn from $P(\mathrm{X})$.

Reiterating using more formal notation, in the first part of this chapter we review key concepts from three bodies of literature that formalize the process of learning

about the world from data:

- **Statistical learning theory** provides theoretical guarantees on rates of convergence of a learning algorithm (i.e., a process for selecting a function from a function family) to an optimal function for a given learning problem; in other words, how close are the obtained classification functions to the optimal one that can be achieved. Statistical learning theory furthermore informs how well the learned classification function will generalize to unseen data. Results on these questions are relative to a specific learning algorithm and function family, and may be distribution (i.e., $P(\mathrm{X}, Y)$)-specific or distribution-agnostic; the emphasis is on the function family, or hypothesis class, and the learned function.

- **Information theoretic divergence measures** provide theoretical guarantees around how difficult a classification problem is, based on the specific distribution $P(\mathrm{X}, Y)$ underlying the data generation process for that classification problem. The results are agnostic to the specific function chosen for future classification; emphasis is on the underlying joint probability density $P(\mathrm{X}, Y)$, and the optimal performance that *any* learning algorithm can achieve for the probability density defining the classification problem.

- **Measures of classification complexity** (renamed *measures of dataset complexity* in the recent literature) provide empirical estimates on the difficulty of a classification problem, both for specific function families and in a function-agnostic matter. These measures approach the problem of quantifying classification difficulty from the opposite direction, using empirical, rather than theoretical, methods applied to a specific finite sample from $P(\mathrm{X}, Y)$.

The next three sections discuss each of these three bodies of literature in turn.

3.2 Statistical Learning Theory

A foundational text in statistical learning theory is Vapnik (1995), which presents a readable summary on key concepts originally presented in Vapnik and Chervonenkis (1971). Vapnik (1995) reviews the key breakthroughs from a number of heavyweights surrounding the advancements of learning theory up until the mid-1990's, including Andrey Tikhonov, Emanuel Parzen, Andrey Kolmogorv, and Ray Solomonoff, along with himself and Alexey Chervonenkis. We recommend this introductory chapter for a pleasant journey through pivotal discoveries in four areas (ill-posed problems, nonparametric statistics, the law of large numbers in functional space, and algorithmic complexity) that influenced the development of statistical learning theory.

The remaining text provides background and main results on the questions learning theory aims to address, such as "How can one control the rate of convergence (the rate of generalization) of the learning machine?" In answering these and other questions, Vapnik's results were comprehensive, in that he produced asymptotic theoretical results, nonasymptotic theoretical results for both large and small samples, and practical algorithms to which these results could be applied. One of these practical algorithms included the invention of the Support Vector Machine (SVM), (Cortes and Vapnik (1995)), which still receives widespread use today in a diverse set of applications (Cervantes *et al.* (2020)). A succinct overview of Vapnik's main questions and the theories introduced to answer them can be found in Vapnik (1999). We present relevant definitions and a few key results in this section, also drawing on ideas from Bousquet *et al.* (2004).

In order to situate this review in an understandable context, we first pose three questions for which statistical learning theory, in particular the results reviewed here, provides some answers. These questions are posed informally, in order to connect

the type of discussions that machine learning practitioners have on a regular basis to the body of work on statistical learning theory. The questions will also be used to anchor the many results and examples presented throughout this section, which are necessarily to adequately capture key themes from statistical learning theory.

The three questions are:

(1) How do we create models that learn about the world using data?

(2) How close is my model to an optimal one for this problem?

(3) How well will my model work on new data in the future?

We begin with Question (1): How do we create models that learn about the world using data? Statistical learning theory answers this question by (a) introducing a formal notation to discuss relevant aspects, (b) defining the optimization problem to be solved, and (c) providing different processes for approaching this optimization problem. We begin by formalizing the relevant aspects of learning about the world through data.

At a high level, statistical learning theory formalizes the process of *learning about the world through a series of observations,* which consists of making observations, creating a model based on those observations, and using the model to make predictions about future observations. Formally, the process of learning from data assumes three parts: 1) a probability distribution $P(\mathrm{X})$ over $\mathcal{X}$, from which samples $\mathrm{x} \in \mathcal{X}$ are drawn; 2) a supervisor that assigns $y \in \mathcal{Y}$ for each $\mathrm{x} \in \mathcal{X}$ according to $P(Y \mid \mathrm{X})$; and 3) a learning algorithm that can realize a set $\mathcal{F}$ of functions $f : \mathcal{X} \to \mathcal{Y}$. $\mathcal{F}$ is called the *hypothesis class*, also called a *concept class* in Probably Approximately Correct learning theory, or a *function class* in more recent literature.

Note that, for consistency with the rest of this thesis, we use $\mathcal{F}$ to denote the set of functions in $\mathcal{Y}^{\mathcal{X}}$, rather than using $\mathcal{F}$ to denote their counterparts in $\{0,1\}^{\{\mathcal{X},\mathcal{Y},f\}}$

that map the function prediction and true outcome to a loss value when f is binary (as is done in Bousquet *et al.* (2004)).

In order to choose a learning algorithm from $\mathcal{F}$, we need to define a formal goal that will delineate which functions from $\mathcal{F}$ are better than others, in other words, to define the optimization problem to be solved. In statistical learning theory, the goal is to obtain f which minimizes the risk $R(f)$ for a given loss function $L(y, f(\mathrm{x}))$:

$$R(f) = \int L(y, f(\mathrm{x}))dP(\mathrm{x}, y). \tag{3.1}$$

The loss function changes depending on the type of inference the learning algorithm is being used for (e.g. classification estimation, regression estimation). A common loss function for the classification setting is the 0-1 loss:

$$L(y, f(\mathrm{x})) = \begin{cases} 0, & f(\mathrm{x}) = y \\ 1, & f(\mathrm{x}) \neq y \end{cases} \tag{3.2}$$

The minimum risk over all measurable functions f is denoted

$$R^* = \inf_f R(f). \tag{3.3}$$

Having introduced (a) the formal notation and (b) the optimization problem to be solved, we now turn to the final component of how to create models that learn about the world through data: a process for how to select a model that solves the optimization problem. Notationally, the function f that represents our selected model of the world is chosen according to a particular process (termed a *principle* in Vapnik (1995)), using observed data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$.

Along with the observed data, the *No Free Lunch* theorem, as described by Bousquet *et al.* (2004), implies that we must make assumptions in order for one function to be better than another. These assumptions include knowledge about how the past observations relate to future observations (in our case, the i.i.d assumption and that

future observations are also i.i.d according to $P(\mathrm{X}, Y)$), and other assumptions on the hypothesis class $\mathcal{F}$ from which we select f. For example, we may assume that $\mathcal{F}$ only contains functions from a specific model family, or that the optimal function should be as simple as possible while still fitting the patterns in the observed data.

Only with such assumptions can we learn a useful model of the world that allows us to make reasonable predictions on future data. This idea is captured in the formula (Bousquet *et al.* (2004))

$$\text{Generalization} = \text{Data} + \text{Knowledge}. \tag{3.4}$$

There are various processes for selecting the function f_n which is optimal according to our observed data, given our assumptions (knowledge). These procedures depend on the *empirical risk* $R_{\text{emp}}(f)$ of a function f, defined as:

$$R_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\mathrm{x}_i)). \tag{3.5}$$

Here we review three processes described in Vapnik (1995) and Bousquet *et al.* (2004) for selecting f_n.

- **Empirical Risk Minimization (ERM)**: Choose $\mathcal{F}$ to be a hypothesis class of lower complexity, containing a family of functions that follow a specified model. (Precise definitions of the complexity of $\mathcal{F}$ will be given later in the chapter). Then choose $f_n \in \mathcal{F}$ that satisfies

  $$f_n = \text{argmin}_{f \in \mathcal{F}} R_{\text{emp}}(f). \tag{3.6}$$

  ERM is a foundational principle from the early work in statistical learning theory; Vapnik (1995) provides necessary and sufficient conditions for the consistency of the ERM principle.

- **Regularization**: Choose a large function class $\mathcal{F}$ of greater complexity, and apply a regularizing term, so that the regularized empirical risk is minimized:

$$f_n = \text{argmin}_{f \in \mathcal{F}} R_{\text{emp}}(f) + \lambda ||f||^2. \tag{3.7}$$

  In practice, we do not usually know what the best value for λ is; this is normally chosen via cross validation. Also of note, the Regularization principle can be recast as an ERM procedure if we make a few modifications, including 1) allowing more flexibility on the hypothesis class $\mathcal{F}$; and 2) defining the loss function over the entire sample rather than pointwise, and including the regularization term directly into this loss function.

- **Structural Risk Minimization (SRM)**: Choose an infinite sequence of hypothesis classes $\{\mathcal{F}_d \mid d = \{1, 2, 3, \dots\}\}$, where each hypothesis class contains functions that follow a particular model specification, and the complexity d of each consecutive hypothesis class increases. Then choose f_n which satisfies

$$f_n = \text{argmin}_{f \in \mathcal{F}_d, d \in \mathbb{N}} R_{\text{emp}}(f) + \text{pen}(d, n), \tag{3.8}$$

  where the penalty term increases with hypothesis classes of increasing complexity d.

In ERM, our simplifying assumption is that the optimal function is from a specific model family, usually containing functions of lower complexity. In the Regularization principle, we inject our assumption that simpler models are better via the penalty term $\lambda ||f||^2$, rather than explicitly selecting a simpler hypothesis class. In SRM, our assumption favoring model simplicity is captured by the structured sequence of model families and the penalty term $\text{pen}(d, n)$.

With these definitions in place, we have introduced a formal setting that, at a high level, answers Question (1), or how we can use data to learn functions that provide

information about the world and allow us to make predictions.

With a learned function in hand, we can ask relevant questions about the learned function (which is often called a "fitted model" rather than a learned function in modern machine learning and statistics). We start by turning to Question (2) from the beginning of this section: how close is my model to an optimal one for this problem?

As a reminder, the function selected by one of the strategies (ERM, SRM, etc.) applied to a finite sample of size n is denoted f_n. Our approximation of the risk of f_n (which is $R(f_n)$) is the empirical risk $R_{\text{emp}}(f_n)$. Using $R(f^*)$ to denote the risk of the best function in $\mathcal{F}$ (meaning $R(f^*) = \inf_{f \in \mathcal{F}} R(f)$), Question (2) can be formalized as seeking information about the difference $R(f_n) - R(f^*)$. We can observe the following as a partial answer to this question:

$$R(f_n) - R^* = [R(f^*) - R^*] + [R(f_n) - R(f^*)]. \tag{3.9}$$

Interestingly, this formula shows the theoretical underpinnings of the well-known bias-variance trade off, and provides a guiding concept for how to balance our assumptions. The first term on the right-hand side (RHS), called the approximation error, measures how close the risk of the best function in our hypothesis class $\mathcal{F}$ is to the optimal risk. The second term on the RHS, the estimation error, measures how close are the actual risk of our empirically optimal function f_n and the risk of the best possible function in $\mathcal{F}$, f^*.

If we choose a large and complex function class $\mathcal{F}$, the approximation error will be smaller because the class is more likely to contain a function closer to the optimal one. However, this function will be more difficult to estimate from limited data, increasing the estimation error from the second term. On the other hand, a small and low-complexity model class will be easier to learn from data, thus having a lower

estimation error, but may result in higher approximation error. Finding the right balance between these two terms is necessary for our assumptions about the world to lead to the best possible function given the data we have observed. The Regularization principle explicitly balances the two using the hyperparameter λ, and similarly for SRM with $\text{pen}(d, n)$.

The best results, in terms of difference between risk of our chosen function $R(f_n)$ and minimum possible risk R^*, are achieved when our hypothesis class $\mathcal{F}$ is sufficiently large to be able to approximate the optimal function well, (low approximation error) and when we have enough data to reasonably estimate the best function from $\mathcal{F}$ (low estimation error).

Equation (3.9) and the ensuing discussion shed light on one aspect of classification difficulty, which is the underlying complexity of the hypothesis class needed to obtain a low approximation error. If the underlying classification problem demands a complex hypothesis class to accurately approximate it, we must compensate with an appropriately large sample size in order to reduce the estimation error in the right-hand term of the RHS; if we fail to do so, we will see a large difference between the risk (performance) of our learned classification function $R(f_n)$ and the optimal risk that can be achieved for this problem R^*.

Next, we turn to Question (3): how well will the learned function make predictions for new data in the future? This question can formalized by comparing the empirical risk of the function on the data at hand $R_{\text{emp}}(f_n)$, defined in Equation (3.5), to the true risk of the function $R(f_n)$, which is defined via the expectation in Equation (3.1). The difference $R(f_n) - R_{\text{emp}}(f_n)$ provides some degree of assurance on how the model will perform on unseen data compared to how it performed in our initial data. In other words, this difference informs how well the model will *generalize*, presuming that the new data is drawn from the same joint density $P(\text{X}, Y)$ from which the

original data is sampled.

A major focus in statistical learning theory is utilizing convergence techniques from functional analysis to prove bounds on different risk quantities, such as the ones in Equation (3.9) or the risk difference $R(f_n) - R_{\text{emp}}(f_n)$. We present a bound on the actual risk of our selected function $R(f_n)$ relative to the empirical risk $R_{\text{emp}}(f_n)$, in order to provide a partial answer to Question (3); interested readers can look ahead to Equation (3.13) for this bound.

A key component of this bound is the relationship between the number of data points available for learning the function (the sample size, n), and the complexity of the hypothesis class of functions $\mathcal{F}$ from which the function is being learned. We previously saw (informally) via Equation (3.9) that the relationship between these two quantities determines how close the learned function is to the optimal one. We will show in this section that this relationship also determines how well we are able to estimate the true performance of the learned algorithm based off of the performance from our finite sample, which is represented by $R(f_n) - R_{\text{emp}}(f_n)$.

In order to make a formal comparison between sample size and hypothesis class complexity, we need to first define two critical concepts, the growth function and the Vapnik-Chervonenkis (VC) dimension. The VC-dimension provides one way to quantify the capacity, or complexity of a hypothesis class. Loosely, the larger the capacity, the greater the diversity of function assignments that can be made by members of the class.

Let $N^{\mathcal{F}}(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n)$ be defined as:

$$N^{\mathcal{F}}(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n) = \text{card}\{(f(\mathrm{x}_1), f(\mathrm{x}_2), \ldots, f(\mathrm{x}_n)) \mid f \in \mathcal{F}\}$$

Intuitively, $N^{\mathcal{F}}(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n)$ counts the number of distinct assignments of the points into two classes, for the sample $\{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n\} \subset \mathcal{X}$, among all of the functions $f \in \mathcal{F}$.

For example, take $\mathcal{X} = \mathbb{R}^2$, and $\mathcal{F}$ the set of all linear classifiers:

$$\mathcal{F} = \{f \mid f(\mathrm{x}) = \mathrm{sign}(\mathrm{w}^T\mathrm{x} + \beta), \beta \in \mathbb{R}, \mathrm{w} \in \mathbb{R}^2\}. \tag{3.10}$$

Then for the three points in Figure 3.1 we have $N^{\mathcal{F}}(\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3) = 8$.

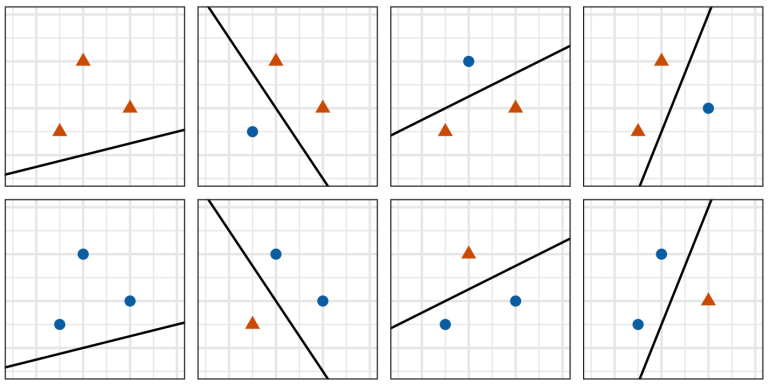

Figure 3.1: Three points in $\mathbb{R}^2$ can be classified in 8 ways by the set of linear classifiers.

If $\mathcal{F}$ is a set of classification functions for a binary classification problem, we immediately observe that $N^{\mathcal{F}}(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n) \leq 2^n$ for all sets of points $\{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n\}$. When $N^{\mathcal{F}}(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n) = 2^n$, that means that $\mathcal{F}$ can generate any classification assignment on this set of points. In this case we say that $\mathcal{F}$ shatters $\{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n\}$.

The *growth function* $G^{\mathcal{F}}(n)$, also known as the shatter coefficient, is defined as:

$$G^{\mathcal{F}}(n) = \sup_{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n} N^{\mathcal{F}}(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n)$$

In other words, $G^{\mathcal{F}}(n)$ is the maximum number of class assignments possible from functions in $\mathcal{F}$,with the maximum taken over all subsets of size n.

In binary classification, we have $G^{\mathcal{F}}(n) \leq 2^n$ for any $\mathcal{F}$. The *Vapnik-Chervonenkis dimension*, or VC dimension, h, is the largest integer n such that $G^{\mathcal{F}}(n) = 2^n$, or such that $\mathcal{F}$ shatters at least one set of n points.

Going back to the example from Figure 3.1, we see that $\mathcal{F}$ can shatter 3 points (because there are $2^3 = 8$ different class assignments from the set of all linear classifiers). However, $\mathcal{F}$ cannot shatter 4 points because there is no linear classifier that

can produce the class assignment in Figure 3.2, the classic XOR problem. The VC dimension of this hypothesis class is 3.

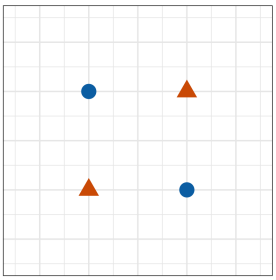

Figure 3.2: There is no linear classifier that can classify the 4 points above into the class assignments shown.

We provide one more example on VC dimension, using the hypothesis class of polynomial classifiers of degree m in $\mathbb{R}^n$. First we rigorously define the set of polynomial classifiers using notation from Anthony (1995). Let the set $\{1, 2, \ldots, n\}$ be denoted $[n]$, and let $[n]^m$ denote the set of all subsets of $[n]$ with at most m elements, with repetition allowed. As an example,

$$[3]^2 = \{\varnothing, \{1\}, \{2\}, \{3\}, \{1,1\}, \{1,2\}, \{1,3\}, \{2,2\}, \{2,3\}, \{3,3\}\}.$$

For each $S \in [n]^m$ and each $\mathrm{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$, the notation x_S denotes the product of x_i for $i \in S$. For example, $x_{\{1,3\}} = x_1 x_3$ and $x_{\{2,2\}} = x_2^2$.

With this notation, a polynomial classifier$f$ is a function $f : \mathbb{R}^n \to \{0, 1\}$ where there are constants w_S (one for each $S \in [n]^m$) such that

$$f(\mathrm{x}) = 1 \iff \sum_{S \in [n]^m} \mathrm{w}_S \mathrm{x}_S > 0.$$

The set of polynomial classifiers in $\mathbb{R}^n$ of degree at most m is denoted $P(n, m)$. Intuitively, polynomial classifiers extend the concept of separating hyperplanes (for linear classifiers) to separating hypersurfaces that are defined by a polynomial equation.

Anthony (1995) proves the following theorem about the VC dimension of $P(n, m)$:

Theorem 1. *For all n, m,*

$$VC\ dim\ (P(n,m)) = \begin{pmatrix} n+m \\ m \end{pmatrix}. \tag{3.11}$$

This theorem supports our earlier assertion that linear classifiers (polynomial classifiers of degree 1) in R^2 have VC dimension 3:

$$\begin{pmatrix} 2+1 \\ 1 \end{pmatrix} = \frac{3!}{1!\,2!} = 3.$$

Furthermore, a polynomial classifier of degree m in $\mathbb{R}^2$ has VC dimension

$$\text{VC dim } (P(2,m)) = \frac{(m+2)(m+1)}{2}. \tag{3.12}$$

As a concrete example, consider the three polynomials in $\mathbb{R}^2$ pictured in Figure 3.3. Moving from left to right in the figure, the polynomials are of degree 3, 5, and 11. Note that this 11-degree polynomial has fewer inflection points (5) than the maximum possible for a polynomial in $\mathbb{R}^2$ of degree 11 (9); however, the higher degree is reflected in different curvature than would be possible with a degree 7 polynomial.

The hypothesis classes of polynomial classifiers in $\mathbb{R}^2$ of degrees 3, 5 and 11 have VC dimension 10, 21, and 78, respectively, as can be easily calculated from Equation (3.12). The greater capacity of the hypothesis classes with higher VC dimension are reflected in how "wiggly" or complex the polynomials contained in those hypothesis classes are. The hypothesis class with a higher VC dimension contains functions covering a greater diversity of classification decisions.

These three particular polynomials will come into play during a simulated data example explored in Section 3.5.

Now that we have a notion for quantifying the capacity of a hypothesis class, or more intuitively, the complexity of the functions contained in that class, we can present one of the key bounds on the empirical risk (Bousquet *et al.* (2004)):

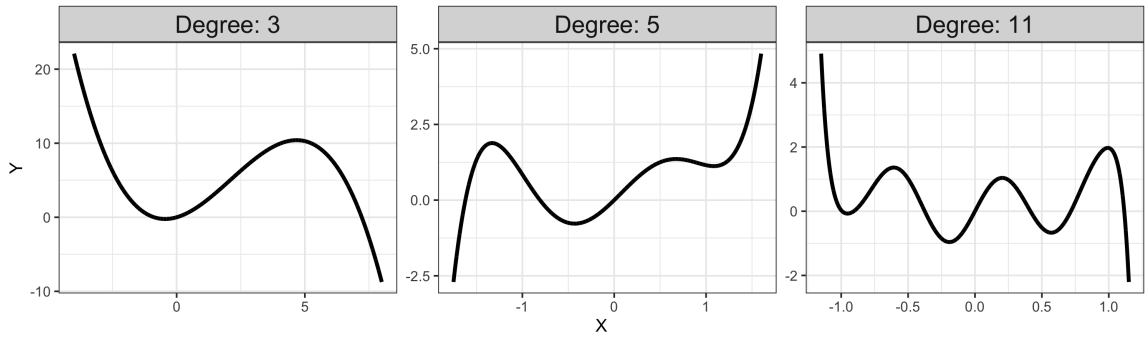

Figure 3.3: Polynomials of degree 3, 5, and 11. Their respective hypothesis classes have VC dimension 10, 21, and 78, respectively.

Theorem 2. *Let $\mathcal{F}$ have VC dimension h. Then with probability at least $1-\delta$,*

$$\forall f \in \mathcal{F}, R(f) \leq R_{emp}(f) + 2\sqrt{2\frac{h \log \frac{2en}{h} + \log \frac{2}{\delta}}{n}} \tag{3.13}$$

To develop greater intuition, we inspect how the right summand of the right-hand side changes when we change individual components. If δ becomes small holding all else constant, meaning we want to be more certain that our bound on the actual risk is true, then the bound becomes less tight and that bounding term becomes larger. (This is akin to the notion in Bayesian inference that if we want the credible interval to contain the estimand with higher probability, the credible interval must become wider.)

The ratio n/h, which represents the ratio of how much data we have to estimate f^* compared to the complexity of the hypothesis class $\mathcal{F}$, determines the size of the left term under the square root, via the term

$$\frac{\log\left(2e\frac{n}{h}\right)}{\frac{n}{h}}. \tag{3.14}$$

When the ratio n/h is large, meaning we have much more data than the capacity of $\mathcal{F}$, then this term becomes very small and the empirical risk is close to the actual risk, with probability $1-\delta$. However, if n/h is small, meaning we are trying to estimate a function from a higher capacity hypothesis class with a small amount of

data, we have worse guarantees on the difference between the empirical risk and the actual risk, and the bound in (3.13) can become large. If the ratio n/h becomes small enough (in particular, close to 0), the term in (3.14) limits to $-\infty$ and the bound is only valid with commensurate decreases in δ, leading to extremely low probabilities of the bound holding.

This bound demonstrates from a theoretical standpoint how decision boundary complexity relates to the difficulty of a classification problem, from the perspective of predicting the performance of a learned algorithm. When the underlying classification problem warrants a high-capacity hypothesis class for accurate approximation (meaning, high decision boundary complexity), there is a proportionate increase in the amount of data needed to guarantee (with any practical usefulness) our estimate of the learned function's true risk. If we do not use appropriate amounts of data for learning high-capacity hypothesis classes, we have a weaker, and in some cases irrelevant in practical terms, guarantee on how the learned algorithm will perform on future data from the same distribution. In practice, the performance of complex functions fitted to small data sets on new data is usually poor (Berisha *et al.* (2021)).

While VC dimension is useful when available, in practice it is difficult to compute, although some work has been done on VC dimensions of more complex hypothesis classes, for example neural networks (Sontag *et al.* (1998), Bartlett *et al.* (2019)).

As a conclusion to our review of relevant concepts from statistical learning theory, we briefly mention other key concepts and related fields, and discuss how they address the three questions presented at the beginning of this section. These ideas expand Vapnik and Chervonenkis' foundational work in creating a rigorous set of mathematical structures within which the concept of learning about the world from data can be comprehended.

Lacking from the statistical-learning theory framework developed by Vapnik are

guarantees on the ability of a specific learning algorithm A to actually learn the optimal function f from the hypothesis class $\mathcal{F}$, given the data and the chosen optimization principle (e.g., ERM, etc.). Valiant (1984) introduced the *Probably Approximately Correct (PAC)* framework, in which he shows that there are certain *concept classes* (a process that determines whether a concept, or predicate, holds for a particular datum), which can probably (i.e. with hig probability) be accurately learned (i.e., are approximately correct) in polynomial time, by a learning algorithm. The emphasis here is on the learning algorithm, and its complexity in terms of execution time. A hypothesis class being PAC-learnable is equivalent to having a finite VC dimension (Blumer *et al.* (1989)), thus PAC provides as an alternative notion for hypothesis class complexity. Topics in PAC are most closely connected to Question (1), in particular, informing how quickly and with what guarantees can we create models that learn about the world. Other key works related to PAC learning theory include Angluin (1988), Haussler (1990), Kearns and Vazirani (1994).

Aside from VC dimension, other alternatives for measuring capacity, or complexity, of a hypothesis class are the *Rademacher complexity*, the *pseudo-dimension* or *Pollard dimension*, and the *fat-shattering dimension*. Each of these areas yields insights that are loosely connected to Questions (2) and (3), as the capacity of a hypothesis class impacts how close the learned function can be to the optimal one, and how well we can estimate the model's performance on future data using a given sample size. Rademacher complexity, first proposed as a complexity measure in Koltchinskii (2001), Bartlett *et al.* (2002), and Mendelson (2002), is an alternative to VC dimension characterized by the ability of functions from the hypothesis class to classify points from the space with random labels. Rademacher complexity was further explored in Bartlett and Mendelson (2002) and Bartlett *et al.* (2005). The fat-shattering dimension(cites: Kearns and Schapire (1990), Bartlett *et al.* (1994), Bartlett (1996),

Alon *et al.* (1997)) and Pollard dimension are both extensions of VC dimension to real valued functions.

We also mention the two related concepts of *stability* and *robustness*. Intuitively, a learning algorithm exhibits stability if a nearly identical training set with a single point removed will produce a similar optimal function using the given learning algorithm. Stability in the context of learning algorithms was first introduced in Rogers and Wagner (1978), Devroye and Wagner (1979), and Devroye and Wagner (1979), and furthered by Bousquet and Elisseeff (2002), Poggio *et al.* (2004), and Mukherjee *et al.* (2006). Robustness, introduced by Xu and Mannor (2012), requires that a testing sample similar to a training sample will have similar performance on the learned function obtained from the learning algorithm applied to data. Both stability and robustness are useful properties of learning algorithms and can be used to improve bounds relating empirical to actual risk, providing additional answers to Question (3); these bounds are typically derived via functional analysis methods of uniform convergence.

To summarize, statistical learning theory offers a theoretical foundation for formalizing the process of learning about the world through data. This body of literature provides information on how close the performance of a model learned from data is to the best performance that could theoretically be achieved for the particular classification problem. It furthermore provides theoretical bounds on the accuracy of the performance estimates obtained for the learned algorithm in a finite sample, in other words, how likely is the performance obtained in the original data to generalize to unseen data.

The impact of decision boundary complexity on classification complexity is also clarified through statistical learning theory's formalisms. In particular, a classification problem with higher underlying decision boundary complexity requires a higher ca-

pacity hypothesis class in order to accurately approximate the decision boundary; this higher capacity hypothesis class in turn demands a commensurate increase in sample size, in order for the screening test to achieve performance close to the best possible performance, and in order for practitioners to adequately estimate performance on unseen data. High decision boundary complexity, combined with small sample sizes n for obtaining the learned model, will result in poor performance relative to the optimal performance for the classification problem regardless of hypothesis class, along with poor model generalization if a high capacity hypothesis class is nonetheless used.

3.3 Information Theoretic Divergence Measures

While statistical learning theory centers on hypothesis classes and the challenges posed by complex decision boundaries, we turn to information theoretic divergence measures for function-agnostic information on a separate aspect of classification complexity: the overlap of the two classes in the joint probability distribution. Class overlap is recognized as one of the most important aspects underlying classification difficulty (Santos *et al.* (2023)). The overlap of the classes determines the limit of the maximum performance that any learning algorithm from any hypothesis class can achieve for that problem, as will be demonstrated shortly.

The section is roughly divided into three parts. In the first part, we define the Bayes error rate and show how class overlap impacts classification difficulty via Bayes error rate. In the second part, we review literature on criteria that can be used for measuring the divergence between two distributions; these divergence measures have a close relationship with class overlap. In the third part, we provide the definition and relevant examples of one specific divergence measure, the Kullback-Leibler (KL) divergence. As a look ahead, in Section 3.5, we will use the KL-divergence to quantify the class overlap and show the clear connection between KL-divergence and the

simulation hyperparameter used to control the class overlap of the simulated data. Finally, we close the section with a comparison of the themes underpinning work from statistical learning theory and information theory.

To begin the first part, we present key definitions. Formally, we have speech features X collected from participants that fall under two class labels, $Y = 1$ (cognitively impaired) or $Y = 0$ (cognitively normal). Their joint probability definition is $P(\mathrm{X}, Y)$. When screening a participant with speech features $\mathrm{X} = \mathrm{x}$, the predicted class and thus the screening decision is usually based on an estimate of the probability $P(Y = 1 \mid \mathrm{x})$. The *Bayes' classifier* for a binary classification problem is the classifier f_{Bayes} that assigns a class based on the following rule:

$$f_{\text{Bayes}}(\mathrm{x}) = \begin{cases} 1 & \text{if } P(Y = 1 \mid \mathrm{x}) > P(Y = 0 \mid \mathrm{x}) \\ 0 & \text{otherwise.} \end{cases} \tag{3.15}$$

The error rate of the Bayes classifier is called the *Bayes error rate* (BER), and it is equal to the risk of the 0-1 loss function for the Bayes classifier:

$$\epsilon_{\text{Bayes}} = \int_{R_1} P(Y = 0 \mid \mathrm{x}) d\mathrm{x} + \int_{R_0} P(Y = 1 \mid \mathrm{x}) d\mathrm{x}, \tag{3.16}$$

where R_1 is the region where $P(Y = 1 \mid \mathrm{x}) > P(Y = 0 \mid \mathrm{x})$, and similarly for R_0. Devroye *et al.* (1996) showed that the BER is the minimum achievable error rate for the given classification problem. The higher the BER, the more difficult the classification problem.

To make the connection between class overlap and BER explicit, consider the following, which follows directly from Bayes' rule:

$$P(Y = 1 \mid \mathrm{x}) = \frac{f(\mathrm{x} \mid Y = 1) P(Y = 1)}{f(\mathrm{x})};$$

$$P(Y = 0 \mid \mathrm{x}) = \frac{f(\mathrm{x} \mid Y = 0) P(Y = 0)}{f(\mathrm{x})}.$$

For ease of notation, we will use the shorthand $f_1(\mathrm{x})p_1$ for $f(\mathrm{x} \mid Y = 1)P(Y = 1)$, and similarly for $f_0(\mathrm{x})p_0$. In order to achieve a BER of 0, perfect separation of the conditional distributions is necessary, since f_1, f_0, p_1, and p_0 are nonnegative; in other words, we would need

$$\int_{R_1} f_0(\mathrm{x})p_0 \, d\mathrm{x} = 0 \qquad \text{and} \qquad \int_{R_0} f_1(\mathrm{x})p_1 \, d\mathrm{x} = 0.$$

On the flip side, the BER will be high when $f_0(\mathrm{x})p_0$ has a large integral over the region R_1, and similarly for $f_1(\mathrm{x})p_1$ over R_0. We use the term *class overlap* to denote the "overlap" of the conditional densities $f_0(\mathrm{x})p_0$ and $f_1(\mathrm{x})p_1$. The higher this overlap, the larger the integrals in (3.16) will be. The smaller this overlap, the lower the BER is and the easier the classification problem is.

This exposition highlights why measuring the distance, or divergence, between probability distributions can provide insights into the difficult of a classification problem in terms of theoretically optimal misclassification rates. More specifically, measures of divergence can be used to provide bounds on the BER. Thus, we are interested in quantitative measures of divergence, and for this we turn to information theory.

In this second part of the section, we present the foundational literature defining a critical class of divergence measures, called f-divergences, and discuss some applications of f-divergences in the classification setting. The seminal text in this area is Ali and Silvey (1966), in which the authors established four criteria that they posited would be useful and natural for a quantity measuring divergence or distance between two probability distributions to satisfy. Ali and Silvey termed functions that satisfied their criteria *coefficients of divergence* in their 1966 paper; they are now known as *Ali-Silvey distances* or *f-divergences*.

From Ali and Silvey (1966), let P_1 and P_2 be two probability measures on the same sample space $(\mathscr{X}, \mathscr{F})$. The four criteria proposed were (Ali and Silvey (1966)):

(1) The coefficient $d(P_1, P_2)$ should be defined for all pairs of measures P_1 and P_2 on the sample space.

(2) Suppose that $y = t(x)$ is a measurable transformation from $(\mathscr{X}, \mathscr{F})$ onto a measure space $(\mathscr{Y}, \mathscr{G})$.Then we should have

$$D(P_1, P_2) \geq d(P_1 t^{-1}, P_2 t^{-1}),$$

where P_i^{-1} denotes the induced measure on $\mathscr{Y}$ corresponding to P_i.

(3) $d(P_1, P_2)$ should take its minimum value when $P_1 = P_2$ and its maximum value when $P_1 \perp P_2$.

(4) Let θ be a real parameter and let $\{P_\theta; \theta \in (a, b)\}$ be a family of equivalent (mutually absolutely continuous) distributions on the real line such that the family of densities $p_\theta(x)$ with respect to a fixed measure μ has monotone likelihood ratio in x. Then if $a < \theta_1 < \theta_2 < \theta_3 < b$, we should have

$$d(P_{\theta_1}, P_{\theta_2}) \leq d(P_{\theta_1}, P_{\theta_3}).$$

The paper proceeds to show that particular measures of divergence, distance, or discrimination, including Jeffrey's divergence (Jeffreys (1946)), Kullback-Leibler divergence (Kullback and Leibler (1951)), and Hellinger distance (Hellinger (1909)), satisfy these properties and are indeed f-divergences. Furthermore, and of interest to our work, they showed the connection between one of the f-divergences and the probability of misclassification in a binary classification problem.

The family of f-divergences have been studied and used extensively in a number of applications, including estimation, classification, detection, compression, database indexing (Basseville (2013)). A comprehensive taxonomy of f-divergences along with other distance measures between probability distributions is given in Cha (2007), with

Basseville (2013) providing an alternative and more recent summary. The family of f-divergences are of particular interest in machine learning, as they can be used to derive bounds on the BER, also termed the *Bayes risk*.

For example, Berisha *et al.* (2016) define a nonparametric divergence measure, the D_p distance, that improves the bounds on the Bayes risk, compared to frequently used f-divergences such as the Bhattacharya distance (Bhattacharyya (1946)). Furthermore, they propose a feature selection method based on D_p distance that improves classification accuracy on a set of pathological speech samples.

Having introduced the concept of f-divergences and their relation to class overlap, classification complexity, and related applications, we now turn to the third part of this section, namely an in-depth example using the Kullback-Leibler divergence. The Kullback-Leibler divergence, or KL divergence, has deep connections to statistics, machine learning, and Bayesian inference. For two probability distributions P and Q of a continuous random variable x, with probability densities p and q, respectively, the KL divergence or relative entropy from Q to P is defined as

$$D_{KL}(P \mid\mid Q) = \int_{-\infty}^{\infty} p(x) \ln \left(\frac{P(x)}{Q(x)} \right) dx \tag{3.17}$$

The KL divergence of two distributions P and Q over a random variable X is the expected value of the log likelihood ratio statistic if X is actually drawn from P. In Bayesian inference, under the assumption Q is the prior probability distribution and P is the posterior, $D_{KL}(P \mid\mid Q)$ is a measure of the information gained by revising beliefs, after conditioning on data, from Q to P.

For an intuitive example of KL divergence, we consider the case of two multivariate Gaussian distributions P_1 and P_2 over $\mathbb{R}^n$ with mean vectors μ_1, μ_2 and covariance matrices Σ_1, Σ_2, respectively. The KL divergence $D_{KL}(P_1 \mid\mid P_2)$ is (Duchi (2007)):

$$D_{KL}(P_1 \mid\mid P_2) = \frac{1}{2} \left(\ln \frac{\det \Sigma_2}{\det \Sigma_1} - n + \mathrm{tr}(\Sigma_2^{-1} \Sigma_1) + (\mu_2 - \mu_1)^T \Sigma_2^{-1} (\mu_2 - \mu_1) \right) \tag{3.18}$$

Figure 3.4 shows two examples of two multivariate Gaussian distributions with different amounts of overlap. For the example on the left,

$$P_1 \sim \mathcal{N}(\mu_1 = (1,1)^T, \Sigma_1 = I_2);$$

$$P_2 \sim \mathcal{N}(\mu_2 = (4,4)^T, \Sigma_2 = I_2),$$

where I_2 is the identity matrix of size 2. For the example on the right,

$$P_1 \sim \mathcal{N}(\mu_1 = (2,2)^T, \Sigma_1 = I_2);$$

$$P_2 \sim \mathcal{N}(\mu_2 = (3,3)^T, \Sigma_2 = I_2).$$

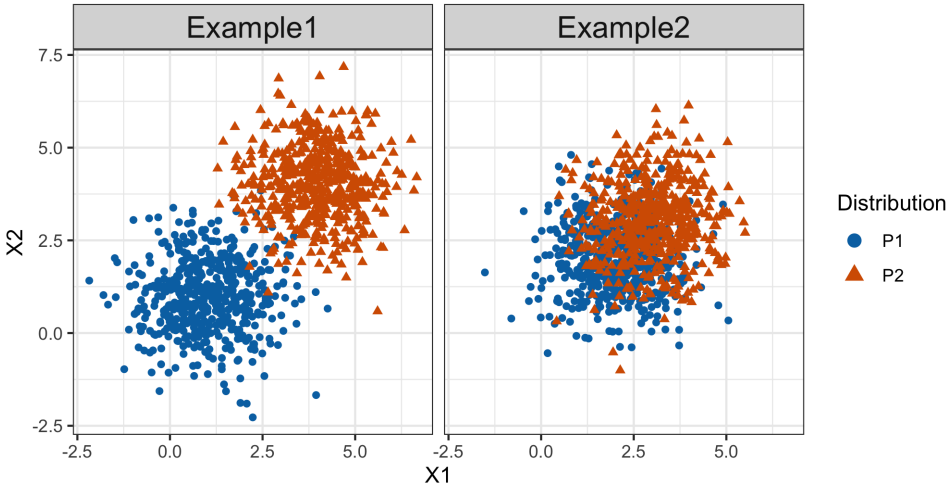

Figure 3.4: Two different examples of the KL divergence between two multivariate Gaussian distributions, P_1 and P_2.

Using the formula in (3.18), we have

$$\begin{aligned} D_{KL}(P_1 \,||\, P_2) &= \frac{1}{2}\left(\ln\frac{\det(I_2)}{\det(I_2)} - 2 + tr(I_2^{-1}I_2) + (3,3)^T I_2^{-1}(3,3)\right) \\ &= \frac{1}{2}\left(\ln(1) - 2 + 2 + (3,3)^T(3,3)\right) \\ &= \frac{1}{2}(18) \\ &= 9, \end{aligned}$$

and similarly

$$\begin{aligned} D_{KL}(P_2 \,||\, P_1) &= \frac{1}{2}\left(\ln\frac{\det(I_2)}{\det(I_2)} - 2 + tr(I_2^{-1}I_2) + (-3,-3)^T I_2^{-1}(-3,-3)\right) \\ &= \frac{1}{2}\left((-3,-3)^T(-3,-3)\right) \\ &= 9. \end{aligned}$$

By a similar calculation, the example in the right plot of Figure 3.4 satisfies

$$D_{KL}(P_1 \,||\, P_2) = D_{KL}(P_1 \,||\, P_2) = 1.$$

From this simple example we see that the distributions with much less overlap have a larger KL divergence.

We look at one more example of KL divergence, for two distributions that do not have a symmetric KL divergence. In Figure 3.5, we see samples from two distributions P_1 and P_2 which are defined as:

$$P_1 \sim \mathcal{N}\left(\mu_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \Sigma_1 = \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}\right)$$

$$P_2 \sim \mathcal{N}\left(\mu_2 = \begin{pmatrix} 4 \\ 4 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 3/2 & 0 \\ 3/2 & 3 \end{pmatrix}\right).$$

To calculate the KL divergences, we first note that $\det(\Sigma_1) = 8$, $\det(\Sigma_2) = 4.5$, and

$$\Sigma_1^{-1} = \begin{pmatrix} 1/4 & 0 \\ 0 & 1/2 \end{pmatrix}, \qquad \Sigma_2^{-1} = \begin{pmatrix} 2/3 & 0 \\ -1/3 & 1/3 \end{pmatrix}.$$

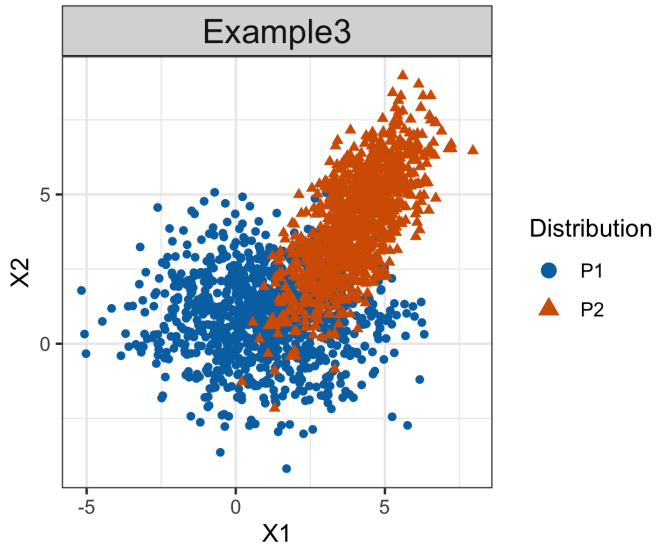

Figure 3.5: An example of two multivariate Guassian distributions P_1 and P_2 for which the KL divergence is not symmetric. This is generally the case for the KL divergence of two distributions.

Then we have

$$\begin{aligned} D_{KL}(P_1 \,||\, P_2) &= \frac{1}{2}\left(\ln\frac{\det(\Sigma_2)}{\det(\Sigma_1)} - 2 + tr(\Sigma_2^{-1}\Sigma_1) + (3,3)\Sigma_2^{-1}(3,3)^T\right) \\ &= \frac{1}{2}\left(\ln\frac{4.5}{8} - 2 + \left(\frac{8}{3}+\frac{2}{3}\right) + (3,3)(2,0)^T\right) \\ &= \frac{1}{2}\left(\ln\frac{4.5}{8} - 2 + \frac{10}{3} + 6\right) \\ &\approx 3.54. \end{aligned}$$

On the other hand,

$$\begin{aligned} D_{KL}(P_2 \,||\, P_1) &= \frac{1}{2}\left(\ln\frac{\det(\Sigma_1)}{\det(\Sigma_2)} - 2 + tr(\Sigma_1^{-1}\Sigma_2) + (-3,-3)\Sigma_1^{-1}(-3,-3)^T\right) \\ &= \frac{1}{2}\left(\ln\frac{8}{4.5} - 2 + \left(\frac{3}{8}+\frac{3}{2}\right) + (-3,-3)(-3/4,-3/2)^T\right) \\ &= \frac{1}{2}\left(\ln\frac{8}{4.5} - 2 + \frac{15}{8} + \frac{9}{4} + \frac{9}{2}\right) \\ &\approx 3.43. \end{aligned}$$

Indeed, we see that in this third example, the distributions are partially overlapped,

more than in Example 1 (left plot in Figure 3.4), which has KL divergence of 9, but less than Example 2 (right plot in Figre 3.4), which has KL divergence of 1.

In this section, we have focused on a very narrow area within information theory, namely the information theoretic divergence measures that have a strong relationship to class overlap and, as a result, to classification complexity. While there are numerous other works discussing f-divergences and information theory more generally, we have kept the scope of this section more focused in order to present some fundamental definitions most related to our use case.

To summarize, class overlap comprises a second aspect, and in many ways the most important one, underpinning the difficulty of the classification problem. The connection between class overlap and classification difficulty comes via the Bayes error rate, as higher class overlap results in a higher BER, and BER determines the best performance that can theoretically be achieved on the classification problem. Divergence measures, particularly f-divergences, can be used as a way to quantify class overlap and to bound the BER; here we have used the KL-divergence as a particular example of an f-divergence.

We close with a final comparison between the results presented in the previous section on statistical learning theory, and the information-theoretic bounds discussed here. The VC-dimension-based bound in Equation (3.13) provides guarantees on how accurate is our estimate of the *risk* of an estimated function we have in hand, after already applying an optimization principle (ERM, SRM, etc.) to data. On the other hand, the divergence-based bounds set limits on the theoretically minimum error rate that is possible to achieve for *any* function, including our learned one. A crude but useful generalization is that one theory (SLT) provides insights into how well we can learn a theoretically optimal function using a finite sample, along with how well we can estimate that function's performance on new data; the other (IT) provides

insights into just how good that optimal function can be.

In relation to our aim of understanding contributors to classification complexity, statistical learning theory offers insights on classification complexity via the complexity of the decision boundary; decision boundary complexity dictates how our algorithm's performance compares to optimal, and how well we can estimate its performance using a finite sample. Information theory, on the other hand, connects classification complexity to the overlap between the two classes being separated in the classification problem.

3.4 Measures of Complexity of Classification Problems

Finally, we review a body of work on purely empirical classification measures, which aim to measure difficulty of the classification problem using quantitative measures that are calculated on the specific sample.

A downside of the theoretical results presented in the previous two sections is that they can be difficult or impossible to calculate for a given sample, a proposed hypothesis class, and a learning algorithm; even estimating information-theoretic divergence measures can be computationally costly, limiting their adoption in practical applications. In contrast, empirical data complexity measures are appealing because they provide a straightforward and computationally feasible approach for quantifying aspects of a particular dataset that relate to classification performance. However, as empirical quantities, they suffer from the same limitations that apply to any finite-sample estimator, including variability (caused by sampling differences) that increases with fewer and fewer datapoints. Furthermore, data complexity measures are susceptible to unexpected behavior in high-dimensional settings, similarly to standard learning algorithms that were originally developed and validated in $n >> p$ scenarios (Berisha *et al.* (2021)). The relationship between the complexity measures and data

dimensionality is further discussed at length in Section 3.6. Nonetheless, they present an alternative method for quantifying classification complexity, and here we provide a review that discusses both their promise and their limitations.

As an overview, this section is also divided into three parts. In the first part, we describe several sets of data complexity measures, along with related work on empirical quantification of decision boundary complexity via topological data analysis. In the second part, we discuss how the standard data complexity measures just introduced relate to the concepts of decision boundary complexity and class overlap. In the third and final part, we present modern-day applications that make use of data complexity measures, and discuss the appropriateness of these use cases.

To begin the first part, we start with the foundational paper in this area, which is Ho and Basu (2002). The authors introduced the concept of classification difficulty of a dataset via 12 *measures of classification complexity*, and grouped these measures in three areas: measures of individual feature overlap, measures of separability of classes, and measures related to the geometry and topology of the class manifolds.

Ho and Basu (2002) computed the measures on more than 800 datasets from the UCI database, (divided into linearly separable and linearly non-separable datasets), along with 100 datasets of increasing sparsity with randomly chosen labels, to analyze the relationship between their proposed complexity measures on these datasets. The question they sought to answer was whether there existed an underlying continuum along one or more of these complexity measures, or a lower dimensional projection of them, which would yield one unified concept of "classification complexity". This question was explored via extensive analysis of their datasets (with three groups of increasing complexity, namely linearly separable, which was less complex than non-linearly separable, which was less complex than the random datasets). They found that there was not one underlying continuum that neatly tied all of the datasets in a

single complexity ordering along a lower-dimension, but still found rich structures in the relationship of the complexity measures for these three subgroups of data.

In subsequent years, these measures of classification complexity, along with new measures not defined in the original 2002 paper (re-branded in conjunction as *measures of data complexity*), found their way into numerous applications, which were neatly summarized in a comprehensive review by Lorena *et al.* (2019). The authors grouped this wider set of classification measures slightly differently than Ho and Basu (2002), separating them into five groups: feature-based measures, linearity measures, neighborhood measures, network measures, and class imbalance measures.

One drawback of the groupings of these measures chosen in Ho and Basu (2002), Lorena *et al.* (2019) and others, are that the groupings do not explicitly relate to the aspect of dataset complexity the measure is providing insights about, but are more related to how the measure is calculated. Santos *et al.* (2023) provided a fresh organization for complexity measures related to class overlap, including some of the original measures from Ho and Basu (2002), along with others. They group their 33 measures of complexity based on the aspect of class overlap about which the measure provides information: feature overlap (overlap of individual features, possibly projected into new dimensions), instance overlap (analysis of a local neighborhood around each individual point), structural overlap (boundary complexity and data morphology, i.e. the internal structure of classes), and multiresolution overlap (measures that recursively assess both global and local complexity information for a combined picture of overlap). The individual 33 measures are described in Santos *et al.* (2022).

In both Santos *et al.* (2022) and Santos *et al.* (2023), the authors perform a special discussion on the separate but important issue of class imbalance, which can exacerbate the not insignificant challenges that class overlap on its own presents.

In our experiments, we calculate the measures of complexity from Lorena *et al.*

(2019), rather than Santos *et al.* (2023) or others, for several reasons. First, the measures from Lorena *et al.* (2019) are available in a maintained and vetted R package `ECoL` (Garcia and Lorena (2019)). Second, the measures have been normalized and had their directionality reversed where necessary, so that they are all computed on a scale from 0 to 1, with 1 being the greatest complexity.

The chosen measures and their descriptions are listed in Table 3.1; we also show the type of insight provided into overlap or complexity, using the terminology given in Santos *et al.* (2023). For the measures not discussed in Santos *et al.* (2023), we list the most appropriate insight, and add a new category *linear complexity* for the linear measures. We also rename *Structural Overlap* to *Structural Overlap & Complexity*, since the measures in this group relate to both the internal structure and topology of the classes (which directly impacts decision boundary complexity) as well as the class overlap itself. Finally, we do not describe the class imbalance measures, as they are not computed in `ECoL`.

For detailed descriptions of these measures and their calculation, see Lorena *et al.* (2019).

We now take a moment to briefly mention relatied advances in topological data analysis (TDA) (Carlsson (2009)), which seek to characterize decision boundary complexity and learning algorithm capacity using mathematical formalisms from topology and algebraic geometry; see, for example, Ramamurthy *et al.* (2019), Guss and Salakhutdinov (2018), and Rieck *et al.* (2018). The general approach is to calculate a *persistent homology*, and use the homology to quantify topological properties, such as number of connected components or number of holes. This quantification can then be applied in service of learning problems, such as choosing a learned algorithm from a model marketplace whose capacity matches the complexity in a given dataset (Ramamurthy *et al.* (2019), Li *et al.* (2020)).

Grouping	Abbr.	Measure	Description	Insight Provided
Feature-based measures	F1	Maximum Fisher's Discriminant Ratio	Determines the maximum discriminative power among each individual feature.	Feature overlap
	F1v	Directional Vector Maximum Fisher's Discriminant Ratio	Determines the data projection with maximum separability.	Feature overlap
	F2	Volume of Overlapping Region	Measures the volume of the overlapping region by multiplying the overlap range of each feature.	Feature overlap
	F3	Maximum Individual Feature Efficiency	Determines the minimum amount of overlap between feature values of different classes.	Feature overlap
	F4	Collective Feature Efficiency	Returns the ratio of examples that could not be separated, using the efficiency of all features.	Feature overlap
Linearity measures	L1	Sum of the Error Distance by Linear Programming	Measures the sum of the distances of incorrectly classified examples to a linear boundary used in their classification.	Linear complexity
	L2	Error Rate of a Linear Classifier	Computes the error rate of the linear SVM classifier.	Linear complexity
	L3	Non-Linearity of a Linear Classifier	Measures the linear error on a set of new synthetic examples generated by interpolating pairs of data examples from the same class, chosen randomly.	Linear complexity
Neighborhood measures	N1	Fraction of Borderline Points	Measures the proportion of examples that are connected to the opposite class by an edge in a Minimum Spanning Tree.	Structural overlap & complexity
	N2	Ratio of Intra/Extra Class Nearest Neighbour Distance	Computes the ratio between the sum of intra-class distances and the sum of extra-class distances.	Structural overlap & complexity
	N3	Error Rate of the Nearest Neighbour Classifier	Measures the error rate of the Nearest Neighbour classifier (1NN), estimated using Leave-One-Out cross-validation.	Instance overlap
	N4	Non-linearity of the Nearest Neighbour Classifier	Measures the 1NN error on a set of new synthetic examples generated the same way as in L3.	Instance overlap
	T1	Fraction of Hyperspheres Covering Data	Computes the ratio of the number of hyperspheres of the same class necessary to cover the data domain, compared to the number of points.	Structural overlap & complexity
	LSC	Local Set Average Cardinality	Determines, for each point, the cardinality of the local set of neighbors closer to it than its nearest enemy; then averages over all points in data.	Structural overlap & complexity
Network measures	Density	Average Density of the Network	Measures the number of edges retained in the graph built from the dataset, normalized by the maximum number of edges between n pairs of data points.	Structural overlap & complexity
	ClsCoef	Clustering Coefficient	Measures the ratio of the number of edges between the neighbors of each point and the maximum number of edges that could possibly exist between them.	Structural overlap & complexity
	Hubs	Hub Score	Scores each node by the number of connections it has to other nodes, weighted by the number of connections these neighbors have.	Structural overlap & complexity

Table 3.1: Data complexity measures from Lorena *et al.* (2019).

A benefit of the TDA approach to classification complexity is that it enjoys a rigorous mathematical formulation, which lends itself to theoretical results on the resulting empirical quantities. To our knowledge, the lack of theoretical results on the empirical measures described in Ho and Basu (2002), Lorena *et al.* (2019), and Santos *et al.* (2022) is a deficiency in the field to date, although this has not hindered the widespread adoption of these measures in many applications.

As further discussion and exploration of TDA-based work would require extensive definitions and formulations, we believe it is outside the scope of this work, and restrict the present analysis on simulated data (section 3.5.4) and speech data (section 4.3.3) to the classification measures listed in Table 3.1.

For the second part of this section, we discuss how the complexity measures from Table 3.1 relate to decision boundary complexity and class overlap, which were extensively discussed in the last two sections on statistical learning theory and divergence measures. F1, F1v, F2, F3, and F4 are related to class overlap, or to the separability of the two classes; however, F1, F1v, and F3 are limited in that this overlap is considered feature-by-feature, rather than looking at the overlap between the conditional joint distributions over the entire feature space. F2 and F4 consider features in tandem, but with some limitations: for F2, the defined overlap regions are parallel to the individual feature axes, and F4 relies on subsequent feature discrimination.

As a note, univariate complexity measures such as F1 or F3 are asymmetrical in the information they provide on data complexity. If a dataset has a single feature that has a high value of Fisher's discriminant ratio (F1) or low class overlap (F3), then the dataset itself has lower complexity in these areas. However, if F1 or F3 indicate high complexity, it may be because there is high class overlap when considering all of the features both individually and collectively, or because the features are individually not separable but do provide separability via (possibly complex) interactions.

L1 and L2 relate to both decision boundary complexity and class overlap; if these measures are low, it means that there is small class overlap, and furthermore the decision boundary is linear (or can be closely approximated by a linear function). If these measures are high, it means that the decision boundary is not linear, and thus more complex. L3 also measures the complexity of the decision boundary, at least in the basic sense of linearity versus nonlinearity; data with more interleaved regions of concavity (a more complex decision boundary) will have a higher error when classifying linearly interpolated points.

The *Linearity measures* are also asymmetrical in their informativeness, since having poor performance on a linear classifier could be due to low class overlap (high separability) with a highly non-linear boundary, or high class overlap with a linear decision boundary. However, if these complexity measures show that the error with a linear classifier is low, it implies lower complexity for the decision boundary.

N1 measures both decision boundary complexity and class overlap on the data as a whole; high class overlap will lead to a high proportion of borderline points, as will a complex decision boundary with less class overlap. N2 measures the dispersion within classes compared to the margin between classes, and is related to class overlap; however, data with classes distributed sparsely along a long thin border can still have a high N2 value, even with little class overlap. N3 and N4 relate to the complexity of the decision boundary in a local region, in particular whether points from different classes are interleaved in the feature space.

T1 relates to both boundary complexity and class overlap. High values of T1 can be obtained from both highly overlapped classes with a simple underlying decision boundary, and from highly separated classes having a complex decision boundary; for example, a dataset with highly separated pockets of classes interspersed inside each other would have a high value of T1. LSC similarly relates to both class overlap and

decision boundary complexity.

Density is more strongly connected to class overlap on the data as a whole; if classes have little overlap, there will be high-density regions with many points from the same class. ClsCoef and Hubs also measure decision boundary complexity (in terms of overlap with nearby neighbors in a local region), using properties of an ϵ-NN graph calculated on the data.

We make a final note on T2, one of the measures from Ho and Basu (2002) (not included in `ECoL`), which calculates the average number of points per dimension. Although not directly related to either decision boundary complexity or overlap of the underlying process generating the data, gives an insight into the ability to learn the underlying decision rule from data, in terms of how sparsely the data are spread among the number of dimensions in the feature space. T2 has a connection to the statistical learning theory bounds of the difference between empirical risk and actual risk, because the larger the number of dimensions p in the input space, the greater the capacity of the hypothesis classes that will be required to be considered (although other aspects of the functions can influence capacity besides just the input dimension, this is one influencing factor). Bounds on the difference of empirical risk compared to actual risk frequently depend precisely on the ratio of the number of data points n to the capacity of the hypothesis class (as in Equation (3.13)); thus there is an indirect relationship between T2 and the statistical learning theory bounds on empirical versus actual risk.

For the third and final part of this section, we discuss the applications of measures of data complexity to problems in data science and machine learning. Lorena *et al.* (2019) provides a review of works that utilize measures of data complexity in specific downstream applications. The four application areas covered are data analysis (domain understanding, data generation), data pre-processing (feature selection, in-

stance selection, noise identification, class imbalance), learning algorithms (domains of competence, algorithm design, algorithm understanding, multiclass decomposition, parameter tuning), and meta-learning (meta-features). See pg. 28 of Lorena *et al.* (2019) for a comprehensive list of publications in these areas.

Our position on the use of data complexity measures in such applications, is that in and of themselves, using the singular dataset on which they are being calculated, the complexity measures cannot provide improvements to classification complexity that overcome the fundamental limits incurred by the data generation process itself and this particular sample of this process. These limits are formalized by theoretical bounds and equations, such as Equations (3.9), (3.13), and (3.16). As a specific example, consider the proposition of using data complexity measures to guide a feature selection process, denoted by the function $f_{feature} : \mathcal{X} \to \mathcal{X}'$, prior to model fitting; for example, suppose several feature selection methods are compared by calculating the classification complexity measures on the selected features, and then the feature selection method with the lowest resulting empirical complexity is chosen. It is tempting to think that one has used the classification complexity measure to fundamentally lower the complexity of the classification problem after feature selection is applied, and that the learned classification model in the space of selected features, $f_{classification} : \mathcal{X}' \to \mathcal{Y}$, will enjoy the advantages of reduced classification complexity, namely a simpler decision boundary and lower class overlap. This idea obscures the fact that the actual function being learned on the data is the composite function $f_{classification} \circ f_{feature}$, and that the theoretical limits on Bayes Error Rate and difference between empirical and actual risk will still apply to this composite function.

We posit the classification measures can be used to fundamentally improve the classification process when they are utilized in a semi-supervised learning context, in which a number of real and artificial datasets are exploited to gain information

about the complexity measures' behavior themselves. This knowledge is then applied in the context of a new dataset, whose properties match the assumptions and scenarios under which in which the information was obtained. This is the mode employed by many of the articles reviewed by Lorena *et al.* (2019); the classification measures are often explored in a large test bed of hundreds or even thousands of datasets, and general principles are obtained that the authors postulate can be applied to new problems. For example, in Luengo and Herrera (2010) and Luengo and Herrera (2015), large scale analyses are performed to determine intervals for combinations of complexity measures, which are called the *domains of competence*. Datasets with classification complexity values inside the domains of competence are likely to have good performance for either a particular (Luengo and Herrera (2010)) or a small family (Luengo and Herrera (2015)) of learning algorithms. The authors demonstrate the generalization of their findings on new datasets not included in the original analyses.

The use of data complexity measures on a large scale analysis to predict the performance of a particular algorithm has also been explored in other contexts, such as feature selection, instance selection, noise filtering, and meta-learning algorithms, including dynamic selection of classifiers (Lorena *et al.* (2019)).

The question naturally arises, if it is valuable to use the classification measures to obtain an indirect measure of expected performance, rather than using actual classification performance metrics themselves from an out-of-sample analysis to benchmark the usability of a particular algorithm for data pre-processing or model selection. We see (at least) two scenarios in which the data complexity measures are preferable. The first scenario is when the learning algorithm to be deployed is extremely computationally expensive, and the calculation of the data complexity measures incurs a substantially lower computational cost. In this case, the complexity measures can be used to screen out planned classification analyses that are expensive to conduct

and are likely to produce unacceptably poor results, thus saving resources. This proposal assumes that the limits of the data complexity measures that predict good or poor performance have been previously assessed using extensive experiments on other datasets.

The other scenario where this approach may be beneficial is when model selection must be done without having the ability to test out the models on a portion of the dataset, due to proprietary restrictions; for example, in the setting of marketplaces for pre-trained neural networks. In this case, using information on classification complexity measures gleaned from extensive secondary analyses can inform which model is most likely to yield good performance based on the classification complexity of the dataset at hand. Again, this approach only appears to be superior when actual testing of the models and direct comparison of classification performance using standard metrics is not possible.

To summarize this section, we have presented several sets of empirical data complexity measures, which are quantities calculated on a specific finite sample that inform different aspects of the dataset. These dataset aspects are presumed to have an impact on classification performance, and include feature overlap, linearity of the decision boundary, local overlap of classes at the level of individual instances, structural overlap between classes on the dataset as a whole, and global complexity via the internal structure of classes.

We discussed which of these data complexity measures are measuring aspects of the dataset related to class overlap, decision boundary complexity, or both, presented applications in which the measures are used, and commented on whether such use cases are appropriate, in our opinion.

As a concluding thought to this section, we point out that while statistical learning theory, information theoretic divergence measures, and data complexity measures

each provide insights about the complexity of the classification problem via class overlap and decision boundary complexity, drawing direct connections between the theoretical results presented in the first two sections and the empirical data complexity measures is challenging. A deeper theoretical foundation for the empirical data complexity measures, and an explicit connection to the results produced by statistical learning theory and information theory would benefit our understanding of these measures, and is a potential direction for future work.

3.5 Classification Complexity on a Simulated Example

In this section, we use a simulated data example to explore how the two aspects of classification difficulty (decision boundary complexity and class overlap) interact, and how they are quantified by the concepts introduced in the preceding three sections. Simulating data provides an exact knowledge of the underlying data generation process, in particular the relevant joint and conditional probability distributions, which in turn allows us to calculate the quantities of VC dimension (measuring decision boundary complexity) and KL divergence (measuring class overlap) for this setting. Moreover, drawing finite samples of data according to this data generation process allows for calculation of empirical data complexity measures from Lorena *et al.* (2019), offering a direct comparison of how the empirical measures respond to increasing levels of class overlap and decision boundary complexity.

In the first subsection, we introduce the data generation process and plot the true decision boundaries and sampled data under different hyperparameter settings for data generation. In the three subsections that follow, we calculate the VC dimension for the decision boundary, the KL-divergence of the conditional distributions, and the empirical data complexity measures listed in Table 3.1, respectively. Each of these calculations are performed under all combinations of hyperparameter settings

determining class overlap and decision boundary complexity in the simulated data.

For the final subsection on calculation of data complexity measures, we also perform a principle component analysis to assess if the lower-dimensional structure of the variation of the empirical measures reflects known directions of increasing classification complexity. *Principle component analysis*, or PCA, is a process of applying a linear transformation to a data matrix $\mathbf{X}$, such that the basis vectors in the new space are in the directions of maximal variation, in a monotonically decreasing fashion. By keeping only the first few transformed vectors (called *principle components*), most of the variation in the original data can be retained, allowing for dimensionality reduction. PCA is also used for easier visualization of the underlying structures of variance in high dimensional data. Principle component analysis was invented by Karl Pearson in Pearson (1901).

Formally, given an $n \times p$ data matrix $\mathbf{X}$, the k^{th} principle component of a $1 \times p$ data vector $\mathbf{x}_i$ is $t_{k(i)} = \mathbf{x}_i \cdot \mathbf{w}_i$, where $\mathbf{w}_i$ is the eigenvector of $\mathbf{X}^T\mathbf{X}$ corresponding to the k^{th}-largest eigenvalue. The full transformation of $\mathbf{X}$ into the principle component space is

$$\mathbf{T} = \mathbf{XW},$$

where $\mathbf{W}$ is the matrix of eigenvectors of $\mathbf{X}^T\mathbf{X}$, and $\mathbf{T}$ consists of the principle components of $\mathbf{X}$. Since the first few principle components contain the directions of maximum variation in the original data, a frequent step is to truncate to only the first L principle components, producing a dimensionality reduction:

$$\mathbf{T_L} = \mathbf{XW_L}.$$

In our example, we calculate the projection of the data complexity measures into the principle component space, and look at underlying complexity structure as a function of the hyperparameters determining the class overlap and decision boundary

complexity of the generated data. This analysis was inspired by similar work in the original paper on data complexity measures by Ho and Basu (2002).

3.5.1 Data Generation Process

Consider the following data generation process (DGP):

$$\begin{aligned} X_1 &\sim \text{Unif}(-1, 1) \\ \epsilon &\sim \mathcal{N}(0, 1) \\ X_2 \mid X_1, \epsilon &\sim [X_1^2 + \sin(\alpha X_1)] + \epsilon \\ Y \mid X_1, X_2 &\sim \text{Bernoulli}\left(\frac{1}{1 + \exp\{-\lambda \cdot [X_2 - (X_1^2 + \sin(\alpha X_1))]\}}\right) \end{aligned}$$

The parameter α governs classification difficulty via complexity of the decision boundary, by modulating the number of changes of concavity along the decision boundary restricted to $\mathcal{X}_1 = [-1, 1]$. This can be seen by calculating the true class boundary in the (X_1, X_2) plane and solving for X_2. The true class boundary is the curve $X_2 = X_1^2 + \sin(\alpha X_1)$, from the following derivation:

$$\begin{aligned} P(Y = 1 \mid X_1, X_2) &= 0.5 \\ \frac{1}{1 + \exp\{-\lambda \cdot [X_2 - (X_1^2 + \sin(\alpha X_1))]\}} &= 0.5 \\ \exp\{-\lambda \cdot [X_2 - (X_1^2 + \sin(\alpha X_1))]\} &= 1 \\ -\lambda \cdot [X_2 - (X_1^2 + \sin(\alpha X_1))] &= 0 \\ X_2 - (X_1^2 + \sin(\alpha X_1) &= 0 \\ X_2 &= X_1^2 + \sin(\alpha X_1). \end{aligned}$$

High values of α decrease the period of the sine function, resulting in a decision boundary between the two classes with higher complexity. On the other hand, low values of α increase the period of the sine function, resulting in a decision boundary

with fewer concavity changes and lower complexity. Figure 3.6 demonstrates the impact of the parameter α on the decision boundary complexity.

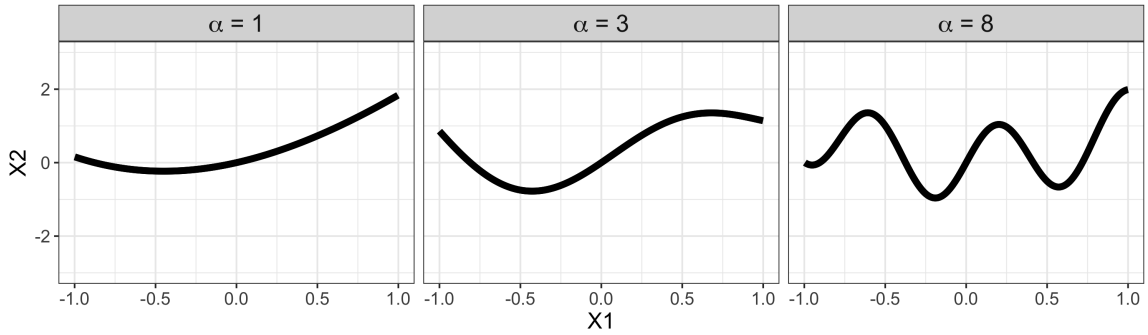

Figure 3.6: Impact of the boundary complexity parameter α.

The parameter λ governs classification difficulty via class overlap, by modulating the impact of the distance $X_2 - (X_1^2 + \sin(\alpha X_1))$ (i.e., the distance from the decision boundary) on the Bernoulli parameter in the distribution of Y. This can be seen by considering how the Bernoulli parameter in the distribution of Y changes in tandem with λ. As λ decreases toward 0, the Bernoulli parameter gets closer to to 0.5, meaning that regardless of the value of x (and regardless of which side of the decision boundary x falls on), there is roughly an equal probability of $P(Y = 1 \mid \text{x})$ and $P(Y = 0 \mid \text{x})$. On the other hand, as λ becomes larger, the value of $-\lambda \cdot [X_2 - (X_1^2 + \sin(\alpha X_1))]$ (the distance from the decision boundary multiplied by $-\lambda$) has a stronger impact on the Bernoulli parameter, and points on either side of the boundary will have a more deterministic probability to belong to one class or the other. This results in lower class overlap and greater class separability.

The impact of λ on class overlap is demonstrated in Figure 3.7, which shows both the true decision boundary and a sample of 1000 points for each (α, λ) combination. Since λ governs the probability of class membership as a function of distance to the decision boundary, we plot different values of λ (rows) in the context of different decision boundary complexities α (columns), rather than collapsing across α values

and plotting the data as a function of λ alone.

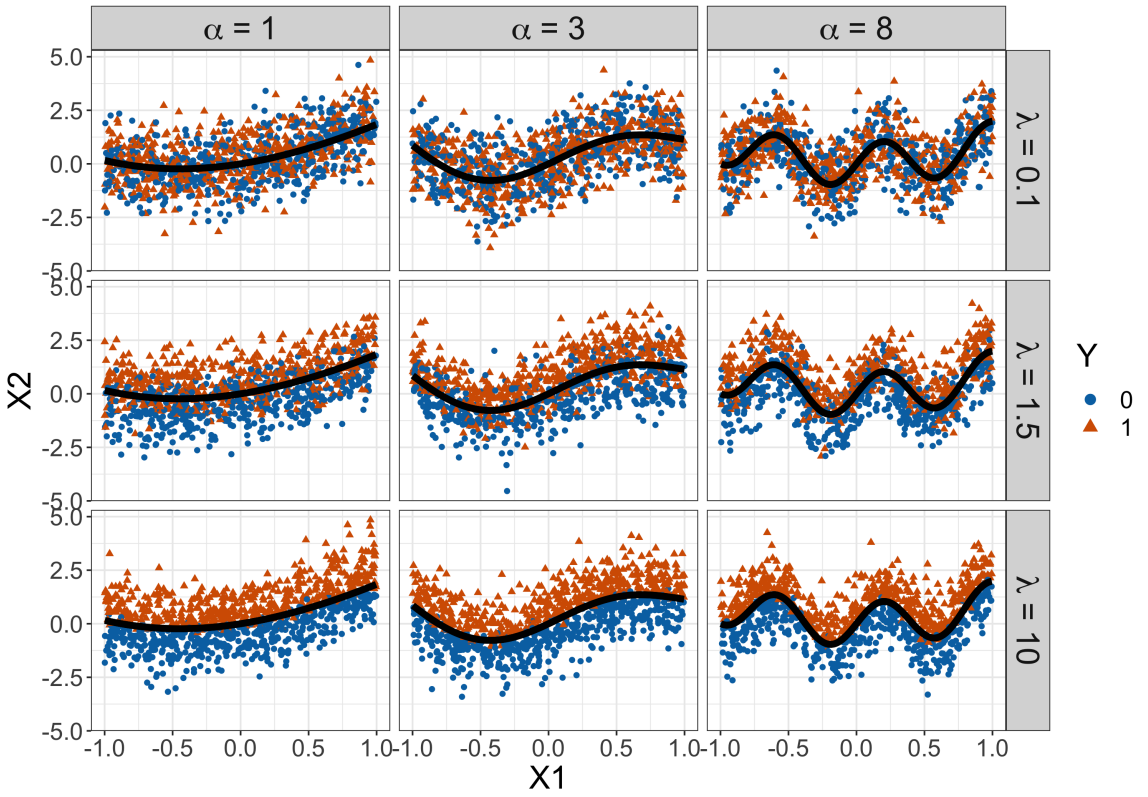

Figure 3.7: Impact of λ and α on classification complexity. As λ increases (moving from top to bottom), distance from the decision boundary is more strongly deterministic of class membership. This leads to lower class overlap and greater class separability.

Next, we explore this DGP by measuring 1) decision boundary complexity via VC dimension, 2) class overlap via KL divergence, and 2) general classification complexity via empirical complexity measures.

3.5.2 VC Dimension of Decision Boundaries

To compare decision boundary complexity via VC dimension, we approximate the three decision boundaries (governed by α) using polynomials of three different degrees. In actuality, all three true decision boundaries arise from the same hypothesis class containing functions with a $\sin(\alpha x)$ term, meaning that the overall hypothesis

class has infinite VC dimension (Bousquet *et al.* (2004)). In order to compare the "wiggly-ness" of the three decision boundaries in a rigorous and meaningful way, we approximate the decision boundary for each complexity subgroup (defined by α) with a function from the hypothesis class of polynomials in $\mathbb{R}^2$ of degree m. In Equation 3.12, we saw that the class of polynomial classifiers in $\mathbb{R}^2$ of degree m has VC dimension $\frac{(m+2)(m+1)}{2}$. This theorem allows for a comparison of decision boundary complexities via a simple calculation.

To summarize, we quantify the complexity of the three decision boundaries corresponding to three values of α, by learning three polynomials that can approximate these boundary functions, and then calculating the VC-dimension of the hypothesis class of polynomials of the given degree for all three cases.

First, recall that from the DGP, the true decision boundary of complexity α is the function f_α defined by

$$f_\alpha : \mathbb{R} \to \mathbb{R}$$
$$f_\alpha(x) = x^2 + \sin(\alpha x).$$

We seek to approximate this polynomial using a function f_m, which is a polynomial in $\mathbb{R}^2$ of degree m. We define the minimum polynomial degree m_α as the degree m of smallest degree polynomial that can approximate f_α within an error tolerance of 0.005, using a sample of size $n = 3000$. The minimum polynomial degree m_α corresponding to the values of α shown in Figures 3.6 and 3.7 ($\alpha \in \{1, 3, 8\}$) are given in Table 3.2.

To support these choices for m_α, we provide the root mean square error between the value $f_\alpha(x)$ of the true decision boundary function and the value $f_m(x)$ of a polynomial of degree m, for $n = 3000$ data points drawn uniformly from $[-1, 1]$. The polynomial f_m is obtained using polynomial regression with the ordinary least squares

Value of α	**Minimum polynomial degree** m_α
$\alpha = 1$	$m_{\alpha=1} = 3$
$\alpha = 3$	$m_{\alpha=3} = 5$
$\alpha = 8$	$m_{\alpha=8} = 11$

Table 3.2: Degree of the smallest degree polynomial that approximates the decision boundary function within the given error tolerance.

(OLS) estimator, calculated using the `lm` function in R version 4.0.5. The root mean square error (RMSE) corresponding to the polynomial of degree m for approximating f_α is

$$\text{RMSE}_\alpha(m) = \sqrt{\sum_{i=1}^{n} (f_\alpha(x_i) - f_m(x_i))^2}. \tag{3.19}$$

With this notation,

$$m_\alpha = \min\{m \mid \text{RMSE}_\alpha(m) \leq 0.005\}.$$

Table 3.3 shows the RMSE between the true decision boundary function f_α of complexity α and the best fit polynomials of degree m_α and $m_\alpha - 1$. As required, m_α is the lowest degree polynomial with an RMSE below the error tolerance threshold of 0.005, going one degree lower to $m_\alpha - 1$ results in an error above the chosen threshold. Note this assumes that polynomials of even lower degree than $m_\alpha - 1$ will have an even greater RMSE.

The error tolerance threshold of 0.005 is motivated by a visual comparison of f_α and the approximating polynomials f_{m_α} and $f_{m_\alpha - 1}$, shown in Figure 3.8. The polynomial resulting from a threshold of 0.005 (namely, a polynomial of degree $m_\alpha = 3, 5, 11$ corresponding to $\alpha = 1, 3, 8$), is visually indistinguishable from the actual decision boundary. However, the polynomials of one lower degree $(2, 4, 10)$, which would be considered "minimal" by choosing a threshold of 0.05 for example, have

Value of α	Value of m_α	RMSE(m_α)	RMSE($m_\alpha - 1$)
$\alpha = 1$	$m_{\alpha=1} = 3$	0.0003	0.02
$\alpha = 3$	$m_{\alpha=3} = 5$	0.003	0.05
$\alpha = 8$	$m_{\alpha=8} = 11$	0.004	0.03

Table 3.3: The polynomial of degree m_α is the smallest degree polynomial with RMSE below 0.005.

visual differences between the fitted polynomial (dashed red line) and the true decision boundary (solid blue line).

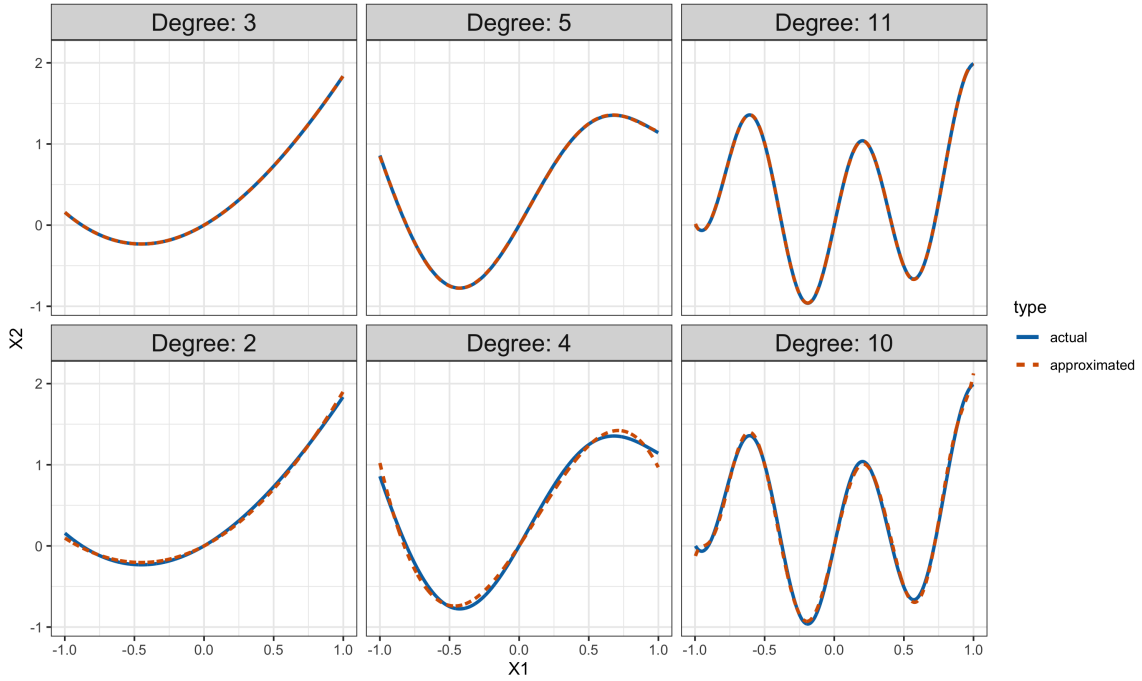

Figure 3.8: The top row shows the actual decision boundary f_α (blue) and the chosen polynomial approximation f_{m_α} (red), for $\alpha \in \{1, 3, 8\}$ (moving left to right). The bottom row shows f_α (blue) and $f_{m_\alpha - 1}$ (red). The lower degree polynomials $f_{m_\alpha - 1}$ are visually different from the true decision boundary f_α (bottom row), whereas f_α and f_{m_α} (top row) are visually indistinguishable.

Figure 3.9 is a visual aid to confirm that the plot of the minimum degree polynomials (degrees $3, 5, 11$) matches their purported degrees, which is not immediately obvious in the plots of these polynomials over a truncated support, as shown in Fig-

ure 3.8. For example, the polynomial of Degree 3 in the top left subplot of Figure 3.8 appears to have no changes of concavity, which would only be possible for an even-degree polynomial. Figure 3.9 clarifies this confusion by showing the chosen minimum degree polynomials (of degrees $3, 5, 11$) over a wider support (top row), and over the support relative to our DGP (bottom row). The dashed rectangles show the truncation of the polynomials to $[-1, 1]$, which is the support of $\mathcal{X}_1$ in the DGP, and the support visualized in Figure 3.8. Notice that these same polynomials were previously presented in Figure 3.3 during our exploration of VC dimension for polynomial classifiers.

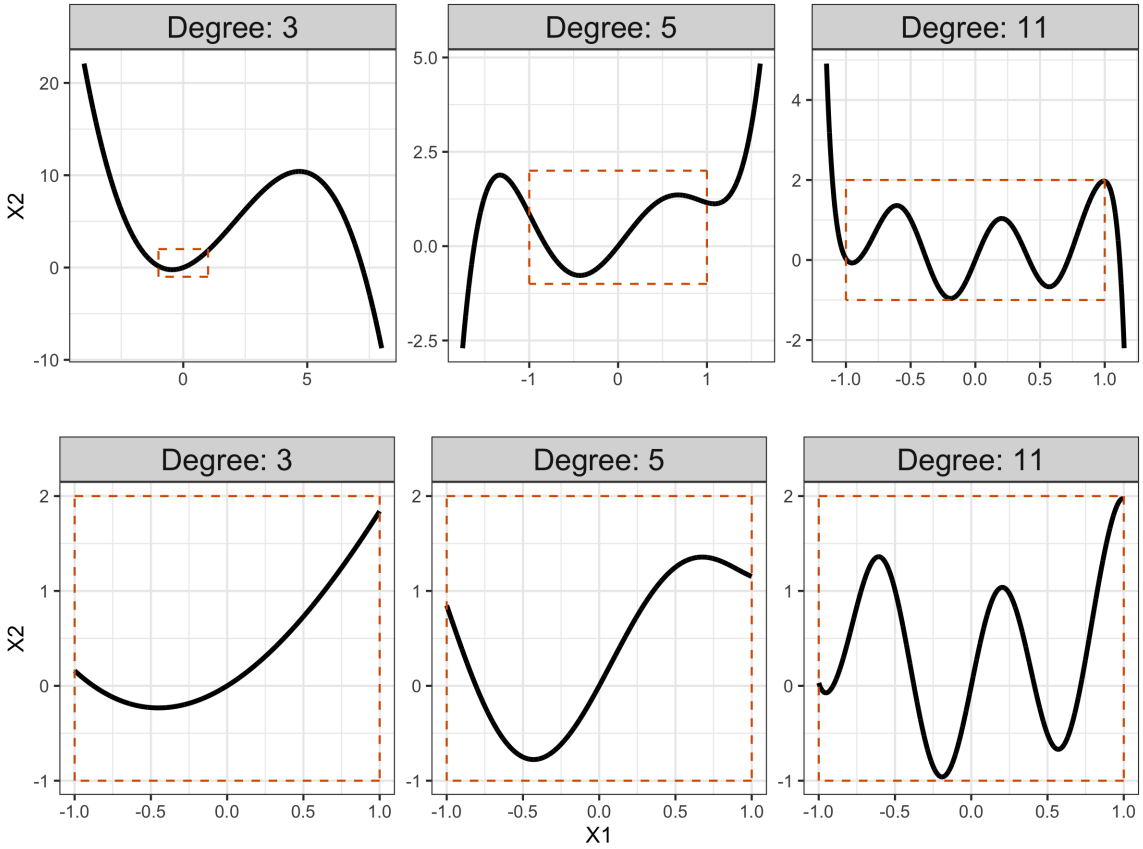

Figure 3.9: The top row shows the polynomials f_{m_α} for $\alpha \in \{1, 3, 8\}$ (from left to right), over a larger support. This top row matches the support from Figure 3.3. The bottom row is a zoomed in version of these polynomials, over support $[-1, 1]$; this is the actual decision boundary for our DGP. The bottom row matches the decision boundaries displayed in Figure 3.8.

The purpose of the preceding approximation exercise was to rigorously show that as α increases, the complexity of the decision boundary also increases in a quantifiable way. Specifically, we see that as α increases from 1 to 3 to 8, the VC dimension of the hypothesis class needed to approximate the decision boundary with high fidelity increases from 10 to 21 to 78.

The increased decision boundary complexity with higher values of α makes the classification problem more difficult, as can be seen if we consider the bias-variance trade-off from Equation (3.9), which we reproduce here for easier reading:

$$R(f_n) - R^* = [R(f^*) - R^*] + [R(f_n) - R(f^*)].$$

It is worth repeating that when the decision boundary has a higher level of underlying complexity (e.g. $\alpha = 8$), it means that, keeping sample size the same, either the approximation error (left term of RHS) or the estimation error (right term of RHS) will increase, relative to the hypothetical scenario of using the same amount of data to solve a classification problem with a lower complexity decision boundary. If we increase the complexity of the hypothesis class to be able to accurately approximate the true decision boundary, the approximation error $R(f^*) - R^*$ will be lower, but the estimation error $R(f_n) - R(f^*)$ will be higher. On the other hand, if we consider a simpler hypothesis class to reduce the estimation error, the higher complexity of the true underlying decision boundary means that the approximation error will increase.

From the point of view of bounding the true risk of the function (which impacts generalization ability), consider, for example, the change in VC dimension from $h = 3$ to $h = 11$ in the bound shown in Equation (3.13). This corresponds to the lowest and highest VC dimensions required for the class overlap settings from our simulated example. For a sample of size $n = 500$, the left term under the square root (shown in (3.14)) increases from 0.04 to 0.1 when the VC dimension h increases from 3 to 11.

As an alternative comparison, if we want the left-term of (3.14) to be below 0.05, we require a sample of size $n = 395$ if the VC dimension is $h = 3$, but a sample of size $n = 1446$ if the VC dimension is $h = 11$.

These examples highlight in a concrete way how decision boundary complexity has a quantifiable impact on the complexity of the underlying classification problem, and that VC dimension is a relevant way to quantify decision boundary complexity.

3.5.3 KL Divergence

Next, we approximate the KL divergence from the conditional distribution

$$f^0_{\alpha,\lambda}(\mathrm{x}) = f_{\alpha,\lambda}(\mathrm{x} \mid Y = 0)$$

to the conditional distribution

$$f^1_{\alpha,\lambda}(\mathrm{x}) = f_{\alpha,\lambda}(\mathrm{x} \mid Y = 1),$$

which is denoted

$$D^{\alpha,\lambda}_{KL}(f^0 \mid\mid f^1).$$

We calculate KL divergence to provide a quantitative measure of the class overlap of the simulated data from our example, and to show that KL divergence has a strong relationship to the known hyperparameter λ that impacts class overlap in our DGP. The calculation is somewhat tedious, but the results are worthwhile.

First, we derive expressions for the densities $f^1_{\alpha,\lambda}$ and $f^0_{\alpha,\lambda}$ using Bayes' rule, the Law of Total Expectation and the Multiplication Rule from basic probability princi-

ples. Overloading notation for f and P,

$$\begin{aligned}
f^1_{\alpha,\lambda}(x_1,x_2) &= f_{\alpha,\lambda}(x_1,x_2 \mid Y=1) \\
&= \frac{P_{\alpha,\lambda}(Y=1 \mid x_1,x_2)\cdot f_{\alpha,\lambda}(x_1,x_2)}{P_{\alpha,\lambda}(Y=1)} \\
&= \frac{P_{\alpha,\lambda}(Y=1 \mid x_1,x_2)\cdot f_{\alpha,\lambda}(x_2 \mid x_1)\cdot f_{\alpha,\lambda}(x_1)}{P_{\alpha,\lambda}(Y=1)} \\
&= \frac{\frac{1}{1+\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]\}}\cdot\frac{1}{\sqrt{2\pi}}\exp\{-\frac{1}{2}[x_2-(x_1^2+\sin(\alpha x_1))]^2\}\cdot\frac{1}{2}}{P_{\alpha,\lambda}(Y=1)} \\
&= \frac{\frac{1}{2\sqrt{2\pi}}\cdot\frac{\exp\{-\frac{1}{2}[x_2-(x_1^2+\sin(\alpha x_1))]^2\}}{1+\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]\}}}{P_{\alpha,\lambda}(Y=1)},
\end{aligned}$$

and

$$\begin{aligned}
f^0_{\alpha,\lambda}(x_1,x_2) &= f_{\alpha,\lambda}(x_1,x_2 \mid Y=0) \\
&= \frac{P_{\alpha,\lambda}(Y=0 \mid x_1,x_2)\cdot f_{\alpha,\lambda}(x_1,x_2)}{P_{\alpha,\lambda}(Y=0)} \\
&= \frac{(1-P_{\alpha,\lambda}(Y=1 \mid x_1,x_2))\cdot f_{\alpha,\lambda}(x_2 \mid x_1)\cdot f_{\alpha,\lambda}(x_1)}{1-P_{\alpha,\lambda}(Y=1)} \\
&= \frac{\left(1-\frac{1}{1+\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]\}}\right)\cdot\frac{1}{2\sqrt{2\pi}}\exp\{-\frac{1}{2}[x_2-(x_1^2+\sin(\alpha x_1))]^2\}}{1-P_{\alpha,\lambda}(Y=1)} \\
&= \frac{\frac{\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]\}}{1+\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]\}}\cdot\frac{1}{2\sqrt{2\pi}}\exp\{-\frac{1}{2}[x_2-(x_1^2+\sin(\alpha x_1))]^2\}}{1-P_{\alpha,\lambda}(Y=1)} \\
&= \frac{\frac{1}{2\sqrt{2\pi}}\cdot\frac{\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]-\frac{1}{2}[x_2-(x_1^2+\sin(\alpha x_1))]^2\}}{1+\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]\}}}{1-P_{\alpha,\lambda}(Y=1)} \\
&= \frac{\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]-\frac{1}{2}[x_2-(x_1^2+\sin(\alpha x_1))]^2\}}{2\sqrt{2\pi}(1-P_{\alpha,\lambda}(Y=1))\cdot(1+\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]\})}.
\end{aligned}$$

To use these densities in an integral approximation for calculating KL divergence, we first need to numerically approximate $P_{\alpha,\lambda}(Y=1)$. We have

$$\begin{aligned}
P_{\alpha,\lambda}(Y=1) &= \int_{\mathcal{X}_1}\int_{\mathcal{X}_2} P_{\alpha,\lambda}(Y=1 \mid x_1,x_2)\cdot f_{\alpha,\lambda}(x_2 \mid x_1)\cdot f_{\alpha,\lambda}(x_1)dx_2dx_1 \\
&= \int_{\mathcal{X}_1}\int_{\mathcal{X}_2} \frac{\exp\{-\frac{1}{2}[x_2-(x_1^2+\sin(\alpha x_1))]^2\}}{2\sqrt{2\pi}(1+\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]\})}dx_2dx_1.
\end{aligned}$$

Table 3.4 provides the approximation of this integral for the 9 combinations of (α, λ) in our simulation study[1].

$P_{\alpha,\lambda}(Y=1)$	$\alpha = 1$	$\alpha = 3$	$\alpha = 8$
$\lambda = 0.1$	0.5	0.5	0.5
$\lambda = 1.5$	0.5	0.5	0.5
$\lambda = 10$	0.466833	0.421735	0.429553

Table 3.4: Numerical approximation for the marginal probabilities $P_{\alpha,\lambda}(Y=1)$.

We now derive a simplified form of the integral representing the KL divergence from $f^1_{\alpha,\lambda}$ to $f^0_{\alpha,\lambda}$. We use the formula from (3.17), along with the preceding derivations (and some intermediate steps) for the conditional densities.

$$
\begin{aligned}
D^{\alpha,\lambda}_{KL}(f^0 \,||\, f^1) &= \int_{\mathcal{X}_1}\int_{\mathcal{X}_2} f^0_{\alpha,\lambda}(x_1,x_2)\ln\left(\frac{f^0_{\alpha,\lambda}(x_1,x_2)}{f^1_{\alpha,\lambda}(x_1,x_2)}\right)dx_2dx_1 \\
&= \int_{\mathcal{X}_1}\int_{\mathcal{X}_2} f^0_{\alpha,\lambda}(x_1,x_2)\ln\left(\frac{\frac{P_{\alpha,\lambda}(Y=0|x_1,x_2)\cdot f_{\alpha,\lambda}(x_1,x_2)}{P_{\alpha,\lambda}(Y=0)}}{\frac{P_{\alpha,\lambda}(Y=1|x_1,x_2)\cdot f_{\alpha,\lambda}(x_1,x_2)}{P_{\alpha,\lambda}(Y=1)}}\right)dx_2dx_1 \\
&= \int_{\mathcal{X}_1}\int_{\mathcal{X}_2} f^0_{\alpha,\lambda}(x_1,x_2)\ln\left(\frac{P_{\alpha,\lambda}(Y=0\mid x_1,x_2)\cdot P_{\alpha,\lambda}(Y=1)}{P_{\alpha,\lambda}(Y=1\mid x_1,x_2)\cdot P_{\alpha,\lambda}(Y=0)}\right)dx_2dx_1 \\
&= \int_{\mathcal{X}_1}\int_{\mathcal{X}_2} f^0_{\alpha,\lambda}(x_1,x_2)\ln\left(\frac{\frac{\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]\}}{1+\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]\}}}{\frac{1}{1+\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]\}}}\cdot\frac{P_{\alpha,\lambda}(Y=1)}{1-P_{\alpha,\lambda}(Y=1)}\right)dx_2dx_1 \\
&= \int_{\mathcal{X}_1}\int_{\mathcal{X}_2} f^0_{\alpha,\lambda}(x_1,x_2) \\
&\qquad \times\ln\left(\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]\}\cdot\frac{P_{\alpha,\lambda}(Y=1)}{1-P_{\alpha,\lambda}(Y=1)}\right)dx_2dx_1 \\
&= \int_{\mathcal{X}_1}\int_{\mathcal{X}_2} \frac{\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]-\frac{1}{2}[x_2-(x_1^2+\sin(\alpha x_1))]^2\}}{2\sqrt{2\pi}(1-P_{\alpha,\lambda}(Y=1)(1+\exp\{-\lambda\cdot[x_2-(x_1^2+\sin(\alpha x_1))]\})} \\
&\qquad \times\left(-\lambda\cdot\left[x_2-(x_1^2+\sin(\alpha x_1))\right]+\ln\left(\frac{P_{\alpha,\lambda}(Y=1)}{1-P_{\alpha,\lambda}(Y=1)}\right)\right)dx_2dx_1.
\end{aligned}
$$

[1]We approximate the double integral using the Wolfram Alpha Double Integral Calculator found at https:www.wolframalpha.comwidgetsview.jsp?id=f5f3cbf14f4f5d6d2085bf2d0fb76e8a.

Table 3.5 shows a numerical approximation of this integral for the same 9 combinations of (α, λ). The values for $P_{\alpha,\lambda}(Y = 1)$ are supplied from Table 3.4.

$D_{KL}^{\alpha,\lambda}(f^0 \mid\mid f^1)$	$\alpha = 1$	$\alpha = 3$	$\alpha = 8$
$\lambda = 0.1$	0.005	0.005	0.005
$\lambda = 1.5$	0.795	0.795	0.795
$\lambda = 10$	7.239	6.516	6.634

Table 3.5: Numerical approximation for the KL divergence from $f_{\alpha,\lambda}^1$ to $f_{\alpha,\lambda}^0$.

In general, the class overlap (as represented by KL divergence) is determined by λ; as λ increases, the conditional distributions are more separable and have a higher KL divergence. For $\lambda = 0.1$ and $\lambda = 1.5$, which represent the subgroups with large and medium class overlap, respectively, we see that the value of α does not make a difference in the approximate KL divergence (to 3 decimal places). The overlap of these subgroups can be seen in the top tow rows from Figure 3.7; here, the region of class overlap covers the entire decision boundary for all three choices of α, hence the exactly shape or complexity of the boundary doesn't have an influence on how separated the classes are.

However, for the subgroups with $\lambda = 10$ which have a very low degree of overlap and high KL divergence, (bottom row of Figure 3.7), the complexity of the decision boundary has a small impact on divergence between conditional class distributions. The subgroup with lowest complexity ($\alpha = 1$) has a slightly higher divergence than the subgroups with greater complexity, which is attributed in part to the differing marginal probabilities $P_{\alpha,\lambda}(Y = 1)$.

The KL divergence calculations provide an objective quantification of the trend in overlap that is visually present in Figure 3.7 as one moves from the bottom row (almost no overlap) to the top row (almost full overlap), which we discuss here in

further depth using Equation (3.16). This equation, reproduced below for easier reading, sheds light on why higher class overlap, quantified in one sense by lower KL divergence, increases the difficulty and complexity of the classification problem.

$$\epsilon_{\text{Bayes}} = \int_{R_1} P(Y = 0 \mid \text{x}) d\text{x} + \int_{R_0} P(Y = 1 \mid \text{x}) d\text{x},$$

In Figure 3.7, the decision boundary shown by the black line demarcates R_1 from R_2. With increasing class overlap (moving from the bottom to the top row), the probability of observing data in the region of the opposite class label gets higher (e.g. $P(Y = 0 \mid \text{x})$ increases in R_1 and vice versa). This increases the integrals in Equation (3.16) (reproduced directly above this paragraph) and thus increases the BER. Since the BER is the theoretically lowest error that can be attained, this directly contributes to the difficulty of the classification problem. While KL-divergence does not directly measure the BER, it is closely related, as are other f-divergences.

This simulated example, in particular the results in Table 3.5, also highlight the subtle interaction between decision boundary complexity and class overlap, where greater decision boundary complexity can lead to greater class overlap even when the probability of being in the wrong class, as a function of distance from the decision boundary, is the same. One explanation for this interaction is that with greater decision boundary complexity, there are more directions in the X domain in which a fixed distance from the decision boundary lands in the region of the opposing class.

The simulated example demonstrates both of our observations at the end of Section 3.3. Greater class overlap contributes to classification complexity by increasing the the theoretically lowest error that can be attained by any learning algorithm on any set of data for that problem, and greater decision boundary complexity contributes by making the theoretically optimal function harder to learn and to estimate performance for, given a fixed sample size.

3.5.4 *Empirical Measures of Data Complexity*

The preceding sections demonstrated how classification difficulty is impacted by both decision boundary complexity and class overlap, which are controlled in our simulated data by the α and λ parameters, respectively.

To understand how these two aspects impact empirical estimates of classification difficulty, we calculate measures of data complexity from Lorena *et al.* (2019) on the 9 subgroups, averaged over 10 draws of the sample with two different sample sizes of increasing order of magnitude: 50 and 500. We use the `ECoL` R package for this analysis. We use only 10 iterations in order to demonstrate the variability of the complexity measures on small sample sizes.

In Table 3.6, we present the results of calculating the complexity measures on datasets of size $n = 50$ generated according to the DGP for all combinations of $\alpha \in \{1, 3, 8\}$ and $\lambda \in \{0.1, 1.5, 10\}$. Table 3.7 shows similar results but for $n = 500$.

Of note, many of the complexity measures yield similar results and are highly correlated; this is to be expected because some of the measures measure very similar quantities. There are several other clear trends. First, the main driver of the complexity measures is the overlap parameter λ. However, within the subgroups with low overlap $\lambda = 10$, the decision boundary complexity parameter α starts to come into play; the complexity measures tend to have a higher value (indicating more complexity) for larger values of α (which correspond to more complicated decision boundaries). Within the $n = 500$ dataset (Table 3.7), this pattern is much more stable across all of the complexity parameters.

We also see that the complexity measures have much higher variance on the smaller dataset $n = 50$ (to be expected); furthermore, within this smaller dataset, for the subgroups where $\lambda = 10$ (the most separability), the complexity measures typically

Table 3.6: Mean (sd) of the 17 data complexity measures, calculated over 10 samples of size $n = 50$ drawn using the DGP above.

α	$\alpha = 1$	$\alpha = 3$	$\alpha = 8$	$\alpha = 1$	$\alpha = 3$	$\alpha = 8$	$\alpha = 1$	$\alpha = 3$	$\alpha = 8$
λ	$\lambda = 0.1$			$\lambda = 1.5$			$\lambda = 10$		
F1	0.98 (0.02)	0.98 (0.025)	0.97 (0.02)	0.85 (0.06)	0.83 (0.086)	0.8 (0.124)	0.53 (0.051)	0.58 (0.043)	0.61 (0.076)
F1v	0.91 (0.086)	0.9 (0.083)	0.87 (0.074)	0.48 (0.109)	0.52 (0.157)	0.5 (0.149)	0.14 (0.024)	0.17 (0.032)	0.28 (0.065)
F2	0.72 (0.143)	0.74 (0.126)	0.75 (0.106)	0.59 (0.142)	0.51 (0.11)	0.57 (0.175)	0.26 (0.077)	0.29 (0.043)	0.38 (0.111)
F3	0.9 (0.033)	0.92 (0.031)	0.91 (0.043)	0.85 (0.074)	0.79 (0.094)	0.81 (0.153)	0.47 (0.077)	0.49 (0.023)	0.65 (0.127)
F4	0.84 (0.042)	0.87 (0.04)	0.84 (0.059)	0.77 (0.076)	0.72 (0.105)	0.72 (0.163)	0.21 (0.149)	0.28 (0.074)	0.56 (0.134)
L1	0.47 (0.023)	0.48 (0.021)	0.47 (0.019)	0.39 (0.022)	0.4 (0.034)	0.39 (0.047)	0.26 (0.022)	0.28 (0.024)	0.33 (0.025)
L2	0.24 (0.012)	0.24 (0.011)	0.23 (0.01)	0.19 (0.013)	0.19 (0.02)	0.19 (0.029)	0.09 (0.012)	0.11 (0.014)	0.15 (0.017)
L3	0.23 (0.018)	0.23 (0.012)	0.23 (0.012)	0.17 (0.021)	0.18 (0.028)	0.18 (0.029)	0.07 (0.008)	0.09 (0.016)	0.13 (0.025)
N1	0.75 (0.1)	0.72 (0.054)	0.75 (0.057)	0.53 (0.094)	0.6 (0.069)	0.59 (0.119)	0.22 (0.072)	0.28 (0.047)	0.42 (0.062)
N2	0.51 (0.051)	0.5 (0.033)	0.52 (0.039)	0.44 (0.035)	0.46 (0.039)	0.43 (0.062)	0.29 (0.037)	0.31 (0.025)	0.37 (0.037)
N3	0.55 (0.113)	0.5 (0.067)	0.56 (0.092)	0.35 (0.087)	0.42 (0.095)	0.36 (0.146)	0.11 (0.064)	0.12 (0.05)	0.23 (0.069)
N4	0.34 (0.09)	0.37 (0.058)	0.31 (0.063)	0.25 (0.062)	0.3 (0.062)	0.25 (0.09)	0.06 (0.035)	0.06 (0.033)	0.13 (0.063)
T1	0.75 (0.077)	0.72 (0.036)	0.73 (0.064)	0.54 (0.101)	0.6 (0.057)	0.58 (0.114)	0.26 (0.057)	0.28 (0.052)	0.45 (0.092)
LSC	0.96 (0.008)	0.96 (0.004)	0.96 (0.005)	0.94 (0.012)	0.95 (0.008)	0.95 (0.015)	0.88 (0.016)	0.9 (0.013)	0.93 (0.011)
Density	0.91 (0.005)	0.91 (0.005)	0.91 (0.005)	0.89 (0.007)	0.9 (0.006)	0.9 (0.01)	0.86 (0.007)	0.87 (0.007)	0.89 (0.007)
ClsCoef	0.44 (0.06)	0.4 (0.048)	0.42 (0.042)	0.37 (0.057)	0.37 (0.052)	0.37 (0.063)	0.34 (0.033)	0.31 (0.035)	0.35 (0.047)
Hubs	0.82 (0.051)	0.86 (0.045)	0.85 (0.034)	0.84 (0.044)	0.86 (0.047)	0.83 (0.048)	0.8 (0.054)	0.84 (0.023)	0.84 (0.021)

have the highest variance with the greatest decision boundary complexity $\alpha = 8$. The patterns in variance reflect typical sampling considerations and highlight our earlier comments that these empirical data measures suffer from the same limitations that other empirical estimates calculated on finite samples do.

The trends discussed above can be observed in Figures 3.10 ($n = 10$), 3.11 ($n = 20$), 3.12 ($n = 50$) and 3.13 ($n = 500$), which include a more extensive set of sample sizes. Each of the subplots shows the values of one of the complexity measures, for each of the 9 combinations of α and λ. The different values of λ are shown in different colors and shapes (each line corresponds to one value of λ), whereas α is shown on the x-axis. Within the Feature Measures and Neighborhood/Network Measures, we chose to plot these particular measures because they were the ones that will have the highest correlation to out of sample performance, in our analysis with a speech

Table 3.7: Mean (sd) of the 17 data complexity measures, calculated over 10 samples of size $n = 500$ drawn using the DGP above.

α	$\alpha = 1$	$\alpha = 3$	$\alpha = 8$	$\alpha = 1$	$\alpha = 3$	$\alpha = 8$	$\alpha = 1$	$\alpha = 3$	$\alpha = 8$
λ	$\lambda = 0.1$			$\lambda = 1.5$			$\lambda = 10$		
F1	0.99 (0.004)	1 (0.005)	1 (0.002)	0.81 (0.018)	0.84 (0.029)	0.81 (0.014)	0.55 (0.026)	0.62 (0.028)	0.61 (0.019)
F1v	0.97 (0.021)	0.98 (0.027)	0.99 (0.01)	0.44 (0.039)	0.48 (0.041)	0.52 (0.023)	0.16 (0.01)	0.2 (0.023)	0.28 (0.015)
F2	0.86 (0.056)	0.85 (0.057)	0.87 (0.067)	0.61 (0.044)	0.65 (0.079)	0.65 (0.092)	0.32 (0.023)	0.33 (0.027)	0.39 (0.047)
F3	0.99 (0.004)	0.99 (0.003)	0.99 (0.004)	0.94 (0.019)	0.94 (0.028)	0.95 (0.023)	0.63 (0.042)	0.64 (0.048)	0.75 (0.027)
F4	0.98 (0.005)	0.99 (0.004)	0.98 (0.007)	0.93 (0.018)	0.93 (0.027)	0.94 (0.022)	0.6 (0.05)	0.63 (0.052)	0.72 (0.03)
L1	0.5 (0.003)	0.5 (0.003)	0.5 (0.001)	0.39 (0.012)	0.4 (0.012)	0.41 (0.006)	0.28 (0.007)	0.3 (0.012)	0.33 (0.009)
L2	0.25 (0.001)	0.25 (0.002)	0.25 (0.001)	0.19 (0.007)	0.2 (0.007)	0.2 (0.003)	0.11 (0.005)	0.13 (0.009)	0.15 (0.004)
L3	0.25 (0.002)	0.25 (0.002)	0.25 (0.001)	0.17 (0.009)	0.18 (0.009)	0.19 (0.004)	0.09 (0.006)	0.1 (0.008)	0.13 (0.005)
N1	0.71 (0.032)	0.72 (0.025)	0.72 (0.02)	0.53 (0.036)	0.53 (0.032)	0.52 (0.028)	0.13 (0.014)	0.13 (0.02)	0.16 (0.022)
N2	0.5 (0.016)	0.49 (0.014)	0.5 (0.007)	0.43 (0.014)	0.43 (0.013)	0.43 (0.011)	0.22 (0.012)	0.22 (0.014)	0.25 (0.014)
N3	0.5 (0.032)	0.49 (0.028)	0.5 (0.016)	0.35 (0.036)	0.37 (0.03)	0.35 (0.022)	0.08 (0.012)	0.08 (0.018)	0.1 (0.021)
N4	0.44 (0.024)	0.42 (0.026)	0.44 (0.018)	0.3 (0.024)	0.32 (0.027)	0.32 (0.027)	0.06 (0.017)	0.09 (0.016)	0.17 (0.025)
T1	0.71 (0.029)	0.72 (0.031)	0.72 (0.026)	0.53 (0.031)	0.53 (0.039)	0.54 (0.025)	0.13 (0.014)	0.14 (0.021)	0.17 (0.022)
LSC	1 (0)	1 (0)	1 (0)	0.99 (0.001)	0.99 (0.001)	0.99 (0.001)	0.95 (0.005)	0.95 (0.004)	0.97 (0.002)
Density	0.91 (0.001)	0.91 (0.001)	0.91 (0.001)	0.9 (0.001)	0.9 (0.002)	0.9 (0.001)	0.87 (0.002)	0.87 (0.002)	0.89 (0.002)
ClsCoef	0.35 (0.004)	0.34 (0.006)	0.32 (0.008)	0.32 (0.008)	0.31 (0.01)	0.3 (0.011)	0.27 (0.003)	0.27 (0.007)	0.27 (0.007)
Hubs	0.88 (0.038)	0.89 (0.031)	0.89 (0.034)	0.89 (0.025)	0.89 (0.017)	0.89 (0.025)	0.85 (0.048)	0.84 (0.039)	0.89 (0.024)

dataset (to be described in greater detail in the next chapter). We also included the Linearity Measures for completeness.

The largest and most obvious change moving from Figure 3.10 ($n = 10$) through to Figure 3.13 ($n = 500$), in other words, as sample size increases, is the size of the error bars indicating the standard error for the mean complexity measure over repeated sampling. As the sample size increases, the variability of the complexity measure outcome over 10 sampling repetitions significantly decreases. This finding emphasizies that the empirical data complexity measures are subject to similar considerations for sampling variability as other empirical estimates calculated on finite samples.

Overall, the pattern between the data complexity measures and the α and λ parameters is clear, at least for the larger sample sizes of $n = 50$ and $n = 500$: the main driver of the complexity measures is the degree of class overlap λ. With high

overlap ($\lambda = 0.1$), the decision boundary complexity is almost irrelevant. However, with sufficiently low overlap ($\lambda = 1.5$ or $\lambda = 10$), the degree of decision boundary complexity (represented by α) can impact the complexity measures. Greater decision boundary complexity (higher α) typically increases the values of the data complexity measures, indicating greater complexity, although in some cases it appears to decrease it; this seems to be a result of sampling variability, as the patterns stabilize for most of the complexity measures with the largest dataset (Figure 3.13).

Since these complexity measures are highly correlated, and in some cases computing almost identical values, we perform Principle Component Analysis (PCA) to explore underlying patterns or lower-dimensional structures in the set of complexity measures, inspired by the insightful analysis done in Ho and Basu (2002). Note that while PCA has limitations when used as a feature engineering technique to improve regression or classification performance (see Jolliffe (1982) and section 4.2 on feature engineering for more details), our use of PCA here, namely, to explore lower dimensional structure underlying a set of features, is entirely appropriate.

To perform the principle component analysis, we first calculate the set of principle components for a 9×17 data matrix. The rows of the data matrix are the 9 hyperparameter configurations of (α, λ) for the data generation process, and the columns of the matrix are the means for each of the 17 data complexity measures, averaged over 50 replications of the DGP. Note that we averaged the complexity measures over 50 iterations of the DGP in order to have a more stable result for the mean complexity measure used for the PCA calculation. Since the rank of the matrix is at most 9, we obtain 9 principle components from this process, and visualize the first four principle components using two 2D plots, similar to the analysis performed in Ho and Basu (2002).

This process is repeated twice, one time for the data complexity measures calcu-

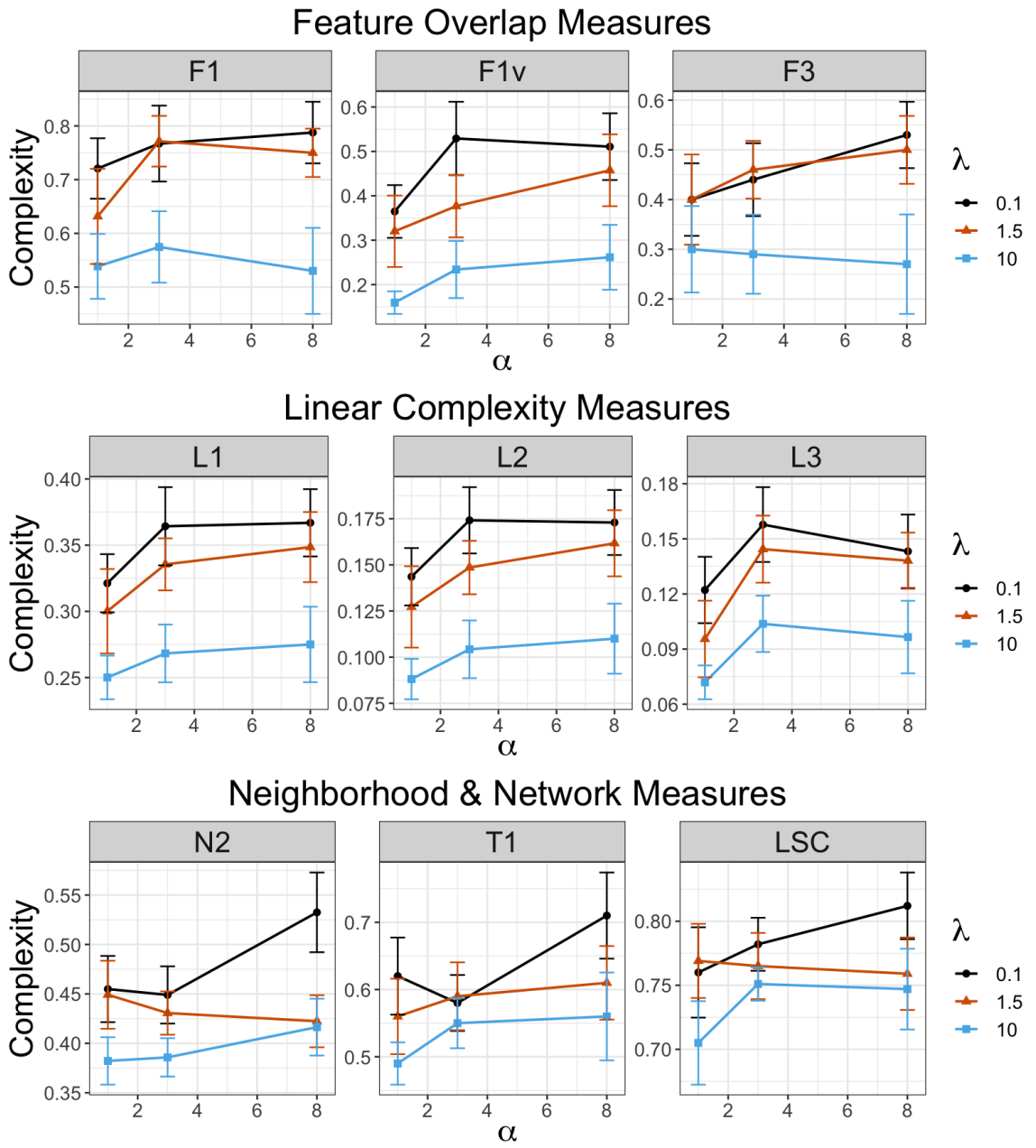

Figure 3.10: Complexity profiles on a subset of complexity measures, calculated for data sets of size $n = 10$ over 10 repetitions. The x axis represents the decision boundary complexity parameter α, with higher α indicating greater complexity. Each line shows complexity values for a different value of the class overlap parameter λ. The complexity measures have extremely high variation due to the small sample size, as shown by large bars indicating the standard error of the mean complexity measure over 10 repetitions of sampling.

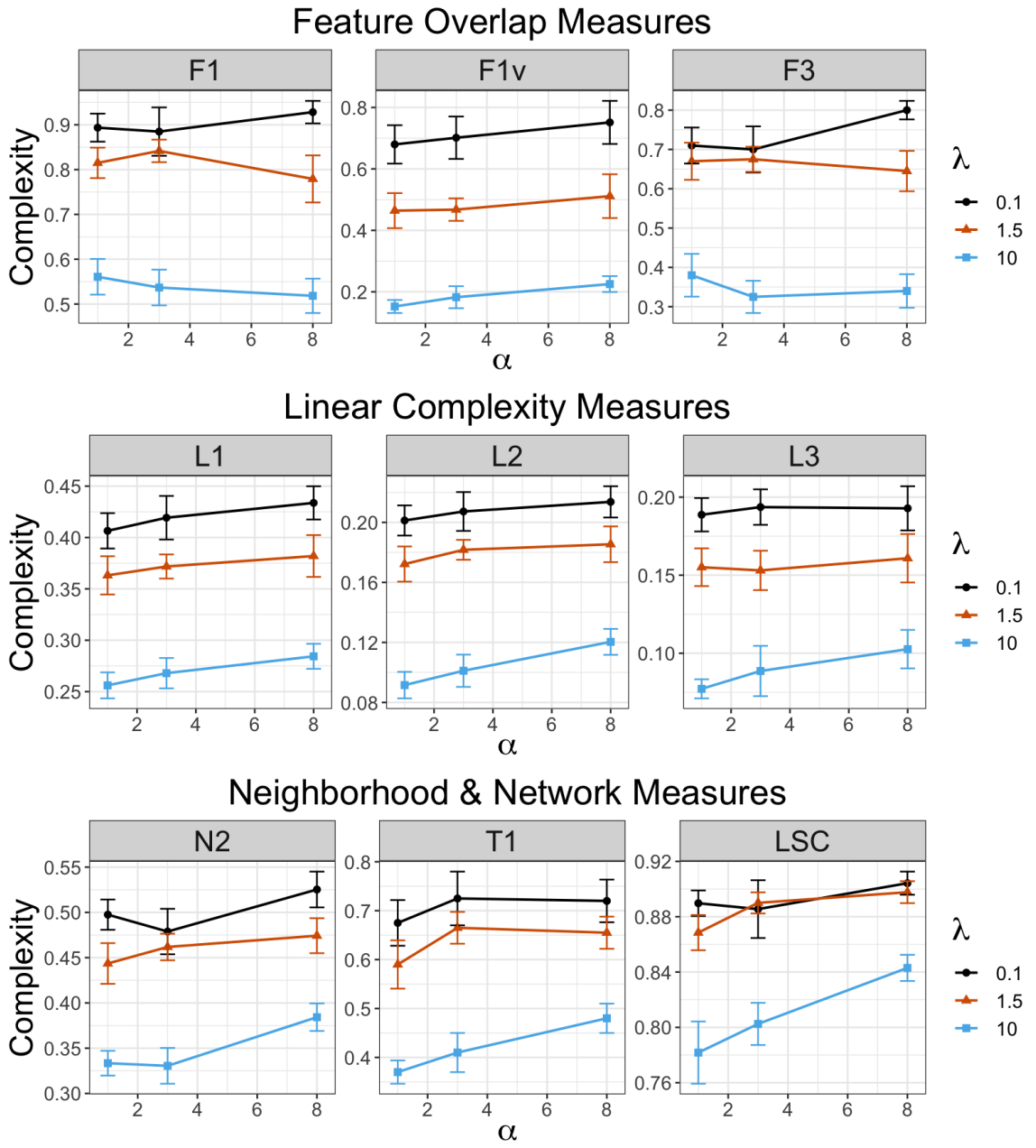

Figure 3.11: Complexity profiles on a subset of complexity measures, calculated for data sets of size $n = 20$ over 10 repetitions. This figure demonstrates less variability (smaller standard error) compared to Figure 3.10, and the patterns of complexity related to the overlap parameter λ and the decision boundary complexity parameter α begin to emerge. These patterns become fully clear by $n = 50$ and are described in more detail in Figure 3.12.

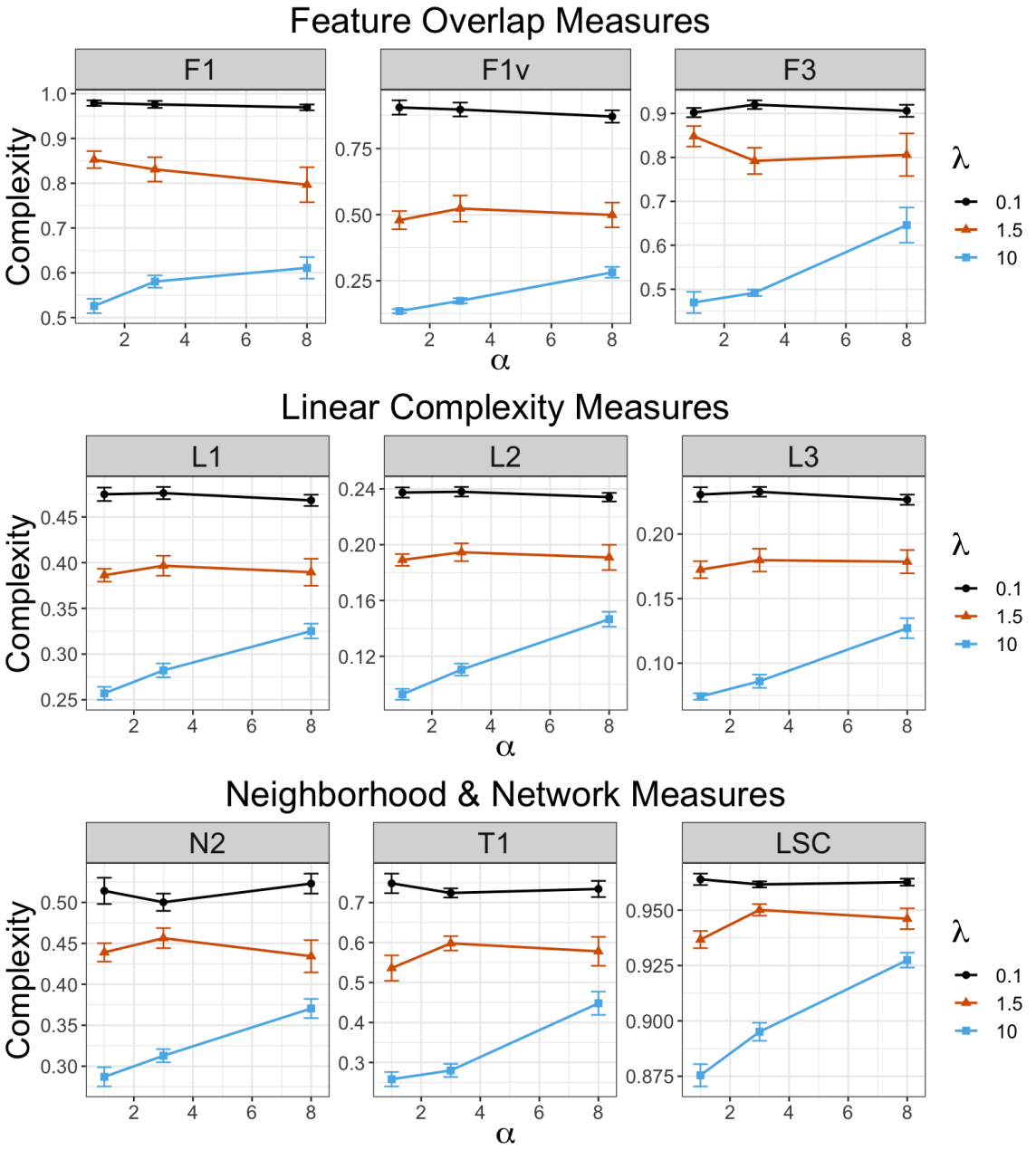

Figure 3.12: Complexity profiles on a subset of complexity measures, calculated for data sets of size $n = 50$ over 10 repetitions. The overall level of the data complexity measure is determined by λ, but within each value of λ (i.e. within each class overlap setting), the decision boundary parameter α has a further impact. The trend in decision boundary complexity is not monotonic in α when there is high class overlap ($\lambda \in \{0.1, 1.5\}$), but when the class overlap is minimal ($\lambda = 10$), there is a clear trend where complexity measures increase with greater decision boundary complexity.

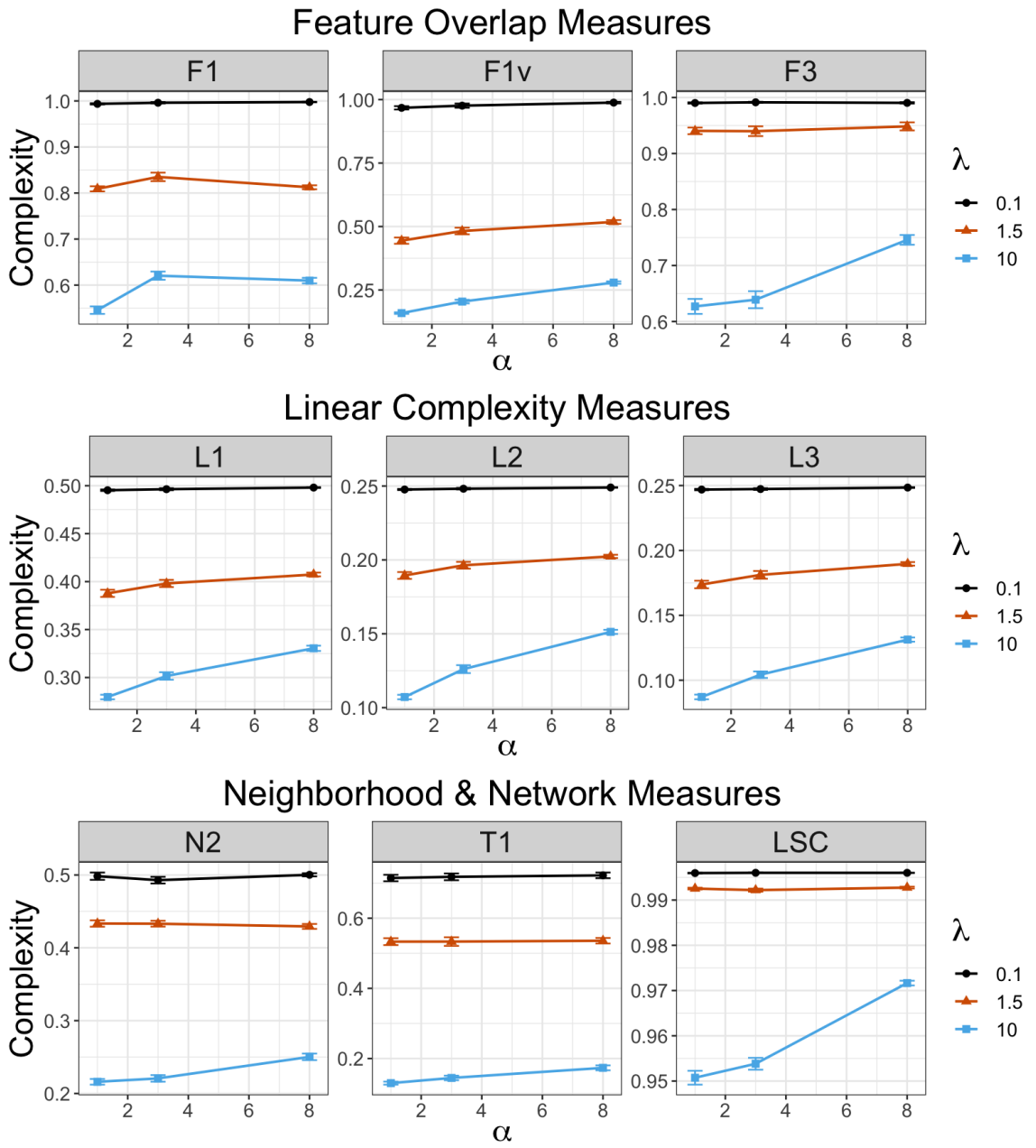

Figure 3.13: Complexity profiles on a subset of complexity measures, calculated for data sets of size $n = 500$ over 10 repetitions. We see similar patterns as in Figure 3.13, but with less variability in the complexity values over the 10 repetitions (as expected with a larger dataset). We also see that the trend in complexity measures as α increases is more stable in the lines representing higher class overlap ($\lambda \in \{0.1, 1.5\}$).

lated on data sets of size $n = 50$, and another time for the data complexity measures calculated on data sets of size $n = 500$. Figures 3.14 and 3.15 show, for data sets of size $n = 50$ and $n = 500$, respectively, the location of the 9 complexity configurations projected onto the subspaces of maximal variation of the data complexity measures. The configurations are labeled by their corresponding decision boundary complexity α and class overlap λ hyperparameters.

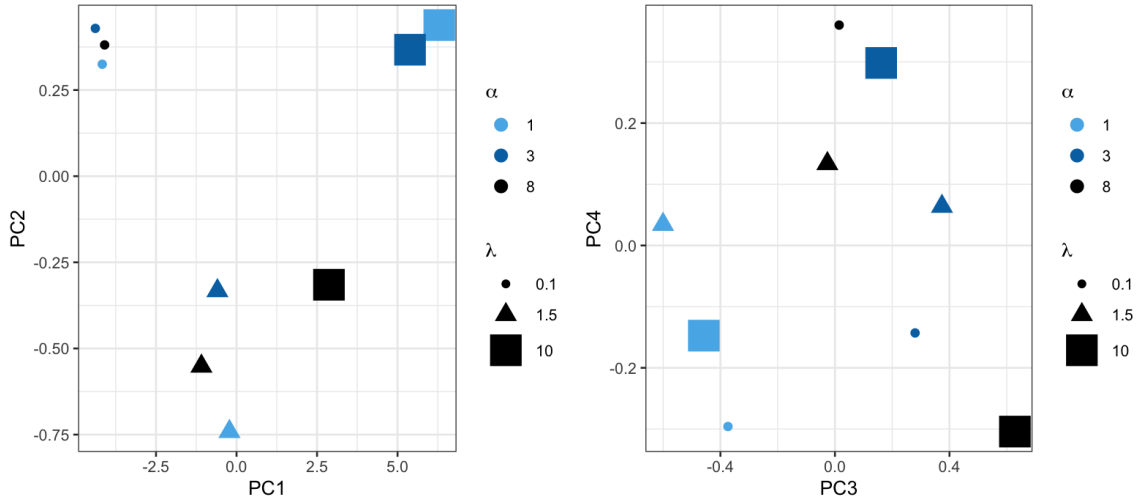

Figure 3.14: Complexity measures for the 9 data subgroups of size $n = 50$ per (α, λ) combination, projected onto the PCA space. Here we visualize the first four principle components.

Looking at the complexity profile for the 9 subgroups with dataset of size $n = 50$ (Figure 3.14), we see that there are 4 clusters in the first two principle components: the subgroups of greatest overlap ($\lambda = 0.1$, small circles, upper left); the subgroups of least class overlap and simplest decision boundary complexity (dark and light blue squares, upper right); and the remaining subgroups, having either medium class overlap, or low class overlap combined with high decision boundary complexity. The first principle component seems entirely related to the class overlap λ, based on the vertical separability of the data in that direction; this is to be expected since the highest amount of variation in the complexity measure values (based on Figure 3.12)

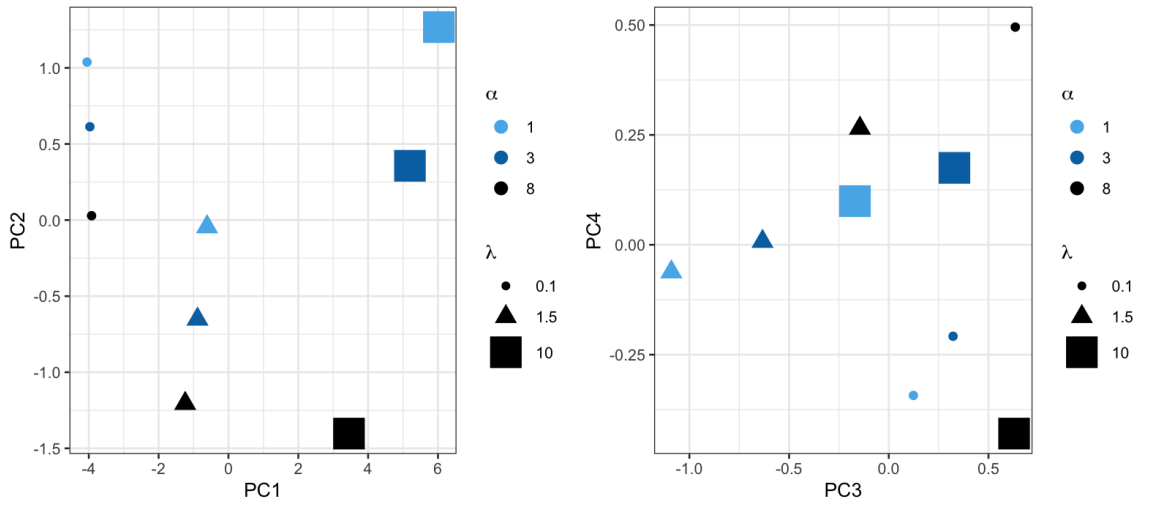

Figure 3.15: Complexity measures for the 9 data subgroups of size $n = 500$ per (α, λ) combination.

is driven by λ. Interestingly, we don't start to see pattern directly attributed to α until we get to the third and fourth principle component, where there is a geometric separation between low decision boundary complexity ($\alpha = 1$) and the other two cases.

When looking at the principle components for the 9 subgroups with a larger dataset of size $n = 500$ (Figure 3.15), it becomes much more obvious that α and λ are responsible for the two main components of complexity (at least, as measured by the complexity measures of Lorena *et al.* (2019)). The first principle component is primarily responsible for the difference by λ, again; however, within each λ subgroup, further complexity is monotonically captured by the second principle component. Thus, PC1 and PC2 capture the interaction effect of these two paramters on the overall complexity of the data, as measured by the 17 data complexity measures. In this case, going to the third and fourth principle components does not reveal any particular further structure or clustering.

To summarize, the PCA analysis demonstrates that the measures of maximal variation in the principle components are primarily related to the class overlap parameter

λ, and secondarily related to the decision boundary complexity parameter α. These patterns are more clearly visible when we have a large sample of data, the $n = 500$ case, which implies that not-obvious patterns of data complexity (i.e., the complexity determined by α) may be obscured in the empirical classification measures for small sample sizes due to sampling variability. Furthermore, take note that these data complexity measures are calculated on datasets with only 2 features, indicating a ratio of data to feature dimension of 25 even in the "small" sample case, which is fairly large. We discuss limitations of data complexity measures for very high dimensional data in the next and final section.

While this exercise in calculating data complexity measures has been confined to a single family of simulated data, the example is insightful because we have levers to directly impact both the class overlap and decision boundary complexity of the generated data. Here we see that the empirical measures of data complexity, which use only the data generated from these DGPs and no knowledge of the underlying distributions used to generate the data, or the true decision boundary, reveal patterns in classification complexity that correspond to these two levers, in particular when using sufficiently large sample sizes and few input features. With a large amount of data relative to the number of features in the data generation process, the data complexity measures provide valuable insights that align with theoretical notions of complexity purported by statistical learning theory and information theory.

3.6 Discussion

In this chapter, we presented a review of relevant literature from statistical learning theory and information theoretic divergence measures, that demonstrated from a theoretical point of view how classification difficulty is impacted by decision boundary complexity and class overlap. When a classification problem has a highly complex de-

cision boundary, a large amount of data is required to be able to learn a classification function that will have favorable properties, from among a high capacity hypothesis class. Lack of such data results in a large discrepancy between the risk (classification performance) of the learned function and the theoretically optimal risk. Furthermore, lack of large data makes the theoretical guarantees on true risk of the function, compared to the empirical risk learned from the data, either very weak (large bounds) or not holding with high probability. On the other hand, when a classification problem has high class overlap, the theoretical limit to the best possible performance of any algorithm, quantified by Bayes error rate, is much higher than when there is low class overlap.

We also reviewed empirical measures of data complexity, which provide a secondary look into classification complexity, offering indirect insights on both class overlap and decision boundary complexity. These measures of data complexity are limited in the information they can provide by the size and representativeness of the sampled data on which they are calculated; they are subject to variability due to sampling variation, particularly in small samples, as shown by the size of the error bars in Figures 3.12 and 3.13.

To demonstrate these concepts in a concrete example, we simulated data where the degree of class overlap and decision boundary complexity was explicitly controlled. We showed via this simulation that both the theoretical and empirical calculations measuring aspects of data complexity (VC dimension, KL divergence, and data complexity measures) aligned with the known complexity differences chosen in the simulation settings.

This work presents a novel look into the connection between notions of classification complexity espoused in the large bodies of literature in statistical learning theory and information theoretic divergence measures. This is also the first work,

to our knowledge, to explicitly compare the complexity quantified by the data complexity measures, to the complexity quantified by the long-standing concepts of VC dimension and KL divergence, in a simulation study where true classification complexity is known a priori.

Although we have focused for much of this chapter on class overlap and decision boundary complexity as separate aspects of classification difficulty, the literature on data complexity measures discusses at length the fact that decision boundary complexity and class overlap are not completely disparate and independent facets of classification complexity. These notions combine together, along with other challenges (e.g., class imbalance, data sparsity, noisy data) to create an overall, multifaceted concept of difficulty. The following quote from Santos *et al.* (2023) provides valuable insights on this point:

> Although we may argue that structural overlap measures focus on data characteristics unrelated to class overlap, in the sense that they describe other general properties of the domains (e.g., geometry, topology, density), we advocate that class overlap cannot be fully understood irrespective of structural information, since the global properties of the domains affect its identification and characterisation.

We reiterate that the nature of the internal structure of the classes and the complexity of the decision boundary inform the nature of the class overlap. Although the classification measures may relate more strongly to either class overlap or decision boundary complexity, many of them draw on both of these, and other aspects, to quantify how difficult a problem is. The complexity measures force us to see that these two aspects that we have focused on separately in the statistical learning theory and information theory contexts are actually quite interconnected. This insight is corroborated by the fact that the decision boundary complexity parameter in our

simulated example, α, had a varying degree of impact (as captured by the empirical measures), based on the underlying level of class overlap λ. Put more simply, the changes in the classification measures in Figures 3.12 and 3.13 are much more stark across the spectrum of α values when the class overlap is low (λ is high).

We finish the chapter with a discussion relating the themes discussed in this chapter to our paper "Digital medicine and the curse of dimensionality" (Berisha *et al.* (2021)). These connections help to place the results presented here in the context of digital health data, which is the application in which they will be utilized in the next chapter.

The key themes of the paper are 1) explaining the curse of dimensionality in the context of large digital data streams, 2) providing insights as to when this curse can and cannot be mitigated, and 3) giving recommendations for best practices to follow in building machine learning models, for the situation where it can be mitigated. As a brief background, the *curse of dimensionality* was first introduced by Richard Bellman as part of his work on dynamic programming (Bellman (1954)), and the term was coined in his book on control systems theory (Bellman (1961)). In modern machine learning, the term is used to refer to the fact that as the dimension of the input data (i.e., the cardinality of $\mathcal{X}$) grows, the same number of data become more and more sparse in the input dimension, thus obscuring patterns and structured relationships between $\mathcal{X}$ and $\mathcal{Y}$; see Theodoridis and Koutroumbas (2006) for more background on the curse of dimensionality.

In the context of digital health data, the curse of dimensionality does not only relate to the actual feature dimension of the incoming data, which can easily number in the millions for digital health streams, but also to the intrinsic dimensionality of the data. Intrinsic dimensionality, for our purposes, can be considered as the capacity of the hypothesis class required to learn an optimal function that matches

the true decision boundary of the underlying classification problem. The presumption in the digital health space is that the hypothesis class of the underlying classification problems attempting to be solved with these digital health streams is very high. This assumption can be attributed to both the signal complexity of the incoming data (in the sense of amount of information contained in the signal, see Zvonkin and Levin (1970) and López-Ruiz *et al.* (1995)), and the multi-faceted sources of information that contribute to human health.

The "curse" of dimensionality can be understood using our findings from statistical learning theory: when a classification problem warrants a high capacity hypothesis class, a proportionately large amount of data is required to learn the optimal function from this hypothesis class with any degree of success. Unfortunately, it is frequently the case in applications of machine learning to digital medicine that models are fit with only hundreds, and frequently just tens, of datapoints (at least for community researchers), owing to the scarcity of publicly available digital health data, and the expense incurred in collecting proprietary digital health datasets (for all but the largest private companies). These sample sizes are insufficient to support learning a high capacity classification function that will possess desirable properties, such as generalization to new data.

In Berisha *et al.* (2021), we demonstrate this point using an intuitive visualization and the concept of "blind spots", but it has been more rigorously demonstrated in the present chapter via the theoretical results presented in Section 3.2, particularly the leading examples of Equations (3.9) and (3.13). While the paper considers three causes for blind spots, this current work does not consider the third, which is a biased sampling strategy that produces data which systematically does not reflect the true nature of the underlying probability distribution $P(\mathrm{X}, Y)$. This can also be considered a form of covariate shift and is outside the scope of the present work.

Note that Equation (3.13) relates to how similar the empirical performance and the actual performance (which will be reflected in future unseen data) of the learned function are. This connection offers direct insights on the concept of generalization, which is a recurring theme in Berisha *et al.* (2021); here, we have shown theoretically that the curse of dimensionality has a direct impact on a model's ability to generalize.

The last section of the paper is devoted to strategies for mitigating the curse of dimensionality where possible, at each stage of model development and validation. The first strategy, applied to the stage of data collection, is to design a thoughtful and appropriate data collection protocol, and this is the main theme of the next chapter. In the next chapter, we provide an extensive analysis on speech data from cognitively normal and impaired participants, which demonstrates the benefit of using maximum performance tasks (recommended in Berisha *et al.* (2021)) for reducing classification complexity, and thereby increasing classification performance.

Another stage of model development around which we do extensive analysis in the next chapter is feature engineering. We provide a formal context in which two recommendations from the paper, namely domain expertise based features and transfer-learning based features, are justified as useful feature engineering strategies.

As a last connection on recommendations from Berisha *et al.* (2021), we discussed at length in this chapter divergence measures, which are recommended in the paper as a way of monitoring covariate shift between data used for model training compared to real-world deployment. While the use case (monitoring covariate shift) is different than our use case of measuring class overlap, the underlying theory related to divergence measures is the same, and the background information on f-divergences applies to this use case as well as our main one.

Finally, we make a note on the impact of dimensionality on the data complexity measures. As these are empirical measures calculated on a finite dataset, they are also

subject to the problems arising with high-feature, small-sample data settings. Because structured patterns and relationships between X and Y can become obscured due to data sparsity, empirical measures meant to quantify these patterns can also become meaningless given a low enough data-to-feature ratio. For example, the complexity measure F2, volume of the overlapping region, is calculated by multiplying together overlap regions of individual features; exponential decay means that this measure quickly tends to 0 in high dimensional datasets, severely reducing its usefulness. Similarly, F4, collective feature efficiency, has less value due to the possibility of datapoints from different classes being able to be spuriously separated from one among hundreds or thousands of features.

The limitations of some of the data complexity measures in high dimensional data are mentioned in Lorena *et al.* (2019), and are also directly referenced in other works on data complexity measures. For example, Mercier *et al.* (2018) note that although there are clear patterns between their classification measure *degOver* and actual classification performance on simulated datasets with simpler class boundaries and fewer data dimensions, the conclusion is less clear for the simulated datasets with higher dimensionality; in this example, the highest dimensional datasets contained only 40 features for 1500 dimensions.

Another limitation of using the data complexity measures for high-dimensional datasets is that many of the large scale analyses are done on datasets mostly having relatively large data-to-feature ratios. For example, Sáez *et al.* (2013) use data complexity measures to predict the usefulness of noise filtering, performing their analysis on 17 classification datasets; the smallest n/p ratio among the 17 is 3.5, but almost all are above 50. This limits our ability to transfer the lessons gained, via the semi-supervised learning approach described at the end of Section 3.4, to only datasets that also have a relatively large data-to-feature ratio. In particular, we cannot imme-

diately apply the knowledge to digital health datasets having thousands of derived features and only hundreds of datapoints.

In the next chapter, as a secondary analysis to analyzing classification performance under specific task and feature engineering protocols, we calculate the same empirical measures of data complexity on each of the datasets and explore their usefulness, paying special attention to the the impact of feature dimensionality on the complexity measure correlation with actual out-of-sample performance.

Chapter 4

TASK ENGINEERING FOR SPEECH-BASED SCREENING TESTS

4.1 Overview

In the last chapter, we explored two aspects of classification problem difficulty via both theoretical and empirical approaches, namely decision boundary complexity and class overlap. Understanding the nature of the factors impacting classification difficulty is critical for designing data-driven screening tests, as the screener performance is a direct consequence of the inherent complexity of the classification problem underlying the condition being screened. In particular, an understanding of the sources of classification complexity can be used to inform prioritization of efforts around designing the test to reduce classification complexity.

The problem of interest in this chapter is designing a speech-based screening test to detect cognitive impairment. In the general setting of digital screening tests for detecting medical conditions, methods for understanding classification complexity are of particular importance. Unlike traditional item-based exams, in which each item is constructed to directly tap into the condition being assessed, a digital data stream in its raw incoming form is usually not directly usable as input to a screening test; each individual sample is meaningless with respect to the underlying condition. Instead, the raw data stream must first be transformed to lower dimensional, meaningful features, which are then fed as input to the classification model that comprises the screening test. Considerations for the best data collection and subsequent feature extraction protocols immediately arise, and an understanding of the factors impacting classification complexity can inform the attention paid to either or both of these steps

during test design.

The novel contributions of this chapter are two-fold. First, we undertake a large scale comparison of the classification complexity underlying different combinations of data collection and feature engineering protocols, on a medical speech dataset consisting of cognitively impaired and cognitively normal participants[1]. The novel contribution of this analysis is to bring attention to the relative impacts that both data collection and feature engineering can have on the classification complexity of the extracted features, in an analysis comparing a larger number and much broader variety of tasks than has been performed in the speech literature to date. We particularly highlight the importance of designing a good data collection protocol, and demonstrate that it is this step of model development and validation that can shift the upper bound of potential classification performance for a given speech-based screening test setting.

Second, we propose a method for systematically and objectively discovering the aspects of the data collection tasks that are driving the performance gains measured in the large scale comparative analysis. These aspects can then be incorporated to a greater degree in further data collection protocol development. This analysis represents a new approach to designing the data collection step when developing a data-driven digital screening test.

Throughout, we bring together ideas from statistical learning theory, design of experiments, and machine learning, to aid in the design of of digital speech-based screening tests for neurological impairment. We use the term *task engineering* to

[1]Throughout this chapter, the analyzed features and tasks include both publicly available and proprietary algorithms. The analysis using proprietary algorithms has been made possible through collaboration with Aural Analytics, Inc., a speech analytics company that seeks to provide meaningful clinical insights for brain health using recorded speech. Aural Analytics was founded by two professors at Arizona State University, Dr. Visar Berisha and Dr. Julie Liss.

describe the process of designing the data collection protocol for the screening test.

For the remainder of this section, we introduce the specific classification problem that we aim to solve with our digital screening test, and provide general background on speech as a digital biomarker.

Human speech has shown impressive potential as a digital biomarker of neurological health (Fagherazzi *et al.* (2021)). In diverse neurodegenerative diseases, including amyotrophic lateral sclerosis (ALS) (Maffei *et al.* (2023)) Parkinson's disease (Rusz *et al.* (2021)), and Alzheimer's disease (de la Fuente Garcia *et al.* (2020)), speech has proven relevant in clinical applications ranging from disease diagnosis to longitudinal tracking of decline over time.

We will consider specifically the task of using speech as a digital biomarker for cognitive decline. The relationship between speech/language and cognition has been demonstrated in myriad studies (Mueller *et al.* (2018), Geraudie *et al.* (2021), Meilán *et al.* (2020), Martínez-Nicolás *et al.* (2021)). To simplify the problem setting, we focus on a speech-based screening test that determines whether the screened individual is cognitively normal (CN) or cognitively impaired (CI), with the cognitively impaired group being recommended for further testing. This reduces the problem to one of binary classification, also known as pattern recognition or pattern classification.

Human speech is sampled at anywhere from 16k to 44k samples per second, meaning that a 1-minute speech audio recording can result in a corresponding feature vector (of discrete amplitude measurements) of size up to 2.64 million. The level of complexity of this raw data make classification using the amplitude features extremely difficult. As a result, and as described earlier in this chapter, the speech signal must be transformed into a lower dimensional feature vector before making a final classification decision.

Two options for feature engineering are possible. In the first, features are first

extracted from the audio, and then provided as input to a separate classification model. Figure 4.1[2] demonstrates this method. The second is to employ an end-to-end model in which feature engineering is done as part of the classification model optimization process. Deep neural networks, such as the Wav2Vec2 (Baevski *et al.* (2020)) and BERT-family models (Devlin *et al.* (2018), Acheampong *et al.* (2021)), employ this mode. See Figure 4.2 for a visual demonstration.

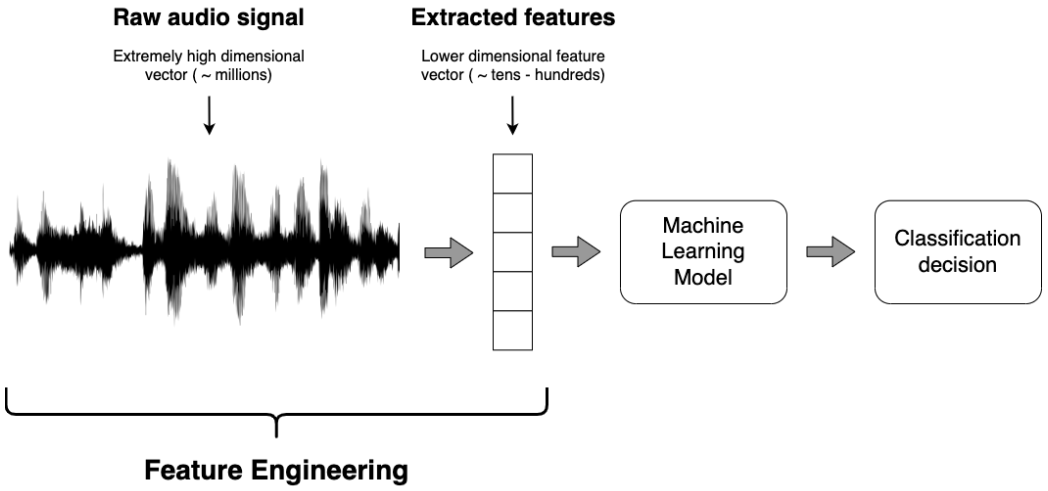

Figure 4.1: Flowchart showing the process of turning a raw audio signal into a classification decision using a separate feature engineering step.

In contrast to the separate feature engineering step, which normally utilizes hand-crafted features designed using clinical insight and domain expertise, the end-to-end approach requires staggering amounts of data (e.g., thousands of hours of speech) to train the network to produce meaningful speech representations in a purely data-driven manner. As clinical speech data is extremely limited (e.g., tens to hundreds of hours of speech in a medical speech dataset), the end-to-end approach is only feasible via transfer learning using models already trained on large quantities of normal speech.

In both of these regimes, the emphasis is on the particular feature engineering methods and classification models; the set of audio samples is considered fixed, often

[2]Image of audio wavefile by Gordon Johnson from Pixaby (https:pixabay.comusersgdj-1086657).

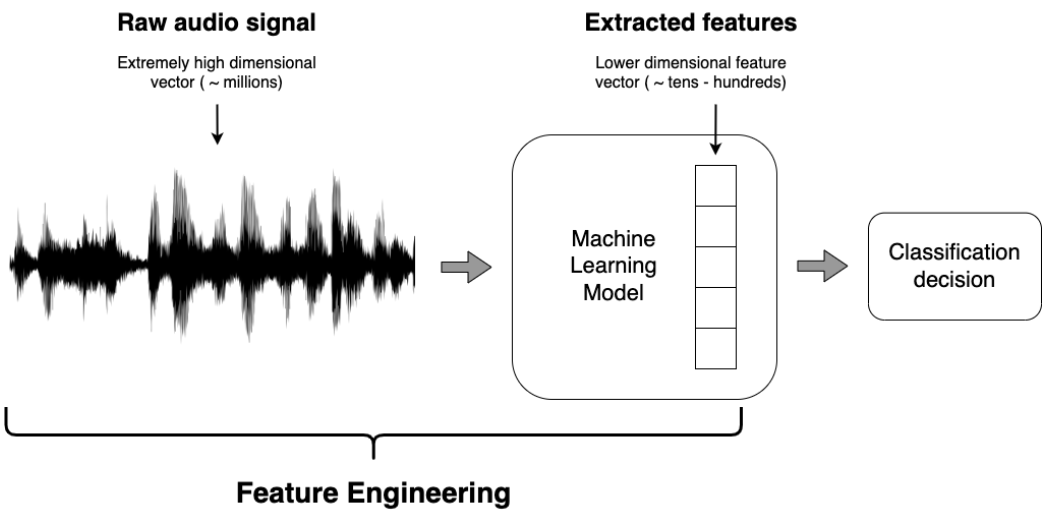

Figure 4.2: Flowchart showing the process of turning a raw audio signal to a classification decision using a built-in feature engineering step.

without particular attention paid to the way in which speech is collected, which we call the *speech elicitation task*. A speech elicitation task is the activity that the participants perform while their speech is being recorded. Passive speech recording involves recording participants as they naturally do whatever they ordinarily do; getting a large amount of passively recorded speech induces understandable privacy concerns. The alternative to passively collected speech is actively collected speech, in which the participant is asked to perform a specific task, and their responses to that task are recorded. Examples of speech elicitation tasks include reading a sentence or paragraph, describing a picture, or being presented with a series of words, pictures, or a story and recalling that content after presentation.

The intense focus on feature engineering and its encompassing activities, and the minimal attention paid to task engineering, is likely due to the pure difficulty for most speech scientists of performing in-house studies to collect speech using speech elicitation tasks of their own design. This difficulty is severely compounded when seeking to design a speech-based screening test for difficult-to-access clinical popula-

tions having a particular medical diagnosis. Speech scientists often must settle for whatever collected speech is available, either via publicly available repositories, or via proprietary datasets owned by their particular academic or professional networks.

One of the major contributions of this chapter is to highlight the need for a paradigm shift, in which practitioners designing speech-based screening tests consider engineering of the speech elicitation task (i.e., the data collection protocol) as a crucial first step in algorithm design. It is our fundamental conviction that the greatest amount of utility for using speech as a clinical tool will be achieved by careful engineering of both the speech elicitation task *and* the extracted speech features and classification models, rather than relying on feature engineering or model exploration in isolation; this is demonstrated in extensive analyses to follow. Figure 4.3 shows a visual representation of this paradigm. This figure is a comparator to the regime of feature engineering as a separate step (Figure 4.1), but the same principle applies to the end-to-end regime (Figure 4.2).

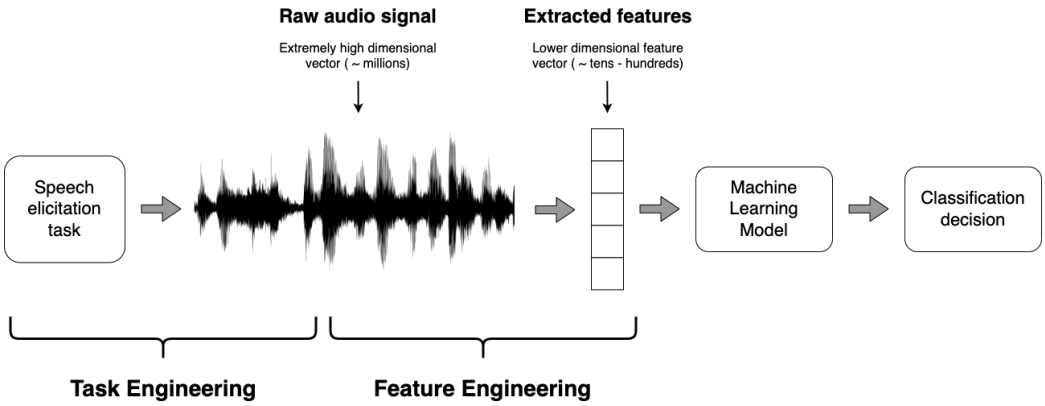

Figure 4.3: Flowchart emphasizing the engineering of the speech elicitation task as a crucial first step to speech-based screening test design.

To describe our goal in the language of the previous chapter, both task+feature engineering or feature engineering alone, the usefulness of the resulting speech features for a speech-based screening test is based on whether the engineered features reduce

the complexity of the classification problem of predicting cognitive status from the engineered features. We rely on our insights on classification complexity from the previous chapter to assess the difficulty of the classification problem resulting from the extracted features. While classification complexity encompasses many facets, we are particularly interested in whether the resulting features reduce classification complexity via smaller class overlap and lower decision boundary complexity.

To close this introduction, we discuss the connections between speech elicitation task design and two related fields: neuropsychological testing and design of experiments.

Although novel in the speech-based machine learning community, the idea of designing targeted tests that can differentiate between populations of interest has a long and rich history in neuropsychological cognitive testing. When referred for cognitive assessment, patients are not passively observed by a neurologist, but rather are asked to perform a series of specific tasks that tax particular function of different cognitive domains. Thus, our emphasis on careful design of the speech elicitation tasks used in a speech-based cognitive screening test has a foundation in existing scientific literature, and one of our contributions is bringing existing ideas from traditional cognitive assessment to the speech machine learning community. See Section 4.2.2 for a deep dive into examples of traditional cognitive assessments and their design.

On the other hand, the phrase "task design" immediately evokes for statisticians the broad field of Design of Experiments (DOE). A natural question arises: what prevents existing methods from DOE from being immediately applied to our use case? Briefly, DOE differs in two respects from our setup of designing speech-based elicitation tests. First, the causal chain between the independent and dependent variables in the model is reversed compared to DOE. Our outcome variable (cognitive impairment status) causally impacts the independent variables (speech features); the

reverse is the case in the traditional DOE setting. Second, we cannot directly sample the model inputs (speech features) at locations of interest, for example at locations that maximize variance in some way. Instead, we must decide how to change the speech elicitation task, and then simply observe how the subsequent speech features change as a result. These ideas, along with a more thorough review of DOE, are detailed in Section 4.4.1.

In the next section of this chapter, we lay out some formal definitions, and provide a brief literature review on feature engineering in speech analysis and task engineering in standard cognitive testing. We also discuss the ability of feature and task engineering to reduce the classification complexity of the resulting features, drawing on concepts from the previous chapter to formalize the discussion. In the third section of this chapter, we perform the large scale analysis comparing classification complexity under different task and feature engineering protocols. Finally, in the last section, we propose a method for deriving automatic insights that can be used to guide future task engineering. We compare a decision tree approach with a Bayesian treed classification method to obtain objective insights into the characteristics of the speech elicitation task that produce simpler classification problems from the resulting speech features.

4.2 Background on Feature and Task Engineering

In this section we first present prior work on feature engineering, in particular engineering of speech features from speech data. We subsequently review the sparse literature on speech elicitation task design, and compare to the process of designing neuropsychological cognitive batteries that has been established over the course of decades. Finally, we highlight how, and in what context, feature and task engineering can be undertaken to improve the properties of the resulting classification

problem, which in turn impacts the performance of a screening test designed to detect cognitively impaired individuals.

4.2.1 Feature Engineering in Applications to Speech

In typical classification or regression problems, feature engineering or feature selection is a frequently used preprocessing step for reducing the dimension of the input space. The nominal purpose of feature engineering or selection is to reduce model variability, thereby increasing performance on unseen data via the classic bias-variance tradeoff (Hastie *et al.* (2009), Section 2.9). In speech analysis, before this step can take place, one must first transform the amplitude values of the recording in the time domain, into a much lower dimensional set of speech features that contain meaningful information from the speech sample. Feature selection on the amplitude values themselves is meaningless, as each individual discrete time sample is only useful in the context of the signal in the surrounding time frame.

Formally, Z will denote a random vector of amplitude values comprising the original audio recording; X is a random vector representing a set of speech features extracted from this original audio. The support of Z is $\mathcal{Z}$, with a single instantiation denoted z. Similarly, X has support $\mathcal{X}$, with x denoting a single instance of these features for a particular audio recording.

The function $f_{feature} : \mathcal{Z} \to \mathcal{X}$ is the mathematical transformation that turns the audio recording into a set of speech features. For a speech recording sampled at a rate of 16k Hertz, the cardinality of $\mathcal{Z}$ is $16000 \cdot L$, where L is the maximum recording duration (in seconds) for the speech elicitation task from which $\mathcal{Z}$ is recorded. The cardinality of $\mathcal{X}$ depends on the particular speech feature engineering approach employed, but typically ranges from tens to thousands.

The function $f_{feature}$ is a transformation such that the resulting features $\mathcal{X}$ repre-

sent an informative component of speech production. Two types of feature engineering approaches we will discuss are so-called hand-crafted features (shown in the diagram in Figure 4.1) and built-in features calculated as part of an end-to-end classification model (shown in Figure 4.2).

Hand-crafted features based on the acoustic properties of the recording have been explored for decades in signal processing applications Kahrs and Brandenburg (1998), Crocker (1998)). Classic acoustic features include jitter, shimmer, fundamental frequency F0, and harmonic-to-noise ratio ((Bielamowicz *et al.* (1996)), along with Mel-Frequency Cepstral Coefficients (MFCCs) (Mermelstein (1976)). Other hand-crafted features include clinically-relevant and interpretable measures of speech, such as speech duration, speaking rate, or number of pauses (Meilán *et al.* (2020)). Hand-crafted language features can also be extracted using an intermediary transformation from the audio recording to a transcript of the words that were spoken; features based on language can then be extracted from the transcript, such as word counts, part-of-speech counts and ratios, or type-to-token ratio (Bucks *et al.* (2000)).

In recent years, end-to-end models based on Deep Neural Networks (DNN) that utilize the original speech recording Z as input have surpassed the performance of models that rely on hand-crafted features, in applications including automatic speech recognition (Chiu *et al.* (2018)), speaker recognition (Bai and Zhang (2021)), and emotion recognition (Papakostas *et al.* (2017)). Examples of the most successful and popular DNN models, which are publicly available to download and have been trained on hundreds and in some cases thousands of hours of speech, include Wav2Vec2 from Facebook (Baevski *et al.* (2020)) GPT4 from OpenAI (OpenAI (2023)) and its predecessors, and the BERT family (Devlin *et al.* (2018), Acheampong *et al.* (2021)). Wav2Vec2 uses the audio as input, whereas GPT3 and the BERT family use transcripts of the speech recording as input. Although these models are intended to

be used after fine tuning on a small amount of additional speech (or transcripts), the neural network can also function as a feature extractor, by sending an audio file or transcript through in a feed-forward pass and extracting the values from the nodes in an intermediate or last layer of the network. These DNN-based features can then be treated as a typical set of speech features and evaluated alongside sets of hand-crafted features.

Although feature engineering (both hand-crafted and built-in) reduces the dimensionality of the input space from millions to tens or thousands, further mathematical transformations (feature engineering) or feature selection can be performed to further reduce the input space prior to model fitting. We will combine all such operations in a function denoted as $f_{select} : \mathcal{X} \rightarrow \mathcal{X}'$; we use the subscript *select* for simplicity, although this step may also involve further mathematical transformations, such as Principle Component Analysis (PCA), rather than pure feature selection.

For a comprehensive review on best practices for feature selection and feature engineering, we refer readers to Section 3.3 of Hastie *et al.* (2009) for a discussion on feature selection in linear models, and to Zheng and Casari (2018) for a more general and modern treatment.

4.2.2 Task Engineering and Cognitive Battery Design

Here, we introduce relevant notation and prior work on speech elicitation task design, along with design of gold standard tests for measuring cognition. As described in Section 4.1, audio recordings can be obtained using an active speech elicitation task, in which the participant is asked to perform a specific task, or a passive speech elicitation task. In this analysis we will focus on comparing different types of active speech elicitation tasks.

Formally, each audio recording Z is produced by asking a participant to perform

a specific speech elicitation task $\mathrm{T} = (T_1, T_2, \ldots, T_m)$. The task is comprised of components T_i, which are meta-features describing a particular aspect of the speech elicitation task. One instance of a task realization is t, and the support of the task space is denoted $\mathcal{T}$. Examples of the task meta-features include the recording duration, whether or not specific cognitive domains are taxed (e.g. memory, retrieval, visuospatial reasoning), or other characteristics of the stimuli that the participant is responding to during the task.

In applications involving automated speech analysis, works describing the process of designing and refining a speech elicitation task to be used in a particular speech-based screening context are essentially nonexistent, to the best of our knowledge. The closest literature involves formal comparisons of different speech elicitation tasks for the same learning problem; these are themselves scant, and mostly focused on connected speech tasks. *Connected speech* means continuous speaking in full sentences, similar to normal conversation. Sajjadi *et al.* (2012) explored semi-structured interviews compared to picture descriptions, and found that the interviews elicited greater differences in morpho-syntactic features, whereas the picture descriptions were more sensitive to semantic features. Beltrami *et al.* (2016) compared picture description tasks to two personal narrative tasks (describing a typical work day and describing a dream) for Italian speakers and found that the picture description task had higher accuracy using features derived from the picture description task. Seçkin and Savaş (2023) compared the differences in extracted features between three different picture description tasks on healthy Turkish-speaking individuals only, and found that the Accident Scene and Picnic pictures had superiority for different sets of features, and were both better than the Cookie Theft picture. Bose *et al.* (2022) compared a wordless story telling task based on a picture book (Frog Story) to a picture description task, and found that the Frog Story task detected differences between CN and AD

participants on a larger number of features.

Clarke *et al.* (2021) performed the most wide-spread comparison to our knowledge; they compared five different connected speech tasks on CN, AD, and MCI (mild cognitive impairment) participants, reporting classification metrics for the different tasks on multiple classification sub-problems. The authors also analyzed which features were the most useful for different types of connected speech tasks. The explicit emphasis on the importance of task selection in speech-based classification analysis, along with the impact of the speech elicitation context on the extracted features and their subsequent usefulness, is the first of its kind and is closely related to the present work, at least among our investigation of prior literature. Employing a similar emphasis on task importance, Martínez-Ferreiro (2022) discuss connected speech tasks compared to naming tasks for the purpose of eliciting naming deficits. The authors explicitly analyze the usefulness of the two task types individually and in conjunction, for the purposes of longitudinal monitoring and cross-sectional group differentiation, and concluded from a systematic review that the combination of both task types was most effective in achieving these goals.

Of the works discussed above, the few works that stress the importance of task selection are all very recent. Noticeably, all of these studies were performing pure task comparison, and at most providing insights on which features work better for which tasks, rather than suggesting directions for future task design based on an empirical and clinically-informed analysis. The current work is the first work to propose proactive speech elicitation task *design* (rather than comparison alone) using ideas from both machine learning and neuropsychology, via automatic discovery of the elements of the speech elicitation task that may be driving performance differences.

The likely cause for the dearth of literature in design of speech elicitation tasks is the difficulty of acquiring speech recordings on a task of one's choosing, particularly for

a classification problem in a medical application, such as a cognition-based screening test. Most machine learning researchers provide results on "found speech", meaning either publicly available speech repositories, or proprietary speech datasets to which they have access, as evidenced by the deluge of machine learning publications related to speech-based classification problems on standard datasets such as DementiaBank (Becker *et al.* (1994)) or ADReSS (Luz and MacWhinney (2020)). As a result, in the speech-based machine learning community, the emphasis is placed on best practices for feature engineering, model selection, and model training, rather than on designing a speech elicitation task that produces good audio for the classification problem in the first place.

Although test design is not a deeply embedded concept in machine learning applications of speech, the concept of design is not new in the field of cognitive neuropsychology. There are a number of neuropsychological batteries that make up a standard cognitive assessment administered when cognitive concerns arise. Examples of commonly administered batteries include the Mini-Mental State Examination (MMSE) (Folstein *et al.* (1975), Tombaugh and McIntyre (1992)), the Boston Naming Test (Goodglass and Kaplan (1972)), the ADAS-Cog assessment (W G Rosen (1984), Cano *et al.* (2010)), the more recent Montreal Cognitive Assessment (MoCA) (Nasreddine *et al.* (2005)), and others. Each of these batteries has been carefully designed and subsequently refined over the course of years to ensure validity, reliability, and sensitivity to cognitive impairment.

For demonstration, we briefly review the original design and validation of the MMSE and MoCA, two of the most commonly administered cognitive assessments, along with a short form of the Boston Naming Test (Lansing *et al.* (1999)).

The MMSE was first introduced in Folstein *et al.* (1975), in which the authors describe the "Mini-Mental State" (MMS, originally) as an abbreviated cognitive as-

sessment that can separate patients with cognitive impairment from those with normal cognition, as well as track longitudinal changes. The MMSE was designed to assess a wide range of cognitive function in a very short time (hence "Mini"), lasting only 5-10 minutes. A brief yet wide-ranging and easily accessible test, the authors postulated, would alleviate two concerns: 1) lack of access to intensive memory clinics for most patients, and 2) difficulty for patients with dementia to complete very lengthy neuropsychological exams. Besides brevity, the other goal in designing the MMSE was to offer insights on cognitive aspects that impact everyday functioning: orientation, memory, reading, and writing. Other standard tests from that time included tasks that taxed cognitive abilities necessary for school or work, such as digit symbol recognition or vocabulary-based tests, rather than ability of the patient to care for themselves.

In summary, the MMSE was designed to 1) provide information on patients' cognitive function necessary for everyday functioning, 2) take a short time to complete, and 3) be able to be readily adopted in varied clinical settings, providing greater accessibility to objective cognitive assessment.

The designers of the MoCA test, while also adopting the goal of brevity, sought to create a test for evaluating early cognitive complaints (Nasreddine *et al.* (2005)). The early stage of cognitive decline is difficult to detect with the gold standard MMSE, as it is geared to provide insights during the later stages of Alzheimer's disease and related dementias (ADRD) (Nasreddine *et al.* (2005)). The authors started designing the test using an initial version which taxed 10 cognitive domains related to mild cognitive impairment; this initial version was based on the authors' clinical experience. Some items with poor discrimination between cognitively impaired and normal patients were subsequently identified and replaced over the course of 5 years of use in-clinic.

Both the MMSE and MoCA were validated via reliability studies using repeated

assessment within a short period of time, along with comparisons to existing gold-standard clinical assessments, and sensitivity and specificity in discriminating between patients with cognitive impairment or normal cognition in prospective studies (Folstein *et al.* (1975), Nasreddine *et al.* (2005)).

Rather than use classification metrics for battery validation in a post-hoc analysis, Lansing *et al.* (1999) proposed a short form of the Boston Naming Test (BNT) that used these criteria for test design itself. The authors performed a stepwise discriminant analysis (with the target to discriminate between normal control (NC) and Alzheimer's disease (AD) groups) to arrive at a subset of 22 items for potential inclusion from the original 60-item. They furthermore discarded several items to achieve gender parity, arriving at a 15-item short form. In this example, the aim of the test design was to create an abbreviated version of the test that removed gender bias while maintaining the discrimination of the original long form.

In the examples above, the overarching themes of test design are creation of a test according to criteria specific to the context in which that test will be used. In the case of the MMSE, the goal was a short, objective test that would inform cognitive function in the late stages of ADRD. For the MoCA, the authors aimed to design a test for early stage cognitive impairment. In the short-form BNT, the authors chose items based on ability to discriminate between normal controls (NC) and patients with Alzheimer's disease (AD), while removing gender-biased items. In all of the examples, exam brevity was of major importance.

These examples highlight that speech-based screening tests for cognitive impairment should be engineered, or designed, to meet specific criteria that are important to the context in which the test will be administered. Considerations such as patient burden from lengthy tests, in particular for cognitively impaired patients, necessitate reasonably brief assessments, a continuing theme from our work on screening tests for

youth delinquency. Furthermore, cognitive tests should be tailored to assess cognitive functions important for a specific indication or stage of impairment.

While a speech task itself may be known to tax neurological function known to decline with cognitive impairment, the process of feature engineering should capitalize on this knowledge in order to make the screening test based on these features discriminative and, ideally, interpretable.These ideas are explored more formally in the next subsection.

4.2.3 Theoretical Impact of Feature and Task Engineering on Classification Complexity

In the previous subsections, we presented prior work on the process of speech feature engineering, speech elicitation task engineering, and cognitive test design, and pointed out that these engineering steps should be concentrated toward achieving a specific goal in the screening test setting. The targeted goal we will concentrate both task and feature engineering efforts on, for the time being, is to make the screening test as accurate as possible. This in and of itself is a challenging goal for a speech-based screening test to achieve, and is a worthy aim for a baseline screening test. Additional considerations such as test length (number of speech elicitation tasks) are of secondary importance for the moment. In order to achieve good classification performance, we seek to reduce the complexity of the classification problem implied by predicting cognitive status using the engineered speech features.

We previously discussed how increasing the class separability and reducing the complexity of the decision boundary are two ways of reducing classification complexity. Reducing classification complexity leads to improved model performance, along with better generalization on unseen data. Here we provide several more definitions to allow for a formal discussion around how and whether task and feature engineering

can impact classification complexity.

The class label, or cognitive group to which the participant who provided the audio recording belongs, is a random variable Y, with support $\mathcal{Y} = \{0, 1\}$; a single realization is denoted y. We use $Y = 0$ to denote participants from the cognitively normal (CN) group, and $Y = 1$ to denote participants who are cognitively impaired (CI), in particular who have been diagnosed with Alzheimer's disease.

The joint distribution we are interested in is

$$\begin{aligned} P(\mathrm{Z}, \mathrm{T}, Y) &= P(\mathrm{Z} \mid \mathrm{T}, Y) P(\mathrm{T} \mid Y) P(Y) \\ &= P(\mathrm{Z} \mid \mathrm{T}, Y) P(\mathrm{T}) P(Y), \end{aligned}$$

since the task presented to the participant is independent of diagnosis. We seek less overlap between the conditional distributions $P(\mathrm{Z} \mid \mathrm{T}, Y = 1)$ and $P(\mathrm{Z} \mid \mathrm{T}, Y = 0)$, which we previously measured using KL-divergence. We also prefer a simpler decision boundary, which is the surface defined by the equation

$$P(\mathrm{Z} \mid \mathrm{T}, Y = 1) = P(\mathrm{Z} \mid \mathrm{T}, Y = 0);$$

we previously measured decision boundary complexity by the VC dimension of the function family needed to accurately approximate the decision boundary function.

With these definitions in place, we take a moment to discuss how both feature and task engineering can impact the complexity of the underlying classification problem.

Recall that fitting a classification model on audio recordings Z according to the Regularization principle is done by finding $f_n : \mathcal{Z} \to \mathcal{Y}$ from a given hypothesis class $\mathcal{F}$ such that

$$f_n = \mathrm{argmin}_{f \in \mathcal{F}} R_{\mathrm{emp}}(f) + \lambda ||f||^2. \tag{4.1}$$

When we consider that 1) the size of the input space $\mathcal{Z}$ is on the order of millions, 2) the input features in their original form Z (amplitude of the speech signal at each

discrete time point) are not likely to be individually comparable from one audio sample to the next, and 3) the number of training examples in a *medical* speech dataset is usually on the order of tens to hundreds, choosing an appropriate hypothesis class $\mathcal{F}$ and then solving Equation (4.1) is an incredibly complicated classification problem. Recalling the bounds on empirical risk from eq. (3.13), having data size n on the order of hundreds (or tens) severely limits the complexity of the hypothesis class from which we can optimize f_n, and still reliably estimate its true risk $R(f_n)$. But, looking at the bias-variance trade-off in eq. (3.9), using a hypothesis class of limited complexity in order to reduce the estimation error (because of limited sample size), will result in a larger approximation error, due to the complex nature of the speech signal in its original form.

This dilemma is purportedly solved by adding a first step of feature engineering $f_{feature}$ to transform the process into a two-part optimization problem: find $f_{feature} : \mathcal{Z} \to \mathcal{X}$ and $g : \mathcal{X} \to \mathcal{Y}$ such that the composite function $g \circ f_{feature} : \mathcal{Z} \to \mathcal{Y}$ minimizes $R_{\text{emp}}(f) + \lambda||f||^2$ for $f = g \circ f_{feature}$. Because the function $f_{feature}$ typically transforms the speech signal into a much lower dimension (on the order of tens to thousands), the reasoning goes, the new problem of finding $g : \mathcal{X} \to \mathcal{Z}$ has a much more favorable data-to-feature ratio for reducing both terms in eq. (3.9) simultaneously.

Without further assumptions, this reasoning taken at face value obfuscates the fact that the feature transformation function $f_{feature}$ is itself a part of the function being optimized. The actual complexity of the hypothesis space containing the learned function is the hypothesis space containing $g \circ f_{feature}$, not g.

We can also look at this problem from the perspective of class overlap, which determines the theoretically lowest error that can be obtained, namely the Bayes error rate shown in eq. (3.16). Without extra assumptions on $f_{feature}$, we are still

limited in eq. (3.16) by the separability of the conditional distributions in the original space of audio recordings $\mathcal{Z}$, not $\mathcal{X}$, since $\mathcal{Z}$ is the domain of the composite function $g \circ f_{feature}$, which is the actual function being learned.

Thus, if the feature engineering function $f_{feature}$ is obtained in a purely empirical, data-driven manner from the same small set of data, without any additional assumptions in place, the fundamental complexity of the task at hand (in relation to eq. (3.9) and eq. (3.16)) is unchanged. Although the optimization process may be more successful, due to optimization techniques that can be applied to composite functions with specific forms that are not available in a standalone function optimization (see, for example Astudillo and Frazier (2019)), we still face a problem of the same complexity. The composite function $g \circ f_{feature}$ may either not be sufficiently complex to approximate the true optimum f^*, or if the complexity is sufficiently high, we may not be able to be reliably estimate its true risk on our small dataset; furthermore, the overlap of the conditional distributions $P(\mathrm{Z} \mid \mathrm{T}, Y = 1)$ and $P(\mathrm{Z} \mid \mathrm{T}, Y = 1)$ remains the same limiting factor.

A partial solution to this dilemma can be found by considering again the formula from (Bousquet *et al.* (2004)):

$$\text{Generalization} = \text{Data} + \text{Knowledge}. \tag{4.2}$$

In order for the feature engineering step $f_{feature}$ to reduce classification complexity, we need to add either more Data, or more Knowledge. These two terms relate to the two approaches to feature engineering described in the beginning of this section. Hand-crafted feature engineering relies on the Knowledge of the usefulness of the extracted features; this Knowledge is often call *domain expertise* (Berisha *et al.* (2021)). Domain expertise implies a scientific understanding of how particular aspects of speech and language are known to decline in a particular condition (in this

case, Alzheimer's disease). If those aspects of speech and language are known to differ in healthy versus impaired populations, and there is a reliable way of extracting those particular components of speech or language for audio recordings elicited by the speech task, then performing that feature engineering as a preprocessing step is likely to reduce the classification complexity. The feature engineering $f_{feature}$ reduces classification complexity by transforming the amplitude values Z into a low-dimensional (*reducing boundary complexity*) and group-differentiating (*increasing class separability*) set of speech features.

Built-in features acquired via transfer learning or semi supervised learning alleviate the dilemma by adding more Data. The end-to-end models that function as feature extractors are trained to perform informative speech-based tasks, on secondary datasets containing hundreds or thousands of hours of speech from thousands of participants. These secondary datasets are typically orders of magnitude larger than what is available in the medical speech dataset, which is used for the final classification problem. Thus, the built-in feature engineering approach reduces the classification complexity by discovering informative features via large, secondary datasets.

To summarize the preceding comments, feature engineering is helpful for reducing classification complexity precisely when there *is* external input, either Data or Knowledge, that a particular feature engineering function $f_{feature}$ is a useful transformation. Useful in this context means that $f_{feature}$ transforms the problem from the high dimensional and complex space of functions $f : \mathcal{Z} \to \mathcal{Y}$ to a lower dimensional problem setting $g : \mathcal{X} \to \mathcal{Y}$, that has better classification properties. The problem in the transformed space may have greater separability of class distributions $P(\mathrm{X} \mid \mathrm{T}, Y = 1)$ and $P(\mathrm{X} \mid \mathrm{T}, Y = 0)$ compared to the separability of $P(\mathrm{Z} \mid \mathrm{T}, Y = 1)$ and $P(\mathrm{Z} \mid \mathrm{T}, Y = 0)$; or the transformed problem may have lower complexity of the underlying decision boundary surface $P(\mathrm{X} \mid \mathrm{T}, Y = 1) = P(\mathrm{X} \mid \mathrm{T}, Y = 0)$ compared

to the surface $P(\mathrm{Z} \mid \mathrm{T}, Y = 1) = P(\mathrm{Z} \mid \mathrm{T}, Y = 0)$, meaning that the trade-off between approximation error and estimation error is shifted to a more favorable level. Either the Knowledge gained via domain expertise, or the Data gained via transfer learning on secondary datasets (or a different method of adding Knowledge or Data which we haven't covered) are required to ensure the problem in the transformed space $g : \mathcal{X} \to \mathcal{Y}$ will have lower classification complexity.

While feature engineering can reduce classification complexity if these assumptions hold, its effectiveness in doing so is still fundamentally limited by the distribution of the original speech audio recordings, $P(\mathrm{Z} \mid \mathrm{T}, Y)$. Unlike feature engineering, task engineering can fundamentally shift this distribution, because the conditional distributions $P(\mathrm{Z} \mid \mathrm{T}, Y = 1)$ and $P(\mathrm{Z} \mid \mathrm{T}, Y = 0)$ are conditioned precisely on the chosen speech elicitation task T. While certain measurable patient characteristics such as age, height, or genetic information are fixed, we can decide what type of speech we want to collect in assessing a potential neurological condition. An appropriate comparison would be a blood test, where the composition of blood varies continuously based on a multitude of factors (food and water intake, exercise, caffeine intake, alcohol intake, etc). Specific blood tests are only useful when the blood sample is taken in a particular context, e.g. in a fasting state; the context in which the blood biomarker is sampled impacts the usefulness of the test, and is explicitly controlled for in order to create a useful interpretation for the sample.

Similarly, speech is not a fixed attribute that can only be measured in a singular and objective manner; the context under which the speech is collected determines the make-up of the resulting sample, just as in the blood test. For example, consider the following two speech elicitation tasks for use in classifying between cognitively normal (CN) patients with normal memory function, and cognitively impaired (CI) patients who suffer from memory complaints. In Task 1, participants are shown a picture of

a single object (e.g. a car) on their mobile device, and asked to recall the object on the next screen; they are recorded while they try to remember and name the object. In Task 2, participants are shown 15 objects, one at a time, and are asked to recall as many as they can on the screen following the final object. Because Task 2 has a higher level of difficulty, the differences in how CN and CI participants perform the task are likely to be larger in Task 2 than in Task 1. Task 1 may still contain some subtle components of speech that can be used to differentiate the participants; for example, time taken to name the object, number of filler words (like "uh" or "um"), or other aspects of speech production that could be automatically extracted from a DNN-based feature extractor. However, for Task 2, in addition to the Task 1 features, one can measure the number of objects named, serial position effects such as recency and primacy (Weitzner and Calamia (2020)), number and length of pauses taken in between objects, prosody metrics capturing whether they used a listing intonation, etc. Task 2 is likely to elicit a richer and more variable (between groups) speech signal, producing a higher ceiling for information that can be extracted via feature engineering.

An appropriate feature engineering algorithm $f_{feature}$ will still be necessary in order to use the resulting audio samples from the engineered task in a classification model in any meaningful way. However, if task engineering is allowed, the task can be modified to assess those aspects of speech and language production that are known to differ for participants of differing cognitive function. Subsequent development of feature engineering algorithms $f_{feature}$ can be undertaken to exploit the information known to be in the task, according to scientific and clinical domain expertise. The task could be similarly engineered to tax elements of speech production about which the transfer learning features carry information, as determined by the original speech-related task that the transfer learning model was trained on. In either case, with a

more fundamentally differentiating speech task, there is an expanded pool of feature extraction options that are likely to reduce classification complexity, based on the task-paired Knowledge or Data with which they are engineered.

To summarize this subsection, task engineering can impact classification complexity by shifting the distribution of the speech samples in the original amplitude space $\mathcal{Z}$. It is the distribution $P(\mathrm{Z} \mid \mathrm{T}, Y)$ that impacts the lower bound on classification complexity which any subsequent feature engineering approach can achieve. Feature engineering in and of itself cannot reduce classification complexity when the decisions made around the feature engineering protocol are purely empirical on the same dataset to be used for classification. However, feature engineering driven by domain knowledge or prior work on large secondary datasets can be used to reduce the complexity of the classification problem; performing this process jointly with task engineering is the avenue of greatest potential improvement for screening test performance.

4.3 Impact of Feature and Task Engineering on Classification Complexity for a Speech-Based Screening Test

In this section, we analyze the impact of feature engineering, task engineering, and combined feature and task engineering on the complexity of the classification problem to separate between CN and CI participants. The goal with both feature and task engineering is to extract useful speech features that make the underlying classification problem easier, via increased class separability and reduced complexity of the decision boundary. The analyses in this section demonstrate the claims in the previous section's discussion: task engineering determines the minimal classification complexity that can be achieved from a set of audio recordings, but good feature engineering is required to extract the maximum amount of useful information from the audio. The two approaches (task and feature engineering) must be undertaken in

tandem to produce optimal features with reduced classification complexity.

We measure the difficulty of a particular set of speech features, extracted from a particular speech elicitation task, via two empirical analyses. First, we measure out-of-sample model performance classifying between CN and CI participants using speech collected from an observational study on Alzheimer's disease. Second, we compare empirical measures of data complexity defined in section 3.4 to the obtained out-of-sample model performance, and show that some of these classification measures provide useful insights into classification complexity.

4.3.1 Data Description

The data for the following analysis is a proprietary dataset owned by Aural Analytics, Inc. The data was collected as part of the Bio-Hermes study run by the Global Alzheimer's Platform Foundation. In my role as a Machine Learning Scientist at Aural Analytics, Inc., I was granted limited permission to access that data. The collaboration between Arizona State University and Aural Analytics has allowed for the demonstration of our ideas on a valuable dataset of speech collected from participants with probable Alzheimer's disease, Mild Cognitive Impairment, and Normal Cognition.

For the following analysis, we investigated the problem of separating between participants who are cognitively normal (CN) and cognitively impaired (CI), specifically who are diagnosed with probable Alzheimer's disease (AD). The data consisted of 211 participants with NC and 30 participants with AD, leaving $n = 241$ participants in total. Participants provided speech samples on 13 different speech elicitation tasks. The tasks are described in Table 4.1.

Task Name	**Abbreviation**	**Description**
Category Naming	CategNam	Participants name as many items as possible from a displayed category.

Visual Naming	VisualNam	Participants are shown an array of objects and asked to name them.
Visual Search	VisualSearch	Participants are shown the same array, but asked to name *only* objects that meet a specific orthographic (spelling) criteria.
Object Recall	ObjectRecall	Participants are presented a series of objects one-by-one and asked to recall them after presentation.
Immediate Word Recall	WordImmed	Participants are presented a series of target words one-by-one and asked to recall them after presentation.
Delayed Word Recall	WordDelay	After performing an intervening task, participants are asked to recall the same set of target words, without another presentation.
Word Recognition	WordRecog	Participants are shown an array of words containing the target words and distractor words, and are asked to name *only* the target words presented earlier.
Immediate Story Recall	StoryImmed	Participants are asked to recall everything they can remember from a story after presentation.
Delayed Story Recall	StoryDelay	After an intervening task, participants are asked to recall everything they can remember from the story again, without a second presentation.
Picture Description	PicDescr	Participants are shown a cartoon image of a scene, and asked to describe everything they see going on in the scene.
Sentence Reading	Sentence	Participants are asked to read aloud a series of sentences.
Diadochokinetic Rate	DDK	Participants are asked to repeat the word "buttercup" as quickly and clearly as they can in a limited amount of time.
Phonation Task	Phonation	Participants are asked to take a deep breath and hold out the "ahhh" sound for as long as possible.

Table 4.1: Descriptions of 13 speech elicitation tasks used in the analysis.

Study participants provided speech recordings on each of these 13 tasks during a speech session. Speech elicitations were recorded on a tablet device. We then extracted five sets of speech features using both open source speech feature extraction algorithms and proprietary speech algorithms owned by Aural Analytics. The speech

feature sets used for the analysis are described in Table 4.2.

Feature Set Name	Size	Description
OpenSmile (eGeMAPSv02)	88	Open source standard acoustic features.
OpenSmile (emobase)	988	Open source acoustic features, geared toward emotion recognition.
Talk2Me	140	Open source transcript-based features.
Wav2Vec2	1024	Output of the last layer of the Facebook Large 960h Wav2Vec2 model.
Clinical *(Proprietary)*	15	Hand-crafted features measuring clinically relevant and interpretable components of speech.

Table 4.2: Descriptions of 5 speech feature extraction methods used in the analysis.

Both OpenSmile feature sets and the Talk2Me feature set are obtained from open-source Python packages for speech feature extraction. The OpenSmile features (Eyben *et al.* (2010)) are acoustic-based, measuring traditional signal processing speech features. The Talk2Me features (Komeili *et al.* (2019)) are calculated using only a transcription of the audio recording, and consist of lexicosyntatic features.

The Wav2Vec2 feature set (Baevski *et al.* (2020)) consists of features extracted as the last layer of an end-to-end deep learning model. Here we use a publicly available pre-trained model without fine tuning, both for a reduction in computational time and so that the approach can be replicated by others on a publicly available model. Due to memory constraints, the Wav2Vec2 features were only calculated on the first 30 seconds of each audio recording.

The final feature set, the Clinical features, are a carefully designed set of interpretable features calculated from proprietary algorithms developed at Aural Analytics. The Clinical features measure a small number of clinically-relevant derived measures, which generally capture three aspects related to how a participant performs a speech elicitation task: accuracy, strategy, and timing. Accuracy relates to

how well the participant captures the content they are engaging with during the task. For example, in the Object Recall task, one of the Accuracy metrics is the participant's score of how many objects they correctly recalled. Strategy features measure **how** the participant actually performs the task; examples include features related to lexical density or grammatical constructs. Finally, timing metrics measure **how quickly** the participant performs a task, and other features related to rate and prosody.

The Clinical and Talk2Me feature sets were not calculated on the Phonation task, due to unavailability of transcripts of the audio recordings for that task. All other task-feature datasets were calculated. In total, 63 (i.e. $12 \times 5 + 1 \times 3$) combinations of tasks and features were evaluated.

Each combination of a speech feature set calculated on a particular speech elicitation task can be viewed as a separate dataset of audio to be used for a classification task; we call each such set of speech data as a task-feature dataset. Every task-feature dataset is a candidate for inclusion in a speech-based screening test for cognitive impairment. The screening test can be designed as a classification model fit to the speech features extracted from that particular speech elicitation task, with the outcome variable being whether or not the participant is cognitively impaired. The goal of the feature and task engineering is to reduce the complexity of the classification task implied under that task-feature dataset.

In the next two sections, we evaluate the complexity of the classification problem posed by a particular task-feature dataset in two ways. First, we look at out-of-sample classification performance when fitting a series of classification models to each task-feature dataset. Second, we calculate the data complexity measures for each task-feature dataset, and analyze 1) how the data complexity measures compare to the out-of-sample performance; 2) how the data complexity measures differ among

task-feature datasets; and 3) what kind of structures the task-feature datasets form in the underlying complexity space.

4.3.2 Out-of-Sample Empirical Analysis

We performed a cross-validation analysis to determine the out-of-sample performance of the best out of a series of classification models. Recall that the aim of this analysis is to determine the comparative classification complexity for different combinations of task and feature engineering protocols.

We measured out-of-sample performance using the Area Under the Curve (AUC) metric on a repeated cross validation (CV) procedure. We used 3 folds for the cross validation step due to limited sample size in the cognitively impaired class; the 3-fold CV was repeated 10 times. The folds for cross validation were created using stratified sampling.

The AUC metric has known limitations. For example, two vastly different ROC curves can have the same single AUC number, and AUC in and of itself does not provide a classification decision until a final cutoff has been chosen, which is what determines the final sensitivity and specificity of the test. However, because AUC is a standard performance metric reported in medical applications of binary classification using speech, we choose to report it here.

Here we describe the procedure performed during a single repetition of the repeated CV analysis, on one task-feature dataset consisting of speech features X extracted from speech elicitation task T.

After splitting the data into 3 folds, for each of the folds we performed a pipeline of five steps, which are shown in Table 4.3: 1) data preprocessing on the training folds; 2) (optional) feature selection or feature transformation on the training folds; 3) model fitting on the training folds; 4) data preprocessing on the testing fold, using

parameters from the training folds; and 5) model prediction on the preprocessed features from the testing fold.

Step	Name	Data Used	Description
(1)	Data Preprocessing	Training	Centering, scaling, removing highly correlated variables, removing variables with near-zero variance
(2)	Feature Selection or Feature Transformation *(Optional)*	Training	Performing principle component analysis (PCA), recursive feature elimination (RFE), or univariate filter-based feature selection (Filtering)
(3)	Model Fitting	Training	Fitting each of 7 models
(4)	Data Preprocessing	Testing	Preprocessing test data using parameters from Steps (1) and (2)
(5)	Model Prediction	Testing	Prediction on preprocessed test data using models from Step (3)

Table 4.3: Steps performed for each of 3 folds (within each of 10 repetitions) for the classification analysis. The predictions from each testing fold were used to calculate out-of-sample AUC on the entire dataset, for each of the 10 repetitions.

During Step (1) of the pipeline, standard data preprocessing steps were performed, including scaling and centering the data to have mean 0 and standard deviation 1, removing highly correlated variables, and removing variables with near-zero variance (NZV variables). The preprocessing transformations $f^{\text{train}}_{\text{preprocess}}$ were stored for later use on the testing data, in Step (4).

During Step (2) of the pipeline, we performed an optional additional feature engineering or feature selection step on the original speech features before model fitting. Although the speech feature sets are themselves obtained using predetermined feature engineering algorithms (i.e. mathematical functions) applied to the original amplitude values from the audio recordings, some of the speech feature sets are very large ($>$ 1k features) and are likely to contain uninformative "junk" features. Thus, according to standard practice in machine learning (Berisha *et al.* (2021)), additional feature engineering or feature selection on these speech feature sets could potentially be helpful in improving out-of-sample performance.

We compared three optional methods for feature transformation or selection: prin-

ciple component analysis (PCA), recursive feature elimination (RFE), and univariate filter-based feature selection. For the PCA-based feature transformation, a principle component analysis was performed, and principle components corresponding to eigenvalues above the default threshold of 0.95 were retained for subsequent model fitting. For the RFE feature selection, a Random Forest (RF) model was fit to sets of l features, $l = \{p, p-1, \ldots, \ell\}$, where at each step, the least important feature according to the RF variable importance metric was removed until the number of features was reduced to a predetermined size ℓ; these ℓ features were then used for subsequent model fitting. Within each training fold, a repeated CV process was performed to determine the best feature set size out of $\ell \in \{10, 25, 50, 100, p\}$ (where p represents using all of the original speech features for that feature set). For the univariate filter-based feature selection, the individually best ℓ features were used for subsequent model fitting, where "best" was determined using performance on an RF model fit using a single feature; the final feature set size ℓ was once again determined from among the options $\ell \in \{10, 25, 50, 100, p\}$ using repeated CV within the training fold.

While there are countless options for feature transformation and/or feature selection, the methods compared represent a diverse and commonly used set of algorithms. As there were many other hyper-parameters of this analysis that were systematically changed for comparison, we believe this set of feature selection/transformation approaches is sufficient to provide a sense of the decrease (or increase) in the classification complexity of each task-feature dataset, that can be achieved via additional feature selection or transformation of the engineered speech features.

During Step (3) of the pipeline, we fit each of 7 classification models using the preprocessed (and optionally transformed or down-selected) features from the training data. Taking inspiration from Smith *et al.* (2014), we chose 7 models from a diverse

set of model families for this model-fitting stage. We used the same models from Smith *et al.* (2014) where possible, as these models were shown in Smith *et al.* (2014) to produce the largest variety of predictions according to the dendogram in Figure 1 of Smith *et al.* (2014). The models used from Smith *et al.* (2014) include Naive Bayes, Random Forest, 5-nearest neighbors (5-NN), and Decision Tree C5.0. We substituted CART for RIPPER, Logistic Regression for MLP, and did not assess the RIDOR or LWL clusters of the dendogram due to unavailable substitutes in the `caret` package, which was used for performing the pipeline. We also included the Bayesian Additive Regression Tress (BART) model (Chipman *et al.* (2010)), which has been shown to achieve good performance with little parameter tuning in simulation studies and empirical data examples (Hill *et al.* (2020)). In total, the 7 models calculated were: Naive Bayes, Random Forest, 5-NN, C5.0, CART, Logistic Regression, and BART.

During Step (4), the preprocessing operations from the training folds $f^{\text{train}}_{\text{preprocess}}$ were applied to the speech features in the testing fold. Calculating the preprocessing parameters on the training data alone for each fold, rather than on the entire dataset, prevented data leakage whereby information from the testing data used during model creation might artificially inflate model performance.

Finally, during Step (5), each of the fitted models was used to predict the outcomes for the testing fold. We stored the predicted probabilities, rather than a class prediction, to allow for calculation of the AUC metric.

Figure 4.4 provides a visualization of this process being performed on a single fold within a single repetition of the analysis.

After performing this process on each of the three folds, the out-of-sample predictions from all folds were used to calculate a single out-of-sample AUC for the full dataset, for each of the 7 classification models. Figure 4.5 provides a visualization of going from the predicted probabilities for all folds to an AUC score for each model.

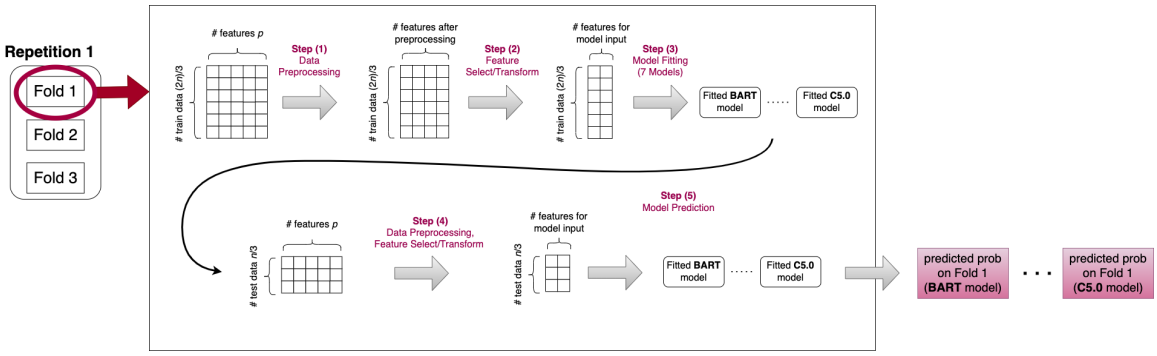

Figure 4.4: Visualization of Steps (1) through (5) being performed on a single task-feature dataset. The top row represents operations performed on the training data; the bottom row represents operations on the testing data.

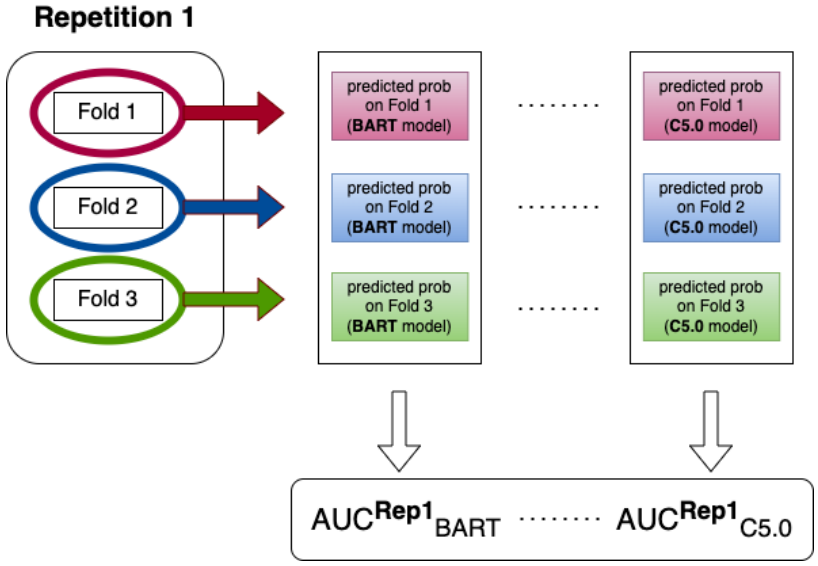

Figure 4.5: Visualization of combining the predictions from all three folds to calculate an AUC score for each model, for a single repetition of the CV analysis.

This entire process was repeated 10 times, producing in total 10 out-of-sample AUC values for each of the 7 classification models. Ten repetitions were used so we could assess the variability of model performance due to sampling variability during fold creation; with an imbalanced data set having a small minority class, the model performance can significantly differ depending on the fold composition and the model used. Figure 4.6 provides a visualization of the process going from 10 repetitions to a distribution of AUC values for each model.

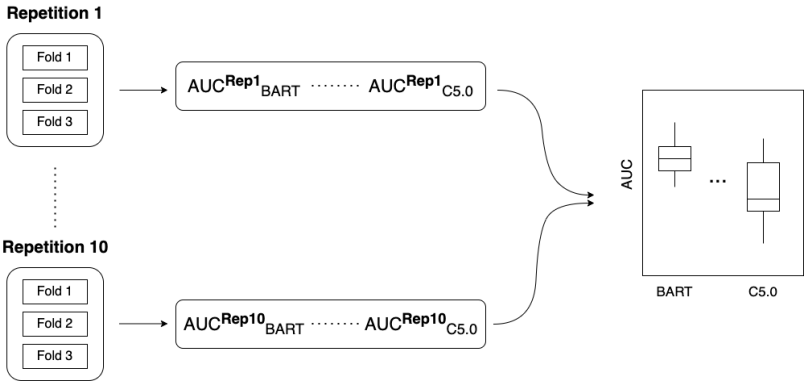

Figure 4.6: Visualization of summarizing the AUC scores from 10 repetitions on all 7 models into a boxplot of the AUC distribution for each model.

Finally, we repeated this analysis 63 times for each of the 63 task-feature datasets combinations. As an example of the AUC results for a single task-feature dataset, Figure 4.7 shows the the distributions of the AUC values from the 7 classification models fit to the Clinical feature set on the Visual Naming task, using the original speech feature set for model fitting (i.e. Step (2) was omitted and no further feature selection or transformation was performed on the Clinical speech features). Each boxplot shows the distribution of out-of-sample AUC values for that classification model over the 10 repetitions.

For this particular task-feature dataset, we see a few patterns in the classification results. First, the tree ensemble methods (BART and RF), Naive Bayes, and to a slightly lesser extent logistic regression show the best out-of-sample performance; 5-NN and the single tree methods show the worse performance. Furthermore, the tree ensemble methods and Naive Bayes have lower AUC variability over the 10 repetitions, whereas logistic regression, 5-NN, and the single decision tree methods (CART and C5.0) have higher AUC variability. The patterns in AUC variability (ensemble methods having lower variability, and KNN, logistic regression, and decision trees having higher variability) corroborate prior work on the variability of out of sample performance for different classification model families (Berisha *et al.* (2021), Drucker

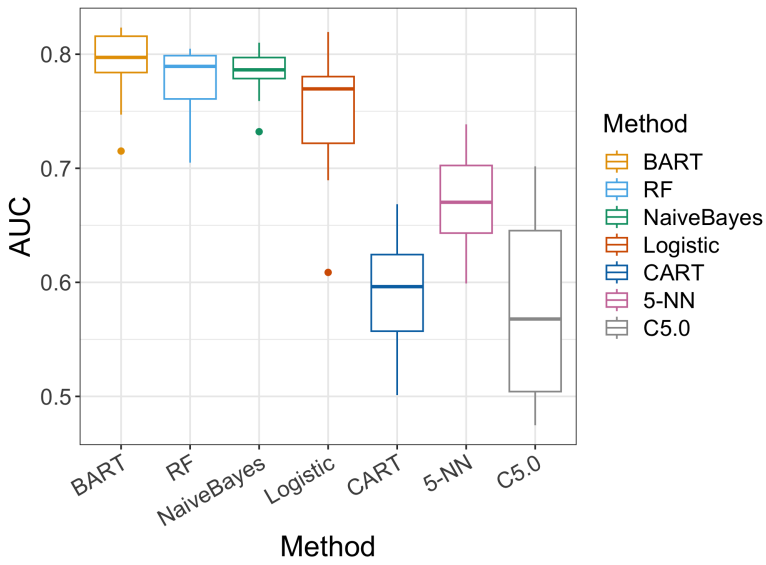

Figure 4.7: Distribution of AUC scores over 10 repetitions for the 7 classification models, for the data set of Clinical features extracted from the Visual Naming task. Ensemble methods and Naive Bayes have the lowest AUC variance; single decision trees, logistic regression, and KNN have the highest AUC variance.

et al. (1994)).

Figures 4.8 and 4.9 show these same distributions for each of the 63 task-feature datasets. While this exact pattern of model performance does not hold for every single task-feature dataset, the general trend can be seen in most. The comparative performance of the models was not the main point of this analysis, but of note, the BART model has the highest AUC distribution for 41 out of 63 task-feature datasets, with RF being highest for 17 datasets, Logistic Regression for 4 and Naive Bayes for 1 dataset. The plots for the Clinical and Talk2Me feature sets on the Phonation task are blank, as these feature sets were not calculated on the Phonation task due to lack of transcripts available for this task.

In order to digest this information in a useful and comprehensible way, we summarized the distributions of the AUC values over the 7 models into a single AUC value representing the overall complexity of the data set defined by the particular task and feature set. This summary allows for comparing the classification complexity of

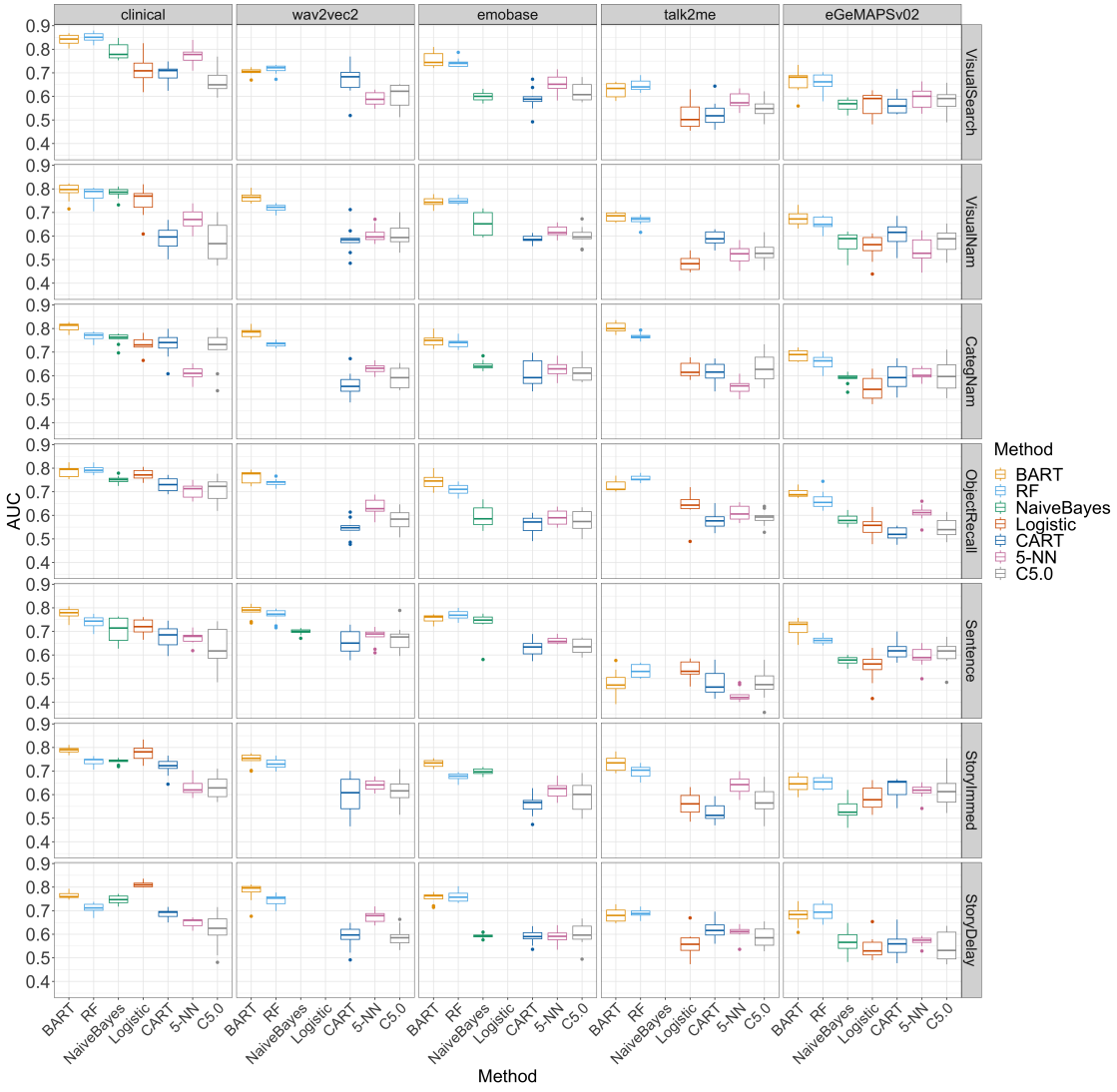

Figure 4.8: Distribution of AUC scores over 10 repetitions for the 7 classification models, for the all task-feature datasets from the first 7 tasks: Visual Search, Visual Naming, Category Naming, Object Recall, Sentence, Story Recall Immediate, Story Recall Delayed.

the task-feature datasets over different levers: the task engineering method (Visual Search, Sentence Reading, etc.) or the speech feature engineering method (Talk2Me, Wav2Vec2, etc.). Essentially, this summarization procedure allows for an easier visual comparison of the underlying classification complexity of each task or feature set, quantified by a summary AUC score.. This visualization allows for comparing

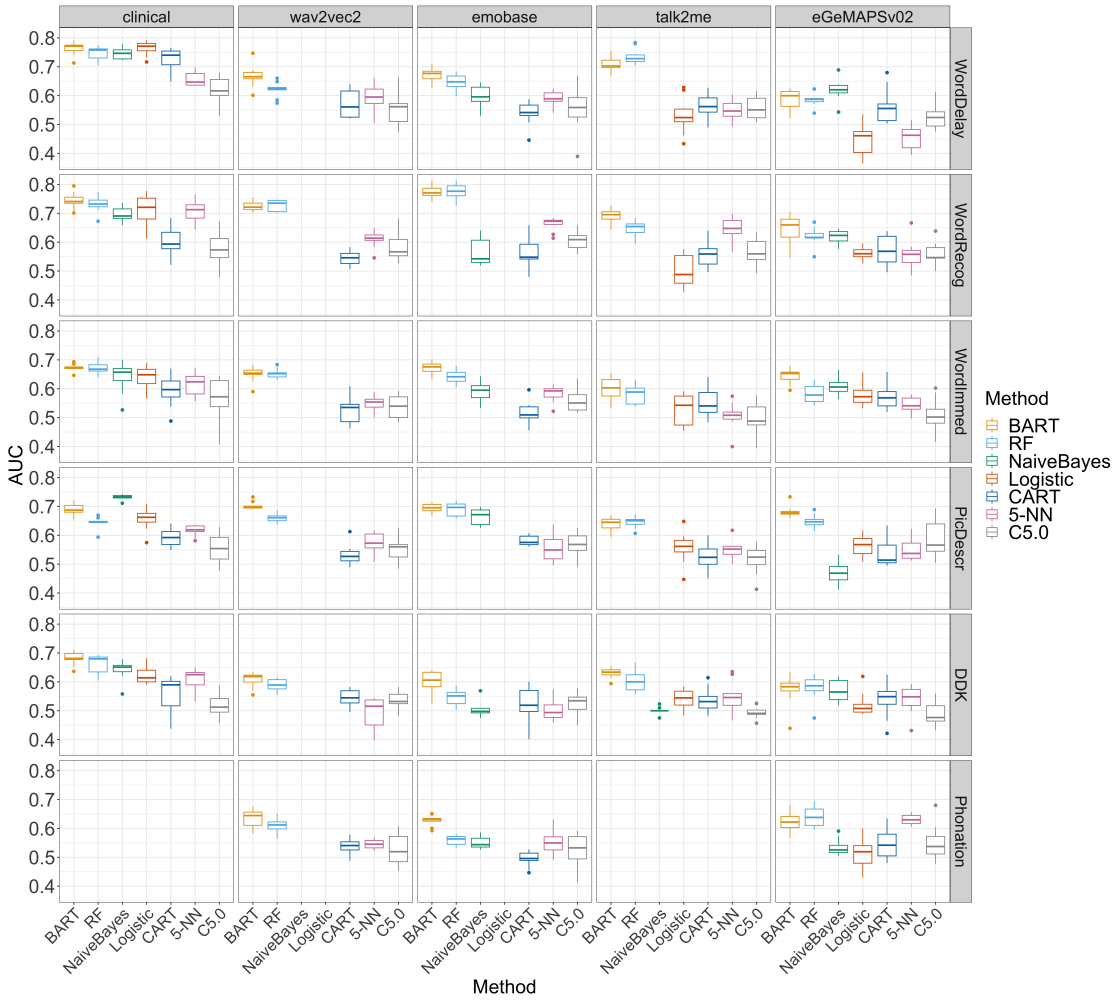

Figure 4.9: Distribution of AUC scores over 10 repetitions for the 7 classification models, for the all task-feature datasets from the last 6 tasks: Delayed Word Recall, Word Recognition, Immediate Word Recall, Picture Description, DDK, Phonation.

whether the main driver of complexity is task engineering, feature engineering, or a combination of the two.

To create this summarizing AUC score for each of the subplots in Figure 4.8, we first took the median out-of-sample AUC from the 10 repetitions, for each of the 7 classification model types. This number represents a typical AUC for that classification model; using the median prevents this typical AUC from being skewed by high or low outliers. Next, we took the maximum of these median AUCs among all of

the 7 classification methods, which represents the best "typical" model performance that could be achieved for this task-feature dataset. This provided a final AUC score for each task-feature dataset, which was used for further analysis. Figure 4.10 provides a visual representation of this process for the results from the Clinical features calculated on the Visual Naming task.

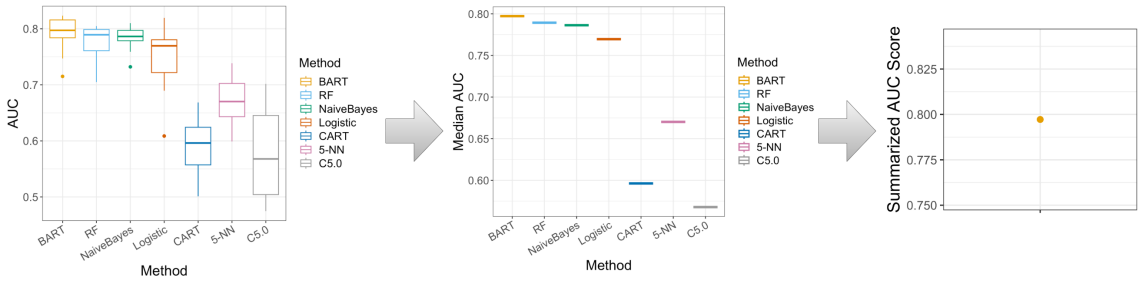

Figure 4.10: Flowchart showing the summarization of 10 AUC scores over 7 classification models into a single score, for a given task-feature dataset. First, the median of the 10 AUCs is extracted for each method. Then, the maximum of the median AUCs is calculated as the summarized AUC score.

Using this summarization method, we can compare a single value representing out-of-sample classification performance for each task-feature dataset. Figure 4.11 shows this summary AUC value for each task-feature combination, using the original speech features (no feature selection or transformation in Step (2)). Each subplot summarizes a single column (task) from Figure 4.8; within a single subplot, each point represents the highest median AUC value for a particular row (feature set) from that column (task), out of all of the classification models compared. For each task (subplot), the highest AUC score over all the feature sets is shown by a dashed gray line.

Figure 4.11 provides a succinct visual overview of the classification complexity (as measured by out-of-sample AUC), posed by using a particular set of speech features X extracted from a particular speech elicitation task T to classify between cognitively normal participants ($Y = 0$) and cognitively impaired participants with Alzheimer's

disease ($Y = 1$). While the classification complexity posed by a particular task-feature set is impacted by both the task and the feature set, we claim that the speech elicitation task determines the fundamental upper bound on model performance that can be achieved using speech features engineered from audio on that task.

As discussed above, engineering of the task T impacts the relationship between the two conditional distributions of the resulting audio recordings, $P(\mathrm{Z} \mid \mathrm{T}, Y = 0)$ and $P(\mathrm{Z} \mid \mathrm{T}, Y = 1)$. This in turn allows greater opportunities for Knowledge via domain expertise or Data via related transfer learning protocols to produce feature engineering functions $f_{feature} : \mathcal{Z} \to \mathcal{X}$ that capture group differences in a particular task.

If a task T provides a context leading to similar distributions of audio recordings $P(\mathrm{Z} \mid \mathrm{T}, Y = 0)$ and $P(\mathrm{Z} \mid \mathrm{T}, Y = 1)$, then finding a feature engineering algorithm $f_{feature} : \mathcal{Z} \to \mathcal{X}$ such that $P(\mathrm{X} \mid \mathrm{T}, Y = 0)$ and $P(\mathrm{X} \mid \mathrm{T}, Y = 1)$ have high separability is extremely difficult, if not impossible. This concept can be seen empirically in Figure 4.11.

For example, in the Phonation task, participants take a deep breath and hold the sound "ahhh" for as long as they can. This task would likely be performed differently by healthy participants and those with a diagnosed respiratory disease (e.g. chronic obstructive pulmonary disease), but respiration is not the primary speech function impacted by Alzheimer's disease. Thus, we do not expect any particular speech feature algorithm to produce high class separability, when the speech features are calculated on audio recordings that are fundamentally similar between participants from the two outcome groups (CN and CI). Indeed, we see in Figure 4.11 that both the hand crafted features using traditional signal processing (OpenSmile Emobase and OpenSmile eGeMAPSv02) and the end-to-end features from a Deep Neural Network (Wav2Vec2) have similarly poor AUC on this classification problem for the Phonation

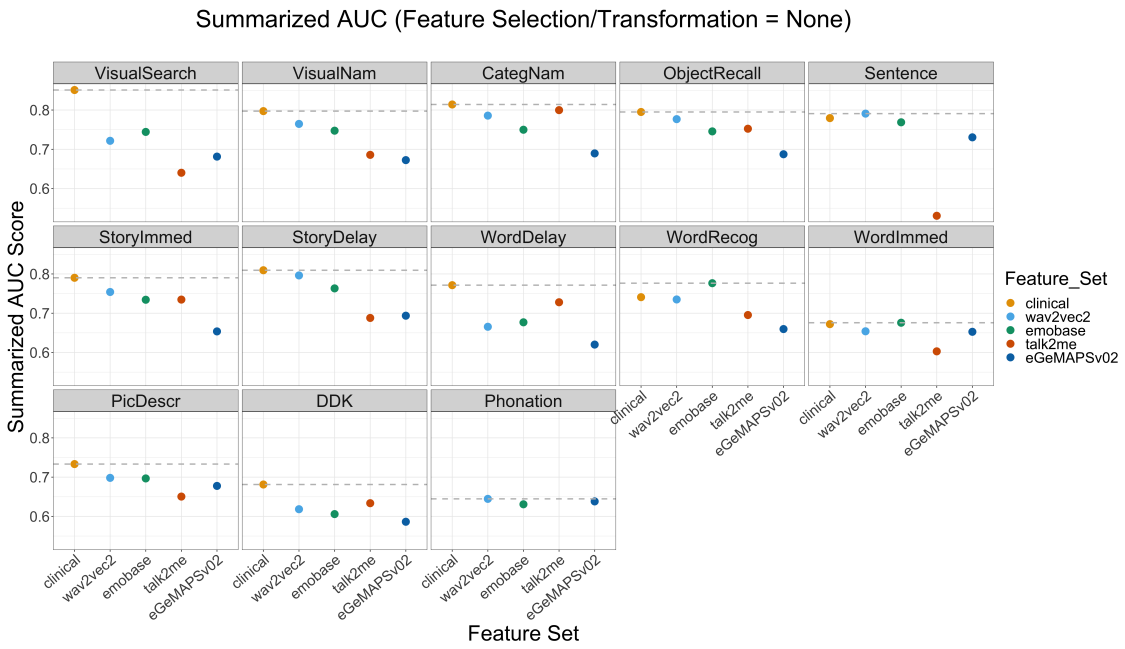

Figure 4.11: Out-of-sample performance summarized for all tasks and features, with no additional feature selection or transformation on the original speech feature sets. The task determines the maximum potential out-of-sample performance, but good feature engineering is required to be able to extract the maximum classification performance from that task.

task.

A similar story can be seen for the DDK task, in which participants are asked to repeat the word "buttercup" as quickly and clearly as they can in a short period of time. On this task, we see in Figure 4.11 that the Clinical features produce a slightly higher max AUC than the other feature sets, probably because the Clinical features are the only feature set that includes the count of how many times the participant repeated the word "buttercup". As Alzheimer's disease has been shown to impact motor function in speech (Meilán *et al.* (2020)), it is reasonable that this feature set would be able to differentiate slightly better between cognitive groups, compared to feature sets measuring vocal quality alone, or grammatical structures on transcripts of one repeated word. More formally, the DDK task has a greater degree of fundamental class separability in the conditional distributions of the audio

recordings $P(\mathrm{Z} \mid \mathrm{T}, Y = 0)$ and $P(\mathrm{Z} \mid \mathrm{T}, Y = 1)$ compared to the Phonation task. A feature engineering method $f_{feature} : \mathcal{Z} \rightarrow \mathcal{X}$, that utilizes domain expertise regarding subtle impacts to motor function in Alzheimer's disease, has reduced classification complexity compared to other, less informative feature sets. However, the ultimate classification complexity for this task is still relatively high (i.e. the upper bound of AUC is low), because it is not taxing cognitive functions that are primarily impacted by the disease being screened.

Tasks that directly tax a neurological function known to be impacted in Alzheimer's disease have a higher upper bound on the summarized AUC score. The conditional distributions of the original audio for this task, $P(\mathrm{Z} \mid \mathrm{T}, Y = 0)$ and $P(\mathrm{Z} \mid \mathrm{T}, Y = 1)$, are different enough between groups that useful feature engineering algorithms $f_{feature}$ can be found. For example, the Visual Search, Visual Naming, and Category Naming tasks all require word retrieval, which is known to decline with greater levels of cognitive impairment (Huff *et al.* (1986)). The existence of a higher max AUC score on these tasks in Figure 4.11 means that participants from different cognitive groups perform the task in a sufficiently differently way; there exists at least one set of speech features that is measuring components of speech and language that differ between the groups in a generalizable manner. More formally, the feature engineering algorithm $f_{feature} : \mathcal{Z} \rightarrow \mathcal{X}$, where $f_{feature}$ is the set of algorithms for the Clinical features, transforms the problem into a problem space with greater class separability and lower decision boundary complexity.

Creating a differentiating speech elicitation task is not enough by itself, however. Figure 4.11 demonstrates that good feature engineering is still necessary in order to extract the maximum classification performance from the audio. For example, in the Sentence Reading task, there is a difference of almost 25 percentage points between the AUC achieved by the Clinical, Wav2Vec2, and Emobase features compared to the

Talk2me features. The Talk2me features (which measure grammatical structures and language complexity based on the transcript) are uninformative on a task in which all of the participants read the same sentence; the distributions $P(\mathrm{X} \mid \mathrm{T}, Y = 1)$ and $P(\mathrm{X} \mid \mathrm{T}, Y = 0)$ will have huge overlap. The Knowledge that deems these features useful in other tasks like Story Recall, where participants are speaking spontaneously, does not apply to this particular task.

On the tasks with lower AUC, such as DDK and Phonation, the absence of a high performing feature set among the handful compared does not conclusively prove the lack of existence of such a feature set. However, the diversity of the speech feature sets compared in terms of both input (audio, transcript, or both) and method of extraction (hand-crafted vs. end-to-end), along with a clinical understanding of the aspects of speech production known to change in the indication studied here (Alzheimer's disease), points to the more likely explanation being that the task itself produces more fundamentally similar speech for the two groups being classified.

The high-level summary of Figure 4.11 is that task engineering, specific to the classification problem for which the speech recordings will be used, is required in order to produce audio recordings in which participants from different groups perform the task in predictably different ways. Additionally, speech feature engineering algorithms that capture the *aspects* of speech in which the participants differ on that particular task, are required in order to achieve maximum classification performance using audio recordings from that task.

Figure 4.12 provides a stylized demonstration of how the speech elicitation task can change the underlying classification complexity of the resulting speech, and thus produce vastly different summary AUC scores. In this example, Speech Task #1 elicits speech from a region in which the conditional distributions over speech features $P(\mathrm{X} \mid \mathrm{T}, Y = 1)$ and $P(\mathrm{X} \mid \mathrm{T}, Y = 0)$ have a complicated decision boundary and high

class overlap. On the other hand, Speech Task #2 pushes the elicited speech to a different region, in which the conditional speech feature distributions have a simpler decision boundary and lower class overlap. This may be the case, for example, if Speech Task #1 is a task unrelated to cognition that both classes perform similarly, and Speech Task #2 is a cognitively challenging task that pushes both groups to higher values of both X_1 and X_2, but for which the cognitively normal group can perform at an even higher level compared to the cognitively impaired group. While this stylized example in two dimensions cannot convey the true complexity of the classification problem in the much higher dimensional space $\mathcal{X}$ of real speech data, it demonstrates how careful task design paired with intelligent feature engineering can produce speech features with better classification properties.

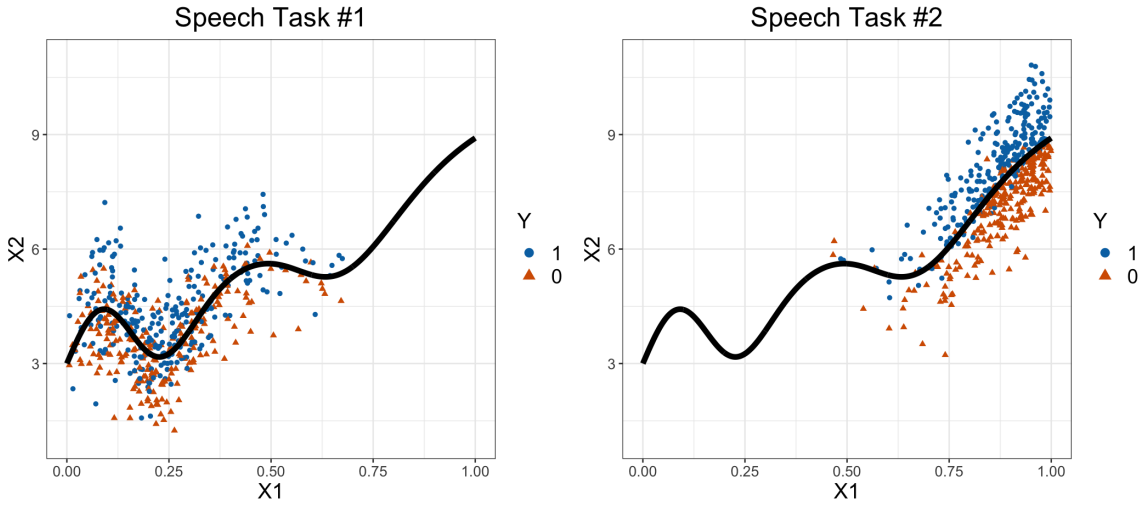

Figure 4.12: A stylized example of two different speech elicitation tasks. Speech Task #1 elicits speech in a region of the support for which both classes perform similarly, leading to high class overlap and a complicated decision boundary. Speech Task #2, for example a cognitively taxing maximum performance task, elicits speech in a region of the support for which the cognitively unimpaired class achieves higher performance, leading to better classification properties of lower class overlap and decision boundary complexity.

In the next part of this section, we explore the impact of additional feature selection or transformation on top of the original speech features. Figure 4.13 shows a plot

having a similar structure as Figure 4.11, except that we provide multiple summarized AUC results per task-feature dataset instead of just one point. The cluster of points in the same color, for a given feature set, indicate the maximum AUC produced by using either the original speech features (square marker), or a set of features obtained via further transformation (PCA, diamond marker) or selection (RFE, circle marker; Filtering, triangle marker) of the original speech features. This additional processing was described in Step (2) of Table 4.3.

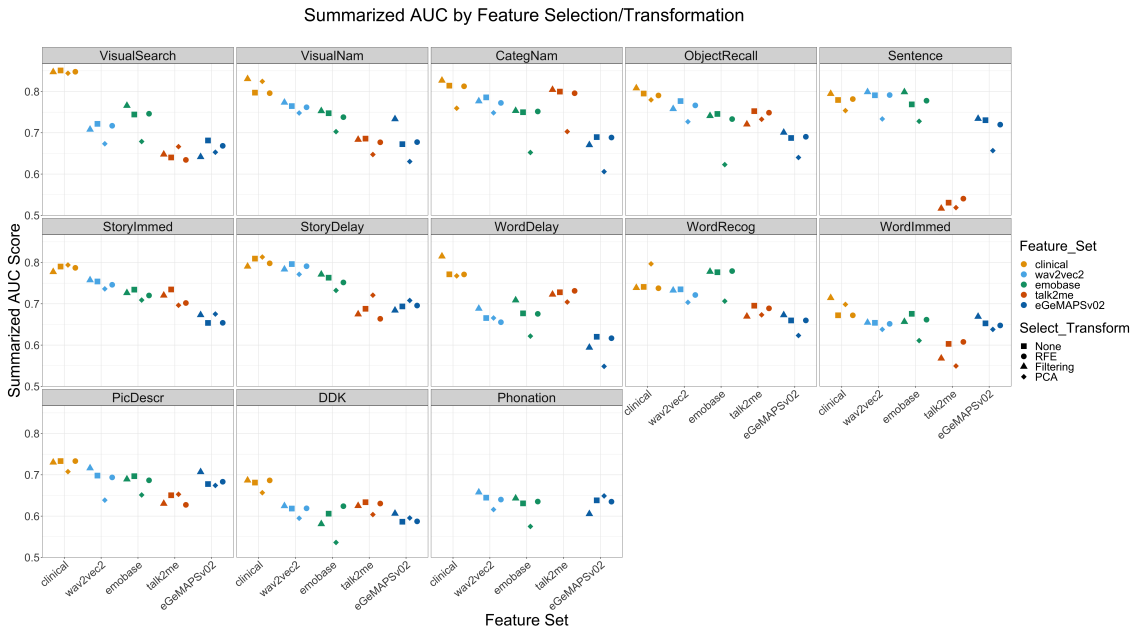

Figure 4.13: Out-of-sample performance summarized for all tasks and features, comparing different options for additional feature selection or transformation on top of the engineered speech features. The relative performance of different task and feature set combinations from Figure 4.11 is largely unchanged; PCA is notably worse than the other feature selection methods.

The patterns in relative performance of the feature sets and tasks are largely unchanged. For the most part, using either the original features, RFE, or Filtering produces similar results; the cluster of points appears to be anchored at a level determined by the original speech feature set, rather than the particular feature selection method used. The only feature transformation method with substantially poorer

performance on several of the task-feature datasets is Principle Component Analysis (PCA).

In the language of Section 4.2, we see the following situation: the original feature engineering function $f_{feature}$ transforms the problem into a new feature space; the classification complexity in that new feature space $g : \mathcal{X} \to Y$ is determined by the relevance of the information (Knowledge or Data) provided by the feature set X on the task T. This sets a baseline for the class separability and decision boundary complexity determined by the conditional distributions of the resulting speech features, $P(\mathrm{X} \mid \mathrm{T}, Y = 0)$ and $P(\mathrm{X} \mid \mathrm{T}, Y = 1)$. Feature sets that are known to extract an aspect of cognitive function measured on the task, or to contain information from a related speech task via transfer learning, impart additional information such that the classification problem in the transformed space of speech features has lower complexity.

With the baseline level of classification complexity determined by the original feature set transformation $f_{feature} : \mathcal{Z} \to \mathcal{X}$, further feature selection or transformations $f_{select} : \mathcal{X} \to \mathcal{X}'$, driven by a purely empirical process with no additional domain expertise or secondary dataset to add additional information, does not significantly reduce the classification complexity in a consistent way. The resulting problem has similar performance as the original features in most cases.

Mathematically, the solution to finding $g : \mathcal{X} \to \mathcal{Y}$ such that

$$g = \mathrm{argmin}_{f \in \mathcal{F}} R_{\mathrm{emp}}(f) + \lambda ||f||^2,$$

will have a similar performance as the solution to finding a composite function $g' \circ f_{select}$, with $f_{select} : \mathcal{X} \to \mathcal{X}'$ and $g' : \mathcal{X}' \to Y$, such that

$$g' \circ f_{select} = \mathrm{argmin}_{f \in \mathcal{F}} R_{\mathrm{emp}}(f) + \lambda ||f||^2,$$

because both are optimized using the same small dataset.

The impact of empirical feature selection and transformation on the Talk2me features calculated for the Sentence task (essentially, no impact) is a good demonstration of this phenomenon in our dataset.

PCA has substantially lower performance in some cases (21 out of 63 combinations), which is unsurprising given the known perils of using PCA as a feature preprocessing step (Jolliffe (1982)). While PCA, RFE and Filtering all provide strictly less information to the model than the full, original feature set (i.e. they are all lossy transformations), the features selected via RFE and Filtering are still related to the outcome variable, either individually or in concert with other selected features. Thus, RFE and Filtering usually maintain the performance of the original features, and in some cases improve it.

In contrast, the PCA features are derived based on the directions of greatest variance in the input features alone; their usefulness depends on whether or not the key PCA assumption is satisfied, namely that the directions of greatest variation in the input features capture variability that is important relative to the outcome classes. If that assumption were somehow known to be true a priori, this would be an example of a third type of additional information (besides domain expertise and transfer learning) that could lead to feature transformations which significantly reduce the classification complexity. In some of the task-feature datasets, however,it appears that the directions of greatest variability (at least above the 0.95 threshold for keeping the principle components) are not sufficient to capture the information that was present in the original feature set; the assumption does not hold. See, for example, the OpenSmile Emobase and OpenSmile eGeMAPSv02 feature sets on the Category Naming task (middle subplot in the top row of Figure 4.13).

In summary, while some additional feature selection and transformation approaches slightly improve classification performance for some of the task-feature datasets, there

is no systematic pattern whereby a particular feature selection or transformation approach consistently and substantially improves performance; the PCA transformation in particular seems to substantially reduce it in a third of the cases. Although this can be attributed to us not trying the "right" feature selection or transformation method, we believe this phenomenon is caused by the lack of outside knowledge or data usage provided by the f_{select} function in our case.

We take a final moment to remark on the novel contribution that this analysis produces in and of itself. We have performed a large scale classification analysis, comparing a large set of highly diverse speech elicitation tasks and five different speech feature engineering algorithms. The speech elicitation tasks include connected speech, naming, and phonating tasks, unlike past work comparing different speech tasks on the same set of participants, which were mainly limited to different connected speech tasks. Furthermore, we have calculated classification performance using a highly diverse set of classification models, in order to reduce the impact of results being determined solely by a lack of fit between the patterns in the particular feature-task dataset and the hypothesis class implied by each of the learning algorithms.

Beyond the basic comparison of which tasks and feature sets provide better performance, we have used the results to draw conclusions about the higher level impact of task or feature engineering on classification complexity. Finally, we have placed these results within the rigorous contexts of drivers of classification complexity, presented in the last chapter on statistical learning theory and information-theoretic divergence measures.

4.3.3 Data Complexity Measures Analysis

As a final step to assess patterns in classification complexity for each task-feature dataset, we calculate the set of data complexity measures outlined in table 3.1 on

each task-feature dataset.

The results demonstrated some limitations in the data complexity measures as originally postured, which we discussed at the end of the previous chapter. In Ho and Basu (2002), Figure 5b (average number of points per dimension axis) shows that a large majority of the real datasets considered had at least 10 points per dimension. However, some of the data complexity measures can provide a misleading measure of simplicity when the number of features relative to datapoints is large. As the classification measures are calculated only on the original dataset and do not incorporate any technique for data splitting or subsampling, patterns that may seem to indicate lower complexity (for some of the measures) are patterns that would not generalize if applied to new data.

For example, the F4 measure calculates the number of datapoints that cannot be separated using all of the features in a greedy manner. With many features compared to number of datapoints, observations may be able to be separated using a spurious relationship from one of the many features, but this does not necessarily indicate a simple classification problem in which patterns observed in the dataset will generalize to new data.

In light of these limitations, we demonstrate results for just the subset of the complexity measures that had the highest and lowest correlation with the out-of-sample AUC scores shown in Figure 4.11. We discuss aspects of how these complexity measures are calculated that are likely to lead to them being informative on future model performance. This analysis contributes to the growing body of literature that seeks to understand how the measures of data complexity relate to out-of-sample classification performance.

First, we look at the correlations between the data complexity measures and the out-of-sample AUC scores for each task-feature dataset. The classification measures

listed in table 3.1 were calculated for each task-feature dataset, and compared to the summarized AUC score from Section 4.3.2 for that task-feature dataset.

Figure 4.14 shows a heatmap of the Pearson correlation between each of the complexity measures and the out-of-sample AUC performance, calculated for each complexity measure over all 63 task-feature datasets. Comparing correlation across complexity measures is a fair comparison, because the set of data complexity measures calculated in `ECoL` are standardized to fall between 0 and 1, with lower scores indicating lower complexity.

Table 4.4 shows the correlations along with p-values indicating whether the correlation is statistically significant based on the sample size. While p-values have inherent limitations and should not be taken as a binary indication of result validity based on a particular threshold (Karpen (2017)), the value is nonetheless useful on a continuous scale in gauging how seriously to take the positive and negative correlations shown in Figure 4.14.

Since a high AUC score is associated with lower complexity and improved model performance, negative correlations indicate good agreement between which tasks and feature sets the *complexity measure* quantifies as being complex, and which tasks and feature sets the *AUC score* quantifies as being complex.

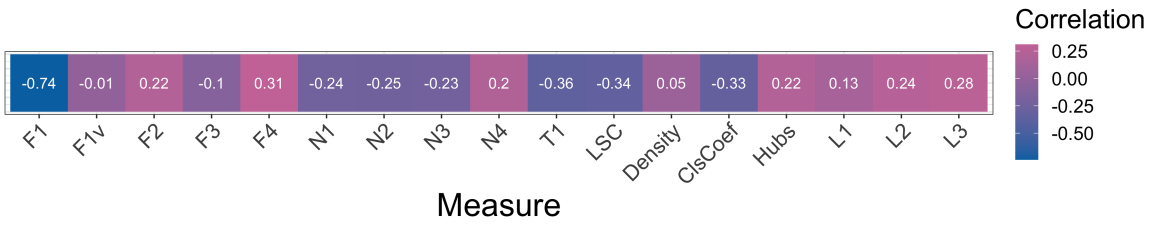

Figure 4.14: Heatmap showing correlation of each of the complexity measures with out-of-sample AUC scores.

The F1 measure has the largest negative correlation with the AUC scores, and

Measure	Correlation with AUC	$p-$Value
F1	-0.74	$< .00001$
F1v	-0.01	.938
F2	0.22	.083
F3	-0.1	.436
F4	0.31	.015
L1	0.13	.443
L2	0.24	.152
L3	0.28	.093
N1	-0.24	.058
N2	-0.25	.048
N3	-0.23	.070
N4	0.2	.116
T1	-0.36	.004
LSC	-0.34	.006
Density	0.05	.697
ClsCoef	-0.33	.008
Hubs	0.22	.083

Table 4.4: Measures of data complexity, their correlation with out-of-sample AUC scores, and p-values indicating statistical significance of the correlations.

several other measures have a moderate negative correlation (N1, N2, N3, T1, LSC, ClsCoef). To do a deeper dive into these correlations, we evaluate how the data complexity measures correlate with the AUC scores when calculating correlation by task or feature set group, rather than over all task-feature datasets. If the pattern of negative correlation holds when calculated only for a particular task or feature set, then this classification measure provides information on complexity that is consistent with what we see in out-of-sample experiments, at least for this analysis.

Figure 4.15 shows the correlations of each complexity measure with the AUC score

when we group the data by feature set. Note that the L1, L2, and L3 measures were not calculated on the wav2vec2 and emobase feature sets, due to having more features than datapoints causing an indeterminate solution. The F1, N1, N2, N3, and LSC measures retain the negative correlation, but T1 and ClsCoef have one feature set each in which the pattern is reversed.

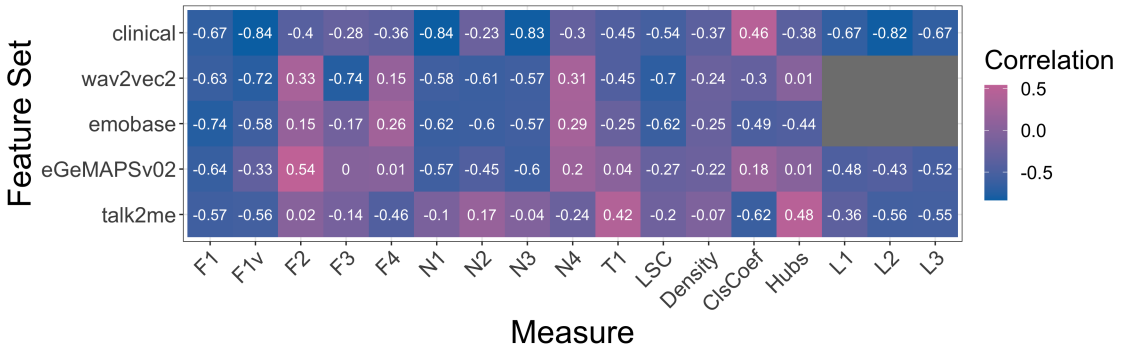

Figure 4.15: Heatmap showing correlation of each of the complexity measures with out-of-sample AUC scores; correlations are calculated for each complexity measure on the datasets corresponding to a single feature set.

The linear complexity measures have negative correlations within each feature set, despite having strong positive correlations overall. This indicates that L1, L2, and L3 are useful measures in providing a relative gauge of classification complexity for a fixed feature engineering approach, but the complexity values across different feature sets are not comparable on an absolute scale. Similarly, the F1v measure has very strong correlation with AUC within each feature set, but taken on the data as a whole the correlation is almost 0, once again indicating this is a good relative measure for a fixed feature engineering approach, but potentially not a good absolute measure of complexity for feature sets of vastly different sizes and compositions.

Finally, Figure 4.16 shows the same type of correlation heatmap, but the correlation for each measure is calculated on the subset of datasets coming from a particular task. Thus, each correlation value measures the linear relationship between the data

complexity measure calculated on all feature sets for one task, compared to the out-of-sample AUC score for all the feature sets on that task. There are many data complexity measures that have the reverse pattern compared to AUC scores as what would be expected, meaning many squares with positive correlations. For these data complexity measures, including F2, F4, N4, and Hubs, the measure indicates one relationship of classification complexity among all feature sets for that task, while the actual AUC performance on out-of-sample data indicates a different complexity relationship among the feature sets.

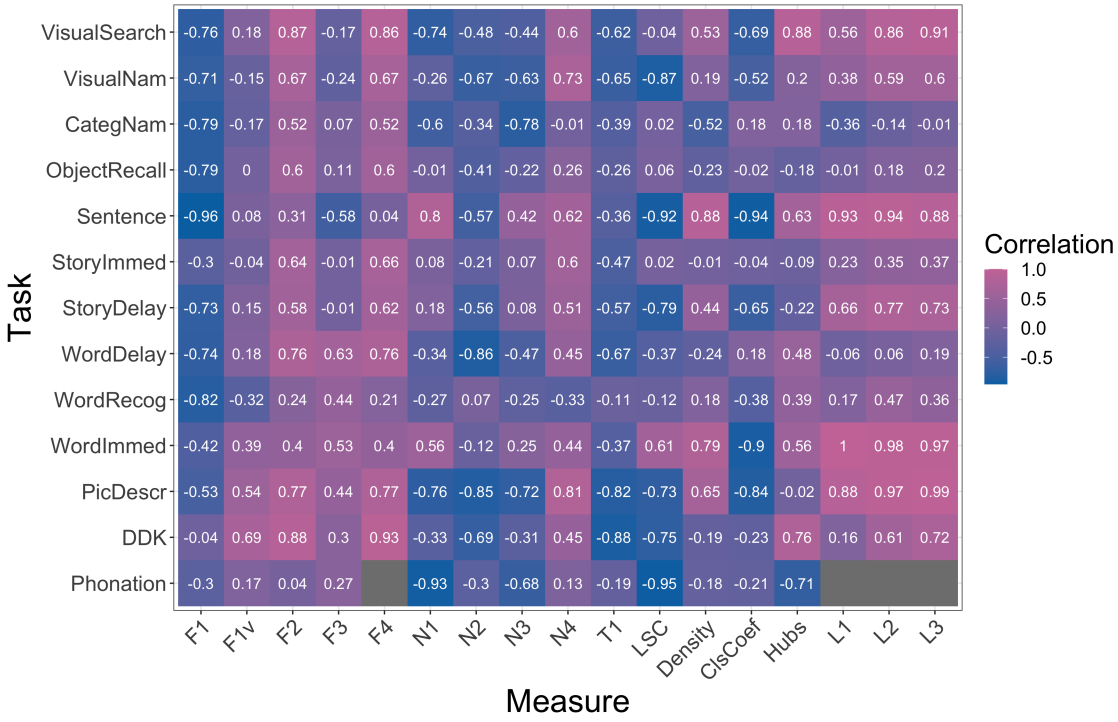

Figure 4.16: Heatmap showing correlation of each of the complexity measures with out-of-sample AUC scores; correlations are calculated for each complexity measure on the datasets corresponding to a single task.

The complexity measures F1, N2, and T1 (and to a lesser extent, ClsCoef) retain the correct direction of correlations for nearly all of the tasks, with near zero correlation for 1 or 2 of the tasks and no statistically significant positive correlations. These

findings indicate that the values of these data complexity measures have a consistent relationship with the patterns observed in out-of-sample performance.

Overall, the story presented with these correlation plots is that many of the classification measures do not correlate well with classification performance, in particular for datasets with a large amount of features or having many "junk" features. Notice that in the correlations by feature set (Figure 4.15), the Clinical feature set has the best correlations overall with the data complexity measures compared to the other feature sets. We theorize this is because the Clinical feature sets produce a more traditional classification problem setting that is seen in tabular datasets found in applications to behavioral and social sciences: a large data-to-feature ratio ($241/15 = 16$), and individually meaningful features that have a reasonable chance of being related to the outcome variable (cognitive status). This is also the classification setting in which the data classification measures have been extensively explored; many of the works reviewed in Lorena *et al.* (2019) vet the classification measures using tens, or in some cases, hundreds, of datasets that fall into this traditional classification problem setting.

Despite the challenge of utilizing the data complexity measures for the speech feature engineering setting, we proceed with a deep dive into a small number of measures that demonstrated good correlation with out-of-sample performance, in order to give a secondary look into the impact of task and feature engineering on classification complexity.

The best performing measure, quantified by correlation to out-of-sample AUC, is the F1 measure, which determines the maximum discriminative power from each of the features individually. Figure 4.17 shows the value of F1 for each of the task and feature set combinations. In this plot, lower values indicate lower classification complexity via lower feature overlap on the best individual feature. While overlap

of single features is not necessarily reflective of the overlap of the multidimensional conditional class distribution as a whole, having at least one dimension with less overlap implies a potentially easier classification problem.

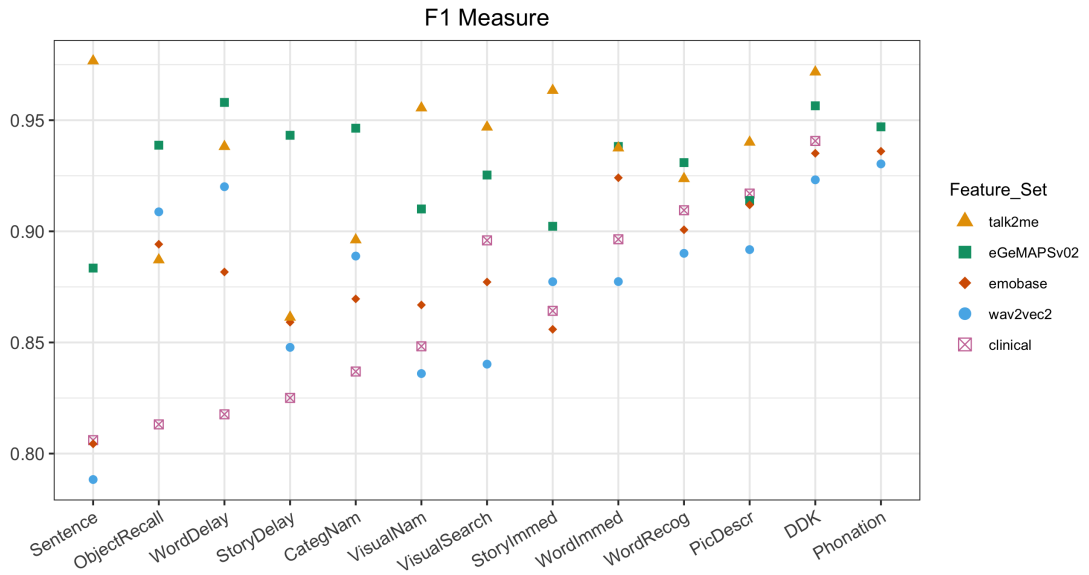

Figure 4.17: Values of the F1 complexity measures on all task-feature datasets. There is a lower limit of data complexity determined by the task; the actual level of complexity within that task is realized based on the specific feature engineering process used to obtain the speech feature set for that task.

The lower bound on classification complexity is determined by the task, similarly to the AUC values shown in Figure 4.11, and within each task, the individual feature engineering approach determines how close that feature set can get to the lowest possible classification complexity (at least, the lowest observed in our experiment). The tasks that were shown to be uninformative for separating between CN and CI via low AUC values (DDK and Phonation) also have the highest complexity using the F1 measure. Furthermore, the Clinical, emobase and Wav2Vec2 feature sets have the lowest complexity for most tasks, whereas the Talk2me and eGeMAPSv02 features have the highest complexity; this agrees with the AUC scores seen previously.

We also take a look at the T1 measure, shown in Figure 4.18. The T1 measure

calculates the number of hyperspheres needed to cover the entire dataset with only one class per hypersphere, normalized by the number of datapoints. There is a similar pattern of classification complexity being lower bounded by task, with individual feature engineering approaches determining how close the dataset gets to that achievable low complexity measure.

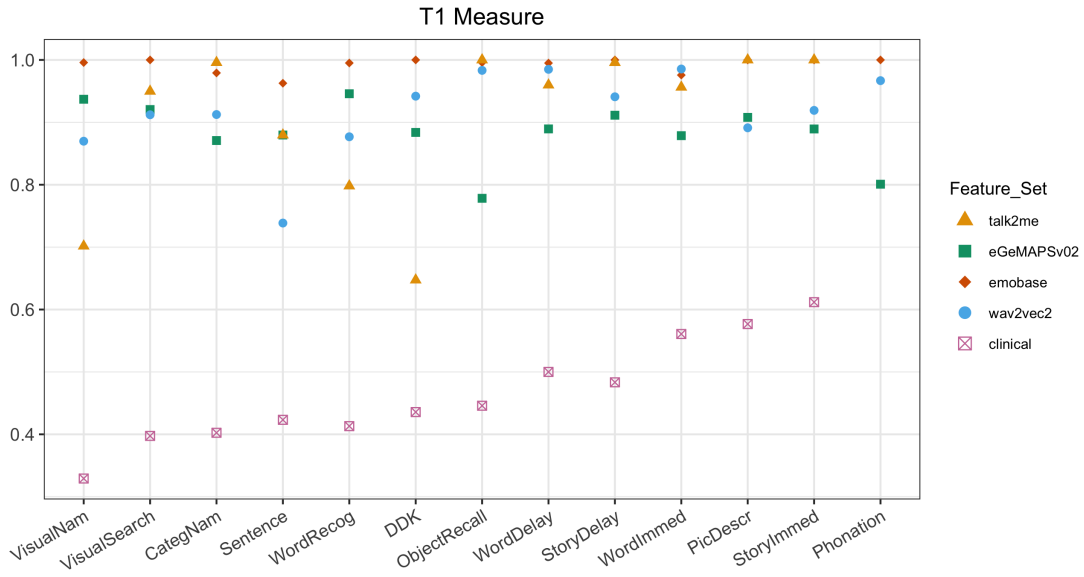

Figure 4.18: Values of the T1 complexity measure on all task-feature datasets.

Finally, as a negative example, Figure 4.19 shows the results of the F2 data complexity measure for each of the task-feature datasets, in both the original scale and log scale. F2 measures the volume of the overlapping region, which is determined by calculating the distance in each feature dimension of overlap, then normalizing by the feature range, and finally multiplying the normalized overlap distances over all features. As discussed in Lorena *et al.* (2019), this is an example of a data complexity measure which is not appropriate to use on a dataset with a large number of features, or to compare for datasets with different numbers of features, due to the measure rapidly approaching 0 as higher numbers of non-overlap distances (all between 0 and

1) are multiplied together. All of the feature sets have a score of essentially 0 for this classification measure, with the exception of the smallest feature set, the Clinical features ($p = 15$). This is understandably a result of F2 experiencing exponential decay in the feature set size. Thus, this is not a recommended measure for comparison on datasets with different numbers of features, or large numbers of features.

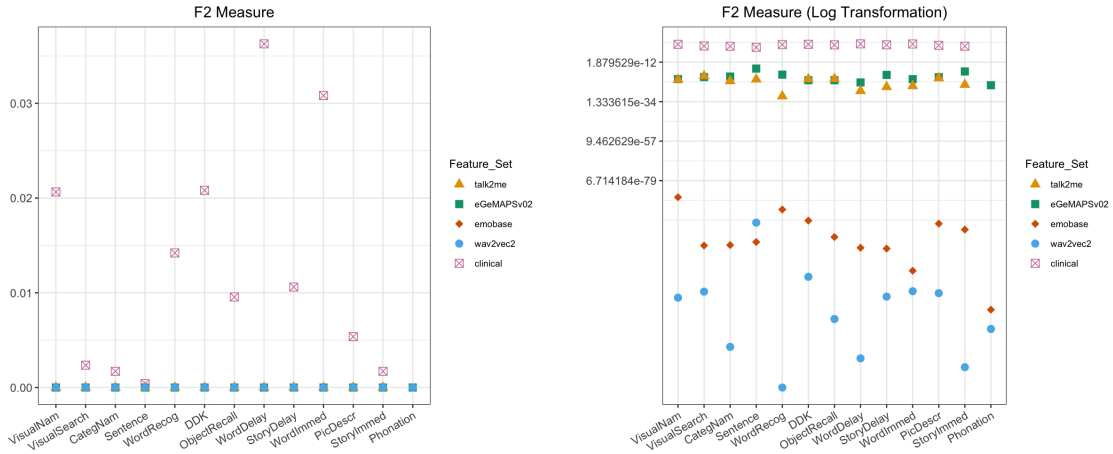

Figure 4.19: Values of the F2 complexity measure on all task-feature datasets, in both the original and log scale of the measure. The only feature set with non-0 values is the Clinincal feature set; all others are skewed by the ratio of size of the feature set to the size of the dataset, and in particular the minority class, leading to a misinformed view of complexity.

As many of the data complexity measures turn out to provide an uninformative view of complexity (marked by a positive correlation in Figures 4.14, 4.15, 4.16), we restrict the rest of the analysis of this section to the data complexity measures that appear to provide useful insights that generalize to model performance on unseen data. To be more specific, we look at how the complexity values cluster in the pairwise complexity space formed by two of the measures from the set F1, N2, and T1. This clustering analysis, similar to the PCA analysis in the previous chapter, aims to use combinations of data complexity measures to discover underlying structures in complexity space that inform how different tasks and features fall along a complexity

continuum.

Figure 4.20 shows the positioning of the data complexity measures in each of these subspaces, colored by feature. There is a strong pattern of the task-feature dataset complexities being clustered together by feature set. The Clinical features occupy the lowest complexity region in the lower left of the plots; as the Clinical features tended to have the highest AUC scores, this explains partially why these measures had the best correlation with out of sample performance.

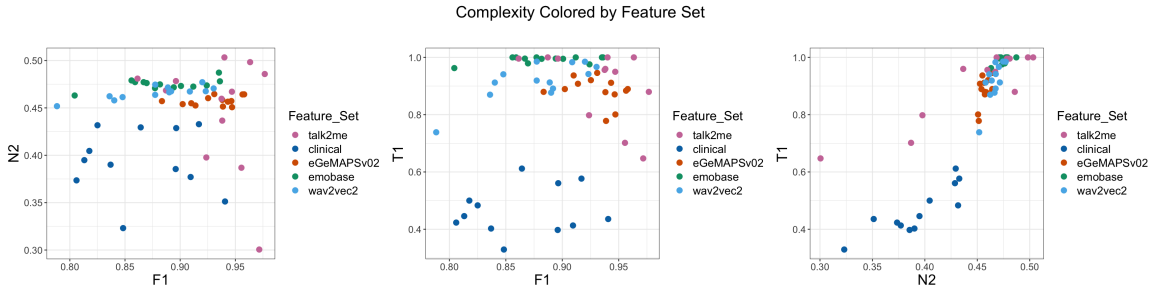

Figure 4.20: Visualization of how the data complexity measures cluster in complexity space. The complexity points have a strong clustering pattern by feature set in the complexity space spanned by F1, N2, and T1.

Figure 4.21 shows a similar kind of plot, but this time colored by task. There are no obvious patterns wherein specific tasks occupy a particular region of the complexity space spanned by these three complexity measures, at least when looking at all of the feature sets combined.

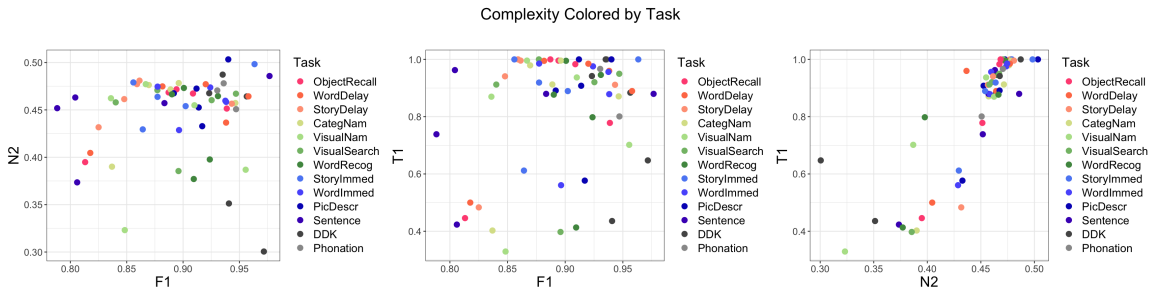

Figure 4.21: Visualization of how the data complexity measures cluster in complexity space. The complexity points do not have a strong clustering pattern by task when looking at all feature sets combined.

When we visualize only the subset of the complexity scores corresponding to a

particular feature set, however, the position of the individual tasks in the complexity space for that feature set can be further analyzed. Investigating which tasks are near each other in the complexity subspaces can spark intuition for why particular types of tasks have a similar level of complexity for the given feature engineering algorithm. We will formalize and automate this process in the final section of this chapter.

Figure 4.22 shows the distribution of complexity points by task, for only the datasets calculated using Clinical features. We see that points that are near each other correspond to tasks with similar levels of cognitive difficulty; for example, the most taxing tasks, Object Recall and the Delayed recall tasks, are by each other, and the tasks that relate to a visual searching component (Visual Search and Word Recognition) are also located near each other in the three complexity subspaces.

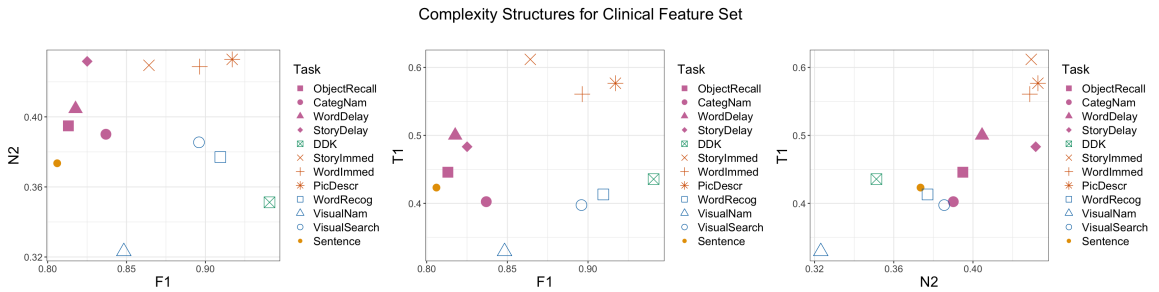

Figure 4.22: Visualization of how the data complexity measures cluster in complexity space., for just the datasets comprised of Clinical features on different tasks.

We see a similar pattern in the Wav2Vec2 features shown in Figure 4.23, except that the tasks located near each other are now more closely related to what type of speech is elicited (spontaneous speech on a description task, vs listing on a recall task, vs naming on a visual task). This may be attributable to Wav2Vec2 features measuring the acoustic properties of the speech, while lacking information related to the cognitive load of each of the tasks and how well the participants captured the content.

The three measures F1, N2, and T1 investigated in the visualization analysis from

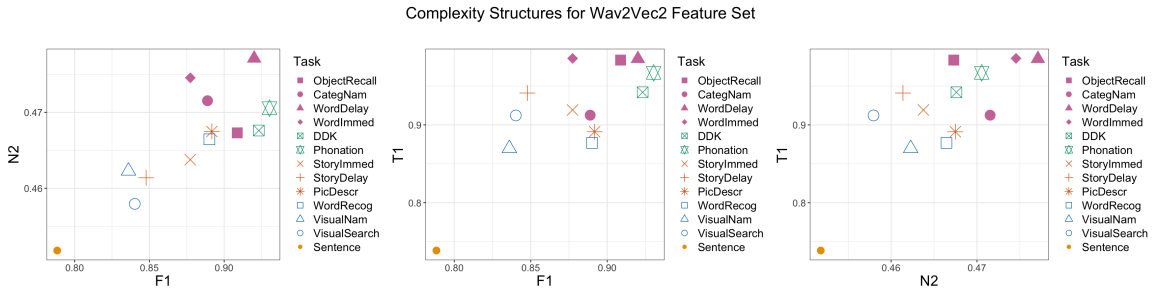

Figure 4.23: Visualization of how the data complexity measures cluster in complexity space., for just the datasets comprised of Wav2Vec2 features on different tasks.

Figures 4.20, 4.21, 4.22 and 4.23 were the ones having the best consistency with out-of-sample AUC scores of the task-feature datasets. F1 has a strong relationship to class overlap, whereas N2 and T1 measure the topology and internal structure of the classes and relate to both class overlap and the complexity of the decision boundary. N2 and T1 specifically measure the clustering structures and distances for points from different classes using a distance-based method, rather than one that considers neighbors vs enemies in a distance-agnostic manner (e.g. N1, N3, N4).

To summarize this section, while the previous analysis was fairly exploratory and qualitative, it was nonetheless an interesting exercise in assessing the data settings for which the measures of classification complexity provide useful information that relate to true model performance. We found that the classification measures are most information on datasets with a small number of interpretable features, rather than one having a large number of features (potentially more than the number of datapoints), some of which may be "junk" features.

Additionally, for the measures that correlated well with out-of-sample performance, we (somewhat unsurprisingly) saw similar patterns in the relative complexity of tasks and feature sets. Overall, the task appeared to be the main limiting factor of classification complexity, but within a given task, the particular feature engineering approach has a large impact on the complexity of the resulting speech features.

Finally, we looked at the structures of task-feature datasets in underlying subspaces spanned by a subset of the data complexity measures. The largest patterns in complexity similarity were determined by the feature set, however, when looking within a single feature set, the relative positioning of tasks within the complexity space reflected the type of information available in that task for the given feature algorithm.

4.4 Speech Elicitation Task Engineering

The previous analyses established that the speech elicitation task, or the context in which the audio recording is collected, is a major driving factor in the underlying complexity of a classification problem implied by the downstream speech features. When designing a speech-based screening test, it therefore remains critically important to carefully consider the context of the speech to be collected, or in other words, the data collection protocol. Joint engineering of both tasks and features is most likely to lead to a speech-based screening test that can detect cognitive impairment with high accuracy, which remains the target of this work.

Thus, in this final section, we concentrate on methods for guiding speech elicitation task engineering and for discovering combinations of features and tasks that are likely to work well together. A novel contribution of this section is the concept of objectively and quantitatively describing a given speech elicitation task using what we term *task meta-features*. To the best of our knowledge, this exercise has not been previously undertaken or even conceived of in the medical speech community. To make the concept explicit, we provide concrete examples of such task meta-features in the context of a speech-based cognitive screening test. While our example focuses on cognition, a similar approach could be undertaken for designing tasks to detect motor impairment, for example in ALS or Parkinson's disease.

Beyond providing a framework for objectively characterizing the speech elicitation task, we pioneer new use cases for two existing machine learning methods, CART (Breiman *et al.* (1984)) and Bayesian treed linear models (Gramacy and Lee (2008)), that leverage the task meta-features for task design. In particular, the proposed methods facilitate insights into which meta-features of the speech elicitation task are driving reductions in classification complexity.

In what follows, we first provide a literature review on key ideas from Design of Experiments, which shares similarities with our goal of engineering (designing) speech-based screening tests. Next, we introduce our CART-based protocol for speech task engineering and show the results on the BioHermes data used for the analysis from the previous section. Following the CART approach, we introduce the Bayesian method for treed linear models, and present results for this same dataset. We conclude with a discussion on the importance of carefully engineering both speech elicitation tasks and speech features for successfully creating a speech-based digital screening test.

4.4.1 Design of Experiments

The phrase "design of screening tests" may easily invoke, for statisticians, the vast field of Design of Experiments (DOE). DOE centers on finding optimal configurations of independent variables among repeated trials of an experiment, when the aim of the experiment is to estimate the impact of systematically modifying these independent variables on an outcome.

The first publication in English relating to optimal design for estimating regression models was Charles Pierce's "Note on the Theory of the Economy of Research" (Pierce (1879)); in 1918, Kirstine Smith published a work on optimal design for limited cases of polynomial regression (Smith (1918)). The roots of DOE in its present form can be traced back to Ronald Fisher's statistical experiments for increasing

agricultural output in the early 20th century (Fisher and Mackenzie (1923), Fisher (1926), Fisher and Wishart (1930)). The seminal text *The Design of Experiments* (Fisher (1935)) christened and laid the foundations for the theory. In subsequent years, DOE underwent further development, with important contributions from Yates (Yates (1937), Yates (1954)), Cox (Cox (1951)), Chernoff (Chernoff (1959)), and Box (Box and Lucas (1959)), to name but a few. Genichi Taguchi developed an approach for applying principles of DOE to manufacturing processes, which subsequently became a major application area for DOE, specifically in product quality and reliability, along with other engineering applications (Taguchi (1962), Taguchi and Phadke (1989), Taguchi (1995)). In more recent decades, DOE has found an audience in widespread application areas, including marketing, pharmaceuticals, energy, flavor engineering, and architecture (Durakovic (2017)).

At its core, Design of Experiments provides a methodology for finding statistically optimal settings when repeated experiments are being performed to test the impact of independent variables on output variables. As a classic example (Hotelling (1944)), consider two experiments in weighing 8 objects using a pan balance: in Experiment 1, each object is weighed individually in one pan, requiring 8 weighings. In Experiment 2, all 8 weighings include different combinations of objects in the right and left pans, and the estimates for each individual object are obtained by linear combinations of the results of the 8 experiments. Both experiments require 8 separate weighings, however, Experiment 2 results in an 8-fold reduction in the variance of the estimate of each object. Thus, DOE is concerned with the combinations of settings of experiments that will produce statistically optimal (to be precisely defined shortly) estimation procedures.

Each of the independent variables that is varied in the experiment is called a *factor*, X_i; the outcome being measured is denoted Y. Different experiment regimes

include One-Factor-At-a-Time (OFAT), in which one factor is varied while the other factors are held constant, full factorial designs, in which all combinations of all factors are tested, and fractional factorial designs, in which a carefully chosen subset of the full factorial designs are executed (Hicks (1964)). Full factorial designs achieve the greatest statistically efficiency in terms of variance reduction in the estimates, but require the highest number of trials and are sometimes infeasible in applications with limited resources for experiment runs. In this case, fractional factorial designs often offer an acceptable trade-off in number of trials compared to variance reduction, and are superior to OFAT experiments in most settings.

DOE aims for optimal experiment configurations, and *optimal* in this context typically implies minimizing, in some sense, the variance around the estimates for X_i in the linear regression $\sum \beta_i X_i = Y$. If more than one factor is used (which is common), instead of a single variance parameter we have the covariance matrix of the β_i, here denoted Σ; in DOE, the inverse of the covariance matrix is called the information matrix (rather than the precision matrix). Different optimality criteria have been derived, which are the solution to an optimization of a functional of the eigenvalues of the information matrix. Examples of such optimality criteria include (Atkinson *et al.* (2007)): *A-optimality* (minimizing the trace of the inverse of the information matrix), *D-optimality* (minimizing the determinant of the information matrix $\Sigma^T\Sigma$), and *E-optimality* (maximizing the minimum eigenvalue of the information matrix).

While the concept of systematically changing input parameters in order to determine their effect on an outcome has similar themes to designing screening tests, there are two key differences. These differences are most easily explained by looking at the causal graphs for the typical DOE setting compared to our setting of designing a speech-based screening test, which are shown in Figures 4.24 and 4.25. In these graphs, nodes show variables related to the problem, and arrows represent a causal

relationship; $A \rightarrow B$ means that the value of A has a causal impact on B.

Figure 4.24: Causal graph underpinning a traditional DOE setting.

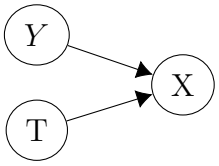

Figure 4.25: Causal graph underpinning the data used in a speech-based cognitive screening test.

The first difference between our setting and DOE is in the direction of the causal relationship between the features in the model (X) and the outcome variable (Y). In DOE, the variables that are modified have a direct causal impact on the outcome; for example, settings for production equipment (X) are varied, and the product is subsequently manufactured under these settings and tested for a particular quality (Y). This causal relationship is demonstrated by the arrow X $\rightarrow Y$ in Figure 4.24. On the other hand, in our setting of speech-based digital screeners, the speech features X are causally downstream of the outcome of interest, the cognitive class membership Y; a patient's status of being cognitively impaired is what influences the distribution of the speech features, not the other way around. This is formally notated by the causal arrow $Y \rightarrow$ X in Figure 4.25.

The other key difference between our setting and DOE is the manner in which we are able to systematically intervene in the system, in order to influence the values of the variables X being used as independent variables to a regression or classification model. In DOE, the inputs to the model can be varied directly, as shown by the causal arrow X $\rightarrow Y$ in Figure 4.24. For the speech-based screening test, we cannot, for example, force patients to produce speech samples with precisely 20 words, speaking at a rate of 4.29 syllables per second. Our lack of direct causal (interventional) control over the speech features X is formally demonstrated by the node for X having

no arrows going out of it in Figure 4.25. However, the speech screening context *does* allow for indirectly influencing the support of the speech feature distribution $\mathcal{X}$, via systematic modifications to the speech elicitation task T. The formal notation we have introduced for the task T emphasizes that the data collection protocol should be carefully controlled to achieve optimal statistical learning, even when the causal relationship between the experiment setting and the features used as model inputs is indirect.

Thus, although DOE is not directly applicable to our problem as formulated, themes from the field are related to our overall notion of design, and can be used to inform the process of speech elicitation task engineering.

An interesting direction for future work would be applying the principles of DOE to obtain optimal configurations of high and low values for task attributes, specifically for the attributes shown later in this section to have the largest impact on screening test performance. These configurations of tasks could then be trialed on small samples of participants, and results compared using DOE methodology, resulting in a decision on the best task configuration for detecting cognitive impairment.

4.4.2 Task Engineering Using CART

As a first method for task engineering, we propose to use the CART algorithm (Breiman *et al.* (1984)) to automatically extract the characteristics of the tasks that lead to reduced classification complexity. We define a set of objective task characteristics that can be evaluated for each task, called *meta-features*. The meta-features and their descriptions are shown in Table 4.5.

We want to understand which, if any, of these meta-features the best performing tasks have in common. In order to do so, we fit a regression tree using the CART algorithm, with these meta-features T_i as the independent variables and the sum-

Task Meta-Feature	Description
Type of speech	Does the task imply listing separate words, connected speech, or phonating?
Memory is taxed	Does the task require the use of memory?
Retrieval is taxed	Does the task require the use of retrieval?
Orthographic transformation is taxed	Does the task require spelling or substantive reading?
Inhibition is taxed	Does the task require the participant to refrain from naming distractor stimuli?
Idea coherence is taxed	Does the task require the participant to discuss complex ideas in a coherent way?
Visual stimuli	Does the task stimuli involve a picture of a scene or object?
Multiple stimuli screens	Are the task stimuli on a single screen, or broken up into several consecutive screens?
Stimuli in array	Do the task stimuli consist of an array of words or objects, or a single one on each screen?
Recording duration	How long does the participant have, at maximum to perform the task?
Maximum number of content units to recall	How many different objects or words does the participant need to obtain from memory, retrieval, or naming?

Table 4.5: Descriptions of the task meta-features that describe objective components of a speech elicitation task.

mary AUC score as the dependent variable. CART is described in more detail in section 2.2.1. The idea behind this approach is to let the CART algorithm automatically determine the properties that the high-performing tasks have in common, and additionally to examine the interactions between task meta-features that reduce classification complexity when used in conjunction. The insights gained from examining the tree can then be applied to designing future tasks that contain these task meta-features. Furthermore, the degree to which the task incorporates the useful meta-features can be extended, where possible.

We fit the regression tree described above on the results from the Clinical feature set as a demonstration, since that feature set had the consistently highest performance

for most of the tasks. The regression tree obtained using the `rpart` R package for model fitting and `rpart.plot` package for tree visualization is shown in Figure 4.26. This regression tree shows specifically which aspects of the speech elicitation tasks reduce classification complexity, under the assumption that the Clinical features will be used in the screening test after task completion. Table 4.6 shows which tasks are included in each of the leaf nodes from the tree in Figure 4.26.

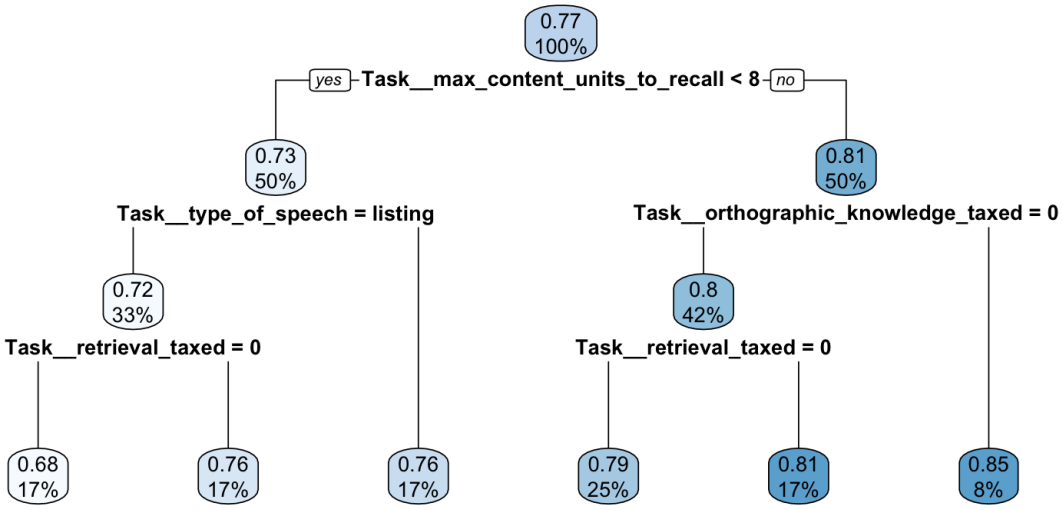

Figure 4.26: Regression tree with the task meta-features as regressors and the summary AUC score as the outcome.

The internal splits and resulting comparative AUCs in the left and right child nodes are reasonable and generate some interesting insights. The first insight is that tasks which require the participant to name or recall a higher number of individual items (or content units, in the case of the story recall task), captured by the `Task__max_content_units_to_recall` meta-feature, lead to greater differentiation between CN and CI individuals. This is a direct confirmation of the discussion in Berisha *et al.* (2021) around the use of maximum performance tasks, namely that requiring the individual to perform a task to their maximum ability can produce speech features with better properties for a classification problem.

Mean AUC	Tasks in Leaf Node
0.68	Diadochokinetic Rate, Immediate Word Recall
0.76	Delayed Word Recall, Word Recognition
0.76	Picture Description, Sentence Reading
0.79	Visual Naming, Object Recall, Immediate Story Recall
0.81	Category Naming, Delayed Story Recall
0.85	Visual Search

Table 4.6: List of the tasks that fall into each of the leaf nodes for the regression tree shown in Figure 4.26. The leaf node groups are listed in order from left-most leaf node to right-most leaf node, and can also be matched via the mean summary AUC score.

Looking further down the tree to the left of the root node and focusing on tasks that have a lower number of content units, the regression tree indicates that tasks involving listing speech type have worse performance than tasks requiring connected speech. (The split on speech type does not include Phonating spech type as an option, since the Clinical features were not calculated on the Phonation task.) Our interpretation of this finding is that tasks that involve listing individual words are the most useful when those words are related to a task with a larger number of content units that must be retrieved, remembered, or individually named.

Within the group of tasks having both a low number of contents units and a listing speech type, taxing retrieval (which is achieved in this case via a delayed recall task) produces higher differentiation compared to not taxing retrieval. This is a straightforward finding, considering that problems with memory and retrieval are one of the hallmarks of Alzheimer's disease (Venneri *et al.* (2008)), but nonetheless useful to have corroborated in the data analysis.

Looking to the right of the root node to tasks that involve a higher number of con-

tent units, the task that involves orthographic transformation (denoted by the meta-feature `Task__orthographic_knowledge_taxed`), which is the Visual Search task, has the highest classification performance. This is consistent with literature showing that orthographic transformation is impaired in Alzheimer's disease (Rodríguez-Ferreiro *et al.* (2014)). For the tasks that do not require orthographic transformation, taxing retrieval produces a higher AUC than not, consistent with the findings in the left-most two leaf nodes.

Since the regression tree is fit only to a single AUC number from each task, the findings are potentially subject to the variability induced by reducing the AUC distributions from Figure 4.8 to a single number. We therefore corroborated these findings by performing the same repeated cross validation described in Section 4.3.2, but rather than performing it separately for each task-feature dataset, we combined the task-feature datasets included in the same leaf node into one dataset before model fitting within each repetition. In other words, for each repetition, if the leaf included e.g. two tasks, we took the dataset with $2n$ rows and 15 columns (two rows per individual containing their feature values on the two tasks), and split this dataset into 3 folds for cross-validated model fitting and predicted. Grouped and stratified cross validation was performed to ensure the same participant was not included in both the training and test dataset using features from different tasks. Furthermore, we only performed this analysis using the BART model, as it was shown to have the highest performance in the majority of task-feature datasets.

The results are shown in Figure 4.27, with the boxplot showing the AUC scores over 25 repetitions, and the black dot showing the mean out-of-sample AUC calculated for each leaf node. We used 25 repetitions rather than 10, due to increased availability of computational resources when running the analysis using only one model and one small feature set (Clinical features).

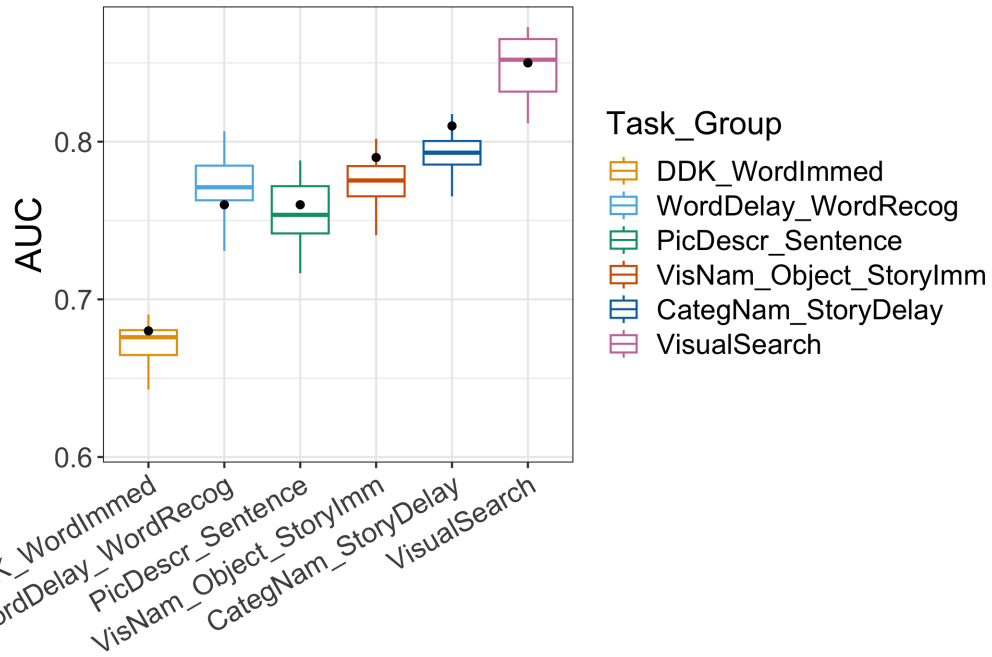

Figure 4.27: AUC distributions over 25 repetitions for each of the task groupings obtained from the leaf nodes in Figure 4.26. The black point shows the AUC score from the regression tree. Overall, the trend aligns with the mean AUCs produced by the regression tree.

The mean AUC from the predicted values in the leaf nodes of Figure 4.26 are always contained within the distribution of AUC scores for the jointly fitted model, shown in Figure 4.27. However, the groups corresponding to the four central leaf nodes have very overlapped AUC distributions, implying that the split for taxing retrieval between leaf nodes 4 and 5 likely does not represent a true difference between these two sets of tasks. However, the general trend matches what was observed in Figure 4.26, and the remaining splits seem to represent real differences in task performance based on meta-features. Overall, averaging the AUCs obtained by fitting separate datasets is a viable approximation for the AUC achieved by fitting the datasets together.

We conclude with remarks on how this method can be used for guiding task engineering. In the concrete case of a speech-based screening test that is used for detect-

ing cognitive impairment, we recommend including tasks that tax at least 8 content units (objects, words, or elements of a recalled story), tax retrieval, and tax orthographic transformation. Practitioners should consider incorporating these elements into speech-based screening tests for cognitive impairment incident to Alzheimer's disease.

Additionally, we recommend designing tasks which incorporate these meta-features to a greater extent; for example, including an even higher number of content units, presenting a stimuli with a more difficult orthographic transformation (e.g., more complex sentences, or more challenging spelling criteria to remember). As pushing the task to be too difficult can result in floor effects, and can also be discouraging for cognitively impaired participants to complete, we recommend experimenting with different task setups via A/B testing on small groups of participants, in order to find the right balance of task difficulty in the general direction recommended here.

To summarize, in the previous section we established that task engineering is a large driver of classification complexity. In this subsection, we have introduced a way to obtain automatic insights into *which* components of the speech elicitation task may be driving the low complexity that we seek. Furthermore, we have presented ideas for turning these automatic insights into new speech elicitation tasks and testing them out in a controlled manner.

4.4.3 Task Engineering Using Treed Probability Models

In the previous section, we saw that the patterns obtained by averaging AUCs over separate models, as compared to directly fitting models on the data from the tasks grouped together, produced similar but not identical results. (The black points in Figure 4.27 are not identical to the median AUC from a model jointly fitted to these tasks). Unfortunately, fitting models to all possible subsets of tasks groupings

would involve performing the repeated CV analysis on $2^{12} = 4096$ different subsets of task groupings; furthermore, even if these results could be obtained, the method for automatically separating the tasks into groups by task meta-features, using the performance of the 4096 task groupings, is not obvious.

Treed probability models provide a method for automatically performing such an analysis. The end result of a treed linear probability model is a tree-based structure, where data are funneled into leaf nodes based on the values of splitting variables and cutpoints at each internal node, and then data within each leaf node are used for fitting a separate linear regression model to the class outcomes Y coded as 0-1 numerical values.

A treed probability model in which the task meta-features are used for internal splitting variables, and the speech features are used for fitting the leaf-node probability models, produces a similar set of insights as the CART-based approach for task engineering. The difference in using treed probability models, rather than single regression trees, is that leaf-node results are based on fitting a single model over data combined from several tasks, instead of averaging over results from individual models on each task.

Our second proposed method for task engineering implements these ideas using Bayesian treed linear models introduced by Gramacy and Lee (2008) and implemented in the `tgp` R package (Gramacy (2007)). Due to computational constraints, we only perform this analysis on the set of Clinical features. Although `tgp` includes functionality for fitting Bayesian treed Gaussian processes, which would potentially improve the results via a more flexible model in the leaf node, due to computational constraints we only used the `btlm` function for Bayesian treed linear models.

We take a moment to remark on the contribution of this section, namely, presenting a novel use case for treed regression and classification methods. In a typical

use of such methods, the same variables are by default used for both partitioning the domain and fitting the model within each partition, letting the data drive which variables should be used for internal splits versus leaf node models. An ideal example for this use case are non-stationary Gaussian processes, which form the main motivation behind the functionality in the `tgp` (**t**reed **G**aussian **p**rocesses) package. While functionality exists for restricting the variables used for internal splits and leaf node models, it can only be utilized via knowledge of specific obsolete parameters for the function call. The use of such functionality does not appear to be standard, judging by 1) its absence from the package vignette and 2) the fact that these parameters are buried in the help page and only included in a tangential control function, rather than in the main package functions.

Our use of treed probability models is novel in several ways. First, not only do we use separate sets of variables for internal splits vs leaf node models using the obscure function parameters mentioned above, but the two sets of variables are not just two subsets of the same input space $\mathcal{X}$. Rather, the two sets of variables have a hierarchical relationship, in which the variables used for internal splits are "meta" variables that determine the distribution of the actual input domain, used for fitting the regression leaf models. In other words, we are not simply choosing $X_1, \ldots, X_i$ for internal nodes and $X_{i+1}, \ldots, X_p$ for leaf node models, from the input vector X. Rather, we are partitioning the domain of meta-features $\mathcal{T}$, and then within each partition of $\mathcal{T}$, fitting a model to the input features $\mathcal{X} \mid \mathcal{T}$ that arise when collected under the setting specified by that partition of the task meta-features.

The second novel contribution of this section is the way in which we use this hierarchical division of meta-variables vs model input variables. We use the fitted treed probability model to gain insights into which part of the meta-feature space $\mathcal{T}$ produces the best data $\mathcal{X} \mid \mathcal{T}$ for a particular classification problem. Thus, we use

an existing method in Bayesian machine learning in a new setting, namely speech elicitation task engineering.

With these remarks in mind, we now turn to describing the specifics of using a treed linear probability model for task engineering. The input to the model is a $(n \cdot n_{\text{tasks}}) \times (p + m)$ matrix $(\mathbf{X}, \mathbf{T})$, where n is the number of participants, n_{tasks} is the number of individual tasks, p is the number of extracted speech features and m is the number of task meta-features. In our example using the Clinical features on the BioHermes data, $n = 241$, $n_{\text{tasks}} = 12$, $p = 15$ and $m = 11$. Each row is a vector comprising the set of speech features and task meta-features from participant i on task j, and is denoted $(\mathrm{x}_{i,\mathrm{t}_j}, \mathrm{t}_j) = (x_{i,\mathrm{t}_j,1}, \ldots, x_{i,\mathrm{t}_j,p}, t_{j,1}, \ldots, t_{j,m})$, where $\mathrm{t}_j = (t_{j,1}, \ldots, t_{j,m})$ is the vector of task meta-features describing the j^{th} task, and $\mathrm{x}_{i,\mathrm{t}_j} = (x_{i,\mathrm{t}_j,1}, \ldots, x_{i,\mathrm{t}_j,p})$ are speech features obtained from a participant performing the j^{th} task. In the function call, we only allow the variables $T_1, \ldots, T_m$ to be used as internal splitting variables, and we only allow the variables $X_{T,1}, \ldots, X_{T,p}$ to be used for fitting the leaf node model. These constraints are achieved using the `splitmin` and `basemax` parameters in the `btlm` function call.

The hierarchical model specification for the Bayesian linear models fitted in the leaf nodes is Gramacy (2007):

$$\begin{aligned} \mathbf{Y} \mid \beta, \sigma^2 &\sim N_n(\mathbf{F}\beta, \sigma^2\mathbb{I}) & \sigma^2 &\sim IG(\alpha_\sigma/2, q_\sigma/2) \\ \beta \mid \sigma^2, \tau^2, \mathbf{W}, \beta_0 &\sim N_p(\beta_0, \sigma^2\tau^2\mathbf{W}) & \tau^2 &\sim IG(\alpha_\tau/2, q_\tau/2) \\ \beta_0 &\sim N_p(\mu, \mathbf{B}) & \mathbf{W}^{-1} &\sim W((\rho\mathbf{V})^{-1}, \rho), \end{aligned}$$

where $\mathbf{F} = (\mathbf{1}, \mathbf{X})$ is the input matrix $\mathbf{X}$ with an intercept added, $\mathbb{I}$ is the $n \times n$ identity matrix, $\mathbf{W}$ is an $m \times m$ matrix, N is the Multivariate Normal distribution, IG is the Inverse-Gamma distribution, and W is the Wishart distribution. The prior parameters α_σ, q_σ, μ, $\mathbf{B}$, ρ, and $\mathbf{V}$ are treated as known. The input matrix $\mathbf{X}$ for

fitting this model consists of speech features from all participants on the tasks that belong to the partition of $\mathcal{T}$ corresponding to the given leaf node; $\mathbf{Y}$ is the outcome vector consisting of 0's and 1's, repeated the same number of times as the number of tasks in the leaf node's partition.

For implementing the tree $\mathfrak{T}$, Gramacy and Lee (2008) use a modification of the tree-generating process from Chipman *et al.* (2010), using the same operations *grow*, *prune*, *change*, *swap*, along with a new operation *rotate* to improve Markov chain mixing (Gramacy and Lee (2008)). The probability of splitting a node η during a *grow* operation is $p_{\text{SPLIT}}(\mathfrak{T}, \eta) = a(1 + q_\eta)^b$, where q_η is the depth of node η, and a and b are parameters determining the overall prior on tree depth. See Gramacy and Lee (2008) for more details on the model specification and MCMC process, which uses reversible-jump MCMC, with some simplifications for greater computational efficiency.

Figure 4.28 shows the Maximum A Posteriori (MAP) tree obtained from fitting the `btlm` function (Bayesian Treed Linear Model) from `tgp` to the Bio-Hermes data. The `tgp` tree-plotting function was modified to return the MAP tree with an AUC value in the leaf node, rather than the estimate of σ which is shown in the implemented version. For each leaf node, the AUC score is the in-sample AUC computed on all of the data from the tasks included in that leaf node based on the task meta-feature splits. The CI probabilities $P(Y = 1 \mid \mathrm{X}, \mathrm{T})$ used for calculating AUC are the predictions using the posterior means for the linear model parameters in the leaf node. Although the AUC values are higher than the out-of-sample AUCs shown in the previous section, the regularization of the model via priors favoring simpler models still prevents complete over-fitting to the training data.

Table 4.7 shows the set of tasks that fall into each of the leaf nodes based on the task meta-feature splitting variables and cutpoints.

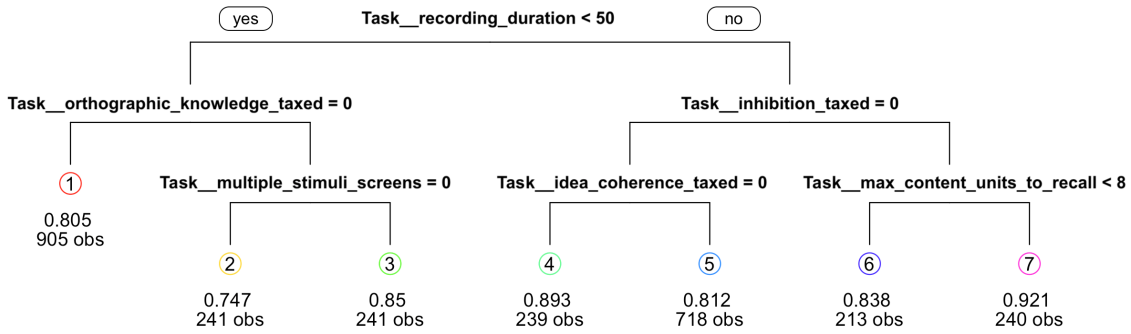

Figure 4.28: MAP tree obtained from fitting a Bayesian treed linear model to the BioHermes speech data. The leaf nodes show the in-sample AUC for the tasks grouped in the leaf nodes, based on the splitting variables shown in the internal nodes. The AUC is obtained using the linear model in each of the leaf nodes to predict the probability of cognitive impairment on the full dataset contained in that node.

Mean AUC	Leaf Node Number	Tasks in Leaf Node
0.805	1	Category Naming, Object Recall Delayed Word Recall, Immediate Word Recall
0.747	2	Diadochokinetic Rate
0.85	3	Sentence Reading
0.893	4	Visual Naming
0.812	5	Immediate Story Recall, Delayed Story Recall Picture Description
0.838	6	Word Recognition
0.921	7	Visual Search

Table 4.7: List of the tasks that fall into each of the leaf nodes for the regression tree shown in Figure 4.28. The leaf node groups are listed in order from left-most leaf node to right-most leaf node, and can also be matched via the posterior in-sample AUC score and leaf number (shown in the colored circle on top of the AUC score in Figure 4.28.

The obtained decision tree can now be explored by comparing the in-sample AUC from predicting on a single, fused dataset (Figure 4.28) to the out-of-sample AUC obtained by averaging over all summary AUC scores for tasks in the leaf nodes (Figure 4.26). Furthermore, the internal nodes can be investigated for insights into which task

meta-features are driving classification complexity.

The predicted in-sample AUCs have some similarities to the out-of-sample AUC achieved on the task groupings from the CART-based approach, at least in terms of relative order. The Diadochokinetic Rate task is clearly inferior than others in terms of lower AUC, and the Visual Search task also appears to have the highest performance. The Delayed and Immediate Word Recall are on the lower end, but unlike in the previous results, Category Naming and Object Recall also fall onto the lower side of performance. The remaining tasks are relatively mixed having intermediate AUC scores.

Interestingly, even with this changed ordering of task performance, we still see some of the same patterns in terms of splits on task meta-features in Figure 4.28 as we did for Figure 4.26. Taxing orthographic knowledge, taxing inhibition, and including a larger number of content units all lead to an increase in performance, i.e. a classification problem with reduced complexity.

We also see other task meta-features making an appearance in the internal nodes for splitting. Making use of multiple stimuli screens leads to improved performance, which aligns with the maximum performance findings, since spreading a task over multiple stimuli screens normally implies a lengthy and sometimes more complex task. The idea coherence split is curious; it appears that for longer tasks that do not require inhibition (meaning participants can speak freely), the listing/naming tasks are superior to those involving idea coherence for separating between Alzheimer's patients and those with normal cognition.

To wrap up our two proposed methods for insight generation, we compare the CART-based method to the method based on Bayesian treed probability models. Aside from the obvious difference of using a Bayesian method compared to a non-Bayesian method, the key difference between the two can be summarized, loosely,

as meta-regression (CART) versus data fusion (`tgp`). In a meta-regression, each sample input is obtained from a single dataset or study; the independent variables are features about the dataset, and the outcome variable is a summary performance metric, traditionally a treatment effect. See Thompson and Higgins (2002) for a deeper background on meta-regression. Here, our dataset inputs are each of the task-feature datasets, and the outcome summary performance metric is the summary AUC score.

On the other hand, data fusion involves combining multiple datasets together as input to the same model, from which a single performance metric summarizing the performance of the combined data is obtained; see Castanedo (2013) for more information and examples of data-fusion. In the Bayesian treed linear model, we fit a single model to multiple combined task-feature datasets and obtain a single AUC for the combined data.

The Bayesian method takes advantage of increased statistical efficiency that is achieved by fitting models with the same number of features on a larger dataset (data fusion), rather than combining results from several models fit to smaller individual datasets (meta-regression). Additionally, the Bayesian method utilizing `tgp` is a novel and satisfying use of treed probability models in an innovative and challenging problem setting.

However, the intense computational demands required for the MCMC make the Bayesian treed probability method infeasible for large datasets with many participants, tasks and features. Therefore, the CART-based approach is recommended for use in a typical setting of speech elicitation task design, in which the number of features (at a minimum) is likely to be large.

4.4.4 *Automatic Insights Guide Future Task Engineering*

Both of the proposed methods (CART-based and Bayesian treed linear models) can be used to guide the process of either refining existing speech elicitation tasks or designing entirely new ones. First, the task meta-features used as internal splitting variables should be interrogated to ensure the obtained insights align with established clinical knowledge. For example, in Figure 4.26, the decision tree indicates that a greater number of content units being named or recalled produces better classification performance. This aligns with the clinical understanding that a more cognitively challenging task will produce a greater difference in performance between cognitively normal and impaired groups.

Beyond individual features, the decision tree provides insights into interactions between the task meta-features that produce better classification results when jointly present. Using Figure 4.26 as an example again, the two splits on the far left side of the plot show that when the speech elicitation type is a listing (naming) task, taxing retrieval via a delayed recall produces better classification performance than not taxing retrieval, which again aligns with clinical understanding of cognitive function. Critically, interactions between task meta-features may not be apparent to researchers using traditional, non-data-driven approaches.

After interrogating (where available) the clinical validity of insights gained from the decision tree, new or refined tasks can be designed to incorporate the beneficial task meta-features (or combinations) to a greater extent. For example, in a word recall or an object recall task, the number of content units can be increased beyond the current amount, choosing 2 or 3 list sizes for comparison. Incorporation of the important meta-feature(s) should be balanced to avoid floor or ceiling effects, and to not deter patient motivation to finish the test by making it too difficult. This balance

can be explored prior to a pilot study via market research with a small number of individuals. An area for future work is to alternatively use fractional factorial designs from DOE to choose combinations of the important meta-features (*factors* in DOE terminology) that should be jointly incorporated into new versions of speech elicitation tasks.

Finally, the new task versions can be compared via A/B testing in small pilot studies, which are then used for selection of the final task to be run in a full validation study.

The reality underlying the recommendations above is that designing a good cognitive screening test is a lengthy and iterative process. Just as traditional cognitive assessments have been subsequently refined over years and multiple validation studies, speech-based cognitive screening tests should be updated and improved over multiple iterations and separately validated in new studies. By undertaking our novel approach to task design, practitioners can move toward reliable and accurate speech-based cognitive screening tests that can be successfully deployed in the healthcare setting.

4.4.5 Joint Engineering of Features and Tasks

As a final addendum to our CART-based method of task engineering, we produce a similar CART tree that can generate ideas about which tasks and feature sets work well together. We first fit a regression tree using the CART algorithm, with the independent variables being two categorical variables containing the task and feature set names, and the dependent variable being the summary AUC score shown in Figure 4.11. Figure 4.29 shows the tree obtained using the `rpart` and `rpart.plot` R packages. For easier visualization, we set the `maxdepth` parameter to 3.

Splitting on the task variable in the root node of the tree in Figure 4.29 is another

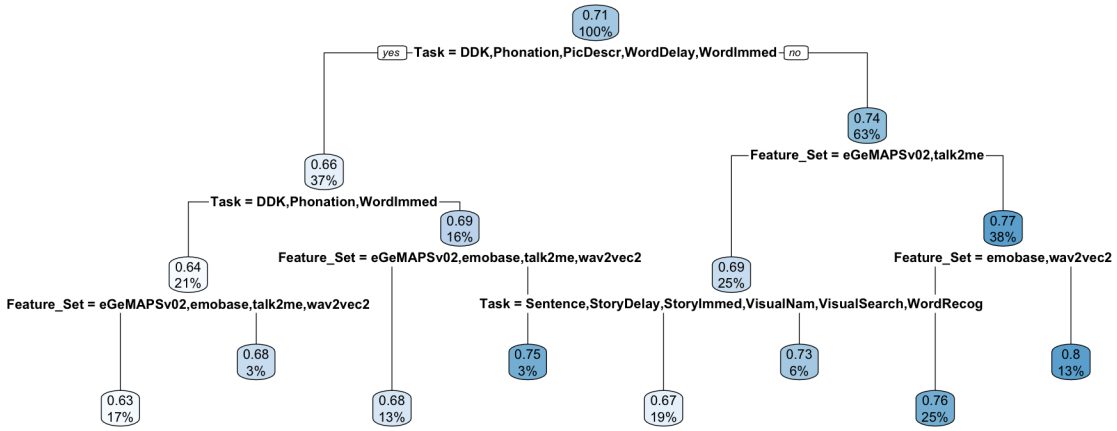

Figure 4.29: Regression tree with `Task` and `Feature_Set` as regressors and the summary AUC score as the outcome being predicted. Splitting on `Task` at the root node means that the `Task` variable contains subgroups having more similar clusters of high and low AUC values than the `Feature_Set` variable.

indication of the importance of good task engineering; it means that splitting on `Task` provides a greater variance reduction in the children nodes combined than splitting on `Feature_Set`. In other words, there are more clustered patterns of high and low AUC scores when task-feature datasets are grouped by task, than when they are grouped by feature set.

The tasks that result in higher AUC scores (over all of the feature sets taken together) are: Sentence Reading, Immediate Story Recall, Delayed Story Recall, Visual Naming, Visual Search, Word Recognition, Category Naming, and Object Recall. Given one of these good tasks (moving to the right child of the leaf node), the next level of AUC improvement is achieved via good feature engineering, as demonstrated by two subsequent splits on the feature set to obtain the best performance in the right-most leaf node. On these better tasks, the eGeMAPSv02 and Talk2me features show the worst performance, Emobase and Wav2Vec2 have better performance, and the Clinical features show the best performance. This corroborates our earlier finding that good task engineering is necessary in order to move the conditional distribu-

tions $P(\mathrm{Z} \mid \mathrm{T}, Y = 1)$ and $P(\mathrm{Z} \mid \mathrm{T}, Y = 0)$ to a region of $\mathcal{Z}$, such that successful speech feature engineering algorithms can be found. Furthermore, it shows that informed speech feature engineering is necessary to transform the original audio from "good" baseline tasks into meaningful and informative speech features with reduced classification complexity. Poor speech feature algorithms $f_{feature}$ on a good task are no better than the best speech feature algorithms on a poor task. This is clear from comparing the AUC of the worst speech feature sets (eGeMAPSv02, Talk2Me) on the set of good tasks (mean AUC = 0.69 on the internal node with a depth of 3, second from the left) to the AUC of the best speech feature set (Clinical) on one of the worst tasks, DDK (second from the left leaf node, AUC = 0.68). Both task and feature engineering must be done in tandem to achieve good classification performance.

Looking to the left side of the root node in Figure 4.29, we have the worse tasks (when considered over all feature sets): DDK, Phonation, Picture Description, Delayed Word Recall, Immediate Word Recall. Here, we see that the very worst tasks for separating cognitive groups, Phonation and DDK, are split off even before splitting on the feature set. In other words, regardless of the feature set under consideration, these tasks showed the worst performance. This additional split on the Phonation and DDK tasks is a secondary visualization of how the maximum AUC score in Figure 4.11 (shown by the dashed line) had the lowest level for the Phonation and DDK, substantially worse than on other tasks.

Lastly, within both the worst performing tasks (DDK, Phonation) and the tasks with medium performance (Picture Description, Immediate Word Recall, Delayed Word Recall), the difference between the Clinical features and the rest of the feature sets is greater than any other grouping of feature sets. Thus,even on tasks with a higher inherent level of classification complexity, good feature engineering can still improve classification performance, though not as much as simply collecting audio on

a better task to begin with.

While the regression tree in Figure 4.29 is simply another way of visualizing and clustering the results from Figure 4.11, it nonetheless provides useful insights, including 1) the overall dominance of task engineering, rather than feature engineering, as the main driver of classification complexity; 2) the importance of good feature engineering to extract maximum classification performance from tasks with a high performance ceiling; 3) the superiority of using a carefully designed small set of speech features that measure clinically interpretable aspects of speech, known to change in the disease being screened, at least in this particular analysis. While this last point is not a main focus of this dissertation, we found it worth mentioning, in particular because it aligns with claims made in Berisha *et al.* (2021).

As a final analysis, we extend the procedure described above by fitting a CART tree using the task meta-features defined in Table 4.5 as independent variables, in place of the task names directly. This analysis allows us to derive insights on which components of the tasks are driving classification complexity, for all feature sets in conjunction. We can also investigate whether individual feature sets work best for tasks with specific meta-features. Figure 4.30 shows the regression tree obtained by fitting the CART algorithm on both the task meta-features and the feature set names.

Similarly to Figure 4.26, the maximum number of content units is the overall driving force behind reduced classification complexity, when taking all feature sets in conjunction. For the eGeMAPSv02 and Talk2me feature sets, we see that taxing retrieval provides an improvement for the low-content-unit tasks, similarly as for the Clinical features in Figure 4.26. On the other hand, having a short recording duration is more impactful for these feature sets on the high-content-unit tasks.

In summary, using the CART approach with both task meta-features and feature set as regressors provides an avenue for discovering task insights that pertain to

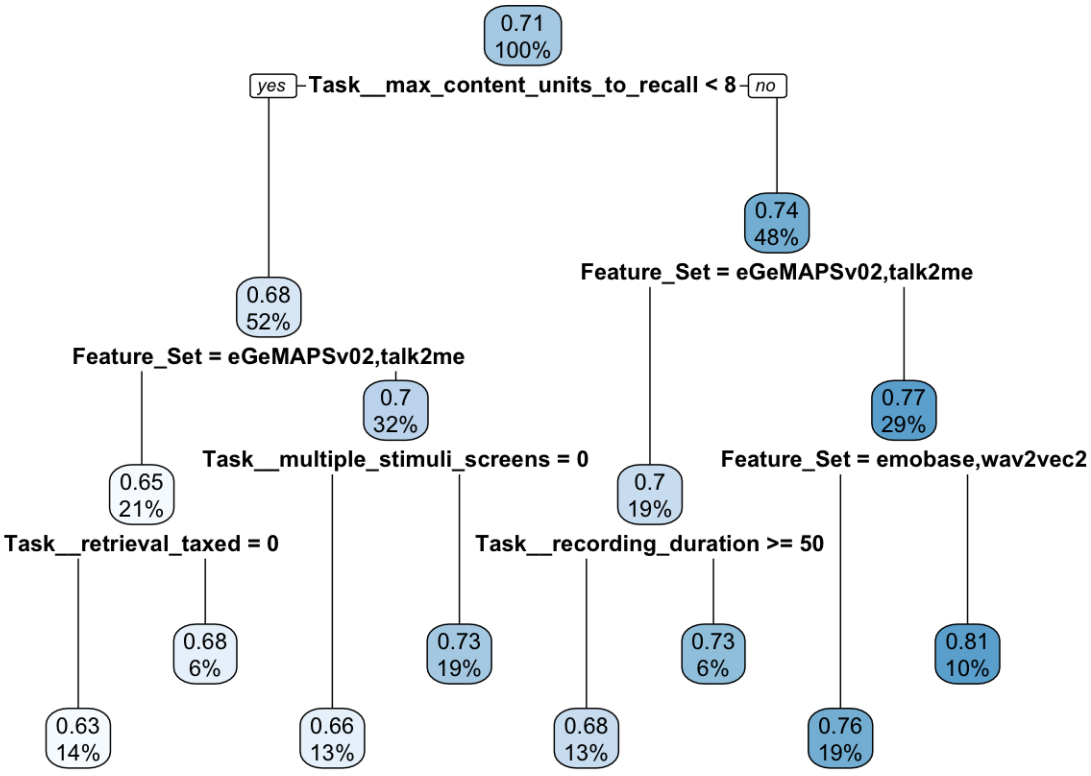

Figure 4.30: Regression tree with the task meta-features and feature set as regressors.

specific feature sets. The obtained trees can generate ideas for which feature sets and task meta-features may work well together in combination, helping guide future task engineering in a feature set-agnostic or feature set-specific manner.

4.4.6 Discussion

The previous analyses use an objective quantification of characteristics of speech elicitation tasks to derive insights on the task components driving classification performance. We proposed a CART-based approach for automatically grouping high-performing tasks using partitions driven by the objective task meta-features, and showed how the obtained tree can be used to generate recommendations for future

speech elicitation tasks that are likely to show good classification performance. We discussed these insights both in a feature set-specific examples (Figure 4.26) and an example involving interactions between both feature sets and task meta-features (Figure 4.30). Furthermore, we proposed an alternative method for obtaining a similar regression tree using a Bayesian treed probability model (Figure 4.28). Both approaches represent an innovative way to use existing methods in service of insight generation driving future task engineering.

There are some caveats to the proposed methods. First of all, the fact that a particular task meta-feature is the best one for separating tasks into high and low AUC buckets, does not mean that that particular meta-feature is singularly responsible for the difference in task performance. The insights gained from the internal splitting nodes should be corroborated by existing clinical evidence in the particular indication being screened, and also by task development and validation activities such as A/B testing, or the exploratory or confirmatory factor analyses typically performed when validating neuropsychological cognitive batteries.

Secondly, the insights derived in the classification tree are subject to the particular task meta-features that the practitioner chooses to define and evaluate for a particular set of tasks. Thus, an initial theory for task meta-features that are driving classification complexity and screening test performance is required to use this method, or at minimum, ideas for objective aspects of a potentially extremely unique set of tasks that are shared among multiple tasks.

Despite these limitations, the ideas presented here offer a concrete path to insight generation for the purposes of speech elicitation task engineering. Our emphasis on the importance of designing good speech elicitation tasks, rather than fully focusing on speech feature engineering and model building, is in and of itself a contribution that advances the work to date on speech-based screening tests used for screening

medical conditions.

The proposed method can additionally be used for choosing between several different versions of the same speech elicitation task that use different stimuli. Rather than comparing classification model performance on 12 different tasks, we could compare performance on 12 versions of the same task, and define a set of meta-features that differentiate between these task versions in an objective way.

While we have named the previous section *Joint task and feature engineering*, the proposed method is limited to using existing feature sets in full for a given task. An exciting direction for future work would be designing an adaptation of the treed linear probability model that incorporates feature selection between *different* feature sets in the tree-splitting process. This could be accomplished, for example, by incorporating Bayesian Lasso models (Park and Casella (2008)) into the leaf nodes, rather than Bayesian linear models. Alternatively, a non-Bayesian approach that uses traditional feature selection methods like the wrapper or filter methods described in the empirical analyses could be considered. The feature selection methods did not seem to have a large impact when used only within a single feature set, but we theorize that these methods may prove more useful when applied to select a small number of features from different feature engineering algorithms. The selected features, applied to the best speech elicitation tasks, could thus combine complementary information from hand-crafted clinically relevant features and built-in DNN-based speech features, potentially making new inroads into reducing classification complexity and achieving higher performance for speech-based cognitive screening tests.

Chapter 5

SUMMARY OF CONTRIBUTION AND FUTURE WORK

This dissertation offers new applications of machine learning in service of designing better screening tests. We focus in particular on two types of screening tests: adaptive tree-based screening tests that select a small number of items for administration from a large item pool, and speech-based screening tests which require participants to perform a speech elicitation task that is recorded and subsequently processed for screening. We demonstrate our novel methods in applications to screening for youth delinquency (adaptive tree-based screening tests) and screening for cognitive impairment (speech-based screening tests).

Our original contribution in Chapter 2 is a new method for designing adaptive tree-based screening tests that are specifically optimized for a particular population and test setting, using an innovative Bayesian decision theory framework. The proposed method offers a principled approach for delineating and evaluating the trade-offs of shortening the original assessment, which is a lengthy test consisting of all the items in the item pool. In providing a new method for tree-based adaptive test design, we extend prior work on both posterior summarization for model selection and youth delinquency risk assessment, incorporating ideas from these separate fields to tackle the problem of designing a tree-based screening test.

Chapter 3 offers a foundational literature review that connects three disparate fields: statistical learning theory, information theory, and empirical data complexity measures. This literature review highlights connections between the fields via the insights they provide into classification complexity, and establishes a unified language and notation for discussing classification complexity in the context of speech

elicitation task design. We compare both theoretical and empirical measures of classification complexity from these separate fields via a carefully designed simulation study that explicitly controls two key aspects of classification complexity. Finally, this review and simulation study provide a language for formal insights on the curse of dimensionality in digital health, which we contribute at the end of the chapter. This discussion expands on ideas from our paper *Digital Medicine and the Curse of Dimensionality* (Berisha *et al.* (2021)).

With the theoretical and empirical understanding of classification complexity established, in Chapter 4 we apply our learnings to the problem of designing a speech-based screening test. Our main contribution from the first half of Chapter 4 is a demonstration via a theoretical discussion and an empirical analysis of the relative merits of task and feature engineering in the context of a speech-based cognitive screening test. The theoretical discussion utilizes the same formal notation about classification complexity presented in Chapter 3. The empirical demonstration supporting our theoretical ideas via a large scale analysis comparing the classification complexity of different speech elicitation tasks and speech feature sets. The combination of theoretical discussion and empirical analysis generate several new insights, the most important of which are:

(1) Speech elicitation task engineering is necessary to reduce the lower limit of classification complexity for the engineered speech features;

(2) Intelligent feature engineering is required to extract maximum classification performance from well-designed speech elicitation tasks;

(3) Feature engineering is most useful when it utilizes external *Knowledge* or *Data*, bringing outside information to the classification problem posed by a particular medical speech dataset.

As a side note, the large scale analysis included a much greater variety of speech elicitation tasks than have been previously compared in the speech literature. Specifically, we compared spontaneous speech, naming, reading, and phonating tasks across the same participants in the same study, whereas prior work was focused on comparing different versions of the same picture description task, or different types of spontaneous speech tasks only.

After demonstrating the fundamental importance of speech elicitation task design (point (1) above), in the latter part of Chapter 4 we propose new methods for designing speech elicitation tasks. First, we introduce the novel concept of task meta-features, which can be used to objectively quantify different aspects of a speech elicitation task. Second, we propose new applications of two different machine learning methods, one based on meta-regression and the other based on data fusion, to gain insights into the meta-features driving classification complexity for a speech elicitation task. We furthermore propose a process for how to use these objective insights to guide future speech task design efforts.

The overall contribution of Chapter 4 is to shine a light on the importance of carefully designing the speech elicitation task underlying a speech-based screening test, and providing specific recommendations to guide the design process. The contributed proposals rely on high-level ideas similar to traditional design of neuropsychological cognitive batteries, but subsequent task iterations are proposed using a combination of clinical guidance and data-driven evidence, rather than clinical intuition alone.

To summarize, the contributions of this dissertation are both novel ideas and novel methods in service of designing better screening tests, which are either conveniently brief, have acceptable accuracy, or both (where possible). Along the way, we have woven together ideas from a multitude of fields, including Bayesian decision theory, machine learning, statistical learning theory, information theory, data complexity

measures, design of experiments, and neuropsychology.

While directions for future work are numerous, we highlight two here. The first direction is creation of publicly available software that can be used to easily implement the method from Chapter 2 for designing tree-based adaptive screening tests. While the code used to produce the presented analysis is publicly available, a more user-friendly interface would allow for greater adoption of the method in important application areas such as education and federal-sponsored survey efforts.

The second direction is expanding the analysis in Chapter 4 to compare several end-to-end classification methods on different speech elicitation tasks, rather than the separate steps of extracting speech features and using them as input to a variety of classification models. Our analysis established the importance of task engineering, but combined task engineering and end-to-end model engineering is an exciting direction for future exploration. While initial results on a limited subset of the speech elicitation tasks showed similar classification performance between end-to-end and separate feature engineering, assessing performance of state-of-the-art models on a wider variety of speech elicitation tasks could yield valuable insights on which combinations of model families and speech elicitation tasks work well together. These insights could then be used to guide efforts toward joint task engineering and model development, maximizing classification performance on speech data collected from a well-designed speech elicitation task.

2020 International Narcotics Control Strategy Report, Volume I: Drug and Chemical Control, U.S. Department of State, Bureau of International Narcotics and Law Enforcement Affairs, retrieved from https://www.state.gov/2020-incsr-volume-i-drug-and-chemical-control-as-submitted-to-congress/ (2020).

Abt, T. and C. Winship, "What works in reducing community violence: A meta-review and field study for the Northern Triangle", Tech. rep., Democracy International, 7600 Wisconsin Avenue, Suite 1010 Bethesda, MD 20814, retrieved from https://www.usaid.gov/sites/default/files/USAID-2016-What-Works-in-Reducing-Community-Violence-Final-Report.pdf (2016).

Acheampong, F. A., H. Nunoo-Mensah and W. Chen, "Transformer models for text-based emotion detection: a review of BERT-based approaches", Artificial Intelligence Review pp. 1–41 (2021).

Ali, S. M. and S. D. Silvey, "A general class of coefficients of divergence of one distribution from another", Journal of the Royal Statistical Society: Series B (Methodological) **28**, 1, 131–142 (1966).

Almond, R. G. and R. J. Mislevy, "Graphical models and computerized adaptive testing", ETS Research Report Series **1998**, 1, i–24, URL `https://doi.org/10.1002/j.2333-8504.1998.tb01753.x` (1998).

Alon, N., S. Ben-David, N. Cesa-Bianchi and D. Haussler, "Scale-sensitive dimensions, uniform convergence, and learnability", Journal of the ACM **44**, 4, 615–631, URL `https://doi.org/10.1145/263867.263927` (1997).

Angluin, D., "Queries and concept learning", Machine Learning **2**, 4, 319–342, URL `https://doi.org/10.1023/a:1022821128753` (1988).

Anthony, M., "Classification by polynomial surfaces", Discrete Applied Mathematics **61**, 2, 91–103 (1995).

Arthur, M. W., J. S. Briney, J. D. Hawkins, R. D. Abbott, B. L. Brooke-Weiss and R. F. Catalano, "Measuring risk and protection in communities using the Communities That Care Youth Survey", Evaluation and Program Planning **30**, 2, 197–211, URL `https://doi.org/10.1016/j.evalprogplan.2007.01.009` (2007).

Arthur, M. W., J. D. Hawkins, J. A. Pollard, R. F. Catalano and A. J. Baglioni, "Measuring risk and protective factors for use, delinquency, and other adolescent problem behaviors: The Communities That Care Youth Survey", Evaluation Review **26**, 6, 575–601, URL `https://doi.org/10.1177/0193841x0202600601` (2002).

Astudillo, R. and P. Frazier, "Bayesian optimization of composite functions", in "International Conference on Machine Learning", pp. 354–363 (PMLR, 2019).

Atkinson, A., A. Donev and R. Tobias, *Optimum Experimental Designs, with SAS*, vol. 34 (OUP Oxford, 2007).

Baevski, A., Y. Zhou, A. Mohamed and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations", Advances in Neural Information Processing Systems **33**, 12449–12460 (2020).

Bai, Z. and X.-L. Zhang, "Speaker recognition based on deep learning: An overview", Neural Networks **140**, 65–99 (2021).

Bartlett, P., "For valid generalization the size of the weights is more important than the size of the network", Advances in Neural Information Processing Systems **9** (1996).

Bartlett, P. L., S. Boucheron and G. Lugosi, "Model selection and error estimation", Machine Learning **48**, 1, 85–113 (2002).

Bartlett, P. L., O. Bousquet and S. Mendelson, "Local Rademacher complexities", The Annals of Statistics **33**, 4, 1497–1537 (2005).

Bartlett, P. L., N. Harvey, C. Liaw and A. Mehrabian, "Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks", The Journal of Machine Learning Research **20**, 1, 2285–2301 (2019).

Bartlett, P. L., P. M. Long and R. C. Williamson, "Fat-shattering and the learnability of real-valued functions", in "Proceedings of the Seventh Annual Conference on Computational Learning Theory", pp. 299–310 (1994).

Bartlett, P. L. and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results", Journal of Machine Learning Research **3**, Nov, 463–482 (2002).

Bashir, A., C. M. Carvalho, P. R. Hahn and M. B. Jones, "Post-processing posteriors over precision matrices to produce sparse graph estimates", Bayesian Analysis **14**, 4, 1075–1090 (2019).

Basseville, M., "Divergence measures for statistical data processing—an annotated bibliography", Signal Processing **93**, 4, 621–633, URL `https://www.sciencedirect.com/science/article/pii/S0165168412003222` (2013).

Becker, J. T., F. Boiler, O. L. Lopez, J. Saxton and K. L. McGonigle, "The natural history of Alzheimer's disease: Description of study cohort and accuracy of diagnosis", Archives of Neurology **51**, 6, 585–594 (1994).

Bellman, R., "The theory of dynamic programming", Bulletin of the American Mathematical Society **60**, 6, 503–515 (1954).

Bellman, R., *Adaptive Control Processes: A Guided Tour* (Princeton University Press, London, 1961).

Beltrami, D., L. Calzà, G. Gagliardi, E. Ghidoni, N. Marcello, R. R. Favretti and F. Tamburini, "Automatic identification of mild cognitive impairment through the analysis of Italian spontaneous speech productions", in "Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)", pp. 2086–2093 (2016).

Berisha, V., C. Krantsevich, P. R. Hahn, S. Hahn, G. Dasarathy, P. Turaga and J. Liss, "Digital medicine and the curse of dimensionality", npj Digital Medicine **4**, 1, 153 (2021).

Berisha, V., A. Wisler, A. O. Hero and A. Spanias, "Empirically estimable classification bounds based on a nonparametric divergence measure", IEEE Transactions on Signal Processing **64**, 3, 580–591, URL `https://doi.org/10.1109/tsp.2015.2477805` (2016).

Berk-Seligson, S., D. Orcés, G. Pizzolitto, M. A. Seligson and C. Wilson, "Impact evaluation: Honduras country report", Tech. rep., The Latin American Public Opinion Project (LAPOP), Vanderbilt University, Nashville, TN (2014).

Bhattacharyya, A., "On a measure of divergence between two multinomial populations", Sankhyā: the Indian Journal of Statistics pp. 401–406 (1946).

Bielamowicz, S., J. Kreiman, B. R. Gerratt, M. S. Dauer and G. S. Berke, "Comparison of voice analysis systems for perturbation measurement", Journal of Speech, Language, and Hearing Research **39**, 1, 126–134 (1996).

Blumer, A., A. Ehrenfeucht, D. Haussler and M. K. Warmuth, "Learnability and the Vapnik-Chervonenkis dimension", Journal of the ACM **36**, 4, 929–965, URL `https://doi.org/10.1145/76359.76371` (1989).

Bock, R. D., "A brief history of item response theory", Educational Measurement: Issues and Practice **16**, 4, 21–33, URL `https://doi.org/10.1111/j.1745-3992.1997.tb00605.x` (1997).

Bock, R. D. and M. Aitkin, "Marginal maximum likelihood estimation of item parameters: Application of an EM algorithm", Psychometrika **46**, 4, 443–459, URL `https://doi.org/10.1007/bf02293801` (1981).

Bock, R. D., R. Gibbons and E. Muraki, "Full-information item factor analysis", Applied Psychological Measurement **12**, 3, 261–280, URL `https://doi.org/10.1177/014662168801200305` (1988).

Bose, A., M. Dutta, N. S. Dash, R. Nandi, A. Dutt and S. Ahmed, "Importance of task selection for connected speech analysis in patients with Alzheimer's disease from an ethnically diverse sample", Journal of Alzheimer's Disease , Preprint, 1–7 (2022).

Bousquet, O., S. Boucheron and G. Lugosi, *Introduction to Statistical Learning Theory*, pp. 169–207 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2004), URL `https://doi.org/10.1007/978-3-540-28650-9_8`.

Bousquet, O. and A. Elisseeff, "Stability and generalization", The Journal of Machine Learning Research **2**, 499–526 (2002).

Box, G. E. and H. L. Lucas, "Design of experiments in non-linear situations", Biometrika **46**, 1/2, 77–90 (1959).

Breiman, L., "Random forests", Machine Learning **45**, 1, 5–32, URL https://doi.org/10.1023/a:1010933404324 (2001).

Breiman, L., J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees*, The Wadsworth and Brooks-Cole Statistics-Probability Series (Taylor & Francis, Boca Raton, FL, 1984).

Bucks, R. S., S. Singh, J. M. Cuerden and G. K. Wilcock, "Analysis of spontaneous, conversational speech in dementia of Alzheimer type: Evaluation of an objective technique for analysing lexical performance", Aphasiology **14**, 1, 71–91 (2000).

Cai, L., "High-dimensional Exploratory Item Factor Analysis by a Metropolis-Hastings Robbins-Monro algorithm", Psychometrika **75**, 1, 33–57, URL https://doi.org/10.1007/s11336-009-9136-x (2010a).

Cai, L., "Metropolis-Hastings Robbins-Monro algorithm for confirmatory item factor analysis", Journal of Educational and Behavioral Statistics **35**, 3, 307–335, URL https://doi.org/10.3102/1076998609353115 (2010b).

Cano, S. J., H. B. Posner, M. L. Moline, S. W. Hurt, J. Swartz, T. Hsu and J. C. Hobart, "The ADAS-Cog in Alzheimer's disease clinical trials: Psychometric evaluation of the sum and its parts", Journal of Neurology, Neurosurgery & Psychiatry **81**, 12, 1363–1368 (2010).

Carlsson, G., "Topology and data", Bulletin of the American Mathematical Society **46**, 2, 255–308 (2009).

Carvalho, C. M., *Structure and Sparsity in High-Dimensional Multivariate Analysis*, Ph.D. thesis, Institute of Statistics and Decision Sciences, Duke University (2006).

Castanedo, F., "A review of data fusion techniques", The Scientific World Journal **2013** (2013).

Cervantes, J., F. Garcia-Lamont, L. Rodríguez-Mazahua and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends", Neurocomputing **408**, 189–215, URL https://doi.org/10.1016/j.neucom.2019.10.118 (2020).

Cha, S.-H., "Comprehensive survey on distance/similarity measures between probability density functions", City **1**, 2, 1 (2007).

Chalmers, R. P., "Generating adaptive and non-adaptive test interfaces for multidimensional item response theory applications", Journal of Statistical Software **71**, 5, 1–39 (2016).

Chang, H.-H., "Understanding computerized adaptive testing: From Robbins-Monro to Lord and beyond", in "The SAGE Handbook of Quantitative Methodology for the Social Sciences", pp. 118–135 (SAGE Publications, Inc., Thousand Oaks, CA, 2004), URL https://doi.org/10.4135/9781412986311.n7.

Chang, H.-H., "Psychometrics behind computerized adaptive testing", Psychometrika **80**, 1, 1–20, URL https://doi.org/10.1007/s11336-014-9401-5 (2015).

Chawla, N. V., K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique", Journal of Artificial Intelligence Research **16**, 321–357, URL https://doi.org/10.1613/jair.953 (2002).

Chernoff, H., "Sequential design of experiments", The Annals of Mathematical Statistics **30**, 3, 755–770 (1959).

Chipman, H. A., E. I. George and R. E. McCulloch, "BART: Bayesian additive regression trees", The Annals of Applied Statistics **4**, 1, 266–298, URL https://doi.org/10.1214/09-aoas285 (2010).

Chiu, C.-C., T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models", in "2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)", pp. 4774–4778 (IEEE, 2018).

Chouldechova, A., "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments", Big Data **5**, 2 (2017).

Chouldechova, A., D. Benavides-Prado, O. Fialko and R. Vaithianathan, "A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions", in "Proceedings of the 1st Conference on Fairness, Accountability and Transparency", vol. 81, pp. 134–148 (2018).

Chouldechova, A. and K. Lum, "The present and future risk of AI in pre-trial risk assessments", Tech. rep., Safety & Justice Challenge (2020).

Clarke, N., T. R. Barrick and P. Garrard, "A comparison of connected speech tasks for detecting early Alzheimer's disease and mild cognitive impairment using natural language processing and machine learning", Frontiers in Computer Science **3**, 634360 (2021).

Cortes, C. and V. Vapnik, "Support-vector networks", Machine Learning **20**, 3, 273–297, URL https://doi.org/10.1007/bf00994018 (1995).

Cox, D., "Some systematic experimental designs", Biometrika **38**, 3/4, 312–323 (1951).

Crocker, M. J., *Handbook of Acoustics* (John Wiley & Sons, 1998).

de Ayala, R. J., *The Theory and Practice of Item Response Theory.*, Methodology in the Social Sciences (The Guilford Press., 2009).

de la Fuente Garcia, S., C. W. Ritchie and S. Luz, "Artificial intelligence, speech, and language processing approaches to monitoring Alzheimer's disease: A systematic review", Journal of Alzheimer's Disease **78**, 4, 1547–1574 (2020).

Delgado-Gomez, D., E. Baca-Garcia, D. Aguado, P. Courtet and J. Lopez-Castroman, "Computerized adaptive test vs. decision trees: Development of a support decision system to identify suicidal behavior", Journal of Affective Disorders **206**, 204–209, URL https://doi.org/10.1016/j.jad.2016.07.032 (2016).

Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding", arXiv preprint arXiv:1810.04805 (2018).

Devroye, L., L. Györfi and G. Lugosi, *A Probabilistic Theory of Pattern Recognition* (Springer New York, 1996), URL https://doi.org/10.1007/978-1-4612-0711-5.

Devroye, L. and T. Wagner, "Distribution-free inequalities for the deleted and holdout error estimates", IEEE Transactions on Information Theory **25**, 2, 202–207 (1979).

Dirven, L., , M. Groenvold, M. J. B. Taphoorn, T. Conroy, K. A. Tomaszewski, T. Young and M. A. Petersen, "Psychometric evaluation of an item bank for computerized adaptive testing of the EORTC QLQ-c30 cognitive functioning dimension in cancer patients", Quality of Life Research **26**, 11, 2919–2929, URL https://doi.org/10.1007/s11136-017-1648-8 (2017).

Drucker, H., C. Cortes, L. D. Jackel, Y. LeCun and V. Vapnik, "Boosting and other ensemble methods", Neural Computation **6**, 6, 1289–1301 (1994).

Duchi, J., "Derivations for linear algebra and optimization", Berkeley, California **3**, 1, 2325–5870 (2007).

Durakovic, B., "Design of experiments application, concepts, examples: State of the art", Periodicals of Engineering and Natural Sciences **5**, 3 (2017).

Eckhouse, L., K. Lum, C. Conti-Cook and J. Ciccolini, "Layers of bias: A unified approach for understanding problems with risk assessment", Criminal Justice and Behavior **46**, 2, 185–209, URL https://doi.org/10.1177/0093854818811379 (2018).

Embretson, S. E. and S. P. Reise, *Item Response Theory for Psychologists*, Multivariate Applications Books Series (Lawrence Erlbaum Associates, Inc., Mahwah, NJ, US, 2000).

Eyben, F., M. Wöllmer and B. Schuller, "openSMILE: the Munich versatile and fast open-source audio feature extractor", in "Proceedings of the 18th ACM international conference on Multimedia", pp. 1459–1462 (2010).

Fagherazzi, G., A. Fischer, M. Ismael and V. Despotovic, "Voice for health: The use of vocal biomarkers from research to clinical practice", Digital Biomarkers **5**, 1, 78–88, URL https://doi.org/10.1159/000515346 (2021).

Fisher, R. and J. Wishart, "The arrangement of field experiments and the statistical reduction of the results.", Imperial Bureau of Soil Science, Technical Communication **10**, 23 (1930).

Fisher, R. A., "The arrangement of field experiments", Journal of the Ministry of Agriculture **33**, 503–515 (1926).

Fisher, R. A., *The Design of Experiments* (Oliver and Boyd, Edinburgh, 1935).

Fisher, R. A. and W. A. Mackenzie, "Studies in crop variation. II. The manurial response of different potato varieties", The Journal of Agricultural Science **13**, 3, 311–320 (1923).

Folstein, M. F., S. E. Folstein and P. R. McHugh, "'Mini-mental state': a practical method for grading the cognitive state of patients for the clinician", Journal of Psychiatric Research **12**, 3, 189–198 (1975).

Frey, A. and N.-N. Seitz, "Multidimensional adaptive testing in educational and psychological measurement: Current state and future challenges", Studies in Educational Evaluation **35**, 2–3, 89–94, URL `https://doi.org/10.1016/j.stueduc.2009.10.007` (2009).

Gabry, J., D. Simpson, A. Vehtari, M. Betancourt and A. Gelman, "Visualization in Bayesian workflow", Journal of the Royal Statistical Society: Series A **182**, 2, 389–402 (2019).

Garcia, L. and A. Lorena, *ECoL: Complexity Measures for Supervised Problems*, R package version 0.3.0 (2019).

Gelman, A., X.-L. Meng and H. Stern, "Posteripr predictive assessment of model fitness via realized discrepancies", Statistica Sinica **6**, 733–807 (1996).

Geraudie, A., P. Battista, A. M. García, I. E. Allen, Z. A. Miller, M. L. Gorno-Tempini and M. Montembeault, "Speech and language impairments in behavioral variant frontotemporal dementia: A systematic review", Neuroscience & Biobehavioral Reviews **131**, 1076–1095 (2021).

Gibbons, R. D., R. D. Bock, D. Hedeker, D. J. Weiss, E. Segawa, D. K. Bhaumik, D. J. Kupfer, E. Frank, V. J. Grochocinski and A. Stover, "Full-information item bifactor analysis of graded response data", Applied Psychological Measurement **31**, 1, 4–19, URL `https://doi.org/10.1177/0146621606289485` (2007).

Gibbons, R. D. and D. R. Hedeker, "Full-information item bi-factor analysis", Psychometrika **57**, 3, 423–436, URL `https://doi.org/10.1007/bf02295430` (1992).

Gibbons, R. D., G. Hooker, M. D. Finkelman, D. J. Weiss, P. A. Pilkonis, E. Frank, T. Moore and D. J. Kupfer, "The Computerized Adaptive Diagnostic test for Major Depressive Disorder (CAD-MDD)", The Journal of Clinical Psychiatry **74**, 07, 669–674, URL `https://doi.org/10.4088/jcp.12m08338` (2013).

Gibbons, R. D. and J. Wang, Personal communication (2019).

Gibbons, R. D., D. J. Weiss, E. Frank and D. Kupfer, "Computerized adaptive diagnosis and testing of mental health disorders", Annual Review of Clinical Psychology **12**, 1, 83–104, URL `https://doi.org/10.1146/annurev-clinpsy-021815-093634` (2016).

Goodglass, H. and E. Kaplan, *The Assessment of Aphasia and Related Disorders* (Lea & Febiger, 1972).

Gramacy, R. B., "tgp: an R package for Bayesian nonstationary, semiparametric nonlinear regression and design by treed Gaussian process models", Journal of Statistical Software **19**, 1–46 (2007).

Gramacy, R. B. and H. K. H. Lee, "Bayesian treed Gaussian process models with an application to computer modeling", Journal of the American Statistical Association **103**, 483, 1119–1130 (2008).

Guss, W. H. and R. Salakhutdinov, "On characterizing the capacity of neural networks using algebraic topology", arXiv preprint arXiv:1802.04443 (2018).

Hahn, P. R. and C. M. Carvalho, "Decoupling shrinkage and selection in Bayesian linear models: A posterior summary perspective", Journal of the American Statistical Association **110**, 509, 435–448, URL `https://doi.org/10.1080/01621459.2014.993077` (2015).

Haley, S. M., P. Ni, L. H. Ludlow and M. A. Fragala-Pinkham, "Measurement precision and efficiency of multidimensional computer adaptive testing of physical functioning using the Pediatric Evaluation of Disability Inventory", Archives of Physical Medicine and Rehabilitation **87**, 9, 1223–1229, URL `https://doi.org/10.1016/j.apmr.2006.05.018` (2006).

Hambleton, R. K., H. Swaminathan and H. J. Rogers, *Fundamentals of Item Response Theory* (Sage Publications, Inc., 1991).

Hare, T., J. C. Guzman and L. Miller-Graff, "Identifying high-risk young adults for violence prevention: A validation of psychometric and social scales in Honduras", Journal of Crime and Justice **41**, 5, 627–642, URL `https://doi.org/10.1080/0735648x.2018.1446184` (2018).

Hastie, T., R. Tibshirani, J. H. Friedman and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2 (Springer, 2009).

Haussler, D., "Probably approximately correct learning", in "AAAI-90", pp. 1101–1108 (1990).

Hawkins, J. D., R. F. Catalano and J. Y. Miller, "Risk and protective factors for alcohol and other drug problems in adolescence and early adulthood: Implications for substance abuse prevention", Psychological Bulletin **112**, 1, 64–105, URL `https://doi.org/10.1037/0033-2909.112.1.64` (1992).

He, J., S. Yalov and P. R. Hahn, "Xbart: Accelerated Bayesian additive regression trees", arXiv preprint arXiv:1810.02215 (2018).

Hellinger, E., "Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen.", Journal Für die Reine und Angewandte Mathematik **1909**, 136, 210–271 (1909).

Hennigan, K. M., K. A. Kolnick, F. Vindel and C. L. Maxson, "Targeting youth at risk for gang involvement: Validation of a gang risk assessment to support individualized secondary prevention", Children and Youth Services Review **56**, 86–96, URL https://doi.org/10.1016/j.childyouth.2015.07.002 (2015).

Hennigan, K. M., C. L. Maxson, D. C. Sloane, K. A. Kolnick and F. Vindel, "Identifying high-risk youth for secondary gang prevention", Journal of Crime and Justice **37**, 1, 104–128, URL https://doi.org/10.1080/0735648x.2013.831208 (2014).

Hicks, C. R., *Fundamental Concepts in the Design of Experiments* (Holt, Rinehart and Winston, New York, 1964).

Higginson, A., K. Benier, Y. Shenderovich, L. Bedford, L. Mazerolle and J. Murray, "Factors associated with youth gang membership in low- and middle-income countries: A systematic review", Campbell Systematic Reviews **14**, 1, 1–128 (2018).

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

Ho, T. K. and M. Basu, "Complexity measures of supervised classification problems", IEEE Transactions on Pattern Analysis and Machine Intelligence **24**, 3, 289–300 (2002).

Hotelling, H., "Some improvements in weighing and other experimental techniques", The Annals of Mathematical Statistics **15**, 3, 297–306 (1944).

Howell, J. C. and A. E. Jr., "Moving risk factors into developmental theories of gang membership", Youth Violence and Juvenile Justice **3**, 4, 334–354 (2005).

Huff, F. J., S. Corkin and J. H. Growdon, "Semantic impairment and anomia in Alzheimer's disease", Brain and Language **28**, 2, 235–249 (1986).

Jeffreys, H., "An invariant form for the prior probability in estimation problems", Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences **186**, 1007, 453–461 (1946).

Johndrow, J. E. and K. Lum, "An algorithm for removing sensitive information: Application to race-independent recidivism prediction", The Annals of Applied Statistics **13**, 1, URL https://doi.org/10.1214/18-aoas1201 (2019).

Jolliffe, I. T., "A note on the use of principal components in regression", Journal of the Royal Statistical Society Series C: Applied Statistics **31**, 3, 300–303 (1982).

Kahrs, M. and K. Brandenburg, *Applications of Digital Signal Processing to Audio and Acoustics* (Springer Science & Business Media, 1998).

Karpen, S. C., "P value problems", American Journal of Pharmaceutical Education **81**, 9, 6570, URL https://doi.org/10.5688/ajpe6570 (2017).

Katz, C. M., H. Cheon, E. C. Hedberg and S. H. Decker, "Impact of family-based secondary prevention programming on risk, resilience, and delinquency: A 6-month follow up within a randomized control trial in Honduras", Justice Quarterly pp. 1–26, URL https://doi.org/10.1080/07418825.2021.1967425 (2021).

Katz, C. M. and A. M. Fox, "Risk and protective factors associated with gang-involved youth in Trinidad and Tobago", Revista Panamericana de Salud Pública **27**, 3, 187–202 (2010).

Kearns, M. and R. Schapire, "Efficient distribution-free learning of probabilistic concepts", in "Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science", (IEEE Comput. Soc. Press, 1990), URL https://doi.org/10.1109/fscs.1990.89557.

Kearns, M. J. and U. Vazirani, *An Introduction to Computational Learning Theory* (MIT press, 1994).

Koltchinskii, V., "Rademacher penalties and structural risk minimization", IEEE Transactions on Information Theory **47**, 5, 1902–1914 (2001).

Komeili, M., C. Pou-Prom, D. Liaqat, K. C. Fraser, M. Yancheva and F. Rudzicz, "Talk2Me: Automated linguistic data collection for personal assessment", PloS One **14**, 3, e0212342 (2019).

Krantsevich, C., P. R. Hahn, Y. Zheng and C. Katz, "Bayesian decision theory for tree-based adaptive screening tests with an application to youth delinquency", The Annals of Applied Statistics **17**, 2, 1038–63 (2023).

Krohn, M. D. and T. P. Thornberry, *The Long View of Crime: A Synthesis of Longitudinal Research*, chap. Longitudinal perspectives on adolescent street gangs, pp. 128–160 (Springer, New York, 2008).

Kullback, S. and R. A. Leibler, "On information and sufficiency", The Annals of Mathematical Statistics **22**, 1, 79–86 (1951).

Lansing, A. E., R. J. Ivnik, C. M. Cullum and C. Randolph, "An empirically derived short form of the Boston naming test", Archives of Clinical Neuropsychology **14**, 6, 481–487 (1999).

Li, W., G. Dasarathy, K. Natesan Ramamurthy and V. Berisha, "Finding the homology of decision boundaries with active learning", Advances in Neural Information Processing Systems **33**, 8355–8365 (2020).

Loh, W.-Y., "Classification and regression trees", WIREs Data Mining and Knowledge Discovery **1**, 1, 14–23, URL https://doi.org/10.1002/widm.8 (2011).

López-Ruiz, R., H. L. Mancini and X. Calbet, "A statistical measure of complexity", Physics Letters A **209**, 5-6, 321–326 (1995).

Lord, F. M., *Applications of Item Response Theory to Practical Testing Problems.* (Lawrence Erlbaum Associates, Inc., 365 Broadway, Hillsdale, NJ, 1980), iSBN: 0-89859-006-X.

Lord, F. M., M. R. Novick and A. Birnbaum, *Statistical Theories of Mental Test Scores* (Information Age Publishing, 1968).

Lorena, A. C., L. P. Garcia, J. Lehmann, M. C. Souto and T. K. Ho, "How complex is your classification problem? A survey on measuring classification complexity", ACM Computing Surveys (CSUR) **52**, 5, 1–34 (2019).

Luengo, J. and F. Herrera, "Domains of competence of fuzzy rule based classification systems with data complexity measures: A case of study using a fuzzy hybrid genetic based machine learning method", Fuzzy Sets and Systems **161**, 1, 3–19 (2010).

Luengo, J. and F. Herrera, "An automatic extraction method of the domains of competence for learning classifiers using data complexity measures", Knowledge and Information Systems **42**, 147–180 (2015).

Luz, H.-F. F. S. d. l. F. D., S. and B. MacWhinney, "Alzheimer's dementia recognition through spontaneous speech: The ADReSS challenge", in "Proceedings of Interspeech", pp. 2172–2176 (2020).

Maffei, M. F., J. R. Green, O. Murton, Y. Yunusova, H. P. Rowe, F. Wehbe, K. Diana, K. Nicholson, J. D. Berry and K. P. Connaghan, "Acoustic measures of dysphonia in amyotrophic lateral sclerosis", Journal of Speech, Language, and Hearing Research pp. 1–16, URL `https://doi.org/10.1044/2022_jslhr-22-00363` (2023).

Magis, D. and J. R. Barrada, "Computerized adaptive testing with R: Recent updates of the package catR", Journal of Statistical Software **76**, Code Snippet 1, URL `https://doi.org/10.18637/jss.v076.c01` (2017).

Magis, D. and G. Raîche, "Random generation of response patterns under computerized adaptive testing with the R package catR", Journal of Statistical Software **48**, 8, URL `https://doi.org/10.18637/jss.v048.i08` (2012).

Maguire, E. R., W. Wells and C. M. Katz, "Measuring community risk and protective factors for adolescent problem behaviors: Evidence from a developing nation", Journal of Research in Crime and Delinquency **48**, 594–620 (2011).

Martínez-Ferreiro, S., "Naming as a window to word retrieval changes in healthy and pathological ageing: Methodological considerations", International Journal of Language & Communication Disorders (2022).

Martínez-Nicolás, I., T. E. Llorente, F. Martínez-Sánchez and J. J. G. Meilán, "Ten years of research on automatic voice and speech analysis of people with Alzheimer's disease and mild cognitive impairment: A systematic review article", Frontiers in Psychology **12**, 620251 (2021).

Meilán, J. J., F. Martínez-Sánchez, I. Martínez-Nicolás, T. E. Llorente and J. Carro, "Changes in the rhythm of speech difference between people with nondegenerative mild cognitive impairment and with preclinical dementia", Behavioural Neurology **2020** (2020).

Mendelson, S., "Rademacher averages and phase transitions in Glivenko-Cantelli classes", IEEE Transactions on Information Theory **48**, 1, 251–263 (2002).

Mercier, M., M. S. Santos, P. H. Abreu, C. Soares, J. P. Soares and J. Santos, "Analysing the footprint of classifiers in overlapped and imbalanced contexts", in "Advances in Intelligent Data Analysis XVII: 17th International Symposium, IDA 2018,'s-Hertogenbosch, The Netherlands, October 24–26, 2018, Proceedings", pp. 200–212 (Springer, 2018).

Mermelstein, P., "Distance measures for speech recognition, psychological and instrumental", Pattern Recognition and Artificial Intelligence **116**, 374–388 (1976).

Meyer, P. J., "U.S. strategy for engagement in Central America: Policy issues for Congress", CRS Report R44812, Congressional Research Service, retrieved from https://crsreports.congress.gov/product/pdf/R/R44812 (2019).

Michel, P., K. Baumstarck, A. Loundou, B. Ghattas, P. Auquier and L. Boyer, "Computerized adaptive testing with decision regression trees: An alternative to item response theory for quality of life measurement in multiple sclerosis", Patient Preference and Adherence **Volume 12**, 1043–1053, URL `https://doi.org/10.2147/ppa.s162206` (2018).

Milborrow, S., *rpart.plot: Plot 'rpart' Models: An Enhanced Version of 'plot.rpart'.*, R package version 3.1.0. https://CRAN.R-project.org/package=rpart.plot (2021).

Mueller, K. D., B. Hermann, J. Mecollari and L. S. Turkstra, "Connected speech and language in mild cognitive impairment and Alzheimer's disease: A review of picture description tasks", Journal of Clinical and Experimental Neuropsychology **40**, 9, 917–939, URL `https://doi.org/10.1080/13803395.2018.1446513` (2018).

Mukherjee, S., P. Niyogi, T. Poggio and R. Rifkin, "Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization", Advances in Computational Mathematics **25**, 1, 161–193 (2006).

Muraki, E. and J. E. Carlson, "Full-information factor analysis for polytomous item responses", Applied Psychological Measurement **19**, 1, 73–90, URL `https://doi.org/10.1177/014662169501900109` (1995).

Murray, J., Y. Shenderovich, F. Gardner, C. Mikton, J. H. Derzon, J. Liu and M. Eisner, "Risk factors for antisocial behavior in low- and middle-income countries: A systematic review of longitudinal studies", Crime and Justice **47**, 1, 255–364, URL `https://doi.org/10.1086/696590` (2018).

Murray, J. S., "Log-linear Bayesian additive regression trees for multinomial logistic and count regression models", Journal of the American Statistical Association pp. 1–35, URL `https://doi.org/10.1080/01621459.2020.1813587` (2020).

Murray, J. S., D. B. Dunson, L. Carin and J. E. Lucas, "Bayesian Gaussian copula factor models for mixed data", Journal of the American Statistical Association **108**, 502, 656–665, URL https://doi.org/10.1080/01621459.2012.762328 (2013).

Nasreddine, Z. S., N. A. Phillips, V. Bédirian, S. Charbonneau, V. Whitehead, I. Collin, J. L. Cummings and H. Chertkow, "The Montreal Cognitive Assessment, MoCA: A brief screening tool for mild cognitive impairment", Journal of the American Geriatrics Society **53**, 4, 695–699 (2005).

OpenAI, "GPT-4 technical report", URL https://arxiv.org/abs/2303.08774 (2023).

Paap, M. C. S., K. A. Kroeze, C. A. W. Glas, C. B. Terwee, J. van der Palen and B. P. Veldkamp, "Measuring patient-reported outcomes adaptively: Multidimensionality matters!", Applied Psychological Measurement **42**, 5, 327–342, URL https://doi.org/10.1177/0146621617733954 (2017).

Papakostas, M., E. Spyrou, T. Giannakopoulos, G. Siantikos, D. Sgouropoulos, P. Mylonas and F. Makedon, "Deep visual attributes vs. hand-crafted audio features on multidomain speech emotion recognition", Computation **5**, 2, 26 (2017).

Park, T. and G. Casella, "The Bayesian lasso", Journal of the American Statistical Association **103**, 482, 681–686 (2008).

Parmigiani, G. and L. Y. T. Inoue, *Decision Theory: Principles and Approaches* (Wiley Blackwell, 2010).

Pearson, K., "LIII. On lines and planes of closest fit to systems of points in space", The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**, 11, 559–572 (1901).

Pierce, C. S., "Note on the theory of the economy of research", in "United States Coast Survey for the fiscal year ending June 1876", (US Government Printing Office, 1879).

Poggio, T., R. Rifkin, S. Mukherjee and P. Niyogi, "General conditions for predictivity in learning theory", Nature **428**, 6981, 419–422, URL https://doi.org/10.1038/nature02341 (2004).

Puelz, D., P. R. Hahn and C. M. Carvalho, "Variable selection in seemingly unrelated regressions with random predictors", Bayesian Analysis **12**, 4, 969–989 (2017).

Ramamurthy, K. N., K. Varshney and K. Mody, "Topological data analysis of decision boundaries with application to model selection", in "International Conference on Machine Learning", pp. 5351–5360 (PMLR, 2019).

Rieck, B., M. Togninalli, C. Bock, M. Moor, M. Horn, T. Gumbsch and K. Borgwardt, "Neural persistence: A complexity measure for deep neural networks using algebraic topology", arXiv preprint arXiv:1812.09764 (2018).

Rodríguez-Ferreiro, J., C. Martínez, A.-J. Pérez-Carbajal and F. Cuetos, "Neural correlates of spelling difficulties in Alzheimer's disease", Neuropsychologia **65**, 12–17 (2014).

Rogers, W. H. and T. J. Wagner, "A finite sample distribution-free performance bound for local discrimination rules", The Annals of Statistics pp. 506–514 (1978).

Rudner, L. M., "Demystifying the GMAT: Computer adaptive testing", Graduate Management Admission Council: Deans Digest (2010).

Rusz, J., J. Hlavnička and M. N. et al, "Speech biomarkers in rapid eye movement sleep behavior disorder and Parkinson disease", Annals of Neurology **90**, 1, 62–75, URL `https://doi.org/10.1002/ana.26085` (2021).

Sáez, J. A., J. Luengo and F. Herrera, "Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification", Pattern Recognition **46**, 1, 355–364 (2013).

Sajjadi, S. A., K. Patterson, M. Tomek and P. J. Nestor, "Abnormalities of connected speech in semantic dementia vs Alzheimer's disease", Aphasiology **26**, 6, 847–866 (2012).

Sands, W. A., B. K. Waters and J. R. McBride, *Computerized Adaptive Testing: From Inquiry to Operation* (American Psychological Association, 1997), URL `https://doi.org/10.1037/10244-000`.

Santos, M. S., P. H. Abreu, N. Japkowicz, A. Fernández and J. Santos, "A unifying view of class overlap and imbalance: Key concepts, multi-view panorama, and open avenues for research", Information Fusion **89**, 228–253, URL `https://doi.org/10.1016/j.inffus.2022.08.017` (2023).

Santos, M. S., P. H. Abreu, N. Japkowicz, A. Fernández, C. Soares, S. Wilk and J. Santos, "On the joint-effect of class imbalance and overlap: A critical review", Artificial Intelligence Review pp. 1–69 (2022).

Seçkin, M. and M. Savaş, "Picnic, accident or cookies? A systematic approach to guide the selection of the picture definition tasks in linguistic assessment", Archives of Clinical Neuropsychology (2023).

Smith, K., "On the standard deviations of adjusted and interpolated values of an observed polynomial function and its constants and the guidance they give towards a proper choice of the distribution of observations", Biometrika **12**, 1/2, 1–85 (1918).

Smith, M. R., T. Martinez and C. Giraud-Carrier, "An instance level analysis of data complexity", Machine Learning **95**, 2, 225–256, URL `https://doi.org/10.1007/s10994-013-5422-z` (2014).

Sontag, E. D. *et al.*, "VC dimension of neural networks", NATO ASI Series F Computer and Systems Sciences **168**, 69–96 (1998).

Taguchi, G., *Studies on Mathematical Statistics for Quality Control*, Ph.D. thesis, Kyushu University (1962).

Taguchi, G., "Quality engineering (Taguchi methods) for the development of electronic circuit technology", IEEE Transactions on Reliability **44**, 2, 225–229 (1995).

Taguchi, G. and M. S. Phadke, "Quality engineering through design optimization", Quality Control, Robust Design, and the Taguchi Method pp. 77–96 (1989).

Theodoridis, S. and K. Koutroumbas, *Pattern Recognition* (Elsevier, 2006).

Thompson, S. G. and J. P. Higgins, "How should meta-regression analyses be undertaken and interpreted?", Statistics in Medicine **21**, 11, 1559–1573 (2002).

Tombaugh, T. N. and N. J. McIntyre, "The mini-mental state examination: a comprehensive review", Journal of the American Geriatrics Society **40**, 9, 922–935 (1992).

UNODC, "UNODC statistics", Https://dataunodc.un.org/ (2018).

Valiant, L. G., "A theory of the learnable", Communications of the ACM **27**, 11, 1134–1142, URL `https://doi.org/10.1145/1968.1972` (1984).

van der Linden, W. J., "Some new developments in adaptive testing technology", Zeitschrift für Psychologie / Journal of Psychology **216**, 1, 3–11, URL `https://doi.org/10.1027/0044-3409.216.1.3` (2008).

van der Linden, W. J. and R. K. Hambleton, "Item response theory: Brief history, common models, and extensions", in "Handbook of Modern Item Response Theory", pp. 1–28 (Springer New York, 1997), URL `https://doi.org/10.1007/978-1-4757-2691-6_1`.

Vapnik, V., "An overview of statistical learning theory", IEEE Transactions on Neural Networks **10**, 5, 988–999, URL `https://doi.org/10.1109/72.788640` (1999).

Vapnik, V. N., *The Nature of Statistical Learning Theory* (Springer New York, 1995), URL `https://doi.org/10.1007/978-1-4757-2440-0`.

Vapnik, V. N. and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities", Theory of Probability and Applications **16**, 264–280, [in Russian] (1971).

Venneri, A., W. J. McGeown, H. M. Hietanen, C. Guerrini, A. W. Ellis and M. F. Shanks, "The anatomical bases of semantic retrieval deficits in early Alzheimer's disease", Neuropsychologia **46**, 2, 497–510 (2008).

W G Rosen, K. L. D., R C Mohs, "A new rating scale for Alzheimer's disease", American Journal of Psychiatry **141**, 11, 1356–1364, URL `https://doi.org/10.1176/ajp.141.11.1356` (1984).

Wainer, H., *Computerized Adaptive Testing: A Primer* (Lawrence Erlbaum Associates Publishers, Mahwah, NJ, 2000).

Wang, C. and H.-H. Chang, "Item selection in multidimensional computerized adaptive testing—gaining information from different angles", Psychometrika **76**, 3, 363–384, URL https://doi.org/10.1007/s11336-011-9215-7 (2011).

Wang, C., H.-H. Chang and K. A. Boughton, "Deriving stopping rules for multidimensional computerized adaptive testing", Applied Psychological Measurement **37**, 2, 99–122, URL https://doi.org/10.1177/0146621612463422 (2012).

Wang, M. and P. Hahn, "Accelerated Bayesian additive regression trees for fast multiclass classification", In Preparation (2021).

Webb, V. J., L. E. Nuño and C. Katz, "Influence of risk and protective factors on school-aged youth involvement with gangs, guns, and delinquency: Findings from the El Salvador Youth Survey", Tech. rep., Center for Violence Prevention and Community Safety, Arizona State University (2016).

Weerman, F. M., C. L. Maxson, F.-A. Esbensen, J. Aldridge, J. Medina and F. van Gemert, "Eurogang program manual", Tech. rep., University of Missouri at St Louis, St Louis, MO (2009).

Weitzner, D. S. and M. Calamia, "Serial position effects on list learning tasks in mild cognitive impairment and Alzheimer's disease", Neuropsychology **34**, 4, 467 (2020).

Woody, S., C. M. Carvalho and J. S. Murray, "Model interpretation through lower-dimensional posterior summarization", arXiv preprint arXiv:1905.07103 (2019).

Xu, H. and S. Mannor, "Robustness and generalization", Machine Learning **86**, 3, 391–423, URL https://doi.org/10.1007/s10994-011-5268-1 (2012).

Yao, L., M. Pommerich and D. O. Segall, "Using multidimensional CAT to administer a short, yet precise, screening test", Applied Psychological Measurement **38**, 8, 614–631, URL https://doi.org/10.1177/0146621614541514 (2014).

Yates, F., "The design and analysis of factorial experiments", (1937).

Yates, F., "The analysis of experiments containing different crop rotations", Biometrics pp. 324–346 (1954).

Zheng, A. and A. Casari, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists* (" O'Reilly Media, Inc.", 2018).

Zheng, Y., H. Cheon and C. M. Katz, "Using machine learning methods to develop a short tree-based adaptive classification test: Case study with a high dimensional item pool and imbalanced data", Applied Psychological Measurement (2020).

Zvonkin, A. K. and L. A. Levin, "The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms", Russian Mathematical Surveys **25**, 6, 83 (1970).

APPENDIX A

PERMISSION TO USE PREVIOUSLY PUBLISHED WORK

Each of the co-authors of Krantsevich *et al.* (2023), namely P. Richard Hahn, Yi Zheng, and Charles Katz, have given their written permissions for the use of material from Krantsevich *et al.* (2023) in this Dissertation.

APPENDIX B

CHANGING THE ACTION SPACE

In Chapter 2 we present one method for populating the action space with a single adaptive test of length m. First, we calibrate a regression tree T_m^* to synthetic data sampled from the posterior predictive distribution, using the maxIPP algorithm with maxIPP $= m$. Then we choose a threshold which is optimized relative to T_m^*. This is only one possible heuristic for obtaining a tree-based adaptive test with at most m questions. We can change the tree-growing algorithm, the data the algorithm is applied to, the way the threshold is chosen; or, we can calibrate a classification tree directly instead of using a two-stage regression tree + cutoff approach. Here we present results for several such alternatives.

First, we obtain different adaptive tests varying the parameters above, using the entire IMC data set. We compare two stopping criteria for growing the regression tree: 1) growing to a maximum depth; 2) growing very deep using maxIPP then pruning back (proposed method, described in section 2.2.1). We furthermore compare these tree-growing algorithms applied to synthetic data generated via two different processes:

(1) Item responses generated via local perturbations ("Perturb") and "at-risk" probabilities obtained using a Random Forest model ("RF"), as in Gibbons *et al.* (2013).

(2) Item responses sampled from a Gaussian copula factor model ("GCFM") and "at-risk" probabilities obtained using a logistic Accelerated Bayesian Additive Regression Trees ("XBART") model. This is our proposed method, described in Chapter 2, with data notated $\{\tilde{\mathrm{x}}_k, \bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\mathrm{x}}_k)\}$, $1 \leq k \leq M$.

For each of these two regression trees, we obtain an optimal cutoff using data $\{\tilde{\mathrm{x}}_k, \tilde{y}_k\}_{k=1}^M$.

We also consider the simpler approach of calibrating a classification tree (via maximum depth) that predicts "at-risk" status directly, rather than the regression tree + cutoff approach. We calibrate three classification trees using the following synthetic datasets:

(1) Item responses generated via local perturbations ("Perturb") and risk classes ("at-risk" = 1, "not-at-risk" = 0) obtained using RF. This data uses the same item responses and the same fitted model as above, but extracting class predictions rather than probability of being in the "at-risk" class.

(2) Synthetic item responses and classes $\{\tilde{\mathrm{x}}_k, \tilde{y}_k\}$, $1 \leq k \leq M$.

(3) Synthetic item responses and utility-based synthetic classes $\{\tilde{\mathrm{x}}_k, \gamma_k^*\}$, $1 \leq k \leq M$, described below.

A reviewer suggested classification tree method (3). Unlike the first two synthetic datasets, which do not incorporate the utility function at all, this method still approximately maximizes the utility function (subject to the number of items constraint), and allows for fine-tuning w based on desired levels of sensitivity and specificity.

The synthetic class outcomes γ_k^* are defined as:

$$\gamma_k^* = \gamma(\tilde{\mathrm{x}_k}) \quad = \begin{cases} 1 & \text{if } \bar{\mathbb{E}}(\tilde{Y} \mid \tilde{\mathrm{x}}_k) \geq \frac{U_0}{U_0+U_1} \\ 0 & \text{otherwise} \end{cases},$$

where U_0 and U_1 are defined in section 2.3.2. If one was not required to use a decision tree for classifying participants as "at-risk" or "not-at-risk", this class assignment would maximize the expected posterior utility point-wise. Thus, a classification tree calibrated to this third synthetic dataset will approximately optimize the expected utility function.

In summary, the settings compared for different adaptive tests are:

(1) Regression Tree + Cutoff Methods
   (a) Stopping criterion:
      - maxIPP & pruning (maxIPP)
      - maximum depth (maxDepth)
   (b) Data for calibrating (item responses + predicted probabilities):
      - local perturbations + Random Forest (Perturb + RF)
      - Gaussian copula factor model + accelerated Bayesian additive regression trees (GCFM + XBART)

(2) Classification Tree Method
   (a) Data for calibrating (item responses + predicted classes):
      - Perturb + RF
      - GCFM + XBART
      - GCFM + utility-based outcomes (GCFM + Utility)

In total this leaves four regression tree methods and three classification tree methods. For γ being each of the seven methods, we computed the expected utility draw $\mathbb{E}U_{\theta^{(j)}}(\gamma)$ over the j^{th} sample population $\{\tilde{\mathrm{x}}_{ij}, \tilde{y}_{ij}\}_{i=1}^N$ using Eqn (4). We similarly computed the j^{th} draw of expected utility of the non-shortened assessment $\gamma^*(\cdot) = \mathrm{Thr}_{C^*}(\bar{\mathbb{E}}(\tilde{Y} \mid \cdot))$, and the difference between the two expected utilities is $\Delta_{\theta^{(j)},m}$.

Figure B.1 shows the boxplots representing the distribution of $\Delta_{\theta,m}$ for each of these seven methods, for a utility function weight of $w = 0.5$. The maxIPP/maximum depth criteria are grouped as "Number of Items".

First, we emphasize that Figure B.1 is not intended to demonstrate the ultimate superiority of any method, but rather to assess which method best approximates our implementation of an optimal (in terms of expected utility) screening instrument that uses all items. We make a direct comparison of these methods on out-of-sample data shortly.

The most striking result in Figure B.1 is the superiority of the utility-based methods (all 4 regression-based methods, and the utility-based classification method) at reproducing the utility of the full-item instrument. This is unsurprising because the other two classification methods were not created with the utility function in mind; they were simply calibrated to synthetic data obtained from models fit to the IMC

Figure B.1: Utility difference plots when we change the way of populating the action space.

data. The IMC data is highly imbalanced, so these two classification trees tend to predict almost all youth to be "not-at-risk", and they have low expected utility compared to the non-sparse action.

The maxIPP and maximum depth stopping criteria produce similar results. We expect there to be a greater difference using maxIPP for an application in which multiple strata of the same question lead to to substantially different outcomes, whereas in this particular application, we feel that differences in item responses are mainly important in the "high-low" sense. However, the maxIPP process seems to do at least as well as maximum depth.

We also derived out of sample sensitivity and specificity for all 7 of these methods in order to do a fair comparison between them. The results are shown in the tables, at the end of this Appendix, one table per utility function (parameterized by w).

Sensitivity is quite poor for the non-utility based classification methods, presumably because the IMC data is highly imbalanced. As a consequence, synthetic data used from models fitted to this imbalanced data also have very few "at-risk" predictions, and so these classification trees in turn make very few "at-risk" predictions. We note that the method Classification – maxDepth – Perturb + RF was the method used for designing the tree-based adaptive tests in Gibbons *et al.* (2016), according to our personal communication with the authors (Gibbons and Wang (2019)). While their method seems to work well for problems such as screening for Major Depressive Disorder, in our application, which has extremely imbalanced class outcomes, it performs quite poorly.

These results highlight the benefit of optimizing the adaptive test to the utility function, either using a regression tree and a separately optimized cutoff, or using a classification tree calibrated to utility-based class labels. In both of these methods, we are able to make finer adjustments to the screening procedure and balance sensitivity and specificity to be in line with desired ranges for a particular group.

For all of the utility-based methods (all regression methods and the utility-based cutoff method), increasing w improves the sensitivity and decreases the specificity, as expected. The regression methods using GCFM + XBART and the utility based classification methods are two different ways of obtaining an adaptive test that approximates the same utility function using the same fitted models, the only difference being whether the utility is optimized point-wise or globally.

We see three advantages of our proposed two-step (global optimization) method:

(1) We believe the two-step formulation is more intuitive, because the

connection to sensitivity and specificity in the quantity used to optimize the threshold (eq. (2.4) in Chapter 2) is more direct.

(2) The two-step process allows one to visualize the impact of the weight w via ROC curves, as in Figure fig. 2.6. Since the predicted probabilities do not change, and the utility function is optimized via the threshold function Thr_{C_T}, one can plot a ROC curve of the predicted probabilities. This allows for directly see how changing w results in different thresholds and corresponding Sensitivity and Specificity.

(3) In terms of practical implementation, it is much more efficient to fit only one regression tree and then separately optimize the threshold for many values of w, compared to fitting a new classification tree for every new value of w one would like to examine. Since our MCMC sample contains 1 million synthetic data, this is actually a substantial computational speedup when many values of w are under consideration.

# Items	Tree Type	Criterion	w	Calibration Data	Sens	Spec
2	Classification	maxDepth	-	Perturb + RF	0.007	1.000
2	Classification	maxDepth	-	GCFM + XBART	0.000	1.000
3	Classification	maxDepth	-	Perturb + RF	0.007	1.000
3	Classification	maxDepth	-	GCFM + XBART	0.000	1.000
4	Classification	maxDepth	-	Perturb + RF	0.014	1.000
4	Classification	maxDepth	-	GCFM + XBART	0.000	1.000
5	Classification	maxDepth	-	Perturb + RF	0.014	0.999
5	Classification	maxDepth	-	GCFM + XBART	0.000	1.000
6	Classification	maxDepth	-	Perturb + RF	0.028	0.997
6	Classification	maxDepth	-	GCFM + XBART	0.000	1.000
7	Classification	maxDepth	-	Perturb + RF	0.021	0.997
7	Classification	maxDepth	-	GCFM + XBART	0.000	1.000
8	Classification	maxDepth	-	Perturb + RF	0.028	0.997
8	Classification	maxDepth	-	GCFM + XBART	0.000	0.999
9	Classification	maxDepth	-	Perturb + RF	0.021	0.997
9	Classification	maxDepth	-	GCFM + XBART	0.000	0.999
10	Classification	maxDepth	-	Perturb + RF	0.028	0.995
10	Classification	maxDepth	-	GCFM + XBART	0.042	0.994
11	Classification	maxDepth	-	Perturb + RF	0.021	0.994
11	Classification	maxDepth	-	GCFM + XBART	0.049	0.992
12	Classification	maxDepth	-	Perturb + RF	0.021	0.995
12	Classification	maxDepth	-	GCFM + XBART	0.062	0.980
13	Classification	maxDepth	-	Perturb + RF	0.028	0.994
13	Classification	maxDepth	-	GCFM + XBART	0.097	0.976
14	Classification	maxDepth	-	Perturb + RF	0.035	0.994
14	Classification	maxDepth	-	GCFM + XBART	0.132	0.967
15	Classification	maxDepth	-	Perturb + RF	0.035	0.992
15	Classification	maxDepth	-	GCFM + XBART	0.160	0.955

# Items	Tree Type	Criterion	w	Calibration Data	Sens	Spec
2	Regression + Cutoff	maxIPP	0.4	Perturb + RF	0.319	0.908
2	Regression + Cutoff	maxDepth	0.4	Perturb + RF	0.319	0.908
2	Regression + Cutoff	maxIPP	0.4	GCFM + XBART	0.340	0.857
2	Regression + Cutoff	maxDepth	0.4	GCFM + XBART	0.167	0.940
2	Classification	maxDepth	0.4	GCFM + Utility	0.167	0.940
3	Regression + Cutoff	maxIPP	0.4	Perturb + RF	0.306	0.907
3	Regression + Cutoff	maxDepth	0.4	Perturb + RF	0.389	0.885
3	Regression + Cutoff	maxIPP	0.4	GCFM + XBART	0.396	0.841
3	Regression + Cutoff	maxDepth	0.4	GCFM + XBART	0.312	0.891
3	Classification	maxDepth	0.4	GCFM + Utility	0.326	0.879
4	Regression + Cutoff	maxIPP	0.4	Perturb + RF	0.444	0.848
4	Regression + Cutoff	maxDepth	0.4	Perturb + RF	0.444	0.848
4	Regression + Cutoff	maxIPP	0.4	GCFM + XBART	0.424	0.841
4	Regression + Cutoff	maxDepth	0.4	GCFM + XBART	0.354	0.872
4	Classification	maxDepth	0.4	GCFM + Utility	0.354	0.860
5	Regression + Cutoff	maxIPP	0.4	Perturb + RF	0.438	0.866
5	Regression + Cutoff	maxDepth	0.4	Perturb + RF	0.382	0.888
5	Regression + Cutoff	maxIPP	0.4	GCFM + XBART	0.465	0.834
5	Regression + Cutoff	maxDepth	0.4	GCFM + XBART	0.438	0.833
5	Classification	maxDepth	0.4	GCFM + Utility	0.417	0.861
6	Regression + Cutoff	maxIPP	0.4	Perturb + RF	0.514	0.859
6	Regression + Cutoff	maxDepth	0.4	Perturb + RF	0.472	0.875
6	Regression + Cutoff	maxIPP	0.4	GCFM + XBART	0.507	0.816
6	Regression + Cutoff	maxDepth	0.4	GCFM + XBART	0.535	0.797

6	Classification	maxDepth	0.4	GCFM + Utility	0.438	0.853
7	Regression + Cutoff	maxIPP	0.4	Perturb + RF	0.562	0.852
7	Regression + Cutoff	maxDepth	0.4	Perturb + RF	0.521	0.865
7	Regression + Cutoff	maxIPP	0.4	GCFM + XBART	0.500	0.823
7	Regression + Cutoff	maxDepth	0.4	GCFM + XBART	0.486	0.826
7	Classification	maxDepth	0.4	GCFM + Utility	0.500	0.835
8	Regression + Cutoff	maxIPP	0.4	Perturb + RF	0.500	0.858
8	Regression + Cutoff	maxDepth	0.4	Perturb + RF	0.451	0.869
8	Regression + Cutoff	maxIPP	0.4	GCFM + XBART	0.569	0.780
8	Regression + Cutoff	maxDepth	0.4	GCFM + XBART	0.535	0.796
8	Classification	maxDepth	0.4	GCFM + Utility	0.507	0.834
9	Regression + Cutoff	maxIPP	0.4	Perturb + RF	0.493	0.865
9	Regression + Cutoff	maxDepth	0.4	Perturb + RF	0.479	0.866
9	Regression + Cutoff	maxIPP	0.4	GCFM + XBART	0.514	0.812
9	Regression + Cutoff	maxDepth	0.4	GCFM + XBART	0.486	0.827
9	Classification	maxDepth	0.4	GCFM + Utility	0.507	0.828
10	Regression + Cutoff	maxIPP	0.4	Perturb + RF	0.507	0.854
10	Regression + Cutoff	maxDepth	0.4	Perturb + RF	0.521	0.850
10	Regression + Cutoff	maxIPP	0.4	GCFM + XBART	0.528	0.799
10	Regression + Cutoff	maxDepth	0.4	GCFM + XBART	0.528	0.790
10	Classification	maxDepth	0.4	GCFM + Utility	0.535	0.828
11	Regression + Cutoff	maxIPP	0.4	Perturb + RF	0.479	0.864
11	Regression + Cutoff	maxDepth	0.4	Perturb + RF	0.479	0.861
11	Regression + Cutoff	maxIPP	0.4	GCFM + XBART	0.528	0.803
11	Regression + Cutoff	maxDepth	0.4	GCFM + XBART	0.549	0.795
11	Classification	maxDepth	0.4	GCFM + Utility	0.535	0.824
12	Regression + Cutoff	maxIPP	0.4	Perturb + RF	0.514	0.846
12	Regression + Cutoff	maxDepth	0.4	Perturb + RF	0.521	0.841
12	Regression + Cutoff	maxIPP	0.4	GCFM + XBART	0.528	0.803
12	Regression + Cutoff	maxDepth	0.4	GCFM + XBART	0.521	0.803
12	Classification	maxDepth	0.4	GCFM + Utility	0.521	0.815
13	Regression + Cutoff	maxIPP	0.4	Perturb + RF	0.514	0.846
13	Regression + Cutoff	maxDepth	0.4	Perturb + RF	0.556	0.821
13	Regression + Cutoff	maxIPP	0.4	GCFM + XBART	0.528	0.803
13	Regression + Cutoff	maxDepth	0.4	GCFM + XBART	0.528	0.807
13	Classification	maxDepth	0.4	GCFM + Utility	0.521	0.835
14	Regression + Cutoff	maxIPP	0.4	Perturb + RF	0.514	0.846
14	Regression + Cutoff	maxDepth	0.4	Perturb + RF	0.549	0.819
14	Regression + Cutoff	maxIPP	0.4	GCFM + XBART	0.528	0.803
14	Regression + Cutoff	maxDepth	0.4	GCFM + XBART	0.542	0.785
14	Classification	maxDepth	0.4	GCFM + Utility	0.549	0.823
15	Regression + Cutoff	maxIPP	0.4	Perturb + RF	0.514	0.846
15	Regression + Cutoff	maxDepth	0.4	Perturb + RF	0.479	0.823
15	Regression + Cutoff	maxIPP	0.4	GCFM + XBART	0.528	0.803
15	Regression + Cutoff	maxDepth	0.4	GCFM + XBART	0.535	0.795
15	Classification	maxDepth	0.4	GCFM + Utility	0.562	0.822

# Items	Tree Type	Criterion	w	Calibration Data	Sens	Spec
2	Regression + Cutoff	maxIPP	0.5	Perturb + RF	0.583	0.748
2	Regression + Cutoff	maxDepth	0.5	Perturb + RF	0.868	0.465
2	Regression + Cutoff	maxIPP	0.5	GCFM + XBART	0.750	0.534
2	Regression + Cutoff	maxDepth	0.5	GCFM + XBART	0.840	0.425
2	Classification	maxDepth	0.5	GCFM + Utility	0.549	0.722
3	Regression + Cutoff	maxIPP	0.5	Perturb + RF	0.750	0.606
3	Regression + Cutoff	maxDepth	0.5	Perturb + RF	0.688	0.658
3	Regression + Cutoff	maxIPP	0.5	GCFM + XBART	0.778	0.522

3	Regression + Cutoff	maxDepth	0.5	GCFM + XBART	0.660	0.659
3	Classification	maxDepth	0.5	GCFM + Utility	0.583	0.691
4	Regression + Cutoff	maxIPP	0.5	Perturb + RF	0.792	0.627
4	Regression + Cutoff	maxDepth	0.5	Perturb + RF	0.750	0.671
4	Regression + Cutoff	maxIPP	0.5	GCFM + XBART	0.708	0.652
4	Regression + Cutoff	maxDepth	0.5	GCFM + XBART	0.757	0.584
4	Classification	maxDepth	0.5	GCFM + Utility	0.660	0.677
5	Regression + Cutoff	maxIPP	0.5	Perturb + RF	0.764	0.614
5	Regression + Cutoff	maxDepth	0.5	Perturb + RF	0.806	0.566
5	Regression + Cutoff	maxIPP	0.5	GCFM + XBART	0.743	0.618
5	Regression + Cutoff	maxDepth	0.5	GCFM + XBART	0.750	0.598
5	Classification	maxDepth	0.5	GCFM + Utility	0.743	0.593
6	Regression + Cutoff	maxIPP	0.5	Perturb + RF	0.750	0.661
6	Regression + Cutoff	maxDepth	0.5	Perturb + RF	0.799	0.601
6	Regression + Cutoff	maxIPP	0.5	GCFM + XBART	0.771	0.587
6	Regression + Cutoff	maxDepth	0.5	GCFM + XBART	0.764	0.573
6	Classification	maxDepth	0.5	GCFM + Utility	0.694	0.633
7	Regression + Cutoff	maxIPP	0.5	Perturb + RF	0.840	0.587
7	Regression + Cutoff	maxDepth	0.5	Perturb + RF	0.785	0.668
7	Regression + Cutoff	maxIPP	0.5	GCFM + XBART	0.729	0.612
7	Regression + Cutoff	maxDepth	0.5	GCFM + XBART	0.715	0.640
7	Classification	maxDepth	0.5	GCFM + Utility	0.708	0.641
8	Regression + Cutoff	maxIPP	0.5	Perturb + RF	0.799	0.594
8	Regression + Cutoff	maxDepth	0.5	Perturb + RF	0.778	0.638
8	Regression + Cutoff	maxIPP	0.5	GCFM + XBART	0.743	0.622
8	Regression + Cutoff	maxDepth	0.5	GCFM + XBART	0.778	0.566
8	Classification	maxDepth	0.5	GCFM + Utility	0.729	0.624
9	Regression + Cutoff	maxIPP	0.5	Perturb + RF	0.799	0.581
9	Regression + Cutoff	maxDepth	0.5	Perturb + RF	0.792	0.592
9	Regression + Cutoff	maxIPP	0.5	GCFM + XBART	0.750	0.609
9	Regression + Cutoff	maxDepth	0.5	GCFM + XBART	0.750	0.606
9	Classification	maxDepth	0.5	GCFM + Utility	0.736	0.634
10	Regression + Cutoff	maxIPP	0.5	Perturb + RF	0.806	0.571
10	Regression + Cutoff	maxDepth	0.5	Perturb + RF	0.806	0.577
10	Regression + Cutoff	maxIPP	0.5	GCFM + XBART	0.757	0.603
10	Regression + Cutoff	maxDepth	0.5	GCFM + XBART	0.750	0.601
10	Classification	maxDepth	0.5	GCFM + Utility	0.771	0.621
11	Regression + Cutoff	maxIPP	0.5	Perturb + RF	0.847	0.532
11	Regression + Cutoff	maxDepth	0.5	Perturb + RF	0.847	0.539
11	Regression + Cutoff	maxIPP	0.5	GCFM + XBART	0.757	0.598
11	Regression + Cutoff	maxDepth	0.5	GCFM + XBART	0.771	0.608
11	Classification	maxDepth	0.5	GCFM + Utility	0.764	0.622
12	Regression + Cutoff	maxIPP	0.5	Perturb + RF	0.854	0.523
12	Regression + Cutoff	maxDepth	0.5	Perturb + RF	0.819	0.572
12	Regression + Cutoff	maxIPP	0.5	GCFM + XBART	0.757	0.600
12	Regression + Cutoff	maxDepth	0.5	GCFM + XBART	0.764	0.608
12	Classification	maxDepth	0.5	GCFM + Utility	0.757	0.627
13	Regression + Cutoff	maxIPP	0.5	Perturb + RF	0.812	0.596
13	Regression + Cutoff	maxDepth	0.5	Perturb + RF	0.812	0.616
13	Regression + Cutoff	maxIPP	0.5	GCFM + XBART	0.757	0.600
13	Regression + Cutoff	maxDepth	0.5	GCFM + XBART	0.778	0.586
13	Classification	maxDepth	0.5	GCFM + Utility	0.757	0.641
14	Regression + Cutoff	maxIPP	0.5	Perturb + RF	0.868	0.529
14	Regression + Cutoff	maxDepth	0.5	Perturb + RF	0.826	0.596
14	Regression + Cutoff	maxIPP	0.5	GCFM + XBART	0.757	0.600
14	Regression + Cutoff	maxDepth	0.5	GCFM + XBART	0.750	0.598

14	Classification	maxDepth	0.5	GCFM + Utility	0.750	0.635
15	Regression + Cutoff	maxIPP	0.5	Perturb + RF	0.868	0.529
15	Regression + Cutoff	maxDepth	0.5	Perturb + RF	0.861	0.549
15	Regression + Cutoff	maxIPP	0.5	GCFM + XBART	0.757	0.600
15	Regression + Cutoff	maxDepth	0.5	GCFM + XBART	0.715	0.612
15	Classification	maxDepth	0.5	GCFM + Utility	0.757	0.630

# Items	Tree Type	Criterion	w	Calibration Data	Sens	Spec
2	Regression + Cutoff	maxIPP	0.6	Perturb + RF	0.924	0.364
2	Regression + Cutoff	maxDepth	0.6	Perturb + RF	1.000	0.000
2	Regression + Cutoff	maxIPP	0.6	GCFM + XBART	0.868	0.370
2	Regression + Cutoff	maxDepth	0.6	GCFM + XBART	0.840	0.425
2	Classification	maxDepth	0.6	GCFM + Utility	0.840	0.443
3	Regression + Cutoff	maxIPP	0.6	Perturb + RF	0.938	0.326
3	Regression + Cutoff	maxDepth	0.6	Perturb + RF	1.000	0.000
3	Regression + Cutoff	maxIPP	0.6	GCFM + XBART	0.882	0.355
3	Regression + Cutoff	maxDepth	0.6	GCFM + XBART	0.882	0.355
3	Classification	maxDepth	0.6	GCFM + Utility	0.896	0.392
4	Regression + Cutoff	maxIPP	0.6	Perturb + RF	0.979	0.250
4	Regression + Cutoff	maxDepth	0.6	Perturb + RF	0.972	0.290
4	Regression + Cutoff	maxIPP	0.6	GCFM + XBART	0.910	0.323
4	Regression + Cutoff	maxDepth	0.6	GCFM + XBART	0.910	0.323
4	Classification	maxDepth	0.6	GCFM + Utility	0.854	0.433
5	Regression + Cutoff	maxIPP	0.6	Perturb + RF	0.986	0.212
5	Regression + Cutoff	maxDepth	0.6	Perturb + RF	0.979	0.268
5	Regression + Cutoff	maxIPP	0.6	GCFM + XBART	0.944	0.276
5	Regression + Cutoff	maxDepth	0.6	GCFM + XBART	0.910	0.323
5	Classification	maxDepth	0.6	GCFM + Utility	0.868	0.444
6	Regression + Cutoff	maxIPP	0.6	Perturb + RF	0.986	0.206
6	Regression + Cutoff	maxDepth	0.6	Perturb + RF	0.979	0.261
6	Regression + Cutoff	maxIPP	0.6	GCFM + XBART	0.931	0.300
6	Regression + Cutoff	maxDepth	0.6	GCFM + XBART	0.938	0.310
6	Classification	maxDepth	0.6	GCFM + Utility	0.924	0.382
7	Regression + Cutoff	maxIPP	0.6	Perturb + RF	0.972	0.266
7	Regression + Cutoff	maxDepth	0.6	Perturb + RF	0.979	0.233
7	Regression + Cutoff	maxIPP	0.6	GCFM + XBART	0.931	0.353
7	Regression + Cutoff	maxDepth	0.6	GCFM + XBART	0.924	0.339
7	Classification	maxDepth	0.6	GCFM + Utility	0.896	0.387
8	Regression + Cutoff	maxIPP	0.6	Perturb + RF	0.972	0.285
8	Regression + Cutoff	maxDepth	0.6	Perturb + RF	0.986	0.213
8	Regression + Cutoff	maxIPP	0.6	GCFM + XBART	0.931	0.340
8	Regression + Cutoff	maxDepth	0.6	GCFM + XBART	0.917	0.354
8	Classification	maxDepth	0.6	GCFM + Utility	0.910	0.399
9	Regression + Cutoff	maxIPP	0.6	Perturb + RF	0.972	0.312
9	Regression + Cutoff	maxDepth	0.6	Perturb + RF	0.958	0.337
9	Regression + Cutoff	maxIPP	0.6	GCFM + XBART	0.924	0.354
9	Regression + Cutoff	maxDepth	0.6	GCFM + XBART	0.938	0.352
9	Classification	maxDepth	0.6	GCFM + Utility	0.917	0.400
10	Regression + Cutoff	maxIPP	0.6	Perturb + RF	0.979	0.278
10	Regression + Cutoff	maxDepth	0.6	Perturb + RF	0.979	0.304
10	Regression + Cutoff	maxIPP	0.6	GCFM + XBART	0.938	0.348
10	Regression + Cutoff	maxDepth	0.6	GCFM + XBART	0.944	0.363
10	Classification	maxDepth	0.6	GCFM + Utility	0.938	0.391
11	Regression + Cutoff	maxIPP	0.6	Perturb + RF	0.979	0.266
11	Regression + Cutoff	maxDepth	0.6	Perturb + RF	0.979	0.280
11	Regression + Cutoff	maxIPP	0.6	GCFM + XBART	0.931	0.374

11	Regression + Cutoff	maxDepth	0.6	GCFM + XBART	0.944	0.377
11	Classification	maxDepth	0.6	GCFM + Utility	0.889	0.408
12	Regression + Cutoff	maxIPP	0.6	Perturb + RF	0.979	0.271
12	Regression + Cutoff	maxDepth	0.6	Perturb + RF	0.972	0.292
12	Regression + Cutoff	maxIPP	0.6	GCFM + XBART	0.931	0.362
12	Regression + Cutoff	maxDepth	0.6	GCFM + XBART	0.944	0.368
12	Classification	maxDepth	0.6	GCFM + Utility	0.882	0.410
13	Regression + Cutoff	maxIPP	0.6	Perturb + RF	0.986	0.260
13	Regression + Cutoff	maxDepth	0.6	Perturb + RF	0.979	0.279
13	Regression + Cutoff	maxIPP	0.6	GCFM + XBART	0.931	0.376
13	Regression + Cutoff	maxDepth	0.6	GCFM + XBART	0.944	0.337
13	Classification	maxDepth	0.6	GCFM + Utility	0.889	0.401
14	Regression + Cutoff	maxIPP	0.6	Perturb + RF	0.986	0.262
14	Regression + Cutoff	maxDepth	0.6	Perturb + RF	0.951	0.319
14	Regression + Cutoff	maxIPP	0.6	GCFM + XBART	0.931	0.372
14	Regression + Cutoff	maxDepth	0.6	GCFM + XBART	0.917	0.397
14	Classification	maxDepth	0.6	GCFM + Utility	0.917	0.408
15	Regression + Cutoff	maxIPP	0.6	Perturb + RF	0.986	0.256
15	Regression + Cutoff	maxDepth	0.6	Perturb + RF	0.951	0.300
15	Regression + Cutoff	maxIPP	0.6	GCFM + XBART	0.931	0.372
15	Regression + Cutoff	maxDepth	0.6	GCFM + XBART	0.931	0.378
15	Classification	maxDepth	0.6	GCFM + Utility	0.910	0.410