2017

Analysis Tools for Small and Big Data Problems

Juan Chen

Analysis Tools for Small and Big Data Problems

Juan Chen

Dissertation submitted
to the Eberly College of Arts and Sciences
at West Virginia University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in
Computational Statistics

Kenneth J. Ryan, Ph.D., Chair
Mark V. Culp, Ph.D., Chair
Erdogan Gunel, Ph.D.
Casey Jelsema, Ph.D.
Matthew Valenti, Ph.D.

Department of Statistics

Morgantown, West Virginia
2017

Keywords: Kernel Regression; Ordinal Data; Repeatability and Reproducibility (R&R); Reproducing Kernel Hilbert Space (RKHS); Semi-Supervised Learning

Abstract

Analysis Tools for Small and Big Data Problems

Juan Chen

The dissertation focuses on two separate problems. Each is informed by real-world applications. The first problem involves the assessment of an ordinal measurement system in a manufacturing setting. A random-effects model is proposed that is applicable to this repeatability and reproducibility context, and a Bayesian framework is adopted to facilitate inference. This first problem is an example of an analysis tool to solve a small data problem.

The second problem involves statistical machine learning applied to big data problems. As more and more data become available, a need increases to automate the ability to identify particularly relevant features in a prediction or forecasting context. This often involves expanding features using kernel functions to better facilitate predictive capabilities. Simultaneously, there are often manifolds embedded within big data structures that can be exploited to improve predictive performance on real data sets. Bringing together manifold learning with kernel methods provides a powerful and novel tool developed in this dissertation.

This dissertation has the advantage of contributing to a more-classical problem in statistics involving ordinal data and to cutting edge machine learning techniques for the analysis of big data. It is our contention that statisticians need to understand both problem types. The novel tools developed here are demonstrated on practical applications with strong results.

Acknowledgements

I wish to express my special appreciation and thanks to Dr. Kenneth J. Ryan and Dr. Mark V. Culp. I was extremely fortunate to have you both as my advisors. I thank you for all of your encouragement, patience, and motivation. Your continuous guidance and direction helped me grow as a researcher and led to the completion of this dissertation.

In addition, I thank the other members of my dissertation committee: Dr. Erdogan Gunel, Dr. Casey Jelsema, and Dr. Mathew Valenti. Your cooperation, thoughtful suggestions and comments, and expertise and patience helped to make the final research phase of my Ph.D. studies a memorable experience.

I am grateful for the financial support provided to me during my studies at West Virginia University. Dr. E. James Harner helped start this journey by offering me a Teaching Assistantship throughout my M.S. program. Dr. Michael Mays and Dr. Mark V. Culp provided Research Assistantships during my Ph.D. program.

I also extend my gratitude to the other Professors and Staff in the Department of Statistics: Ms. Barbara Dailey, Mr. A.B. Billings, Dr. Erin R. Leatherman, and Dr. Robert Mnatsakanov. You enriched my program of study and experiences while living in Morgantown.

And finally, last but by no means least, I thank my dad and mom. You provided me with vital moral and emotional support throughout my life. And most of all to my loving, patient, and supportive husband Shi and my handsome, cute, and clever son Jayden: Your company, encouragement, constant care, support, and understanding from the beginning through the final stages of my Ph.D. studies are so appreciated.

SUMMARY

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This dissertation investigates two challenging statistical problems. The first problem is the assessment of ordinal measurement systems. An ordinal measurement system classifies population items into ordered groups, e.g., "poor," "satisfactory," or "good." The ordinal response model of de Mast and van Wieringen (2010) is the starting point for this effort, and we extend this seminal work in two fundamental directions. First, we extend their modeling framework to account for operators (i.e., the individuals classifying items on the ordinal scale) as random effects, and our proposed Bayesian framework makes this particularly straightforward. Second, Vardeman and VanValkenburg (1999) surveyed the literature on gauge repeatability and reproducibility (R&R) in the context of a linear random-effects model for a continuous response, and we use the terminology from this work to define the concept of R&R carefully for an ordinal response. Chapter 2 is a journal-ready paper addressing these ordinal R&R statistical challenges and is currently under review for publication.

The second problem involves statistical machine learning. Predictive performance as

opposed to interpretation is often the application of a machine learner. A typical example is the supervised learning problem of n complete data pairs $(\boldsymbol{x}_i, y_i)$ for $i = 1, \ldots, n$, i.e., a feature data vector $\boldsymbol{x}_i \in \mathbb{R}^p$ and its corresponding response y_i. This type of data source can be organized in the familiar form of an $n \times p$ feature data matrix $\boldsymbol{X}$ and $n \times 1$ response vector $\boldsymbol{y}$. The goal is then to predict the response $y_0 = \text{NA}$ of a new observation given its corresponding feature vector $\boldsymbol{x}_0$. A challenge is that the '$\boldsymbol{X}$' matrix inputed to the user may not directly be in the most useful form for prediction. This well-known concept is presumably a point of emphasis in an introductory course on linear regression analysis. If $\boldsymbol{X}$ represents a design matrix, additional columns (e.g., square terms for quadratic trends) may need to be appended to $\boldsymbol{X}$ to form an adequate model matrix before applying a standard linear regression subroutine.

In the modern era, there is still a need for the skilled analyst who can handle routine regression analyses and provide interpretative value on smaller data sets, but the focus of the machine learning portion of this dissertation is to automate elements of a prediction process. Two competing schools of thought on processing the initially recorded feature information $\boldsymbol{X}$ are dimensionality reduction and expansion. Dimensionality reduction includes kernel PCA (Boser et al., 1992) and Laplacian approaches (Belkin and Niyogi, 2003). Such methods ignore or marginalize irrelevant directions in $\boldsymbol{X}$ for the prediction task at hand (Kung, 2014). This can be done on the rows and/or columns of $\boldsymbol{X}$.

On the other hand, methods for dimensionality expansion include localized estimators. The idea is to take a small p problem and capture the intrinsic local structure to improve performance. Early examples of this type include k-Nearest Neighbors (k-NN), the Nadaraya-Waston kernel, and LOWESS methods (Hastie et al., 2009). In this direction, kernel methods are established as having led to some of the most powerful machine learning techniques (Kung, 2014).

Chapter 3 justifies a reproducing kernel Hilbert space (RKHS) setting as a dimensional-

ity expansion approach that optimizes over a set of functions. As motivation for this RKHS framework, a discussion of smoothing splines, a related dimensionality expansion technique, is in Section 3.1. While splines have strong theoretical underpinnings when $p = 1$, they, unlike the RKHS framework, neither extend naturally to larger p nor predict well on real data. The presentation of the RKHS framework in Section 3.2 introduces the reader to the so-called 'kernel trick,' which provides a finite data kernel regression optimization that is equivalent to the Hilbert space optimization of interest. In Section 3.3, the sequential minimal optimization (SMO) is summarized (Platt, 1998). While this SMO heuristic gets around a quadratic programming problem and enables the fitting of an SVM, Chapter 3 culminates in the negative result of Section 3.3.3. The related complications to the loss function fit by the SMO, such as the so-called ε-sensitive loss function, underperform on real data benchmarks, and this directs our search for practical methods in Chapter 4 on optimization problems with standard loss functions such as square error loss for regression or logistic loss for classification. Our novelty and contribution in Chapter 4 comes from developing penalty functions that effectively boost performance on real data.

In this regard, Chapter 4 proposes two novel prediction methods: a Safe Semi-Supervised Kernel Model (S3KM) and an anchor graph S3KM (AS3KM). Both methods are principled on a RKHS via the 'kernel trick.' In addition, each can directly help assess the potential of semi-supervised learning. Under a semi-supervised paradigm, a full feature matrix $\boldsymbol{X}$ may be available, but some proper subset of the responses is missing, and this partitions the index set $i = 1, \ldots, n$ for the n observations into the labeled and unlabeled sets, where the unlabeled set is defined as all observations with a missing response, i.e., $\{i : y_i = \text{NA}\} \subset \{1, \ldots, n\}$.

There are a number of ways to motivate semi-supervised learning. For example, the $\boldsymbol{x}$-data may be readily available or cheaper than the response y. A example is credit scoring. All potential customers who applied for a loan in the past submitted their application containing the $\boldsymbol{x}$-data, but suppose the bank now wants to start offering loans to a new customer segment.

If this customer type was previously denied loans, then a default response on the loan yes ($y = 1$) or no ($y = 0$) is not available for the new customer segment of interest.

Semi-supervised approaches hold the promise of using all the available information in the labeled and unlabeled sets to improve performance, and there are number of approaches in the literature for how one might go about this. One such concept is based on the cluster assumption (Chapelle et al., 2006a). This uses the $\boldsymbol{X}$ data to implicitly find manifolds (i.e., clusters) and assumes that the manifolds have predictive value (Hein et al., 2005). In a sense, much of this semi-supervised literature was created in a bubble. Simulations hand picked the probability models (or tuning parameter values) to make a proposed technique flourish, and restrictive semi-supervised smoothness assumptions reinforced the need for these manifold or gap finding methods (Lafferty and Wasserman, 2008). Such activity was extensive and included graph cutting (Wang et al., 2013), graph regularization (Zhou et al., 2004; Belkin et al., 2006; Culp and Ryan, 2013), S3VM methods (Chapelle et al., 2006a, 2008), and several other approaches (Chapelle et al., 2006b), but did not necessarily translate into methods ready to handle real data challenges.

On the other hand, a supervised learner computes a prediction rule from only the labeled data, but these methods have a longer and more practical history. For example, supervised packages such as `caret` (Kuhn, 2014) have computationally efficient and robust cross-validation (CV) procedures and perform well on real (and noisy) data challenges. This helps motivates the concept of *safe* semi-supervised approaches, i.e., semi-supervised approaches that perform comparable to or better than a supervised counterpart. The proposed S3KM and AS3KM have a built-in safety feature. Sections 4.3 and 4.4 include an analysis of the turning parameter settings, some of which default to a safe and well-established supervised baseline or alternative.

The type of gap finding semi-supervised approaches mentioned earlier perform poorly when semi-supervised assumptions are even slightly perturbed on real data (Culp and Ryan,

2013; Singh et al., 2009). They often result in jagged classifications rules that are highly sensitive to noise. The resulting degradation in performance is much worse than that for supervised learning (Fernández-Delgado et al., 2014). In spite of this, the scale and availability of unlabeled data makes the use of semi-supervised learning very appealing in applications such as drug discovery, text analysis, and bioinformatics. The proposed S3KM and AS3KM in Chapter 4 extend semi-supervised optimization paradigms for graph penalization (Zhou et al., 2004; Belkin et al., 2006; Chapelle et al., 2006b) into the safe arena. This dissertation conclusion with the summary and future research directions described in Chapter 5.

CHAPTER 2

RANDOM EFFECTS MODELS FOR REPEATABILITY AND REPRODUCIBILITY OF ORDINAL MEASUREMENTS

We use a Bayesian inferential approach to analyze ordinal repeatability and reproducibility (R&R) data using the De Mast–Van Wieringen model (de Mast and van Wieringen, 2010). We also consider a population of raters by extending the De Mast–Van Wieringen model to random effects and define match-probability-based measures to decompose R&R into contributions due to repeatability and due to reproducibility. These extensions are illustrated with the De Mast–Van Wieringen R&R study data, although our motivation for this work comes from a need to analyze ordinal data in a proprietary context.

Keywords: Bayesian, Dirichlet distribution, fixed effects, Markov chain Monte Carlo.

2.1 Introduction

Ordinal data arise often in business and industry. For example, when visual inspection is required to test for defects in a manufacturing context, the measurement scale of "poor," "fair," "good," "excellent" might be employed. As is the case with numerical measurements, precision of ordinal measurements is important for quality control. Unlike the numerical case, however, the state of the science for the assessment of repeatability and reproducibility (R&R) in the context of ordinal data is not as well-developed. Standard methods for the design and analysis of gauge R&R studies are well-known (see Burdick et al. (2005) for a review); the lack of sufficient methods to carry out R&R analyses on ordinal data was thoughtfully outlined by de Mast and van Wieringen (2010) in their seminal paper proposing a latent variable model to assess R&R of ordinal measurements. Their frequentist inferential approach provides estimates of R&R for a fixed group of operators. This paper offers an important extension to allow for random effects in the model, enabling us to treat the operators as a sample from a larger population of operators, which likely is the case in many applications. Allowing for the inclusion of random effects will enable prediction of R&R for a new operator for whom we currently do not have data. This implementation of this extension is achieved through a novel application of a Bayesian inferential approach described in Section 2.3.

The random effects model presented in Section 2.3 also applies to situations where the response is distributed over a finite set of numbers. For example, consider a manufacturing setting where operators look for the presence or absence of $H-1$ features on units coming off an assembly line. It may be more appropriate to model the distribution of the number of features present on a unit $0,1,\ldots,H-1$ with a multinomial as opposed to a binomial

distribution because the trials corresponding to each feature are not necessarily identically distributed.

Before defining the random effects modeling extension in Section 2.3, the fixed effects model of de Mast and van Wieringen (2010) is briefly described in Section 2.2. Section 2.4 then looks at defining parametric functions to measure R&R in numerical and nominal extremes. Data analyses based on the Section 2.3 model and Section 2.4 measures are given in Section 2.5. The paper conclusions with a follow-up discussion in Section 2.6.

2.2 Latent Variable Model

When operators classify parts according to quality on an ordered scale, for example $\{1 = \text{poor}, 2 = \text{fair}, 3 = \text{good}, 4 = \text{excellent}\}$, the true value of the construct of quality is not directly measured in the classification process but falls somewhere along an underlying continuum. De Mast and van Wieringen (2010) considered the latent value for the quality of a part and proposed a latent variable model conducive to the definition and computation of R&R for ordinal measurements. That model is described here.

For now, assume a balanced design with I parts, J operators, K repeated measurements for each operator/part. Also, let H be the number of categories for classification. Then $Y_{ijk} \in \{1, 2, ..., H\}$ is the category assigned to part i on the kth repetition by operator j. For a fixed part i with a latent value of x, let $q_j(h|x) := P(Y_{ijk} = h|X_i = x)$. Then $q_j(h|x)$ is a function of x from $\mathbb{R}$ to $[0, 1]$ that specifies the probability operator j will label part i as being in category h given the part's true value of x. De Mast and van Wieringen (2010) proposed

$$q_j(h|x) = \frac{\exp(\sum_{m=1}^{h-1} \alpha_j(x - \delta_{jm}))}{\sum_{n=1}^{H} \exp(\sum_{m=1}^{n-1} \alpha_j(x - \delta_{jm}))}. \tag{2.1}$$

We model the cut-point parameters $\boldsymbol{\delta}_j = (\delta_{j1}, \ldots, \delta_{j(H-1)})$ for operator j as ordered, i.e.,

$\delta_{j1} \leq \delta_{j2} \leq \cdots \leq \delta_{j(H-1)}$. In this case, functions $q_j(h|x)$ and $q_j(h+1|x)$ intersect at $x = \delta_{jh}$ for $h = 1, \ldots, H-1$, e.g., see the upper left panel of Figure 2.1. The cut points define operator j's category boundaries such that part i is determined by operator j to most likely be in category h if $\delta_{j(h-1)} < x_i < \delta_{jh}$. The α_j are positive scaling parameters such that smaller (larger) values for α_j correspond to flatter (steeper) curves for operator j. The steeper curves resulting from the larger α_j demonstrate an improved discrimination ability. That is, larger α_j are interpreted as better repeatabilities. Misaligned cut points between two operators indicate problems with reproducibility.

The top four panels in Figure 2.1 are examples and demonstrate the effects of small versus large α and aligned versus misaligned $\boldsymbol{\delta}$. In the upper two panels on the left, the probability curves in Equation (2.1) are basically the same (i.e., aligned cut points resulting in stronger reproducibility), whereas these curves are flat (i.e., small α resulting in weak repeatability). In the panels on the right, the pattern is reversed. The curves are steep (i.e., higher α resulting in stronger repeatability), but the curves are distinct across rows (i.e., misaligned cut points resulting in weaker reproducibility). We revisit Figure 2.1 and its bottom row later in Section 2.4 after defining measures to decompose R&R into proportions.

To fit Model (2.1), de Mast and van Wieringen (2010) transform the response $\{Y_{ijk}\}$ into $\{R_{ijh}\}$ such that $R_{ijh} = |\{k|Y_{ijk} = h\}|$, i.e., the number of repeats out of K for which the jth operator assigns category h to part i. Note that $\sum_{h=1}^{H} q_j(h|x) = 1$, so if $\mathbf{R}_{ij} = (R_{ij1}, \ldots, R_{ijH})$, then

$$\mathbf{R}_{ij}|\boldsymbol{\delta}_j, \alpha_j, X_i = x \overset{\text{ind}}{\sim} \text{Multinomial}(K, (q_1(h|x), \ldots, q_H(h|x))),$$

given an assumption of conditional independence. It is important to note that if the same arbitrary scalar is added to the X_i and the cut points $\boldsymbol{\delta}_j$, the multinomial probabilities defined in Equation (2.1) do not change. That is, the model is unidentifiable. De Mast and van Wieringen (2010) circumvented this identifiability problem by assuming the latent variables

were a random sample from the standard normal distribution, i.e.,

$$X_i \overset{\text{iid}}{\sim} \text{Normal}(0,1). \tag{2.2}$$

Assuming complete data $R_{ijh} = r_{ijh}$ and $X_i = x_i$ are observed for all $i = 1,\ldots,I$, $j = 1,\ldots,J$, and $h = 1,\ldots,H$, the resulting likelihood is

$$\mathscr{L}_{\text{Fixed}}(\boldsymbol{\delta},\boldsymbol{\alpha}) \propto \prod_{i=1}^{I}\left[\phi(x_i)\prod_{j=1}^{J}\prod_{h=1}^{H} q_j(h|x_i)^{r_{ijh}}\right], \tag{2.3}$$

where $\phi(\cdot)$ is the probability density function of the standard normal distribution. De Mast and van Wieringen (2010) used maximum likelihood estimation to estimate the α_j and δ_{jm} parameters by integrating out the latent variables X_i with a Gauss-Hermite numerical integral quadrature rule.

The subscript "Fixed" in $\mathscr{L}_{\text{Fixed}}(\boldsymbol{\delta},\boldsymbol{\alpha})$ from Likelihood (2.3) reflects the fact that operators are treated as fixed effects in this estimation approach. However, in many contexts, it is natural to want to model the operators as a random sample from some larger population of operators; this is routinely done in R&R problems with continuous measurements. However, authors who have tackled the more complex ordinal case, including de Mast and van Wieringen (2010) and Deldossi and Zappa (2014), have thus far focused on establishing R&R estimation approaches in the context of a fixed operator effect. In Section 2.3, we propose a random effects model based on the probability curves in Equation (2.1) and describe our novel use of a Bayesian approach of parameter estimation that treats the latent variables X_i for $i = 1,\ldots,I$ as additional parameters.

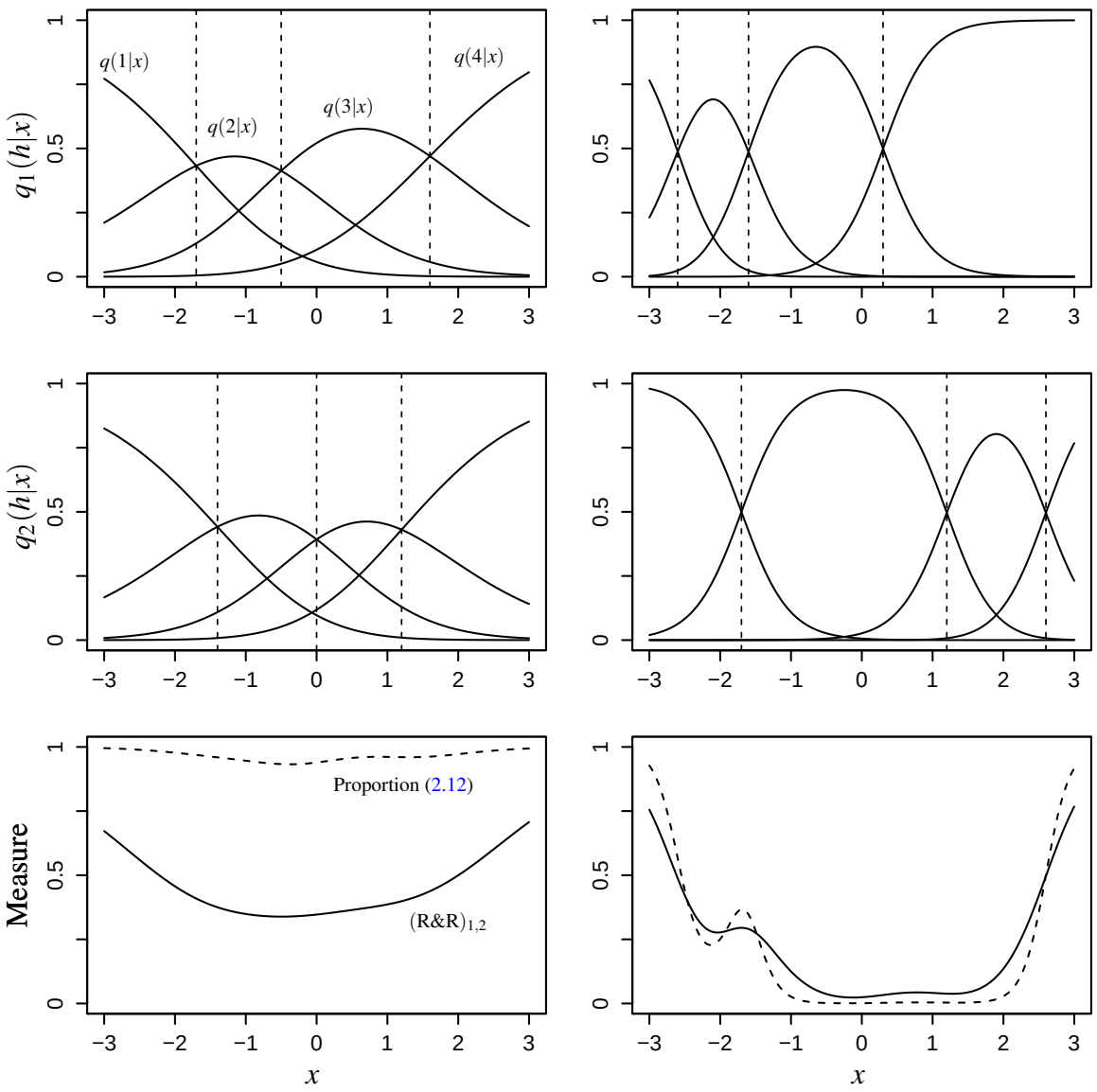


Figure 2.1: The effects of small $\alpha = 1$ with aligned cut points (column 1) versus large $\alpha = 3$ with misaligned cut points (column 2) for two hypothetical pairs of operators. Rows 1 and 2 are the probability curves from Equation (2.1) for operators 1 and 2, respectively, with dashed vertical lines at the cut points, and row 3 displays R&R measures. Each is plotted against the latent part variable x.

2.3 Random Effects Model

In a design of experiments context, a factor is referred to as a *random effect* if its observed levels are a subset of the levels of interest. The observed levels of a random effect are

often modeled as a sample from some population with unknown parameters. In the R&R context of the previous section, we propose to look at both the parts I and raters J as random effects, whereas preceding work focused on treating only the parts as random effects. The raters were treated as fixed effects making inference on the broader population of raters not directly possible. For example, by also treating rater as a random effect, this work will make a predictive inference concerning a new rater $J+1$, for whom no data are available, seamless. This predictive capability is revisited in the examples of Section 2.5. The purpose of this section is to define the needed random effects extension. The likelihood and a general use prior are defined in Sections 2.3.1 and 2.3.2.

2.3.1 The Likelihood

To extend Model (2.1) to a random effects model, we first model the scale parameters as

$$\alpha_j|\mu_\alpha,\tau_\alpha \overset{\text{iid}}{\sim} \text{Log Normal}(\mu_\alpha,\tau_\alpha), \tag{2.4}$$

where μ_α and $\sigma_\alpha = 1/\sqrt{\tau_\alpha}$ are the mean and standard deviation of normally distributed $\log(\alpha_j)$ for $j = 1,\ldots,J$. This notation parameterizes the log normal distribution in terms of its precision $\tau_\alpha = 1/\sigma_\alpha^2$ in order to line up with that used by JAGS (Plummer, 2015) and our Bayes model implementation to come. Larger σ_α (or equivalently smaller τ_α) implies more heterogeneity between the scale parameters of operators, and larger μ_α implies an expectation of steeper curves (e.g., recall the top two panels on the right of Figure 2.1).

Next, a distribution over ordered cut points is defined. This will be accomplished indirectly by putting a distribution on the probability π_{jh} that a randomly selected operator records category h on a randomly selected part. With $\boldsymbol{\pi}_j = (\pi_{j1},\ldots,\pi_{jH})$, let

$$\boldsymbol{\pi}_j|\boldsymbol{\lambda} \overset{\text{iid}}{\sim} \text{Dirichlet}(\boldsymbol{\lambda}) \tag{2.5}$$

for $j = 1, \ldots, J$ and define the transforms

$$\delta_{jm} = \Phi^{-1}\left(\sum_{n=1}^{m} \pi_{jn}\right) \tag{2.6}$$

for $j = 1, \ldots, J$ and $m = 1, \ldots, H-1$, where $\Phi^{-1}(\cdot)$ is the inverse of the cumulative distribution function of a standard normal random variable. The support of Distribution (2.5) is non-negative vectors of length H summing to one, and the positive parameters $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_H)$ control the mean and variance of the components of $\boldsymbol{\pi}_j$, i.e.,

$$\begin{aligned} \mathbb{E}[\pi_{jh}|\boldsymbol{\lambda}] &= \frac{\lambda_h}{\lambda_0}, \text{ where } \lambda_0 = \sum_{h=1}^{H} \lambda_h, \\ \mathbb{V}\text{ar}(\pi_{jh}|\boldsymbol{\lambda}) &= \frac{\lambda_h(\lambda_0 - \lambda_h)}{\lambda_0^2(\lambda_0 + 1)}, \end{aligned}$$

by the known properties of the Dirichlet distribution. For example, all $\lambda_h = \lambda$ for some positive scalar λ implies that the probability vector with equal components of $1/H$ is expected. Large values for λ imply lower variance because the variance formula is a quadratic divided by a cubic polynomial. The induced distribution on cut points is necessarily ordered because of the cumulative sum of probabilities in the Transformations (2.6). If the components of parameter vector $\boldsymbol{\lambda}$ are large (small), we expect the cut points of two randomly selected operators j_1 and j_2 to be aligned (misaligned) and for these two operators to exhibit strong (weak) reproducibility in some definable sense to come.

Applying the above distributions for the scale parameters and cut points, the random effects likelihood can be written as

$$\begin{aligned} \mathscr{L}_{\text{Random}}(\boldsymbol{\delta}, \boldsymbol{\alpha}, \boldsymbol{\lambda}, \mu_\alpha, \sigma_\alpha) \propto\ & \mathscr{L}_{\text{Fixed}}(\boldsymbol{\delta}, \boldsymbol{\alpha}) \times \\ & \prod_{j=1}^{J} \left[\frac{1}{\sigma_\alpha \alpha_j} \phi\left(\frac{\log(\alpha_j) - \mu_\alpha}{\sigma_\alpha}\right) \frac{\Gamma(\lambda_0) \prod_{h=1}^{H} \pi_{jh}^{\lambda_h - 1}}{\prod_{h=1}^{H} \Gamma(\lambda_h)} \right], \end{aligned} \tag{2.7}$$

where $\Gamma(\cdot)$ is the gamma function. The form of the Random Effects Likelihood (2.7) is the Fixed Effects Likelihood times an adjustment. This adjustment promotes a data-based compromise between fixed separate analyses by operator and a single pooled analysis where all operators are assumed to have the same parameters.

2.3.2 A General Use Prior

The Random Effects Likelihood (2.7) is more complex than that for Fixed Effects (2.3), and the numerical analysis required to compute maximum likelihood estimates (MLEs) for either likelihood is non-trivial, is prone to numerical instabilities, and might as a convenience result in the application of large-sample approximate confidence interval procedures. On the other hand, non-linear models with latent variables and their random effects extensions often lend themselves to seamless Bayes implementation with direct calculation of the posterior of any parametric function to quantify its uncertainty. But the price for this ease of implementation is the work needed to test and justify a prior for a Bayesian analysis. When little prior information is available, the concept is to simply stabilize the analysis and in turn to not shrink the Bayesian estimates far from the MLEs. In this regard, we use

$$\begin{aligned} \mu_\alpha &\sim \text{Normal}(\mu_{\mu_\alpha}, \tau_{\mu_\alpha}), \\ \tau_\alpha = 1/\sigma_\alpha^2 &\sim \text{Log Normal}(\mu_{\tau_\alpha}, \tau_{\tau_\alpha}), \\ \lambda_h &\overset{\text{iid}}{\sim} \text{Log Normal}(\mu_\lambda, \tau_\lambda) \text{ for } h = 1, \ldots, H, \end{aligned}$$

and suggest hyperparameter values of $\mu_{\mu_\alpha} = 0.8$, $\tau_{\mu_\alpha} = 0.4$, $\mu_{\tau_\alpha} = 4$, $\tau_{\tau_\alpha} = 0.4$, $\mu_\lambda = 2$, and $\tau_\lambda = 0.2$ for general purpose use. The rationale is based on setting far extremes of the parameter space equal to $\mu \pm 1.96\sigma$ and solving the resulting equations.

- The middle 0.95 prior probability is on $\exp(\mu_\alpha) \in (0.1, 50)$, so the the true median of the distribution $\alpha_j | \mu_\alpha, \tau_\alpha$ is somewhere between very flat $\alpha \approx 0.1$ and very steep

curves $\alpha \approx 50$ (recall Figure 2.1).

- The middle 0.95 prior probability is on $\exp(2 * 1.96\sigma_\alpha) \in (1.1, 10)$. These extremes make the 0.975 quantile of $\alpha_j|\mu_\alpha, \tau_\alpha$ either 1.1 or 10 times that of the 0.025 quantile, so the distribution of α_j from operator-to-operator is either homogeneous (1.1 extreme) or heterogeneous (10 extreme).

- The middle 0.95 prior probability is on $\lambda_k \in (0.1, 500)$, so the prior distribution on cut point distributions spans a range from an aligned (500 extreme) to a misaligned (0.1 extreme) cut point distribution.

Two approaches are used in Section 2.5 to validate the robustness to this prior choice. The first is in the context of a real data analysis. The endpoints of the intervals (0.1,50), (1.1,10), (0.1,500) used to define this prior are changed by an order of magnitude to (0.01,500), (1.01,100), (0.01, 5000) to reset the prior and redo the analysis. This sensitivity analysis demonstrates that inference on parametric functions of interest is unaffected by further spreading out of the prior distribution. The second is a simulation study, based on the real data, and is used to demonstrate the solid frequentist properties of the Bayes method with the suggested general purpose prior. Simulation is further used to investigate ordinal R&R from a design of experiments perspective in terms of choosing I, J, K under the constraint of a fixed number of responses $I * J * K$. Before these results are presented in Section 2.5, Section 2.4 focuses on defining parametric functions that measure R&R.

2.4 R&R Measures

Defining measures for ordinal R&R turns out to be a challenging and subtle task. Deldossi and Zappa (2014) called into question the measures proposed by de Mast and van Wieringen (2010). De Mast et al. (2014) was critical of the use of the heavily cited and very often used

kappa statistic. Our goal in this section is define measures for ordinal R&R that decompose some sensible metric in the response such as a variance or a match probability into percentage components due to repeatability and reproducibility. It is desirable for these measures to reflect the ordinal nature of the scale of the response, but it is seemingly of the utmost importance to first clearly define what is meant by repeatability and reproducibility (R&R). For this reason, Section 2.4.1 discusses a linear model with a continuous response to help clearly define R&R before tackling our goal in Sections 2.4.2 and 2.4.3. Section 2.4.2 focuses on what might be termed a "numerical extreme approach" where one is willing to assign numbers to the ordinal categories, whereas Section 2.4.3 is more of a "nominal extreme" that allows for a number of ways to incorporate the ordered nature of the categories, where the "best" way depends on the application.

2.4.1 Gauge R&R Measures for a Continuous Response

As in Section 2.2, let Y_{ijk} be the kth measurement made by operator j on part i. A two-way random effects linear model is

$$\begin{aligned} Y_{ijk} &= \mu + \alpha_i + \beta_j + \varepsilon_{ijk}, \text{ where} \\ \alpha_i &\overset{\text{iid}}{\sim} \text{Normal}(0, 1/\sigma^2_{\text{part}}), \\ \beta_j &\overset{\text{iid}}{\sim} \text{Normal}(0, 1/\sigma^2_{\text{operator}}), \\ \varepsilon_{ijk} &\overset{\text{iid}}{\sim} \text{Normal}(0, 1/\sigma^2), \end{aligned} \tag{2.8}$$

and all α_i, β_j, and ε_{ijk} are independent. These types of models, possibly also allowing for interactions between parts and operators, are standard in the assessment of gauge R&R for a continuous response (Vardeman and VanValkenburg, 1999).

Following Vardeman and VanValkenburg (1999), the *repeatability variance* is σ^2, and the *reproducibility variance* is $\sigma^2_{\text{operator}}$. Each of these components of variance are better

Example	Univariate Data Set	Expected Sample Variance
1	$Y_{ij1}, Y_{ij2}, \ldots, Y_{ijn}$	$\mathbb{E}[s_1^2] = \sigma^2$
2	$Y_{i11}, Y_{i21}, \ldots, Y_{in1}$	$\mathbb{E}[s_2^2] = \sigma^2_{\text{operator}} + \sigma^2$

Table 2.1: A pair of hypothetical univariate data sets of sample size n are listed along with their expected sample variances under Linear Model (2.8).

understood through the two hypothetical data sets in Table 2.1. In the first, the same operator is asked to measure the same part n times, and the expected sample variance s_1^2 is the repeatability variance σ^2. In the second, n randomly selected operators are asked to measure the same part once each, but the expected sample variance s_2^2 is $\sigma^2_{\text{operator}} + \sigma^2$, i.e., the sum of the R&R variances. This is an important point that will be referred to while developing ordinal measures. That is, we can directly construct/obtain easy-to-understand data sets that can capture repeatability or R&R, but to capture the concept of reproducibility requires a subtraction in the present context, i.e., $\mathbb{E}[s_2^2 - s_1^2] = \sigma^2_{\text{operator}}$.

Another salient point is that the concept of R&R in the engineering literature has historically focused on a fixed part, i.e., consider $\mu + \alpha_i$ to be fixed. In this case,

$$\begin{array}{ccccccc} \underbrace{\mathbb{V}\text{ar}(y_{ijk}|\mu+\alpha_i)} & = & \underbrace{\mathbb{V}\text{ar}(\mathbb{E}[y_{ijk}|\mu+\alpha_i+\beta_j]|\mu+\alpha_i)} & + & \underbrace{\mathbb{E}[\mathbb{V}\text{ar}(y_{ijk}|\mu+\alpha_i+\beta_j)|\mu+\alpha_i]} \\ \sigma^2_{\text{operator}} + \sigma^2 & = & \sigma^2_{\text{operator}} & + & \sigma^2 \end{array}$$

relates the breakdown of variance components in the presented linear model to a well-known variance identity, i.e., $\mathbb{V}\text{ar}(Y) = \mathbb{V}\text{ar}(\mathbb{E}[Y|X]) + \mathbb{E}[\mathbb{V}\text{ar}(Y|X)]$. For the linear model currently under consideration,

$$\begin{aligned} \%\text{ of variance due to repeatability} &= 100 \times \left(\frac{\sigma^2}{\sigma^2_{\text{operator}} + \sigma^2} \right) \% \\ \%\text{ of variance due to reproducibility} &= 100 \times \left(\frac{\sigma^2_{\text{operator}}}{\sigma^2_{\text{operator}} + \sigma^2} \right) \% \end{aligned} \tag{2.9}$$

decomposes a meaningful total, i.e., the variance of Example 2 from Table 2.1, into mean-

ingful percentages. We next look to extend such a breakdown to the context of the model from Section 2.3.

2.4.2 Numerical-Based R&R Measures

Next, we use the model from Section 2.3, i.e., our random effects extension of de Mast and van Wieringen (2010), and also assume that the subject matter expert is willing to assign numbers to the categories of the ordinal response. The equally spaced values of $1,2,\ldots,H$ for $H = 4$ categories will be used for simplicity, although this presentation extends to $H \geq 2$ categories and any ordered (possibly non-equally spaced) values including $0,1,\ldots,H-1$ used in the motivating example of the previous paragraph.

	Category				Numerical		Nominal	
Operator j	1	2	3	4	Mean	Variance	$(\text{Repeatability})_j$	$(\text{R\&R})_{1,2}$
1	0	0.1	0.1	0.8	3.7	0.41	0.66	0.45
2	0	0.1	0.4	0.5	3.4	0.44	0.42	0.45

Table 2.2: Hypothetical distributions for a pair of operators on a fixed part. The additional columns are used to compute the numerical-based measures from Section 2.4.2 given the Likert scale 1-4 and the nominal-based measures from Section 2.4.3.

Decomposing the total variation in this way allows us to compute separately the variation due to repeatability, the variation due to reproducibility, and the proportion of total variation accounted for by each. Consider the example shown in Table 2.2 for a fixed part classified as one of four ordinal categories labeled $1,2,3,4$ by two operators according to the given probability distributions. With this numerical Likert scale (i.e., 1-4) for each operator, we can calculate the mean and the variance of the discrete probability distribution. Then the repeatability variation can be calculated as the mean of the variances, while reproducibility variation is the variance of the means. In this example, the repeatability variance is $(0.41 + 0.44)/2 = 0.425$, and the reproducibility variance is $0.25 * (3.7 - 3.4)^2 = 0.0225$. Thus, repeatability comprises $0.425/(0.425 + 0.0225) \approx 95\%$ of the total variance on the Likert

scale for the part under consideration if operator 1 or 2 is selected at random by the flip of a fair coin. The Table 2.2 example is revisited after defining nominal-based measures in Section 2.4.3.

2.4.3 Nominal-Based R&R Measures

The purpose of this section is to define measures for normal R&R and then extend those to reflect the ordered categories of ordinal data. The basic concept is built off of match probabilities for a pair of responses on the same part, so throughout this development assume a fixed part i with latent value $X_i = x$ is under consideration. Let column vector $\boldsymbol{p}_j = \boldsymbol{p}_j(x) = (p_{j1}, \ldots, p_{jH})^\top$, where $p_{jh} = q_j(h|x)$ are computed from Equation (2.1). The vectors of probabilities $\boldsymbol{p}_j$ for operator j on part i sum to one, i.e., $\boldsymbol{p}_j^\top \vec{1} = 1$, for each $j = 1, 2, \ldots$. A pair of repetitions from the same operator j will be used to define his/her repeatability with the symmetric $H \times H$ matrix of match probabilities $\boldsymbol{p}_j \boldsymbol{p}_j^\top$. Note this matrix satisfies sum $\left(\boldsymbol{p}_j \boldsymbol{p}_j^\top\right) = 1$, and its row h and column h' entry is $P(Y_{ij1} = h \cap Y_{ij2} = h')$. Similarly, a pair of repetitions one from each of a two randomly selected operators j and $j' \neq j$ will be used to define the R&R between these operators. For this purpose, the $H \times H$ outer product matrix $\boldsymbol{p}_j \boldsymbol{p}_{j'}^\top$ will be used. When for example $H = 4$, these matrices have the forms

$$\boldsymbol{p}_j \boldsymbol{p}_j^\top = \begin{bmatrix} p_{j1}p_{j1} & p_{j1}p_{j2} & p_{j1}p_{j3} & p_{j1}p_{j4} \\ p_{j2}p_{j1} & p_{j2}p_{j2} & p_{j2}p_{j3} & p_{j2}p_{j4} \\ p_{j3}p_{j1} & p_{j3}p_{j2} & p_{j3}p_{j3} & p_{j3}p_{j4} \\ p_{j4}p_{j1} & p_{j4}p_{j2} & p_{j4}p_{j3} & p_{j4}p_{j4} \end{bmatrix},$$

$$\boldsymbol{p}_j\boldsymbol{p}_{j'}^\top = \begin{bmatrix} p_{j1}p_{j'1} & p_{j1}p_{j'2} & p_{j1}p_{j'3} & p_{j1}p_{j'4} \\ p_{j2}p_{j'1} & p_{j2}p_{j'2} & p_{j2}p_{j'3} & p_{j2}p_{j'4} \\ p_{j3}p_{j'1} & p_{j3}p_{j'2} & p_{j3}p_{j'3} & p_{j3}p_{j'4} \\ p_{j4}p_{j'1} & p_{j4}p_{j'2} & p_{j4}p_{j'3} & p_{j4}p_{j'4} \end{bmatrix}.$$

Natural measures for repeatability and R&R are to compute match probabilities by simply summing the main diagonals of these outer product matrices, i.e., define

$$(\text{Repeatability})_j = P(Y_{ij1} = Y_{ij2}) = \sum_{h=1}^{H} p_{jh}^2 = \text{tr}\left(\boldsymbol{p}_j\boldsymbol{p}_j^\top\right) = \boldsymbol{p}_j^\top\boldsymbol{p}_j \text{ and} \tag{2.10}$$

$$(\text{R\&R})_{jj'} = P(Y_{ij1} = Y_{ij'1}) = \sum_{h=1}^{H} p_{jh}p_{j'h} = \text{tr}\left(\boldsymbol{p}_j\boldsymbol{p}_{j'}^\top\right) = \boldsymbol{p}_j^\top\boldsymbol{p}_{j'} \tag{2.11}$$

as the repeatability for operator j and the R&R between operators j and j'. Unlike the linear model discussion in Section 2.4.1, repeatability and R&R in the context of the model from Section 2.3 as defined here in Displays (2.10) and (2.11) are operator dependent, so there are distributions for repeatability and R&R across all operators or all pairs of operators. While $0 \leq (\text{R\&R})_{jj'} \leq 1$, it turns out that $1/H \leq (\text{Repeatability})_j \leq 1$. The lowest possible repeatability of $1/H$ occurs if and only if we have the "guessing distribution" that places probability $1/H$ on each ordinal category. The lowest and highest possible R&R values of 0 and 1 occur with degenerate distributions that place probability 1 on different and the same ordinal category.

In general, one might expect that the probability of a match (of 2 ordinal responses) should degrade when more noise is injected into the data collection. Thus, R&R should have a lower probability of a match than repeatability in some obvious sense. This can be shown in the context of Measures (2.10) and (2.11) because the ratio

$$0 \leq (\text{Proportion})_{jj'} = \frac{(\text{R\&R})_{jj'}^2}{(\text{Repeatability})_j \times (\text{Repeatability})_{j'}} \leq 1 \tag{2.12}$$

is guaranteed to be a proportion by applying the Cauchy-Schwarz inequality to the inner product representations on the right of Equations (2.10) and (2.11). The extremes of 0 and 1 in Inequalities (2.12) are achieved if and only if (i) the probability vectors are orthogonal $\boldsymbol{p}_j \perp \boldsymbol{p}_{j'}$, i.e., the probability distributions are completely misaligned and so mismatches can be completely attributed to a lack of reproducibility, or (ii) $\boldsymbol{p}_j = \boldsymbol{p}_{j'}$, i.e., the probability distributions are completely aligned and so mismatches can be completely attributed to a lack of repeatability. Therefore, Proportion (2.12) will be referred to as a *proportion due to repeatability*. For an interpretation of Proportion (2.12) in terms of match probabilities, refer to Table 2.3.

			Repetition k	
			1	2
Operator	j	j	Y_{ij1}	Y_{ij2}
	j'	j'	$Y_{ij'1}$	$Y_{ij'2}$

Table 2.3: A pair of responses from each of a pair of randomly selected operators on a fixed part i. The probability of equal columns is the denominator of Proportion (2.12), and the probability of equal rows is the numerator.

We now briefly revisit examples from previous sections. First, recall Figure 2.1, but now focus of its bottom row of plots. There is low $(\text{R\&R})_{1,2}$ from Display (2.11) in each pair of operators, but for different reasons. The pair on the left has low repeatability (i.e., Proportion (2.12) close to one), whereas the pair on the right has low reproducibility (i.e., Proportion (2.12) close to zero). As for a numerical example, Proportion (2.12) due to repeatability is $0.45^2/(0.66 * 0.42) \approx 0.73$ for the pair of hypothetical operators in Table 2.2.

Nominal Measures (2.10)-(2.12) do not directly reflect the ordinal nature of the underlying response Y, but can be easily adapted to do so in a manner that is consistent with the application. With this purpose in mind, let $\boldsymbol{B}$ be an $H \times H$ symmetric matrix with binary

entries and consider generalized measures of the form

$$(\boldsymbol{B}\text{ Repeatability})_j = \text{sum}\left(\boldsymbol{B}\odot\boldsymbol{p}_j\boldsymbol{p}_j^\top\right) = \text{tr}\left(\boldsymbol{p}_j\boldsymbol{p}_j^\top\boldsymbol{B}\right) = \boldsymbol{p}_j^\top\boldsymbol{B}\boldsymbol{p}_j \text{ and} \quad (2.13)$$

$$(\boldsymbol{B}\text{ R\&R})_{jj'} = \text{sum}\left(\boldsymbol{B}\odot\boldsymbol{p}_j\boldsymbol{p}_{j'}^\top\right) = \text{tr}\left(\boldsymbol{p}_j\boldsymbol{p}_{j'}^\top\boldsymbol{B}\right) = \boldsymbol{p}_j^\top\boldsymbol{B}\boldsymbol{p}_{j'}, \quad (2.14)$$

where $\odot$ represents the Hadamard product (i.e., elementwise multiplication) between matrices of the same dimension. For example, first consider metrics based on a pair of ordinal responses being off by at most m categories along the ordinal scale. These are

$$(\boldsymbol{B}_m\text{ Repeatability})_j = P(|Y_{ij1}-Y_{ij2}|\le m) = \sum_{h=1}^{H}\sum_{h'=\max\{1,h-m\}}^{\min\{H,h+m\}} p_{jh}p_{jh'} \text{ and} \quad (2.15)$$

$$(\boldsymbol{B}_m\text{ R\&R})_{jj'} = P(|Y_{ij1}-Y_{ij'1}|\le m) = \sum_{h=1}^{H}\sum_{h'=\max\{1,h-m\}}^{\min\{H,h+m\}} p_{jh}p_{j'h'}, \quad (2.16)$$

where binary matrices $\boldsymbol{B}_m$ have an entry of 1 if and only if the row and column number differ in absolute value by at most by $m\in\{0,1,\ldots,H-1\}$. If for example $H=4$, then

$$\boldsymbol{B}_0=\boldsymbol{I},\quad \boldsymbol{B}_1=\begin{bmatrix}1&1&0&0\\1&1&1&0\\0&1&1&1\\0&0&1&1\end{bmatrix},\quad \boldsymbol{B}_2=\begin{bmatrix}1&1&1&0\\1&1&1&1\\1&1&1&1\\0&1&1&1\end{bmatrix},\quad \boldsymbol{B}_3=\vec{1}\vec{1}^\top.$$

Clearly, Generalized Measures (2.13) and (2.14) with $\boldsymbol{B}=\boldsymbol{B}_m$ start at Measures (2.10) and (2.11) when $m=0$ and monotonically increase to 1 as $m\in\{0,1,\ldots,H-1\}$ increases. Looking for the smallest value of m such the generalized measures are both close to 1 summarizes the extent of the variation along the ordinal scale.

While we suggest Measures (2.15) and (2.16) with $m\in\{0,1,\ldots,H-1\}$ for general purpose use, we note Generalized Measures (2.13) and (2.14) with a customized choice for

$\boldsymbol{B}$ elicited from the application is preferred. In this regard, consider the partitioning of an $H = 5$-point ordinal scale given by

$$\boldsymbol{B}_{\text{Block}} = \left[\begin{array}{cc|ccc} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{array}\right].$$

To make the choice of $\boldsymbol{B} = \boldsymbol{B}_{\text{Block}}$ relevant, suppose ordinal categories 1 and 2 correspond to parts that can be sold, whereas ordinal categories 3-5 correspond to parts that must be retooled or scrapped altogether. Suppose further the company must supply only parts of level 1 to a customer requiring higher precision inputs, whereas the company can supply parts of levels 1 or 2 to a different customer who does not require the same level of precision. If there is low repeatability and R&R on the original 5-point ordinal scale $\boldsymbol{B} = \boldsymbol{I}$, this is necessarily only a measurement problem for the customer requiring the higher level of precision, because the other customer can still be satisfied if there is high repeatability and R&R on the coarser collapsed scale defined by $\boldsymbol{B} = \boldsymbol{B}_{\text{Block}}$.

It is also worth noting the special case of $\boldsymbol{B}$ such that $\boldsymbol{B} = \tilde{\boldsymbol{B}}^{\top}\tilde{\boldsymbol{B}}$ is nonnegative definite. In this case,

$$0 \leq \frac{(\boldsymbol{B}\ \text{R\&R})_{jj'}^2}{(\boldsymbol{B}\ \text{Repeatability})_j \times (\boldsymbol{B}\ \text{Repeatability})_{j'}} \leq 1 \tag{2.17}$$

follows by applying the Cauchy-Schwarz inequality to the quadratic forms on the right of Equations (2.13) and (2.14). Thus, the generalized measures can admit to a proportion interpretation, where Proportion (2.17) equals 1 if and only if 100% of the variability causing mismatches is due to repeatability. A sufficient condition for $\boldsymbol{B} = \tilde{\boldsymbol{B}}^{\top}\tilde{\boldsymbol{B}}$ is satisfied when $\boldsymbol{B}$ is a block matrix based on any partitioning of the ordinal categories into any number of subsets.

Table 2.4: De Mast–Van Wieringen Follow-up R&R Study ($I = 30$ Parts, $J = 3$ Raters, $K = 2$ Repeats, $H = 4$ Ordinal Categories)

	Part i														
Operator j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	3	4	2	2	4	1	2	3	3	2	2	3	1	2	2
	3	4	2	2	4	1	2	3	3	2	3	3	1	3	2
2	2	4	2	2	4	1	3	3	3	2	3	3	1	3	2
	3	4	2	2	4	1	3	3	3	3	4	3	1	3	2
3	3	4	2	2	4	1	2	3	3	2	3	3	1	2	2
	3	4	2	2	4	1	2	4	3	2	3	3	1	3	2

	Part i														
Operator j	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	4	4	4	3	3	3	1	4	4	2	2	2	2	3	2
	4	4	4	3	3	4	1	4	4	2	2	2	2	3	2
2	4	4	4	3	3	4	1	4	4	2	2	3	2	3	3
	4	4	4	3	3	4	1	4	4	2	2	3	2	3	3
3	4	4	4	3	3	4	1	4	4	2	2	3	2	3	2
	4	4	4	3	3	4	1	4	4	2	2	3	2	3	2

2.5 Demonstrations

This section focuses on bringing together the random effects Bayesian modeling from Section 2.3 and the measures for ordinal R&R from Section 2.4 in order to present a practical data analysis framework. This is done through examples. Section 2.5.1 analyzes a set of ordinal R&R data from de Mast and van Wieringen (2010), listed here in Table 3.1. This analysis is then used to investigate the operating characteristics of the Bayesian inference technique with a related simulation study in Section 2.5.2.

2.5.1 A Real Data Analysis

Given the data in Table 3.1 and the random effects model from Section 2.3 with the general use prior of Section 2.3.2, the Bayesian posterior distribution of the parameters were approximated with Markov chain Monte Carlo (MCMC) sampling. The interested reader is referred to Appendix 2.A for the necessary background information on using MCMC to carry out a Bayesian data analysis. A chain of $B = 10^4$ posterior draws was retained after an

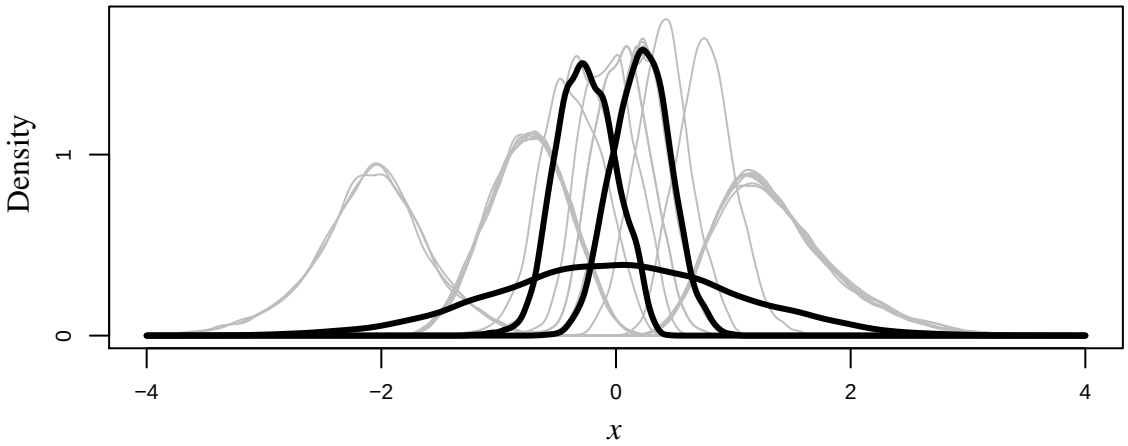


Figure 2.2: Posterior distributions of the latent variables X_i for parts $i = 1, 2, \ldots, 31$. The dark curves are for parts 29-31. Part 31 is the flatter dark curve, whereas the steeper dark curve with the lower mode is part 30.

initial burn in-period of discarding the first $5 * 10^3$ draws. Time series plots of the parameters indicated adequate mixing of the chain. This chain was initialized and generated with `JAGS` (Plummer, 2015) in 13 seconds on a 4 GHz processor. The program, given in Appendix 2.B, is surprisingly concise for such an involved modeling context.

The Bayesian approach directly quantifies the posterior uncertainty in the latent variables X_i; see Figure 2.2. The clustering of the leftmost 3 grey curves with a mode of roughly -2 correspond to the parts rated in ordinal category 1 on each repeat from each rater, i.e., parts $i = 6, 13, 22$ in Table 3.1. The minor differences in these 3 curves are due to MCMC error. There are no data for some new part $i = 31 > 30 = I$ selected at random from the broader part population, so the posterior for X_{31} (e.g., the flattest of the plotted distributions) is the standard normal by Assumption (2.2). Since the posterior distributions of each X_i is outputted by the `JAGS` program, they can be used to help obtain the posterior distributions of any part-dependent R&R measures, and this is done next for parts $i = 29, 30, 31$ highlighted by the black curves in Figure 2.2.

In this regard, Figure 2.3 displays the posterior distributions of R&R measures for parts $i = 29, 30, 31$ and operators $j = 1, 2$. All ordinal responses for part $i = 29$ in the Table 3.1 data

were category 3. As a result, the distributions in the left panel of Figure 2.3 are concentrated on proportions close to 1. The grey repeatability curves are similar for operators $j = 1, 2$. The solid dark R&R curve is focused on lower values as expected by Inequality (2.12), but only on slightly lower values. So, the Proportion (2.12) due to repeatability is especially close to 1 in this boundary case of a part with a constant response.

The story is quite different in the center panel of Figure 2.3 for part $i = 30$. This part was rated as in category 2 on each repeat by operator $i = 1$, but in category 3 for both repeats from operator $i = 2$. So, there is much posterior uncertainty in Proportion (2.12), which looks roughly like the continuous uniform distribution on the interval $(0, 1)$. This is due presumably to the low value of $K = 2$ repeats. It is hard to determine the root cause as repeatability versus reproducibility with such small sample sizes. It would, for example, be more clear that the problem was solely due to repeatability if instead the design had say $K = 20$ repeats with the data having the same pattern of constant (yet distinct) responses by operator. Most parts in the data were like part $i = 29$ with a constant response, so it may not be a surprise that the panel on the right for the R&R of a new part selected at random looks more like the panel on the left. Do, however, notice the higher level of posterior uncertainty for a new part given by longer tails in the skewed-left distributions in the panel on the right when compared to the panel on the left.

Next, the concept of Figure 2.3 and the use of the R&R measures on individual parts is used to define some informative, aggregate measures across all parts. This is done with a pair of considerations in mind. First, we wanted to smooth over the volatility in the part-to-part posterior uncertainty (due to small K) in order to pick up on the general pattern across all parts. Second, we wanted these aggregate measures to be directly related to some easy-to-understand statistics of the actual Table 3.1 data in terms of the conceptual understanding of repeatability and reproducibility laid out in Section 2.4. These aggregate measures (across

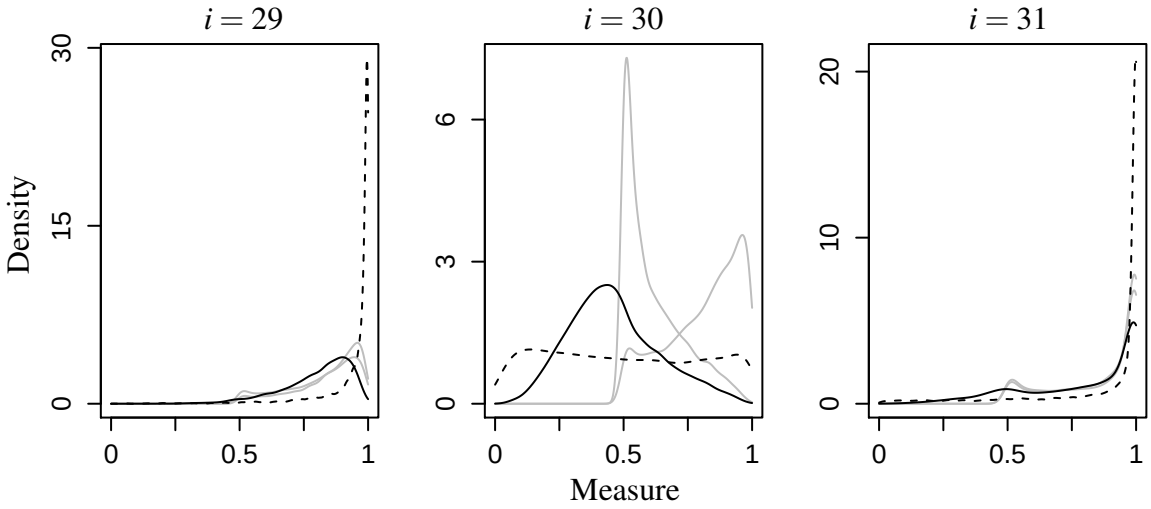


Figure 2.3: Posterior distributions of R&R measures for operators $j = 1,2$ on parts $i = 29$ (left), 30 (middle), and 31 (right). The grey curves are $(\text{Repeatability})_j$ with $j = 1,2$. The solid and dashed black curves are $(\text{R\&R})_{1,2}$ and the Proportion (2.12) due to repeatability, respectively.

all parts $i = 1,\ldots,I$) are

$$\overline{(\text{Repeatability})}_j = \sum_{i=1}^{I} (\text{Repeatability})_j / I \tag{2.18}$$

$$\overline{(\text{R\&R})}_{jj'} = \sum_{i=1}^{I} (\text{R\&R})_{jj'} / I \tag{2.19}$$

$$\overline{(\text{Proportion})}_{jj'} = \sum_{i=1}^{I} \frac{(\text{R\&R})^2_{jj'}}{(\text{Repeatability})_j \times (\text{Repeatability})_{j'}} / I. \tag{2.20}$$

Although not reflected in this notation, it was previously emphasized in Section 2.4 that summands on the right of Measures (2.18)-(2.20) do depend on the part i.

Related basic statistics are the sample proportions

$$\widehat{\overline{(\text{Repeatability})}}_j = \frac{\sum_{i=1}^{I} \sum_{1 \le k < k' \le K} \mathscr{I}_{\{Y_{ijk} = Y_{ijk'}\}}}{I * \binom{K}{2}} \tag{2.21}$$

$$\widehat{\overline{(\text{R\&R})}}_{jj'} = \frac{\sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{k'=1}^{K} \mathscr{I}_{\{Y_{ijk} = Y_{ij'k'}\}}}{I * K^2} \tag{2.22}$$

of matching responses on a given part, where $\mathscr{I}_{\{\cdot\}}$ is the binary indicator variable. The

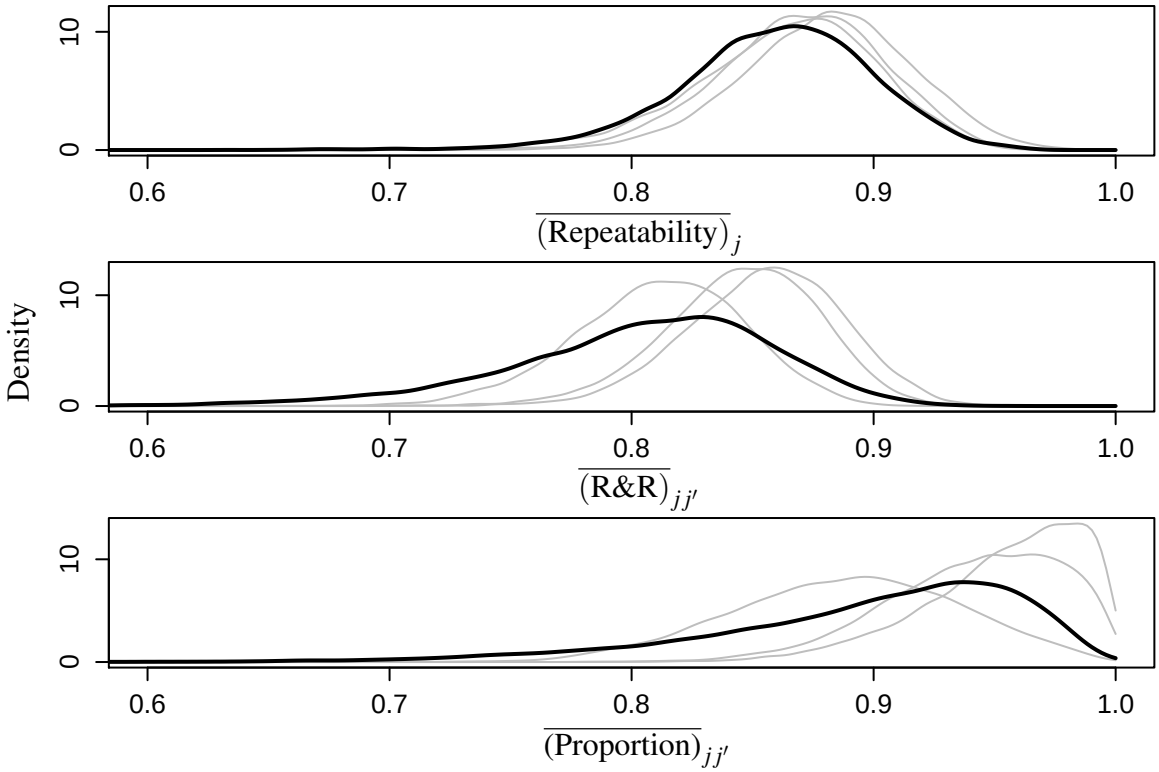


Figure 2.4: Posterior distributions of R&R measures averaged over parts $i = 1,2,\ldots,30$: repeatability (top), R&R (middle), and Proportion (2.20) due to repeatability (bottom). The grey curves are for the operators $j = 1,2,3$ and $j < j' = 2,3$, whereas the black curves are the posterior predictive distributions for new operators $j = 4$ and $j' = 5$.

color coding in Table 3.1 helps quickly see that Statistics (2.21) are $27/30, 27/30, 28/30$ for operators $j = 1,2,3$, and this is closely reflected with posterior uncertainty by the 3 grey curves in the top panel of Figure 2.4. The related posterior predictive distribution for a new operator $i = 4$ on the same $I = 30$ parts is given by the dark curve and is flatter as expected to reflect a reduction in certainty during prediction of a new operator.

To three decimal places, Statistics (2.22) are $0.808, 0.900, 0.850$ for the pairs of operators $(j, j') = (1,2),(1,3),(2,3)$. So, the data suggest that operators $(j, j') = (1,2)$ are a bit more misaligned than the other pairs. This also shows up in the Bayesian analysis in the middle panel of Figure 2.4. The leftmost grey curve corresponds to operators $(j, j') = (1,2)$, although there is uncertainty (i.e., overlapping posteriors). Again, as expected the flattest curve is the dark one corresponding to a pair of new, randomly selected operators $(j, j') = (4,5)$.

The order of operations for Measure (2.20) is of special note. Proportions (2.12) are first computed by part and then these are averaged because computing Measures (2.18) and (2.19) first followed by taking the ratio does not always result in a proportion. Thus, unlike Statistics (2.21) and (2.22), there is no simple estimate for $\overline{(\text{Proportion})}_{jj'}$ due to divide by zero issues. There is no such problem in the Bayesian modeling framework. The leftmost grey curve in the bottom panel of Figure 2.4 corresponds to the pair of operators $(j, j') = (1, 2)$ and indicates reproducibility as being in play more with the discrepancy between this pair of operators. Prediction of Proportion (2.20) due repeatability for a new pair of operators again has the most posterior uncertainty.

As previously mentioned in Section 2.3.2, a sensitivity analysis based on increasing the prior uncertainty by an order of magnitude was conducted to demonstrate the robustness of this data analysis to the general use prior. This looser prior is expected to shrink MLEs less, and the Bayesian analysis of the Table 3.1 data was rerun with this second prior. The resulting versions of Figure 2.2-2.4 based on this second analysis were effectively the same. Thus, the ease of the Bayesian implementation through the concise, numerically stable, and fast `JAGS` code in Appendix 2.B is justified for the analysis of the Table 3.1 data. This analysis also demonstrated some novel inferences of interest in the nominal extreme developed in Section 2.4.3. A quick skim of Table 3.1 shows that pretty much all misalignments on a fixed part are off by $m = \pm 1$ category along the ordinal scale, so it may come as no surprise that producing figures analogous to Figures 2.3 and 2.4 based on $\boldsymbol{B}_m$ repeatability and R&R from Displays (2.15) and (2.16) with $m = 1$ results in all posterior densities piling up mass very close to one, suggesting problems in neither repeatability nor reproducibility if we were willing to accept "being within a category" as "close enough."

2.5.2 A Simulation Study

Simulation can be used to investigate the operating characteristics of an inference procedure (Bayesian or frequentist). For a Bayes procedure, this might be done to validate that the numerical stability due to the prior does not unduly over-shrink the estimates and affect

performance. In this regard, an interesting simulation study is conducted to provide such validation and to assist in a discussion of ordinal R&R studies from the design perspective of picking I, J, K. This simulation study assumes a population of raters with $\mu_\alpha = 2.6$, $\sigma_\alpha = 0.2$, and $\boldsymbol{\lambda} = (11, 44, 29, 40)$. This choice makes this study directly related to the data analysis and application from Section 2.5.1 because $\mu_\alpha = 2.6$, $\sigma_\alpha = 0.2$, and $\boldsymbol{\lambda} = (11, 44, 29, 40)$ are in fact the posterior medians, given the real data in Table 3.1.

To further constrain the design problem, assume a budget (of time or money) that only allows for the collection of $I * J * K = 180$ ordinal responses. For simplicity, designs are further fixed to $K = 2$ repeats to focus on the tradeoff of picking: more parts and fewer operators versus fewer parts with more operators. Since 180 only has 2 as a factor with multiplicity 2, designs with $J = 1, 2, 3, 5$ operators are compared. For each design, 10 data sets were generated, and all 40 data sets were generated in a fraction of a second. The `JAGS` code was then used to obtain a posterior MCMC sample given each simulated data set. These 40 MCMC chains finished in 533 seconds on a 4 GHz processor, and the needed computer time was independent of the design.

Estimation of broader rater population parameters $\mu_\alpha, \sigma_\alpha, \boldsymbol{\lambda}$ is possible because of the Section 2.3 random effects model extension. These parameters are related prediction of a new operator, and so they provide reasonable metrics to optimize the tradeoff of interest: more accurately estimate the parameters of fewer operators with more parts per operator versus investigate more operators with less precision on each operator due to fewer parts.

Bayesian posterior 95% credible intervals (see Appendix 2.A) were used to quantify this tradeoff. It is known that such intervals when based on non-informative priors often hold a frequentist coverage probability close to 0.95, and this occurred in this simulation with our not-strongly-informative, general-use prior from Section 2.3.2. Observed coverage rates for parameters $\mu_\alpha, \sigma_\alpha, \lambda_1, \lambda_2, \lambda_3, \lambda_4$ were 1,1,0.975,0.9,0.95,0.95 over the 40 simulated data sets. By design $J = 1, 2, 3, 5$, there were only 10 generated data sets, and only parameter λ_1 with design $J = 5$ had the lowest observed coverage of $8/10$, so coverage rates appeared roughly independent of the design.

Because of the solid/constant coverage performance in the previous paragraph, it is thus

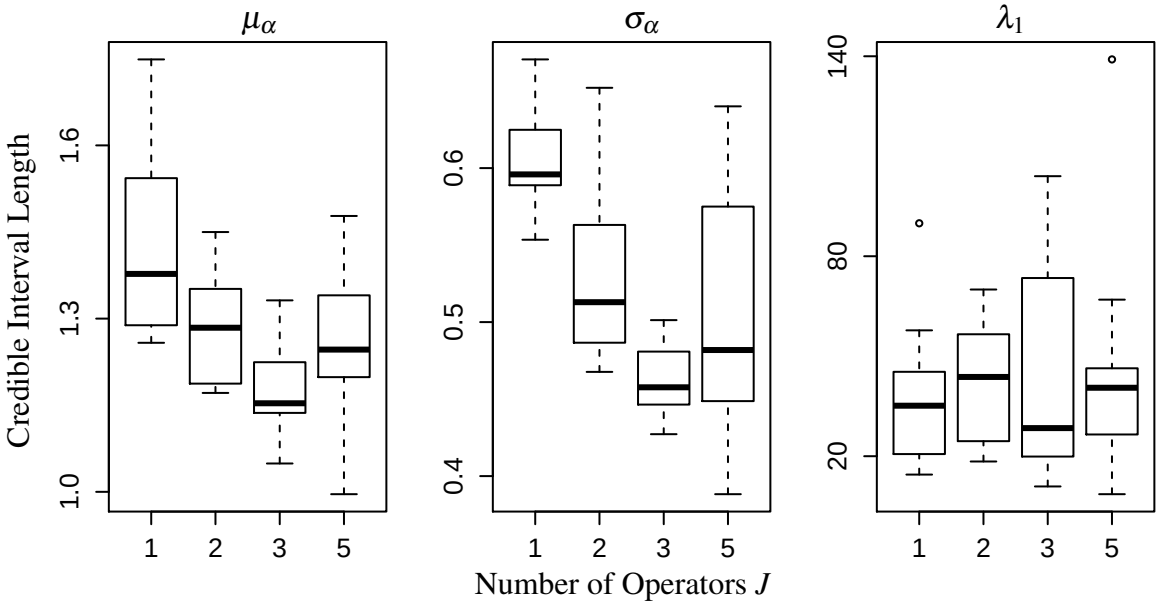


Figure 2.5: Sampling distribution of lengths of 95% Bayesian posterior credible intervals for μ_α (left), σ_α (middle), and λ_1 (right). Designs had $I * J * K = 180$ responses with $K = 2$ repeats. The number of parts I and operators J were varied. Each boxplot is based on 10 simulated data sets from a population of raters with $\mu_\alpha = 2.6$, $\sigma_\alpha = 0.2$, and $\boldsymbol{\lambda} = (11, 44, 29, 40)$.

sensible to compare designs $J = 1, 2, 3, 5$ based on the lengths of the 95% credible intervals. Figure 2.5 displays the results. (The results for parameters $\lambda_2, \lambda_3, \lambda_4$ were not included because they were similar to those for λ_1.) The pattern is clear if you simply compare the medians. The actual design used by de Mast and van Wieringen (2010) with $J = 3$ seems optimal in terms of increasing precision by reducing credible interval length.

2.6 Discussion

The De Mast–Van Wiergin model was extended to a population of raters by specifying distributions for the raters' scaling and cut point parameters. A non-trivial aspect of this modeling was that of defining a distribution over ordered cut points; a transformation of a Dirichlet distribution was used to achieve this endpoint. A Bayesian framework, which treated the latent variable for each part as an additional parameter, facilitated inference with a remarkably concise `JAGS` program. In the example data sets we fit, this program was numerically stable

and had short run times. An additional benefit of the Bayesian paradigm in general is that uncertainty for any parametric function is easily obtained given a posterior MCMC sample.

The Bayesian framework did not, however, remove the complication in an ordinal R&R context of defining parametric functions that accurately convey the meaning of R&R and decompose the distribution of an ordinal response into these two fundamental sources in some meaningful sense. For some initial guidance on this challenge, gauge R&R (with a continuous response and a linear model) helped define measures of ordinal R&R in a numerical extreme. If a subject matter expert is willing to assign numbers to each category, then the variance of the response can be decomposed into percentages due to repeatability and reproducibility. With this background in mind, measures of ordinal R&R were defined through a nominal extreme using match probabilities of two responses on the same part, and the Cauchy-Schwarz inequality was used to define a proportion measure for a pair of operators on a given part that is 0 or 1 if and only if the probability of a match is 0 or the operators have the exact same distribution. Thus, the extreme values for this proportion measure of 0 and 1 suggest that the root cause of mismatches in the ordinal response is due solely to reproducibility or repeatability, respectively.

With the ordinal metrics in hand, the Bayesian method was tested on real and simulated data sets to assess the quality of the estimation procedure and to showcase the value added by these new metrics. The estimation procedure was demonstrated to be robust to prior specification for a real data analysis and was demonstrated to have strong frequentist properties in a related simulation context. The simulations were also used to study ordinal R&R from a design of experiments perspective. One might imagine a fixed budget on the number of responses and want to pick an optimal design in terms of the number of parts, raters, and repeats. Minimizing posterior uncertainty in the parameters for the population of raters provided a natural design of experiments objective and was made possible because of the unified random effects modeling extension from this work. In addition, the extended modeling framework made a number of novel inferences possible. These included inference on a particular part involved in the R&R study or selected at random from the broader part population and inference on a rater or pair of raters from the R&R study or selected at

random from the broader rater population.

2.A Bayesian Data Analysis

Here we briefly summarize the Bayesian approach. A Bayesian analysis combines prior information with observed data $\mathbf{y}$ to produce a posterior distribution for the parameters $\boldsymbol{\theta}$ using Bayes' theorem that takes the form

$$\pi(\boldsymbol{\theta}|\mathbf{y}) = \frac{L(\mathbf{y}|\boldsymbol{\theta})\,\pi(\boldsymbol{\theta})}{\int_{\Theta} L(\mathbf{y}|\boldsymbol{\theta})\,\pi(\boldsymbol{\theta})\,d\boldsymbol{\theta}}, \tag{2.23}$$

where $\boldsymbol{\Theta}$ denotes the range of values for the parameters $\boldsymbol{\theta}$. In the context of the Section 2.3 model,

$$\boldsymbol{\theta} = (\boldsymbol{\delta}, \boldsymbol{\alpha}, \boldsymbol{\lambda}, \mu_\alpha, \sigma_\alpha, X_1, \ldots, X_I), \tag{2.24}$$

and the unobserved latent variables X_i as well as the random effects $\boldsymbol{\delta}, \boldsymbol{\alpha}$ are simply treated as additional parameters from the Bayesian perspective.

The likelihood function denoted by $L(\mathbf{y}|\boldsymbol{\theta})$ describes the probability mass function of $\mathbf{y}$ given the model parameters $\boldsymbol{\theta}$, where $\mathbf{y}$ denotes the vector of observed data $\mathbf{y}$ (here, the counts r_{ijh}). The available information about $\boldsymbol{\theta}$ is initially summarized by the prior distribution $\pi(\boldsymbol{\theta})$.

Bayes' Theorem (2.23) shows how the data and prior information are combined to obtain the posterior distribution of $\boldsymbol{\theta}$ denoted by $\pi(\boldsymbol{\theta}\,|\,\mathbf{y})$. In many applications, an analytical expression for the integral in Equation (2.23) does not exist. Instead, Markov chain Monte Carlo (MCMC) is used to simulate samples $\{\boldsymbol{\theta}^{(b)}, b = 1, \ldots, B\}$ from the posterior distribution $\pi(\boldsymbol{\theta}\,|\,\mathbf{y})$. Such samples with large B (say $B = 10^4$) can be used to accurately reflect the posterior distribution for any parametric function $g(\boldsymbol{\theta})$ of interest. One can simply make a histogram (or kernel density estimate) of the posterior draws $g\left(\boldsymbol{\theta}^{(1)}\right), \ldots, g\left(\boldsymbol{\theta}^{(B)}\right)$ or for example compute the 0.025 and 0.975 quantiles to obtained an equal-tailed 95% posterior credible interval for $g(\boldsymbol{\theta})$. See Casella and George (1992), Chib and Greenberg (1995) and Gelman et al. (2013) for discussions of popular MCMC algorithms.

A parametric function studied in this manner back in Section 2.5 was

$$g(\boldsymbol{\theta}) = \sum_{i=1}^{I} (\text{Repeatability})_j / I$$

built from Equation (2.10) with $j \leq J = 3$ (see the grey curves in the top panel of Figure 2.4). Another example is this same parametric function, but for a new operator $j > J = 3$. The related black curve in the top panel of Figure 2.4, however, requires the concept of the *posterior predictive distribution*. There were no data directly obtained from operator $j = 4$, so his/her parameters are not directly listed in Vector (2.24). In spite of this, predictions for operator $j = 4$ are still directly possible with our random effects model. Operators $j = 1, 2, 3$ are used to estimate the parameters $\mu_\alpha, \tau_\alpha, \boldsymbol{\lambda}$ of the rater population and in turn predict plausible outcomes for operator $j = 4$ as follows. Given $\mu_\alpha^{(b)}, \tau_\alpha^{(b)}$, the posterior predictive distribution for $\alpha_4^{(b)}$ is simulated with independent draws from Distributions (2.4) for each $b = 1, \ldots, B$. Similarly, the posterior predictive distribution for a new operator's cut points is simulated by using the $\boldsymbol{\lambda}^{(b)}$ with Displays (2.5) and (2.6).

2.B JAGS Code

The Bayesian random effects model from Section 2.3 is specified by the `JAGS` code given below in this appendix. This code was used to obtain the posterior samples in Section 2.5 to demonstrate the effectiveness of this approach. This code was called from `R` (R Core Team, 2016) with `RJAGS` (Plummer, 2016) so that posterior samples were readily available in `R` to facilitate our Bayesian data analyses and for example construct Figures 2.2-2.5 with the `density` function in `R`. Parameter nomenclature lines up in a straightforward manner, e.g., `alpha[j]` in the code is α_j back in the mathematical presentation of Section 2.3. A Bayesian fixed effects version of the model from Section 2.3 results if the three lines of code marked `#PRIOR` are removed, so the effort required to extend the Bayesian paradigm from fixed to random effects makes this approach quite advantageous. As previously mentioned in Section 2.3, log normal densities represented by function `dlnorm`(μ, τ) in the code were

parameterized in terms of the mean μ and precision $\tau = 1/\sigma^2$ on the log scale. See the JAGS manual for more details concerning syntax.

```
model{
 for(j in 1:J){
  alpha[j]~dlnorm(mu.alpha,tau.alpha)
  pi[j,1:H]~ddirch(lambda)
  for(h in 1:(H-1)){delta[j,h]<-qnorm(sum(pi[j,1:h]),0,1)}
 }
 for(i in 1:I){
  X[i]~dnorm(0,1)
  for(j in 1:J){
   p[i,j,1]<-1
   for(h in 2:H){p[i,j,h]<-exp(sum(alpha[j]*
                              (X[i]-delta[j,1:(h-1)])))}
   R[i,j,1:H]~dmulti(p[i,j,1:H]/sum(p[i,j,1:H]),K)
  }
 }
 mu.alpha ~dnorm( mu.mu.alpha, tau.mu.alpha)  #PRIOR
 tau.alpha~dlnorm(mu.tau.alpha,tau.tau.alpha) #PRIOR
 for(h in 1:H){lambda[h]~dlnorm(mu.lambda,tau.lambda)} #PRIOR
}
```

CHAPTER 3

LEARNING WITH REPRODUCING KERNEL HILBERT SPACES

Prediction is a fundamental practical problem in (statistical) machine learning. This often involves a large number of feature or predictor variables for some response of interest and a fairly large number of cases upon which to build a *prediction rule*, i.e., a function of the predictors used to approximate the response.

Let $\boldsymbol{x}_i \in \mathbb{R}^p$ be the feature vector and $y_i \in \mathbb{R}$ be the response for observation $i = 1, \ldots, n$. These cases can be concisely represented in a matrix form. In this regard, let $\boldsymbol{X}$ be the $n \times p$ matrix of feature data, which stacks the $\boldsymbol{x}_i$ as its rows and $\boldsymbol{y} = (y_i, ..., y_n)^\top$ be the response vector. The object under this *supervised* setup is to create an effective prediction rule, i.e., estimate a function $f : \mathbb{R}^p \to \mathbb{R}$ to approximate an arbitrary response y_0 corresponding to feature vector $\boldsymbol{x}_0$. In practice, mathematical frameworks for defining and computing learners are often based on algorithms or optimization problems.

In general, *supervised learning* involves the use of *n complete* cases or observations, possibly organized into feature matrix $\boldsymbol{X}$ and response vector $\boldsymbol{y}$ to compute an estimate of f. On the other hand, *semi-supervised learning* involves situations were the full feature matrix $\boldsymbol{X}$

is available, but some proper subset of the components of the response vector $\boldsymbol{y}$ are missing. A concept for semi-supervised learning is to have the data determine if the additional information contained in the feature observations $\boldsymbol{x}_i$ corresponding to missing responses $y_i = \text{NA}$ can lead to an improved estimate for f. To begin, this Chapter investigates prediction rules from a supervised perspective before proposing semi-supervised generalizations in Chapter 4. The focus of this Chapter will involve an optimization problem based on Reproducing Kernel Hilbert Spaces (RKHS).

3.1 Euclidean Space Prediction Rules

In regression with a continuous response, one might use the all-purpose square error loss function

$$L(y_0, f(\boldsymbol{x}_0)) = (y_0 - f(\boldsymbol{x}_0))^2$$

to help define the sought after function f. In this context, the conditional expected value of $y_0|\boldsymbol{x}_0$ minimizes the expected loss or risk, i.e.,

$$f(\boldsymbol{x}_0) = \mathbb{E}[y_0|\boldsymbol{x}_0] = \underset{\text{Functions } \tilde{f}:\mathbb{R}^p \to \mathbb{R}}{\arg\min} \mathbb{E}[L(y_0, \tilde{f}(\boldsymbol{x}_0))].$$

Then data analysis amounts to using $\boldsymbol{X}$ and $\boldsymbol{y}$ to compute an estimate $\hat{f}$ of $\mathbb{E}[y_0|\boldsymbol{x}_0]$, but how exactly one proceeds might be premised on some model-based assumptions.

For example, take the classical linear regression model with $\mathbb{E}[\boldsymbol{y}|\boldsymbol{X}] = \boldsymbol{X}\boldsymbol{\beta}$ with heteroscedastic error terms $\mathbb{V}\text{ar}(\boldsymbol{y}|\boldsymbol{X}) = \sigma^2\boldsymbol{I}$. An estimate for the required function $f(\boldsymbol{x}_0) = \boldsymbol{x}_0^\top\beta$ might be based on the concept of least squares

$$\widehat{\boldsymbol{\beta}}^{(\text{LS})} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min} (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \tag{3.1}$$

to produce the estimate $\hat{f}(\boldsymbol{x}_0) = \boldsymbol{x}_0^\top \widehat{\boldsymbol{\beta}}^{(\text{LS})}$

A method to solve Optimization (3.1) is to simply solve the $\boldsymbol{\beta}$-score of the objective

function (i.e., take the derivative of the objective function with respect to $\boldsymbol{\beta}$, set it equal to a vector of zeros, and solve). If $\boldsymbol{X}^\top\boldsymbol{X}$ is nonsingular, then the unique solution is given by

$$\widehat{\boldsymbol{\beta}}^{(\mathrm{LS})} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$$

and motivates the well-known prediction rule $\hat{f}(\boldsymbol{x}_0) = \boldsymbol{x}_0\widehat{\boldsymbol{\beta}}^{(\mathrm{LS})}$. Let $\mathrm{rank}(\boldsymbol{X}) = r$. Then $\boldsymbol{X}$ has a singular value decomposition

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top,$$

where the $n \times n$ matrix $\boldsymbol{U}$ has orthonormal columns spanning the column space of $\boldsymbol{X}$ (denoted by $\mathscr{C}(\boldsymbol{X})$), the $p \times p$ matrix $\boldsymbol{V}$ has orthonormal columns spanning columns of $\mathscr{C}(\boldsymbol{X}^\top)$, and the $n \times p$ rectangular diagonal matrix $\boldsymbol{D} = [\mathrm{diag}(d_1, d_2, ..., d_p)|\boldsymbol{0}]$ where $d_1 \geq d_2 \geq \cdots \geq d_r > d_{r+1} = \cdots = d_p = 0$. The d_j are the square roots of eigenvalues of $\boldsymbol{X}^\top\boldsymbol{X}$. This decomposition is useful for projecting the response vector $\boldsymbol{y}$ onto $\mathscr{C}(\boldsymbol{X})$ as

$$\widehat{\boldsymbol{f}}^{(\mathrm{OLS})} = \widehat{\boldsymbol{f}}(\boldsymbol{X}) = \begin{pmatrix} \hat{f}(\boldsymbol{x}_1) \\ \vdots \\ \hat{f}(\boldsymbol{x}_n) \end{pmatrix} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top(\boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top)^{-1}\boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^\top\boldsymbol{y} = \boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{y}.$$

For the linear inner product, we have

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle \equiv \sum_{i=1}^{n} \boldsymbol{u}_i\boldsymbol{v}_i = \boldsymbol{u}^\top\boldsymbol{v},$$

so the least squares prediction rule in the Euclidean space $\mathbb{R}^n$ is given by

$$\widehat{\boldsymbol{f}}^{(\mathrm{OLS})} = \boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{y} = \sum_{j=1}^{r} \langle \boldsymbol{u}_j, \boldsymbol{y} \rangle \boldsymbol{u}_j,$$

where $\boldsymbol{u}_j$ is the jth column of $\boldsymbol{U}$. This presentation leads into the Hilbert space generalization to come in Section 3.2.

A shortcoming of least squares estimation under the linear regression model arises when there is so-called multicollinearity, i.e., the feature variables (or columns of $\boldsymbol{X}$) are correlated. When there is multicollinearity, the matrix $\boldsymbol{X}^\top\boldsymbol{X}$ may be close to singular matrix, and as a result, the least-squares estimates becomes highly sensitive to random errors in the observed responses. One way out of this situation is ridge regression, which stabilizes the estimated regression coefficients by shrinking them towards zero. The ridge regression optimization is

$$\widehat{\boldsymbol{\beta}}^{(\text{Ridge})} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\,(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})+\lambda\boldsymbol{\beta}^\top\boldsymbol{\beta}, \tag{3.2}$$

where $\lambda \geq 0$ is a tuning parameter that controls the strength of the shrinking. When $\lambda = 0$, Optimization Problem (3.2) equals Linear Regression (3.1), whereas when $\lambda \to \infty$, an estimate of $\widehat{\boldsymbol{\beta}}^{(\text{Ridge})} = \vec{0}$ results. Compromise values of $\lambda \in (0,\infty)$ balance (i) fitting a linear model of $\boldsymbol{y}$ on $\boldsymbol{X}$ with (ii) coefficient shrinking as seen in the closed-formula

$$\widehat{\boldsymbol{\beta}}^{(\text{Ridge})} = \left(\boldsymbol{X}^\top\boldsymbol{X}+\lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}^\top\boldsymbol{y}.$$

The vector of fits under this prediction rule is $\widehat{\boldsymbol{f}} = \boldsymbol{X}\widehat{\boldsymbol{\beta}}^{(\text{Ridge})} = \boldsymbol{X}\left(\boldsymbol{X}^\top\boldsymbol{X}+\lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}^\top\boldsymbol{y}$, and with the use of the singular value decomposition of $\boldsymbol{X}$, we get

$$\begin{aligned}
\widehat{\boldsymbol{f}}^{(\text{Ridge})} &= \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top\left(\boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top+\lambda I\right)^{-1}\boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^\top\boldsymbol{y} \\
&= \boldsymbol{U}\boldsymbol{D}\left(\boldsymbol{V}^\top\left(\boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top+\lambda I\right)\boldsymbol{V}\right)^{-1}\boldsymbol{D}\boldsymbol{U}^\top\boldsymbol{y} \\
&= \boldsymbol{U}\boldsymbol{D}\left(\boldsymbol{D}^2+\lambda\boldsymbol{I}\right)^{-1}\boldsymbol{D}\boldsymbol{U}^\top\boldsymbol{y} \\
&= \sum_{j=1}^{r}\left(\frac{d_j^2}{d_j^2+\lambda}\right)\langle\boldsymbol{u}_j,\boldsymbol{y}\rangle\boldsymbol{u}_j.
\end{aligned} \tag{3.3}$$

Because $0 < \frac{d_{j+1}^2}{d_{j+1}^2+\lambda} \leq \frac{d_j^2}{d_j^2+\lambda} < 1$, the coefficients of the orthonormal basis vectors $\boldsymbol{u}_j$ used to decompose $\widehat{\boldsymbol{f}}^{(\text{Ridge})}$ are a shrunken version of the coefficients of $\widehat{\boldsymbol{f}}^{(\text{OLS})}$, and the most severe shrinking is enforced along the lower order principal components of $\boldsymbol{X}$.

A way of moving beyond the linear model assumption of $\boldsymbol{f} = \boldsymbol{X}\boldsymbol{\beta}$ is to transform to

feature variables and then a use a linear model in this new space of derived feature data say $h(\boldsymbol{x}_i)^\top = (h_1(\boldsymbol{x}_i), h_2(\boldsymbol{x}_i), ..., h_p(\boldsymbol{x}_i))$. This results in a prediction rule of the form

$$\hat{f}(\boldsymbol{x}_0) = \sum_{j=1}^{p} \hat{\beta}_j h_j(\boldsymbol{x}_0) = h(\boldsymbol{x}_0)^\top \widehat{\boldsymbol{\beta}}.$$

Instead of restricting the set of functions to a minimizer of an empirical loss function (like OLS), many techniques (like ridge regression) are motivated by adding a penalty term to the objective function to be minimized. Let $J(f) \geq 0$ be a term penalizing the "roughness" of the function f. This concept of penalty in some contexts simplifies to a finite data penalty $J_n(\boldsymbol{f})$ on the vector $\boldsymbol{f}$ of n function evaluations $\boldsymbol{f}_i = f(\boldsymbol{x}_i)$ for $i = 1, \ldots, n$. A generic version of this latter option is

$$\widehat{\boldsymbol{f}} = \underset{\boldsymbol{f} \in \mathbb{R}^n}{\arg\min} \, (\boldsymbol{y} - \boldsymbol{f})^\top (\boldsymbol{y} - \boldsymbol{f}) + J_n(\boldsymbol{f}), \tag{3.4}$$

whereas an example of the former in the context of $p = 1$ is smoothing splines

$$f_\lambda(x) = \underset{f \text{ with 2 derivatives}}{\arg\min} \, (\boldsymbol{y} - \boldsymbol{f})^\top (\boldsymbol{y} - \boldsymbol{f}) + \lambda \int_a^b (f''(x))^2 dx \tag{3.5}$$

with a function-based penalty term $J(f) = \lambda \int_a^b (f''(x))^2 dx$ and smoothing parameter $\lambda > 0$.

The solution to Optimization (3.5) is known to be a natural cubic spline

$$f_\lambda(x) = \sum_{j=1}^{n} \beta_j N_j(x)$$

with a second derivative of $f''(x) = \sum_{j=1}^{n} \beta_j N_j''(x)$, and so

$$(f''(x))^2 = \sum_{j=1}^{n} \sum_{l=1}^{n} \beta_j \beta_l N_j''(x) N_l''(x), \tag{3.6}$$

which is just a quadratic form written in summation notation. With a goal of representing the integral of Quadratic Form (3.6) in a matrix representation, let $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_n)^\top$ and $\boldsymbol{H} = [N_j(x_i))]$ and $\boldsymbol{\Omega} = [\int_a^b N_i''(t) N_i''(t) dt]$ be $n \times n$ matrices with $i = 1, \ldots, n$ and $j = 1, \ldots, n$

indexing their rows and columns. Thus, smoothing splines defined by Optimization (3.5) have the finite data representation

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^n}{\arg\min}\,(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\beta})^\top(\boldsymbol{y}-\boldsymbol{H}\boldsymbol{\beta})+\lambda\boldsymbol{\beta}^\top\boldsymbol{\Omega}\boldsymbol{\beta} \tag{3.7}$$

where penalty $J(f)$ on the function f has the finite data representation $J_n(\boldsymbol{f})=\lambda\boldsymbol{\beta}^\top\boldsymbol{\Omega}\boldsymbol{\beta}$ on the function evaluations $\boldsymbol{f}$. The optimal solution of

$$\widehat{\boldsymbol{\beta}}=(\boldsymbol{H}^\top\boldsymbol{H}+\lambda\boldsymbol{\Omega})^{-1}\boldsymbol{H}^\top\boldsymbol{y}$$

to Optimization (3.7) and its corresponding prediction rule of

$$\widehat{\boldsymbol{f}}_\lambda=\boldsymbol{H}(\boldsymbol{H}^\top\boldsymbol{H}+\lambda\boldsymbol{\Omega})^{-1}\boldsymbol{H}^\top\boldsymbol{y}$$

follow by the method used to derive Ridge Regression (3.2). The $n\times n$ nonnegative definite matrix $\boldsymbol{S}_\lambda=\boldsymbol{H}(\boldsymbol{H}^\top\boldsymbol{H}+\lambda\boldsymbol{\Omega})^{-1}\boldsymbol{H}^\top$ is often called a *smoother matrix*.

In general, suppose $J_n(\boldsymbol{f})=\boldsymbol{f}^\top\boldsymbol{K}\boldsymbol{f}$, where $\boldsymbol{K}$ is a known $n\times n$ nonnegative definite penalty matrix. Then the $n\times n$ nonnegative definite smoother $\boldsymbol{S}_\lambda=(\boldsymbol{I}+\lambda\boldsymbol{K})^{-1}$ solves

$$\widehat{\boldsymbol{f}}_\lambda=\boldsymbol{S}_\lambda\boldsymbol{y}=\underset{\boldsymbol{f}}{\arg\min}\,(\boldsymbol{y}-\boldsymbol{f})^\top(\boldsymbol{y}-\boldsymbol{f})+\lambda\boldsymbol{f}^\top\boldsymbol{K}\boldsymbol{f}. \tag{3.8}$$

This symmetric smoother $\boldsymbol{S}_\lambda$ also has a spectral decomposition

$$\boldsymbol{S}_\lambda=\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top=\sum_{j=1}^n d_j\boldsymbol{U}_j\boldsymbol{U}_j^\top,$$

where $d_j\geq 0$ are the eigenvalues with corresponding eigenvectors $\boldsymbol{U}_j$ as the columns of $\boldsymbol{U}$. From this, the representation

$$\widehat{\boldsymbol{f}}_\lambda=\boldsymbol{S}_\lambda\boldsymbol{y}=\left(\sum_{j=1}^n d_j\boldsymbol{U}_j\boldsymbol{U}_j^\top\right)\boldsymbol{y}=\sum_{j=1}^n d_j\langle\boldsymbol{U}_j,\boldsymbol{y}\rangle\boldsymbol{U}_j,$$

makes it clear that the resulting prediction rule satisfies $\widehat{\boldsymbol{f}}_\lambda \in \mathscr{C}(\boldsymbol{U})$.

3.2 Reproducing Kernel Hilbert Spaces with Splines

The Reproducing Kernel Hilbert Space (RKHS) concept is illustrated by extending the smoothing spline case into the more general RKHS function space paradigm when $p = 1$. The underlying rationale for Hilbert space construction is to enforce the notion of smoothness through penalization of functions. This was observed with the smoothing spline in Optimization (3.5) by restricting the function space under examination to twice differentiable functions. This was equivalent to Optimization (3.4), which penalized the function evaluations $\boldsymbol{f} \in \mathbb{R}^n$. The conventional wisdom is that optimization over the Euclidean space is 'overburdensome' due to many bad choices for the sought after function f while the restriction to twice differentiable functions leads to a more desirable solution set. Precisely, we aim to optimally choose an $f \in \mathscr{H}$, where

$$\mathscr{H} = \left\{ h : [a,b] \to \mathbb{R} : h \text{ and } h' \text{ are absolutely continuous and } \int_a^b (h''(x))^2 dx < \infty \right\}.$$

Optimization (3.5) is to be extended to operate on functions from space $\mathscr{H}$.

The Euclidean optimization problem for smoothing splines operated on a vector space with the ℓ_2-norm as its inner product. In order to optimize over space $\mathscr{H}$, an inner product on functions must be constructed. Let $h_1, h_2, h_3 \in \mathscr{H}$, and define constants $a, b \in \mathbb{R}$. In general, an inner product $\langle \cdot, \cdot \rangle$ must satisfy the following 3 properties.

1. **Symmetry:** $\langle h_1, h_2 \rangle = \langle h_2, h_1 \rangle$.
2. **Linearity:** $\langle a * h_1 + b * h_2, h_3 \rangle = a * \langle h_1, h_3 \rangle + b * \langle h_2, h_3 \rangle$.
3. **Nonnegative Definiteness:** $\langle h_1, h_2 \rangle \geq 0$ and $\langle h_1, h_2 \rangle = 0 \Leftrightarrow h_1 = h_2$.

A quick analysis of Optimization (3.5) might initially suggest $\langle h_1, h_2 \rangle_1 = \int_a^b h_1''(x) h_2''(x) dx$ as a good candidate for an inner product on functions, but this is not valid because it does not

satisfy the nonnegative definite criterion, e.g., $\langle x, 2x\rangle_1 = 0$ while $x \neq 2x$. So, the following well-defined inner-product adjusts $\langle \cdot, \cdot \rangle_1$ to

$$\langle h_1, h_2 \rangle \equiv h_1(a)h_2(a) + h_1'(a)h_2'(a) + \int_a^b h_1''(x)h_2''(x)dx,$$

which does indeed satisfy the definition. The space $\mathscr{H}$ together with inner product $\langle \cdot, \cdot \rangle$ is an example of an RKHS.

Next, Optimization (3.5) is to be fully recast in terms of the RKHS construct. First, define the continuous functional $F_t(f) = f(t)$ and linear differential operator $L[f](x) = f''(x)$. Optimization (3.5) is directly extended to

$$\min_{f \in \mathscr{H}} \sum_{i=1}^{n} (\mathbf{y}_i - F_{x_i}[f])^2 + \lambda \int_a^b (L[f](x))^2 dx. \tag{3.9}$$

The Riesz Representation Theorem (Heckman, 2012) provides the key step necessary to solve Optimization (3.9). The result establishes that there exists a function $R_x(\cdot) \in \mathscr{H}$, called a **representer** such that

$$F_x[f] = \langle R_x, f \rangle = f(x), \ \forall f \in \mathscr{H}.$$

It turns out that

$$R_x(z) = 1 + (x-a)(z-a) + R_{1x}(z)$$

with

$$R_{1x}(z) = xz(\min(x,z) - a) - \frac{x+z}{2}\left(\min(x,z)^2 - a^2\right) + \frac{1}{3}\left(\min(x,z)^3 - a^3\right)$$

is in-fact a representer for our particular Hilbert Space $\mathscr{H}$ with corresponding inner product $\langle \cdot, \cdot \rangle$. It can also be shown (Heckman, 2012) that the solution to Optimization (3.9) has the form

$$f(x) = \alpha_0 + \alpha_1 x + \sum_{i=1}^{n} \boldsymbol{\beta}_i R_{1x_i}(x), \tag{3.10}$$

with $\alpha_0,\ \alpha_1 \in \mathbb{R}$. The so-called 'kernel trick' incorporates Representation (3.10) into Optimization (3.9) to produce the equivalent finite data version

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^2,\boldsymbol{\beta}\in\mathbb{R}^n} (\boldsymbol{y}-\boldsymbol{T}\boldsymbol{\alpha}-\boldsymbol{K}\boldsymbol{\beta})^\top (\boldsymbol{y}-\boldsymbol{T}\boldsymbol{\alpha}-\boldsymbol{K}\boldsymbol{\beta})+\lambda\boldsymbol{\beta}^\top\boldsymbol{K}\boldsymbol{\beta},$$

where $\boldsymbol{T}$ is the $n\times 2$ simple linear regression model matrix, and $\boldsymbol{K}$ is the Gram matrix of R_{1x}, i.e., $\boldsymbol{K}_{ij} = R_{1x_i}(x_j)$. This is in-fact a generalized ridge regression problem and can be solved in a similar manner as Optimization (3.7) above.

The result presented here can be directly computed on a $p = 1$ data set. Indeed, the construction can generalize to m-differentiable functions. The challenge, however, in practice is to identify the representer of more general Hilbert spaces and inner products for higher order functions. Extending this to larger p is also possible using additive models or tensor splines (Hastie et al., 2009), but has additional practical challenges. A more fruitful expedition for extending this work to larger p is pursued next using Mercer Kernels. This exposition is well-known, and the ideas presented next have had a significant influence on machine learning leading to some of the best techniques in the field.

3.2.1 Mercer Kernels and Hilbert Space Construction

The exposition for the smoothing spline using RKHS is natural and intuitive. One starts by contemplating the type of function sought and then defines the Hilbert space with an inner product to achieve this goal. The challenge is to determine the exact representer necessary to solve the ensuing penalized regression problem. This step is absolutely necessary and non-trivial. An alternative is to start with the representer and construct a Hilbert space and corresponding inner product using this function. It turns out that this approach is much more powerful in practice, but is not as intuitive. The representer is known in this literature as a **kernel** function. The elegance of this is to bypass the need to find a representer. The final result of this section provides the supervised kernel regression problem which is a generalization of the Smoothing Spline Optimization (3.5) to this setting.

Let C be a compact (closed and bounded) support, i.e., $C \subseteq \mathbb{R}^p$. To begin, define the general function space of all squared integrable functions

$$L_2(C) = \left\{ f : C \to \mathbb{R} : \int_C (f(t))^2 < \infty \right\}$$

with corresponding inner product

$$\langle f, g \rangle_{L_2(C)} = \int_C f(t)g(t)dt < \infty. \tag{3.11}$$

This function space and inner product do not form a RKHS. The goal in this construction is to find a subspace of $L_2(C)$ that restricts to functions in such a way that this subspace with a corresponding norm is indeed a RKHS. Precisely, let $\{\psi_i\}_{i=1}^{\infty}$ be an orthonormal basis of functions that span $L_2(C)$ and project $f \in L_2(C)$ onto this basis, i.e.,

$$f(x) = \sum_{i=1}^{\infty} c_i \psi_i(x)$$

with $c_i = \langle \psi_i, f \rangle_{L_2(C)}$. It is easily seen that $\sum_{i=1}^{\infty} c_i^2 < \infty$ for any function f. The general concept pursued here is that the basis functions $\{\psi_i\}_{i=1}^{\infty}$ and a corresponding sequence $a_1 \geq a_2 \geq \cdots \geq 0$ are chosen so that the set of functions under examinations satisfy the more stringent condition

$$f(x) = \sum_{i=1}^{\infty} c_i \psi_i(x) \text{ and } \sum_{i=1}^{n} \frac{c_i^2}{a_i} < \infty. \tag{3.12}$$

To do this, we require a kernel function.

Define the kernel function $K : C \times C \to \mathbb{R}$ as a symmetric function. The function K is assumed to be nonnegative definite, i.e., for any sequence $\{x_i\}_{i=1}^{n}$ the Gram matrix generated from this kernel onto the sequence is nonnegative definite. Examples of commonly used kernels include those listed below.

- **Linear Kernel**: $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^\top \boldsymbol{y}$.

- **Polynomial Kernel**: $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \left(a\boldsymbol{x}^\top \boldsymbol{y} + b\right)^d$.
- **Gaussian Kernel**: $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \exp\left(-\frac{||\boldsymbol{x}-\boldsymbol{y}||_2^2}{\sigma}\right)$.

Define a orthonormal basis of $L_2(C)$ using a kernel such that

$$\int_C \phi_i(z)k(x,z)dz = \gamma_i \phi_i(x),$$

with $\gamma_1 \geq \gamma_2 \geq \cdots \geq 0$. The function sequence $\phi_i(x)$ are referred to as eigenfunctions with corresponding eigenvalue γ_i. Some intuition for this sequence is that large eigenvalue typically correspond to more 'wiggle' functions $\phi_i(x)$ with respect to kernel K, i.e., we wish to restrict attention by forcing more weight on higher order eigenfunctions which will have a similar effect as ridge regression from Equation (3.3). At any rate, define a function space such that $\mathscr{H}_K \subseteq L_2(C)$ by

$$\mathscr{H}_K = \left\{ f(x) = \sum_{i=1}^{\infty} c_i \phi_i(x) \in L_2(C) \mid \sum_{i=1}^{n} \frac{c_i^2}{\gamma_i} < \infty \right\}, \tag{3.13}$$

which is analogous to Condition (3.12). The corresponding inner product

$$\langle f_1, f_2 \rangle_{\mathscr{H}_K} = \langle \sum_{i=1}^{\infty} c_i \phi_i, \sum_{i=1}^{\infty} d_i \phi_i \rangle_{\mathscr{H}_K} \equiv \sum_{i=1}^{\infty} \frac{c_i d_i}{\gamma_i}$$

and $||f||^2_{\mathscr{H}_K} = \langle f, f \rangle_{\mathscr{H}_K} = \sum_{i=1}^{\infty} \frac{c_i^2}{\gamma_i}$ are given. This is in-fact a RKHS. The projection $f(x) = \sum_{i=1}^{\infty} c_i \phi_i(x) \in \mathscr{H}_K$ onto this basis is called the **primal form** of the function f

It remains to be shown that K is indeed the representer of RKHS $\mathscr{H}_K$ with inner product $\langle \cdot, \cdot \rangle_{\mathscr{H}_K}$. Mercer's theorem establishes that $K(x,\cdot) = \sum\limits_{i=1}^{\infty} \gamma_i \phi_i(\cdot)\phi_i(x) = \sum\limits_{i=1}^{\infty} c_i \phi_i(x)$ with $c_i = \gamma_i \phi_i(\cdot)$ and $\sum\limits_{i=1}^{\infty} \frac{c_i^2}{\gamma_i} = \sum\limits_{i=1}^{\infty} \frac{\gamma_i^2 \phi_i(\cdot)^2}{\gamma_i} = \sum\limits_{i=1}^{\infty} \gamma_i \phi_i(\cdot)\phi_i(\cdot) = K(\cdot,\cdot) < \infty$, so it is verified that $K(x,\cdot) \in \mathscr{H}_K$. Finally, $K(x,\cdot)$ is indeed the representer of evaluation at x in space $\mathscr{H}_K$ since

$$\langle f, K(x,\cdot) \rangle_{\mathscr{H}_K} = \left\langle \sum_{i=1}^{\infty} c_i \phi_i, \sum_{i=1}^{\infty} \gamma_i \phi_i(x) \phi_i \right\rangle_{\mathscr{H}_K} = \sum_{i=1}^{\infty} \frac{c_i \gamma_i \phi_i(x)}{\gamma_i} = \sum_{i=1}^{\infty} c_i \phi_i(x) = f(x).$$

Further,

$$\langle K(z,\cdot),K(x,\cdot)\rangle_{\mathscr{H}_K} = \langle \sum_{i=1}^{\infty} \gamma_i \phi_i(z)\phi_i, \sum_{i=1}^{\infty} \gamma_i \phi_i(x)\phi_i \rangle_{\mathscr{H}_K} = \sum_{i=1}^{\infty} \frac{\gamma_i^2 \phi_i(x)\phi_i(z)}{\gamma_i} = K(x,z)$$

which is the reproducing property of the RKHS.

The **dual form** of a function $f \in \mathscr{H}_K$ is given with

$$f(x) = \sum_{i=1}^{\infty} \boldsymbol{\alpha}_i K(z_i,x), \text{ where } \sum_{i=1}^{\infty} \boldsymbol{\alpha}_i < \infty.$$

It is unclear that the primal and dual forms of a function $f \in \mathscr{H}_K$ are indeed equivalent. To see that they are, choose $\{z_i\}_{i=1}^{\infty}$ and $\{b_i\}_{i=1}^{\infty}$, so then it follows that

$$\begin{aligned} f(x) &= \sum_{i=1}^{\infty} b_i K(z_i,x) \\ &= \sum_{i=1}^{\infty}\sum_{\ell=1}^{\infty} b_i \gamma_\ell \phi_\ell(z_i)\phi_\ell(x) \\ &= \sum_{\ell=1}^{\infty}\sum_{i=1}^{\infty} b_i \gamma_\ell \phi_\ell(z_i)\phi_\ell(x) \\ &= \sum_{\ell=1}^{\infty} c_\ell \phi_\ell(x), \end{aligned}$$

where $c_\ell = \sum_{i=1}^{\infty} b_i \gamma_\ell \phi_\ell(z_i)$ and $\sum_{\ell=1}^{\infty} \frac{c_\ell^2}{\gamma_\ell} = \sum_{\ell=1}^{\infty}\sum_{i=1}^{\infty} \frac{b_i^2 \gamma_\ell^2 \phi_\ell^2(z_i)}{\gamma_\ell} = \sum_{i=1}^{\infty} b_i^2 K(z_i,z_i) < \infty$. From this result, the norm of $f \in \mathscr{H}_K$ can be compactly written as

$$\begin{aligned} ||f||^2_{\mathscr{H}_K} = \langle f,f\rangle_{\mathscr{H}_K} &= \langle \sum_{i=1}^{n} \boldsymbol{\alpha}_I K(z_i,.), \sum_{i=1}^{n} \boldsymbol{\alpha}_i K(z_i,.)\rangle_{\mathscr{H}_K} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{\alpha}_i \boldsymbol{\alpha}_j K(z_i,z_j) \\ &= \boldsymbol{\alpha}^\top \boldsymbol{K} \boldsymbol{\alpha}. \end{aligned} \tag{3.14}$$

The Representation (3.14) for finite data is the basis of the term the 'kernel trick' because it puts practical mathematical machinery in place for Hilbert space optimization.

An RKHS adaptation of Optimization (3.4) to kernel regression is

$$\min_{f \in \mathscr{H}_K} \sum_{i=1}^{n} (\boldsymbol{y}_i - f(x_i))^2 + \lambda ||f||^2_{\mathscr{H}_K}. \tag{3.15}$$

As with the smoothing splines from Section 3.1 above, it turns out that the solution to Optimization (3.15) on finite data is the kernel evaluated at the rows of $\boldsymbol{X}$, i.e.,

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \boldsymbol{\alpha}_i K(\boldsymbol{x}_i, \boldsymbol{x}) = \boldsymbol{K}\alpha,$$

where the $\boldsymbol{K}$ is $n \times n$ kernel Gram matrix for $n \times p$ training data matrix $\boldsymbol{X}$. This result together with Kernel Trick (3.14) gives the equivalent optimization

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} (\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha})^\top (\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}) + \lambda \boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha}$$

having solution $\widehat{\boldsymbol{y}} = \boldsymbol{K}(\boldsymbol{K} + \lambda \boldsymbol{I})^{-1}\boldsymbol{y}$. For classification, one can extend an RKHS optimization to logistic regression and solve this extension with iterative weighted least squares using Bernoulli probability weights.

3.3 Loss Function Mechanics for Kernel Based Approaches

Machine learning often involves an optimization of a generic objective function

$$\text{Loss} + \lambda * \text{Penalty}.$$

The loss function is a non-decreasing function of both the response and learning function. In Section 3.1, Optimization (3.4) had this form with a squared error loss functional $L(\boldsymbol{y}, \boldsymbol{f}) = (\boldsymbol{y} - \boldsymbol{f})^T (\boldsymbol{y} - \boldsymbol{f})$ and general penalty function $J(\boldsymbol{f})$ for regression problems. A logistic loss version uses the logistic loss functional $L(\boldsymbol{y}, \boldsymbol{f}) = \sum_{i=1}^{n} \log\left(1 + e^{-2\boldsymbol{y}_i \boldsymbol{f}_i}\right)$ for classification problems. The advent of powerful machine learning techniques using kernel functions has lead to new loss functions. These functions typically require powerful algorithms to fit. The

general idea is to breakdown the optimization problem into 'primal' and 'dual' forms. The dual form is directly solved. In this Section, we develop two common loss functions typically used with kernel methods. The main goal is to test optimizing these loss functions against the solutions using the classical squared error loss and logistic loss functions. Our goal is to determine the efficacy of these functions on predictive performance. To this endpoint, the benchmark comparison in Section 3.3.3 fully optimized all tuning parameters for both versions.

3.3.1 Support Vector Machines in Classification

Classification is a common problem under examination in machine learning. The Support Vector Machine (SVM) is a well-known kernel approach applied to classification problems and is presented next. Assume 2-level classification with coding $\boldsymbol{y}_i \in \{-1, 1\}$. The goal is to ultimately predict

$$\widehat{\boldsymbol{y}}_i = \begin{cases} 1 & f(\boldsymbol{x}_i) \geq 0 \\ -1 & \text{otherwise.} \end{cases}$$

This requires estimating the prediction rule f.

Geometrically, the SVM attempts to find a hyperplane that separates the response classes. Ideally, the hyperplane is as far away as possible from each classification group, but this is only truly possible when the classes are linearly separable. Assume for now that the classes are separable. A hyperplane is the set of points $\left\{\boldsymbol{x} \in \mathbb{R}^p : \boldsymbol{\omega}^\top \boldsymbol{x} + b = \vec{0}\right\}$. It clearly follows that $\boldsymbol{\omega}$ is orthogonal to any element in this set. The true linear prediction rule is assumed to have form $f(\boldsymbol{x}) = \boldsymbol{\omega}^\top \boldsymbol{x} + b$. The main idea is to choose f in a way to maximize the margin M between the two classes, i.e.,

$$\begin{aligned} \underset{\widetilde{\boldsymbol{\omega}}, b \in \mathbb{R}}{\text{maximize}} \quad & M \\ \text{subject to:} \quad & \boldsymbol{y}_i \left(\widetilde{\boldsymbol{\omega}}^\top \boldsymbol{x}_i + b\right) \geq M \; \forall i, \\ & \left\|\widetilde{\boldsymbol{\omega}}\right\|_2^2 = 1. \end{aligned}$$

This problem can be reformulated to the more convenient and equivalent version

$$\min_{\boldsymbol{\omega},b\in\mathbb{R}} \tfrac{1}{2}||\boldsymbol{\omega}||^2 \text{ subject to } \boldsymbol{y}_i\left(\boldsymbol{\omega}^\top \boldsymbol{x}_i + b\right) \geq 1 \;\forall\; i.$$

This convex optimization problem can be re-expressed in terms of a Lagrangian multiplier to obtain the so-called **primal** functional corresponding to

$$F(\alpha, b, w) = \frac{1}{2}||\boldsymbol{\omega}||^2 - \sum_{i=1}^{n} \boldsymbol{\alpha}_i \left(\boldsymbol{y}_i(\boldsymbol{\omega}^\top \boldsymbol{x}_i + b) - 1\right) \;\forall\; \alpha_i \geq 0.$$

Taking derivative of F with respect to b and $\boldsymbol{\omega}$, the primal form can be converted into a so-called **dual** form functional

$$\begin{aligned} G(\boldsymbol{\alpha}) &= \sum_{i=1}^{n} \boldsymbol{\alpha}_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{y}_i \boldsymbol{y}_j \boldsymbol{x}_i^\top \boldsymbol{x}_j \boldsymbol{\alpha}_i \boldsymbol{\alpha}_j \\ &= \vec{1}^\top \boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{H} \boldsymbol{\alpha}. \end{aligned} \tag{3.16}$$

Maximizing $G(\boldsymbol{\alpha})$ subject to the constraint that $\boldsymbol{\alpha}^\top \boldsymbol{y} = 0$ and $\boldsymbol{\alpha} \geq \vec{0}$ gives solutions $b = \boldsymbol{y}_i - \boldsymbol{\omega}^\top \boldsymbol{x}_i$ for some i and $\boldsymbol{\omega} = \sum_{i=1}^{n} \boldsymbol{y}_i \boldsymbol{\alpha}_i \boldsymbol{x}_i$. In the case when the classes are linearly non-separable, the above derivation requires a modification using slack variables and a cost parameter (Cortes and Vapnik, 1995).

The SVM is easily generalized to a non-linear classifier associated with the RKHS previously developed (Scholkopf & Smola, 2002) using a kernel function K. First, define

$$k(\boldsymbol{x}) = \begin{pmatrix} K(\boldsymbol{x}, \boldsymbol{x}_1) \\ K(\boldsymbol{x}, \boldsymbol{x}_2) \\ \vdots \\ K(\boldsymbol{x}, \boldsymbol{x}_n) \end{pmatrix} = K(\boldsymbol{x}, \cdot).$$

Replacing $\boldsymbol{x}_i$ in the above derivation with $k(\boldsymbol{x}_i)$ leads to a dual form problem that is now non-linear in the optimization functional, i.e., in Equation (3.16) replace $\boldsymbol{\omega}$ with $\boldsymbol{\omega} = \sum_{i=1}^{n} \boldsymbol{\alpha}_i \boldsymbol{y}_i k(\boldsymbol{x}_i)$ and set $\boldsymbol{H} = \left(\boldsymbol{y}_i \boldsymbol{y}_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)$. The prediction rule $f(x) = \langle \boldsymbol{\omega}, k(\boldsymbol{x}) \rangle_{\mathscr{H}_K} + b$ is given with inner-

product

$$\langle \boldsymbol{\omega}, k(\boldsymbol{x}) \rangle_{\mathscr{H}_K} = \langle \sum_{i=1}^{n} \boldsymbol{\alpha}_i \boldsymbol{y}_i k(\boldsymbol{x}_i), k(\boldsymbol{x}) \rangle_{\mathscr{H}_K} = \sum_{i=1}^{n} \boldsymbol{\alpha}_i \boldsymbol{y}_i \langle k(\boldsymbol{x}_i), k(\boldsymbol{x}) \rangle_{\mathscr{H}_K} = \sum_{i=1}^{n} \boldsymbol{\alpha}_i \boldsymbol{y}_i K(\boldsymbol{x}, \boldsymbol{x}_i).$$

So, the prediction rule is $f(\boldsymbol{x}) = \sum_{i=1}^{n} \boldsymbol{\alpha}_i \boldsymbol{y}_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b$, with $b = \boldsymbol{y}_i - \sum_{i=1}^{n} \boldsymbol{\alpha}_i \boldsymbol{y}_i K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for some i. Alternatively, this optimization problem can be formulated in terms of a hinge loss optimization problem where $(1 - \boldsymbol{y}_i f(\boldsymbol{x}_i))_+ = \max(0, 1 - \boldsymbol{y}_i f(\boldsymbol{x}_i))$ is the hinge loss function. The SVM solves

$$\min_{f \in \mathscr{H}_K} \frac{1}{n} \sum_{i=1}^{n} (1 - \boldsymbol{y}_i f(\boldsymbol{x}_i))_+ + \lambda ||f||^2_{\mathscr{H}_K}$$

for a prediction rule $f(\boldsymbol{x})$. Multi-class regression and ordinal regression are both possible generalizations of this framework and have been considered (Hill and Doucet, 2007; Shashua and Levin, 2002).

3.3.2 Sensitive Loss Functions for Regression

The SVM has had a profound impact on the literature (Lin et al., 2002; Hastie et al., 2009). One by-product is the development of hinge loss as an optimization function. This approach and loss function make sense in classification, but are not naturally applicable to regression. One attempt to bridge this gap is the so-called ε-insensitive loss function

$$|\boldsymbol{y} - f(\boldsymbol{x})|_\varepsilon = \sum_{i=1}^{n} (|\boldsymbol{y}_i - f(\boldsymbol{x}_i)| - \varepsilon) \mathbf{1}_{\{|\boldsymbol{y}_i - f(\boldsymbol{x}_i)| > \varepsilon\}}.$$

The learning function is a hyperplane parameterized as $f(\boldsymbol{x}) = \langle \boldsymbol{\omega}, \boldsymbol{x} \rangle + b$. The SVM optimization problem adjusted to regression has form

$$\begin{aligned} \underset{\boldsymbol{\omega}, b}{\text{minimize}} \quad & \tfrac{1}{2}||w||^2 + C \textstyle\sum_{i=1}^{n} (\xi_i + \xi_i^\star) \\ \text{subject to:} \quad & \boldsymbol{y}_i - f(\boldsymbol{x}_i) \leq \varepsilon - \xi_i, \\ & f(\boldsymbol{x}_i) - \boldsymbol{y}_i \leq \varepsilon - \xi_i^\star, \end{aligned}$$

where $\xi_i, \xi_i^\star \geq 0$ are the slack variables and C is the cost parameter. The solution to this problem is referred to as Support Vector Regression (SVR).

The SVR methods is extended to kernel function K whose objective has the form

$$f(\boldsymbol{x}, \boldsymbol{\alpha}^\star, \boldsymbol{\alpha}) = \sum_{i=1}^{n} (\boldsymbol{\alpha}_i^\star - \boldsymbol{\alpha}_i) K(\boldsymbol{x}_i, \boldsymbol{x}) + b.$$

The vector $\boldsymbol{\alpha}^\star$ are the 'positive' Lagrange multipliers while the vector $\boldsymbol{\alpha}$ are the 'negative' Lagrange multipliers. Additional non-negative multipliers v_i and $v_i^\star$ are also defined. The **primal** function is given

$$\begin{aligned} F &= ||\boldsymbol{\omega}||^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^\star) - \sum_{i=1}^{n} \boldsymbol{\alpha}_i (\varepsilon + \xi_i - \boldsymbol{y}_i + \langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle + b) - \\ &\quad \sum_{i=1}^{n} \boldsymbol{\alpha}_i^\star (\varepsilon + \xi_i^\star + \boldsymbol{y}_i - \langle \boldsymbol{\omega}, \boldsymbol{x}_i \rangle - b) - \sum_{i=1}^{n} (v_i \xi_i + v_i^\star \xi_i^\star). \end{aligned}$$

Proceeding as with the SVM, take derivatives of the objective with respect to $\boldsymbol{\omega}, b, \xi_i, \xi_i^\star$ and then reformulate into a corresponding **dual** functional

$$G(\boldsymbol{\alpha}^\star, \boldsymbol{\alpha}) = \frac{1}{2} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star)^\top \boldsymbol{K} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star) + \varepsilon \vec{1}^\top (\boldsymbol{\alpha} + \boldsymbol{\alpha}^\star) - \boldsymbol{y}^\top (\boldsymbol{\alpha} - \boldsymbol{\alpha}^\star)$$

subject to

$$\sum_{i=1}^{n} \boldsymbol{\alpha}_i = \sum_{i=1}^{n} \boldsymbol{\alpha}_i^\star \text{ and } 0 \leq \boldsymbol{\alpha}, \boldsymbol{\alpha}^\star \leq C.$$

From this, the SMO algorithm of Platt (1998) can be used to estimate the $\boldsymbol{\omega}$ and b. The parameters C and ε are to be estimated by cross-validation (CV).

3.3.3 Empirical Demonstrations

Regression and classification benchmarks results are described in this section. The ε-sensitive loss and hinge loss kernel based optimization problems were fit using the `kernlab` package (Karatzoglou et al., 2004) in R. In each example, the corresponding square error loss and logistic loss functions were also fit using in-house software. The goal was to assess how

Table 3.1: Benchmark Data Sets.

Data Set	(n, p)	Type	Response	Reference
Blood	$(208, 134)$	Regress	$\log(\text{BBB})$	Kuhn (2014)
Eye	$(120, 200)$	Regress	$\sqrt{\text{Express}}$	Scheetz et al. (2006)
U.S. News & World Report	$(1004, 20)$	Regress	SAT.ACT	ASA Data Expo '95
Votes	$(435, 16)$	Class	House Vote	Lichman (2013)
Flare	$(1066, 9)$	Class	Solar Flare	Lichman (2013)
German Credit	$(1000, 20)$	Class	Credit Score	Lichman (2013)

real the bottom-line contribution of each complex loss function is to the much simpler loss function. Table 3.1 summarizes each data set used.

For this experiment, the polynomial kernel was fit, $K(\boldsymbol{x}, \boldsymbol{y}) = (a\boldsymbol{x}^\top \boldsymbol{y} + b)^d$. Three-fold CV was used to estimate the parameters on the finite grid

$$(a, b, d, \varepsilon) \in \{0.01, 0.1, 1.0, 1.5\} \times \{0.0, 0.05, 1.0, 2.5\} \times \{1.0, 2.0, 3.0\} \times \{0.05, 0.1\},$$

and the C parameter was chosen over a fine grid of length 27 between 0.05 and 10.0. In each case, the data sets were broken up into training and testing. The training percentages used were 10%, 30%, and 50%. The process was repeated 25 times per training size, and the testing error was recorded.

The results are presented in Figure 4.1. It was somewhat surprising that the complex loss functions made no appreciable difference. In some cases, the performance was actually worse. These result focus our direction in the next chapter. Squared error loss and logistic loss are used to fit our main contribution in that chapter. This study justifies this decision in Chapter 4 to avoid overly complicated loss functions that require additional computational time while not improving performance.

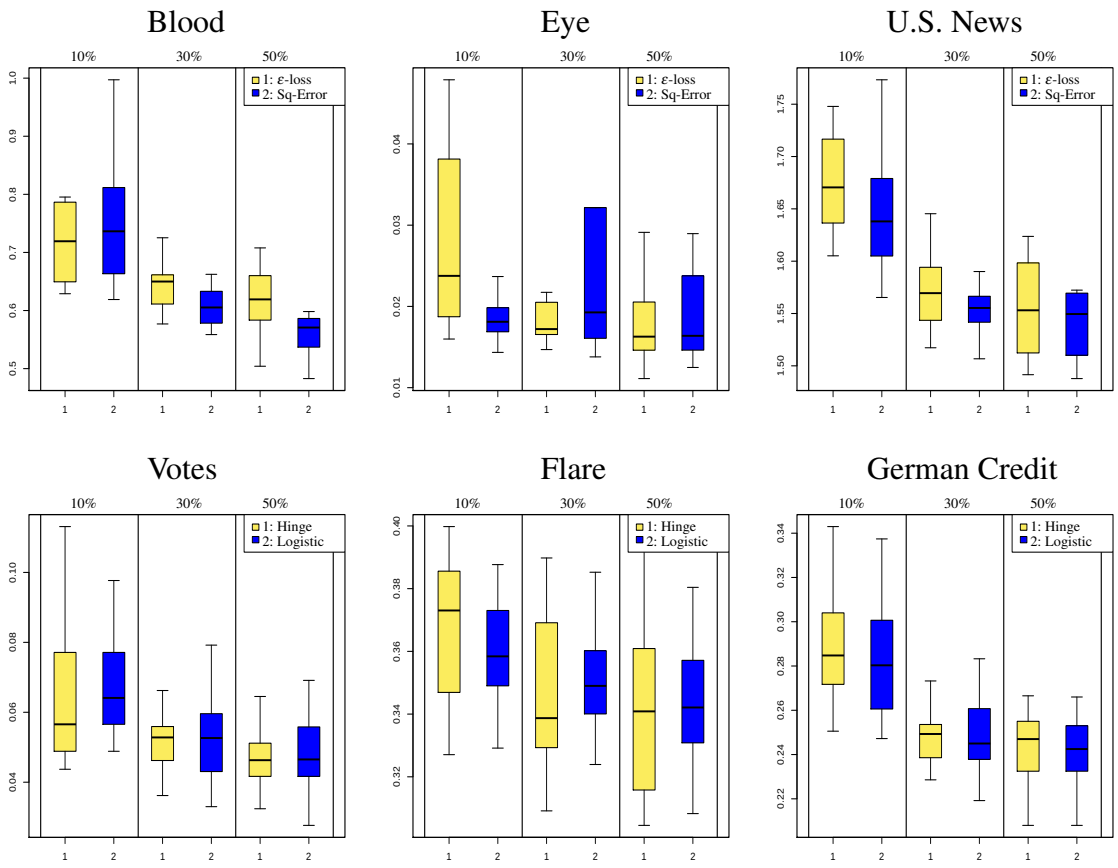


Figure 3.1: Testing Performance on Real Data Sets.

CHAPTER 4

A SAFE MANIFOLD APPROACH TO SEMI-SUPERVISED LEARNING

4.1 Introduction

The main contribution of this chapter is a novel *safe* semi-supervised kernel-based modeling (S3KM) approach. As discussed in Chapter 1, the safety feature of the S3KM is its ability to tradeoff between a semi-supervised learning manifold-based approach and a well-established supervised alternative. First, notational conventions are given in Section 4.2, and a general relationship between ridge and kernel regression is proven in Section 4.3. Then the S3KM for regression with a square error loss is defined in Section 4.4. Next, the S3KM is extended to classification problems in Section 4.5 with a logistic labeled loss, and the resulting optimization is solved by an iterative algorithm based on the square error version in Section 4.4. The S3KM is then extended to an anchor graph S3KM or AS3KM for computation efficiency in Section 4.6. Our novel S3KM and AS3KM methods are compared to the related method of manifold regularization in Section 4.7. All methods are benchmarked on real data in Section 4.8, and these empirical results demonstrate the effectiveness of the S3KM and

AS3KM.

There are many semi-supervised approaches, but few are safe in the manner described here, i.e., they tradeoff between a semi-supervised learning manifold-based approach and a well-established supervised alternative. A safe example is Culp et al. (2009), which promoted safety by preferring the less-wiggly, supervised fit of non-parametric local kernel regression over a more complicated semi-supervised fit unless overruled by a stepwise criterion. Recent work involved non-noisy structured data problems (Li and Zhou, 2011) or a kernel density approach (Kawakita and Jun'ichi, 2014; Culp and Ryan, 2013; Azizyan et al., 2013). In addition, few of the semi-supervised approaches in the literature are actually implemented, robust, and practical for real data problems. These shortcomings justify why our safe method is advantageous during the practical applications in Section 4.8.

4.2 Mathematical Problem Setup and Notation

This section outlines the notational conventions used to define the S3KM later in Section 4.4. Let L and U partition the index set $\{1,\ldots,n\}$ for the n observations into the sets of labeled and unlabeled observations. The technical setup requires that the $m = |L|$ labeled observations $(y_i, \boldsymbol{x}_i)$ for $i \in L$ are independent and identically distributed, where $y_i \in \mathbb{R}$ and $\boldsymbol{x}_i \in \mathbb{R}^p$. An additional $n - m = |U|$ unlabeled observations $\boldsymbol{x}_i$ are also independent and identically distributed (and independent of the labeled data), but their responses y_i for $i \in U$ are not available for training. Based on stacking the $\boldsymbol{x}_i$ as row vectors for $i = 1,\ldots,n$, the full data are represented by an $n \times p$ model matrix $\boldsymbol{X}$ with the row-wise partition

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_L \\ \boldsymbol{X}_U \end{pmatrix},$$

and we tacitly assume a sorting of the data, i.e., with loss of generality the labeled observations come first.

Localized structures within model matrix $\boldsymbol{X}$ can be exploited with a kernel regression

setup. This requires choosing a kernel function $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, which is used to convert $\boldsymbol{X}$ to an $n \times n$ nonnegative definite kernel matrix

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{K}_{LL} & \boldsymbol{K}_{LU} \\ \boldsymbol{K}_{UL} & \boldsymbol{K}_{UU} \end{pmatrix}. \tag{4.1}$$

The entries of kernel matrix $\boldsymbol{K}$ are $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, i.e., simply apply the kernel function K to the ith and jth rows of $\boldsymbol{X}$.

Global manifold structures within model matrix $\boldsymbol{X}$ can be exploited with a graph-based operator such as a graph Laplacian. To induce sparsity, we use a k-nearest neighbors (k-NN) graph. Let $N_k(\boldsymbol{x}_0) \subset \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_n\}$ such that $|N_k(\boldsymbol{x}_0)| = k$ be the neighborhood of any $\boldsymbol{x}_0 \in \mathbb{R}^p$. Then the $n \times n$ distance matrix $\widetilde{\boldsymbol{D}} = [\widetilde{d}_{ij}]$ with entries

$$\tilde{d}_{ij} = \begin{cases} \left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|_2^2 & \text{if } \boldsymbol{x}_j \in N_k(\boldsymbol{x}_i) \text{ or } \boldsymbol{x}_i \in N_k(\boldsymbol{x}_j) \\ \infty & \text{otherwise} \end{cases}$$

is well-defined. This in turn is used to obtain an $n \times n$ adjacency matrix $\boldsymbol{\omega} = [\omega_{ij}]$ with entries

$$\omega_{ij} = \exp\left(-d_{ij}/\sigma^2\right) \tag{4.2}$$

and then its corresponding graph Laplacian

$$\boldsymbol{\Delta} = \operatorname{diag}\left(\boldsymbol{\omega}\vec{1}\right) - \boldsymbol{\omega} = \begin{pmatrix} \boldsymbol{\Delta}_{LL} & \boldsymbol{\Delta}_{LU} \\ \boldsymbol{\Delta}_{UL} & \boldsymbol{\Delta}_{UU} \end{pmatrix}. \tag{4.3}$$

At least hypothetically, there is also an $n \times 1$ response vector $\boldsymbol{y} = [y_i]$ corresponding to $\boldsymbol{X}$. This response also partitions into the m observed responses $\boldsymbol{y}_L$ and $n - m$ *latent* (or unobserved) variables $\boldsymbol{y}_U$, and we adopt the notation

$$\boldsymbol{y}(\boldsymbol{y}_U) = \begin{pmatrix} \boldsymbol{y}_L \\ \boldsymbol{y}_U \end{pmatrix},$$

to emphasize that we don't have $\boldsymbol{y}_U$ (in spite of the fact that goal of a study may be to predict $\boldsymbol{y}_U$). We will also use the diagonal matrix

$$\begin{aligned} \boldsymbol{w} &= \text{diag}(w_1, \ldots, w_n) \\ &= \begin{pmatrix} \boldsymbol{w}_{LL} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{w}_{UU} \end{pmatrix} \end{aligned}$$

comprised of positive observation case-weights $w_i > 0$ for $i = 1, \ldots, n$.

4.3 Supervised Ridge and Kernel Regression Connections

This section culminates in Theorem 1. This result establishes an equivalence between kernel and ridge regression and is also referenced later in Sections 4.4 and 4.6 to interpret of our of novel S3KM and AS3KM methods as induced ridge regressions based on a sort of kernel transformed model matrix.

Supervised approaches only use the labeled data: $\boldsymbol{X}_L, \boldsymbol{y}_L$ (and possibly the $m \times m$ non-negative definite kernel matrix $\boldsymbol{K}_{LL}$ computed from $\boldsymbol{X}_L$). Start with supervised least squares (LS) regression

$$\widehat{\boldsymbol{\beta}}^{(\text{LS})} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left\| \boldsymbol{w}_{LL}^{1/2} (\boldsymbol{y}_L - \boldsymbol{X}_L \boldsymbol{\beta}) \right\|_2^2.$$

For ease of exposition, assume model matrix $\boldsymbol{X}_L$ is of full column rank, so then $\widehat{\boldsymbol{\beta}}^{(\text{LS})} = \left(\boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{X}_L \right)^{-1} \boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{y}_L$. Also let $\{\lambda_i, \boldsymbol{v}_i\}_{i=1}^p$ be the eigen or spectral decomposition of the symmetric matrix $\boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{X}_L$.

Supervised ridge regression

$$\widehat{\boldsymbol{\beta}}^{(\text{Ridge})} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left\| \boldsymbol{w}_{LL}^{1/2} (\boldsymbol{y}_L - \boldsymbol{X}_L \boldsymbol{\beta}) \right\|_2^2 + \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta} \tag{4.4}$$

and has the representation

$$\begin{aligned}\widehat{\boldsymbol{\beta}}^{(\text{Ridge})} &= \left(\boldsymbol{X}_L^\top \boldsymbol{w}_{LL}\boldsymbol{X}_L + \lambda \boldsymbol{I}\right)^{-1}\left(\boldsymbol{X}_L^\top \boldsymbol{w}_{LL}\boldsymbol{X}_L\right)\widehat{\boldsymbol{\beta}}^{(\text{LS})} \\ &= \sum_{i=1}^{p} \frac{\lambda_i}{\lambda_i+\lambda} c_i \boldsymbol{v}_i,\end{aligned}$$

where $\widehat{\boldsymbol{\beta}}^{(\text{LS})} = c_1\boldsymbol{v}_1 + \cdots + c_p\boldsymbol{v}_p$ is projected onto the eigen decomposition of $\boldsymbol{X}_L^\top \boldsymbol{w}_{LL}\boldsymbol{X}_L$, and so shrinking is proportionally more concentrated on the lower order principal components or eigenvectors $\boldsymbol{v}_i$ with the smaller eigenvalues λ_i. The corresponding labeled fits are

$$\begin{aligned}\boldsymbol{X}_L\widehat{\boldsymbol{\beta}}^{(\text{Ridge})} &= \boldsymbol{X}_L\left(\boldsymbol{X}_L^\top \boldsymbol{w}_{LL}\boldsymbol{X}_L + \lambda \boldsymbol{I}\right)^{-1}\left(\boldsymbol{X}_L^\top \boldsymbol{w}_{LL}\boldsymbol{X}_L\right)\widehat{\boldsymbol{\beta}}^{(\text{LS})} \\ &= \sum_{i=1}^{p} \frac{\lambda_i}{\lambda_i+\lambda} c_i \boldsymbol{X}_L\boldsymbol{v}_i.\end{aligned} \tag{4.5}$$

Next, we turn attention to supervised kernel regression

$$\widehat{\widetilde{\boldsymbol{\alpha}}} = \underset{\widetilde{\boldsymbol{\alpha}}\in\mathbb{R}^m}{\arg\min} \left\| \boldsymbol{w}_{LL}^{1/2}(\boldsymbol{y}_L - \boldsymbol{K}_{LL}\widetilde{\boldsymbol{\alpha}})\right\|_2^2 + \lambda \widetilde{\boldsymbol{\alpha}}^\top \boldsymbol{K}_{LL}\widetilde{\boldsymbol{\alpha}}. \tag{4.6}$$

for some choice of nonnegative definite kernel function $K(\cdot,\cdot)$. This yields labeled fits of

$$\begin{aligned}\widehat{\boldsymbol{\eta}}_L &= \boldsymbol{K}_{LL}\widehat{\widetilde{\boldsymbol{\alpha}}} \\ &= \left(\boldsymbol{K}_{LL}\boldsymbol{w}_{LL} + \lambda \boldsymbol{I}\right)^{-1}\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}\boldsymbol{y}_L.\end{aligned}$$

An example of Kernel Regression (4.6) is based on the linear kernel function $K(\boldsymbol{x}_i,\boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$ implying $\boldsymbol{K} = \boldsymbol{X}_L\boldsymbol{X}_L^\top$ is the outer product matrix of $\boldsymbol{X}_L$ and labeled fits vector

$$\widehat{\boldsymbol{\eta}}_L = \left(\boldsymbol{X}_L\boldsymbol{X}_L^\top \boldsymbol{w}_{LL} + \lambda \boldsymbol{I}\right)^{-1}\boldsymbol{X}_L\boldsymbol{X}_L^\top \boldsymbol{w}_{LL}\boldsymbol{y}_L,$$

whereas the Labeled Fits (4.5) from ridge regression directly involve the inner product matrix $\boldsymbol{X}_L^\top \boldsymbol{w}_{LL}\boldsymbol{X}_L$. Kernel Regression Optimization (4.6) is equivalent to Ridge Regression Optimization (4.4) with the constraint of picking an optimal $\widehat{\boldsymbol{\beta}}^{(\text{Ridge})}$ in the row space of $\boldsymbol{X}_L$.

This is easily seen after substitutions $\boldsymbol{\beta} \mapsto \boldsymbol{X}_L^\top \widetilde{\boldsymbol{\alpha}}$ followed by $\boldsymbol{X}_L \boldsymbol{X}_L^\top \mapsto \boldsymbol{K}_{LL}$ into Optimization (4.4). Theorem 2 states this equivalence in terms of the fits and uses a direct proof.

Theorem 1. *The labeled fits* $(\boldsymbol{K}_{LL}\boldsymbol{w}_{LL} + \lambda \boldsymbol{I})^{-1} \boldsymbol{K}_{LL}\boldsymbol{w}_{LL}\boldsymbol{y}_L$ *of kernel regression with the linear kernel* $\boldsymbol{K}_{LL} = \boldsymbol{X}_L \boldsymbol{X}_L^\top$ *equal the labeled ridge regression fits* $\boldsymbol{X}_L \widehat{\boldsymbol{\beta}}^{(Ridge)}$ *for any* $\lambda \geq 0$.

Proof. We need to show that

$$\left(\boldsymbol{X}_L \boldsymbol{X}_L^\top \boldsymbol{w}_{LL} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{X}_L \boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{y}_L = \boldsymbol{X}_L \left(\boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{X}_L + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{y}_L. \tag{4.7}$$

This direct proof hinges on the observation that $\boldsymbol{w}_{LL}^{1/2} \boldsymbol{X}_L \boldsymbol{v}_i$ is an eigenvector of the outer product matrix $\boldsymbol{w}_{LL}^{1/2} \boldsymbol{X}_L \boldsymbol{X}_L^\top \boldsymbol{w}_{LL}^{1/2}$ for $i = 1, \ldots, p$, i.e.,

$$\boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{X}_L \boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i \Rightarrow \boldsymbol{w}_{LL}^{1/2} \boldsymbol{X}_L \boldsymbol{X}_L^\top \boldsymbol{w}_{LL}^{1/2} (\boldsymbol{w}_{LL}^{1/2} \boldsymbol{X}_L \boldsymbol{v}_i) = \boldsymbol{w}_{LL}^{1/2} \boldsymbol{X}_L \lambda_i \boldsymbol{v}_i.$$

With this in mind, the left hand side of Equation (4.7) is

$$\begin{aligned}
\text{Kernel Fits} &= \left(\boldsymbol{X}_L \boldsymbol{X}_L^\top \boldsymbol{w}_{LL} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{X}_L \boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{y}_L \\
&= \left(\boldsymbol{X}_L \boldsymbol{X}_L^\top \boldsymbol{w}_{LL} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{X}_L \left(\boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{X}_L\right) \left(\boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{X}_L\right)^{-1} \boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{y}_L \\
&= \boldsymbol{w}_{LL}^{-1/2} \left(\boldsymbol{w}_{LL}^{1/2} \boldsymbol{X}_L \boldsymbol{X}_L^\top \boldsymbol{w}_{LL}^{1/2} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{w}_{LL}^{1/2} \boldsymbol{X}_L \boldsymbol{X}_L^\top \boldsymbol{w}_{LL}^{1/2} \left(\boldsymbol{w}_{LL}^{1/2} \boldsymbol{X}_L \widehat{\boldsymbol{\beta}}^{(\mathrm{LS})}\right) \\
&= \left[\boldsymbol{w}_{LL}^{-1/2}\right] \left(\boldsymbol{w}_{LL}^{1/2} \boldsymbol{X}_L \boldsymbol{X}_L^\top \boldsymbol{w}_{LL}^{1/2} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{w}_{LL}^{1/2} \boldsymbol{X}_L \boldsymbol{X}_L^\top \boldsymbol{w}_{LL}^{1/2} \left(\sum_{i=1}^{p} c_i \boldsymbol{w}_{LL}^{1/2} \boldsymbol{X}_L \boldsymbol{v}_i\right) \\
&= \left[\boldsymbol{w}_{LL}^{-1/2}\right] \boldsymbol{w}_{LL}^{1/2} \boldsymbol{X}_L \sum_{i=1}^{p} \frac{\lambda_i}{\lambda_i + \lambda} c_i \boldsymbol{v}_i,
\end{aligned}$$

whereas the right hand side of Equation (4.7) is

$$\begin{aligned}
\text{Ridge Fits} &= \boldsymbol{X}_L \left(\boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{X}_L + \lambda \boldsymbol{I} \right)^{-1} \boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{y}_L \\
&= \boldsymbol{X}_L \left(\boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{X}_L + \lambda \boldsymbol{I} \right)^{-1} \boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{X}_L \widehat{\boldsymbol{\beta}}^{(\mathrm{LS})} \\
&= \boldsymbol{X}_L \left(\boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{X}_L + \lambda \boldsymbol{I} \right)^{-1} \boldsymbol{X}_L^\top \boldsymbol{w}_{LL} \boldsymbol{X}_L \sum_{i=1}^{p} c_i \boldsymbol{v}_i \\
&= \boldsymbol{X}_L \sum_{i=1}^{p} \frac{\lambda_i}{\lambda_i + \lambda} c_i \boldsymbol{v}_i \\
&= \text{Kernel Fits.}
\end{aligned}$$

□

Theorem 1 is easily generalized to an arbitrary nonnegative definite kernel function and its resulting $m \times m$ Gram matrix $\boldsymbol{K}_{LL}$ (computed from $\boldsymbol{X}_L$). This follows by taking the eigenvalue decomposition of $\boldsymbol{K}_{LL} = \boldsymbol{\Phi}_{LL} \boldsymbol{\Lambda}_{LL} \boldsymbol{\Phi}_{LL}^\top$ where the top $r \leq m$ eigenvalues are nonzero. The $m \times r$ matrix $\widetilde{\boldsymbol{A}}_L$ is constructed from the top r eigenvalues and eigenvectors so that $\boldsymbol{K}_{LL} = \widetilde{\boldsymbol{A}}_L \widetilde{\boldsymbol{A}}_L^\top$. So, Theorem 1 establishes that kernel regression reduces to ridge regression with an kernel-based induced model matrix substitution of $\widetilde{\boldsymbol{A}}_L$ in place of $\boldsymbol{X}_L$.

4.4 A Safe Semi-Supervised Kernel Model: S3KM

Joint training is a general semi-supervised framework that treats the unknown components of $\boldsymbol{y}_U$ as additional decision variables during optimization. Our focus is the joint training optimization problem

$$\left(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{f}}, \widehat{\boldsymbol{y}}_U \right) = \underset{\boldsymbol{\alpha}, \boldsymbol{f}, \boldsymbol{y}_U}{\arg\min} \left\| \boldsymbol{w}^{1/2} \left(\boldsymbol{y}(\boldsymbol{y}_U) - \boldsymbol{f} - \boldsymbol{K}\boldsymbol{\alpha} \right) \right\|_2^2 + \lambda_1 \boldsymbol{\alpha}^\top \boldsymbol{K} \boldsymbol{\alpha} + \lambda_2 \boldsymbol{f}^\top \boldsymbol{\Delta} \boldsymbol{f} + \gamma \boldsymbol{y}_U^\top \boldsymbol{y}_U \quad (4.8)$$

for some $\lambda_1 \geq 0$, $\lambda_2 \geq 0$, and $\gamma \geq 0$ with corresponding fits of

$$\widehat{\boldsymbol{\eta}} = \widehat{\boldsymbol{f}} + \boldsymbol{K} \widehat{\boldsymbol{\alpha}}. \quad (4.9)$$

The solution to Optimization (4.8) is henceforth referred to as the Safe Semi-Supervised Kernel Model (S3KM). A data analysis involving Optimization (4.8) might be boiled down to a choice of kernel function $k(\cdot,\cdot)$ to produce $\boldsymbol{K}$ as well as estimation of the tuning parameters $\lambda_1, \lambda_2, \gamma, \sigma^2$, where σ^2 from Equation (4.2) is used to construct Laplacian (4.3). This endpoint might be achieved by using Cross-Validation (CV) to estimate $\lambda_1, \lambda_2, \gamma, \sigma^2$ for a number of kernel functions $k(\cdot,\cdot)$. While our focus will often default to the linear kernel function $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$ for ease of presentation, results extend in a natural manner to other kernel functions such as those listed in Section 3.2.1.

Next, we analytically investigate the extremes of the compromise or tradeoff spanned by Optimization (4.8) in the limits as its tuning parameters $\lambda_1, \lambda_2, \gamma$ are set to boundary values of 0 or ∞. This is done to better understand and motivate the need for each term in the objective function of Optimization (4.8). This discussion is broken into Section 4.4.1 for $\lambda_1 = \infty$ and Section 4.4.2 for $\lambda_2 = \infty$.

4.4.1 The S3KM when $\lambda_1 = \infty$

The limit of $\lambda_1 \to \infty$ implies $\boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} \to 0$. This is equivalent to a limit of $\boldsymbol{K}\boldsymbol{\alpha} = \vec{0}$ when $\lambda_1 = \infty$ because $\boldsymbol{K}$ is nonnegative definite. This follows since $\boldsymbol{K} = \boldsymbol{A}\boldsymbol{A}^\top$ for some matrix $\boldsymbol{A}$ and hence

$$\begin{aligned} \boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} = 0 \quad &\Rightarrow \quad \boldsymbol{\alpha}^\top \boldsymbol{A}\boldsymbol{A}^\top \boldsymbol{\alpha} = 0 \\ &\Rightarrow \quad \boldsymbol{A}^\top \boldsymbol{\alpha} = \vec{0} \\ &\Rightarrow \quad \boldsymbol{K}\boldsymbol{\alpha} = \vec{0} \\ &\Rightarrow \quad \boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} = 0. \end{aligned}$$

So, when $\lambda_1 = \infty$, Optimization (4.8) simplifies to a well-understood graph-based optimization

$$\left(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{y}}_U\right) = \underset{\boldsymbol{f}, \boldsymbol{y}_U}{\arg\min} \left\| \boldsymbol{w}^{1/2} \left(\boldsymbol{y}(\boldsymbol{y}_U) - \boldsymbol{f}\right) \right\|_2^2 + \lambda_2 \boldsymbol{f}^\top \boldsymbol{\Delta} \boldsymbol{f} + \gamma \boldsymbol{y}_U^\top \boldsymbol{y}_U; \tag{4.10}$$

for example, see Culp and Ryan (2013) for an in-depth study of a similar graph-based prob-

lem. In the objective function of Optimization (4.10), the vector of decision variables $\boldsymbol{y}_U$ can be easily profiled out as it is only involved in the unlabeled loss and its penalty, i.e.,

$$(\boldsymbol{y}_U - \boldsymbol{f}_U)^\top \boldsymbol{w}_{UU} (\boldsymbol{y}_U - \boldsymbol{f}_U) + \gamma \boldsymbol{y}_U^\top \boldsymbol{y}_U, \tag{4.11}$$

solving the $\boldsymbol{y}_U$-score yields of Objective (4.11) yields

$$\begin{aligned} \boldsymbol{w}_{UU} (\boldsymbol{y}_U - \boldsymbol{f}_U) + \gamma \boldsymbol{y}_U &= \vec{0} \\ \boldsymbol{y}_U &= (\boldsymbol{w}_{UU} + \gamma \boldsymbol{I})^{-1} \boldsymbol{w}_{UU} \boldsymbol{f}_U. \end{aligned} \tag{4.12}$$

Solution (4.12) is the optimal $\boldsymbol{y}_U$ at a given $\boldsymbol{f}_U$, and plugging this into Objective (4.11) can be used to show that

$$\begin{aligned} (\boldsymbol{y}_U - \boldsymbol{f}_U)^\top \boldsymbol{w}_{UU} (\boldsymbol{y}_U - \boldsymbol{f}_U) + \gamma \boldsymbol{y}_U^\top \boldsymbol{y}_U &= \left(\vec{0} - \boldsymbol{f}_U\right)^\top [\boldsymbol{V}_{UU}] \left(\vec{0} - \boldsymbol{f}_U\right), \text{ where} \\ \boldsymbol{V} &= \begin{pmatrix} \boldsymbol{V}_{LL} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{V}_{UU} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{w}_{LL} & \boldsymbol{0} \\ \boldsymbol{0} & \gamma \boldsymbol{w}_{UU} (\boldsymbol{w}_{UU} + \gamma \boldsymbol{I})^{-1} \end{pmatrix}. \end{aligned}$$

Thus, the fits

$$\widehat{\boldsymbol{\eta}} = \widehat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \left(\boldsymbol{y}(\vec{0}) - \boldsymbol{f}\right)^\top \boldsymbol{V} \left(\boldsymbol{y}(\vec{0}) - \boldsymbol{f}\right) + \lambda_2 \boldsymbol{f}^\top \boldsymbol{\Delta} \boldsymbol{f} \tag{4.13}$$

equal the fits $\widehat{\boldsymbol{\eta}} = \widehat{\boldsymbol{f}}$ from Optimization (4.10). Solving the $\boldsymbol{f}$-score of the objective from Optimization (4.13) results in the closed-formula

$$\begin{aligned} -\boldsymbol{V} \left(\boldsymbol{y}(\vec{0}) - \widehat{\boldsymbol{f}}\right) + \lambda_2 \boldsymbol{\Delta} \widehat{\boldsymbol{f}} &= \vec{0} \\ (\boldsymbol{V} + \lambda_2 \boldsymbol{\Delta}) \widehat{\boldsymbol{f}} &= \boldsymbol{V} \boldsymbol{y}(\vec{0}) \\ \widehat{\boldsymbol{f}} &= (\boldsymbol{V} + \lambda_2 \boldsymbol{\Delta})^{-1} \boldsymbol{V} \boldsymbol{y}(\vec{0}). \end{aligned} \tag{4.14}$$

When $\gamma = \infty$, $\boldsymbol{V}_{UU} = \boldsymbol{w}_{UU}$, so $\boldsymbol{V} = \boldsymbol{w}$. In this context with $\gamma = \infty$, the fits $\widehat{\boldsymbol{f}}$ are a manifold

averaging with 0 imputations for the missing unlabeled responses.

As for $\gamma = 0$, partitioning the Equations (4.14) to

$$\begin{pmatrix} \boldsymbol{w}_{LL} + \lambda_2 \boldsymbol{\Delta}_{LL} & \lambda_2 \boldsymbol{\Delta}_{LU} \\ \lambda_2 \boldsymbol{\Delta}_{UL} & \boldsymbol{V}_{UU} + \lambda_2 \boldsymbol{\Delta}_{UU} \end{pmatrix} \begin{pmatrix} \widehat{\boldsymbol{f}}_L \\ \widehat{\boldsymbol{f}}_U \end{pmatrix} = \begin{pmatrix} \boldsymbol{w}_{LL} \boldsymbol{y}_L \\ \vec{0} \end{pmatrix}$$

by the labeled and unlabeled sets is particularly informative. When $\gamma = 0$, $\boldsymbol{V}_{UU} = \boldsymbol{0}$, and the unlabeled fits

$$-\boldsymbol{\Delta}_{UU} \widehat{\boldsymbol{f}}_U = \boldsymbol{\Delta}_{UL} \widehat{\boldsymbol{f}}_L.$$

satisfy a harmonic property. If for example the labeled responses are constant on a (possibly non-elliptical) manifold, this harmonic property uses that particular observed constant response value as the predicted value throughout the manifold on both the labeled and unlabeled cases (Culp and Ryan, 2013).

4.4.2 The S3KM when $\lambda_2 = \infty$

More insight into the varied types of possible predictions obtained from Optimization (4.8) is gleaned when $\lambda_2 = \infty$. Then the vector $\boldsymbol{f}$ is in the null space of the graph Laplacian $\boldsymbol{\Delta}$ from Equation (4.3). Null vectors indicate the connected components of the graph $\boldsymbol{\omega}$ used to compute $\boldsymbol{\Delta}$, and these null vectors represent the manifolds in the feature space (Culp and Ryan, 2013). In the context of a fully connected graph $\boldsymbol{\omega}$, the all ones vectors $\vec{1} \in \mathbb{R}^n$ is a basis for the null space of $\boldsymbol{\Delta}$. In particular, penalty $\lambda_2 \boldsymbol{f}^\top \boldsymbol{\Delta} \boldsymbol{f} = 0$ whenever $\boldsymbol{f} = c\vec{1}$ for some scalar $c \in \mathbb{R}$, so when $\lambda_2 = \infty$, there is no penalty for centering the response with say a weighted mean of $\boldsymbol{y}_L$. In this section, we thus consider

$$\left(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{y}}_U\right) = \underset{\boldsymbol{\alpha}, \boldsymbol{y}_U}{\arg\min} \left\| \boldsymbol{w}^{1/2} \left(\boldsymbol{y}(\boldsymbol{y}_U) - \boldsymbol{K}\boldsymbol{\alpha} \right) \right\|_2^2 + \lambda_1 \boldsymbol{\alpha}^\top \boldsymbol{K} \boldsymbol{\alpha} + \gamma \boldsymbol{y}_U^\top \boldsymbol{y}_U \tag{4.15}$$

with no graph term and its fits of

$$\widehat{\boldsymbol{\eta}} = \boldsymbol{K}\widehat{\boldsymbol{\alpha}} = \begin{pmatrix} \boldsymbol{K}_{LL}\widehat{\boldsymbol{\alpha}}_L + \boldsymbol{K}_{LU}\widehat{\boldsymbol{\alpha}}_U \\ \boldsymbol{K}_{UL}\widehat{\boldsymbol{\alpha}}_L + \boldsymbol{K}_{UU}\widehat{\boldsymbol{\alpha}}_U \end{pmatrix} \tag{4.16}$$

as a proxy of Optimization (4.8) when $\lambda_2 = \infty$.

A Supervised Safety Parameter Setting: In the special case of $\gamma = 0$, Optimization (4.15) reduces to

$$\widehat{\boldsymbol{\alpha}}_0 = \underset{\boldsymbol{\alpha}\in\mathbb{R}^n}{\arg\min} \left\| \boldsymbol{w}_{LL}^{1/2} (\boldsymbol{y}_L - \boldsymbol{K}_{LL}\boldsymbol{\alpha}_L - \boldsymbol{K}_{LU}\boldsymbol{\alpha}_U) \right\|_2^2 + \lambda_1 \boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha}. \tag{4.17}$$

with corresponding fits of

$$\widehat{\boldsymbol{\eta}}_0 = \boldsymbol{K}\widehat{\boldsymbol{\alpha}}_0 = \begin{pmatrix} \boldsymbol{K}_{LL}\widehat{\boldsymbol{\alpha}}_{0L} + \boldsymbol{K}_{LU}\widehat{\boldsymbol{\alpha}}_{0U} \\ \boldsymbol{K}_{UL}\widehat{\boldsymbol{\alpha}}_{0L} + \boldsymbol{K}_{UU}\widehat{\boldsymbol{\alpha}}_{0U} \end{pmatrix}. \tag{4.18}$$

Optimization (4.17) appears to depend on the unlabeled portions of the full kernel matrix $\boldsymbol{K}$ given in Equation (4.1). In spite of this, we establish (later in this section in Theorem 2) that Optimization (4.17) and its Fits (4.18) are equivalent to supervised kernel regression

$$\widehat{\widetilde{\boldsymbol{\alpha}}} = \underset{\widetilde{\boldsymbol{\alpha}}\in\mathbb{R}^m}{\arg\min} \left\| \boldsymbol{w}_{LL}^{1/2} (\boldsymbol{y}_L - \boldsymbol{K}_{LL}\widetilde{\boldsymbol{\alpha}}) \right\|_2^2 + \lambda \widetilde{\boldsymbol{\alpha}}^\top \boldsymbol{K}_{LL}\widetilde{\boldsymbol{\alpha}} \tag{4.19}$$

and its fits

$$\widehat{\boldsymbol{\eta}} = \begin{pmatrix} \boldsymbol{K}_{LL} \\ \boldsymbol{K}_{UL} \end{pmatrix} \widehat{\widetilde{\boldsymbol{\alpha}}}. \tag{4.20}$$

While we do not contend that solution $\widehat{\widetilde{\boldsymbol{\alpha}}}$ is unique, we do provide a concise matrix representation for Fits (4.20) in Lemma 2 that is guaranteed to be unique for any $\lambda > 0$. The proof to Lemma 2 relies on the well-known matrix identity established in Lemma 1.

Lemma 1. *If $\boldsymbol{K}$ is a nonnegative definite matrix with Partition (4.1), then $\boldsymbol{K}_{UL} = \boldsymbol{K}_{UL}\boldsymbol{K}_{LL}^{-}\boldsymbol{K}_{LL}$ for any choice of generalized inverse $\boldsymbol{K}_{LL}^{-}$ of $\boldsymbol{K}_{LL}$ such that $\boldsymbol{K}_{LL} = \boldsymbol{K}_{LL}\boldsymbol{K}_{LL}^{-}\boldsymbol{K}_{LL}$.*

Proof. If

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{I} - \boldsymbol{K}_{LL}^{-}\boldsymbol{K}_{LL} \\ \boldsymbol{0} \end{pmatrix},$$

then

$$\boldsymbol{B}^{\top}\boldsymbol{K}\boldsymbol{B} = \boldsymbol{B}^{\top} \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{K}_{UL} - \boldsymbol{K}_{UL}\boldsymbol{K}_{LL}^{-}\boldsymbol{K}_{LL} \end{pmatrix} = \boldsymbol{0}.$$

Since $\boldsymbol{K}$ nonnegative definite implies $\boldsymbol{K} = \boldsymbol{A}^{\top}\boldsymbol{A}$ for some matrix $\boldsymbol{A}$,

$$\boldsymbol{B}^{\top}\boldsymbol{K}\boldsymbol{B} = \boldsymbol{B}^{\top}\boldsymbol{A}^{\top}\boldsymbol{A}\boldsymbol{B} = \boldsymbol{0} \Rightarrow \boldsymbol{A}\boldsymbol{B} = \boldsymbol{0} \Rightarrow \boldsymbol{A}^{\top}\boldsymbol{A}\boldsymbol{B} = \boldsymbol{0} \Rightarrow \boldsymbol{K}\boldsymbol{B} = \boldsymbol{0}.$$

So, with $\boldsymbol{K}\boldsymbol{B} = \boldsymbol{0}$, we must have

$$\boldsymbol{K}\boldsymbol{B} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{K}_{UL} - \boldsymbol{K}_{UL}\boldsymbol{K}_{LL}^{-}\boldsymbol{K}_{LL} \end{pmatrix} = \boldsymbol{0} \Rightarrow \boldsymbol{K}_{UL} = \boldsymbol{K}_{UL}\boldsymbol{K}_{LL}^{-}\boldsymbol{K}_{LL}.$$

□

Lemma 2. *The full $n \times 1$ Fits (4.20) based on Supervised Optimization (4.19) is*

$$\widehat{\boldsymbol{\eta}} = \begin{pmatrix} \boldsymbol{K}_{LL} \\ \boldsymbol{K}_{UL} \end{pmatrix} \widehat{\widetilde{\boldsymbol{\alpha}}} = \begin{pmatrix} \boldsymbol{K}_{LL} \\ \boldsymbol{K}_{UL} \end{pmatrix} \boldsymbol{w}_{LL}(\boldsymbol{K}_{LL}\boldsymbol{w}_{LL} + \lambda \boldsymbol{I})^{-1}\boldsymbol{y}_L. \tag{4.21}$$

Proof. We start by finding the labeled portion of the fits vector $\widehat{\boldsymbol{\eta}}_L$. To do this, we take the $\widetilde{\boldsymbol{\alpha}}$-score of the objective function and solve for $\boldsymbol{K}_{LL}\widehat{\widetilde{\boldsymbol{\alpha}}}$, i.e.,

$$\begin{aligned} -\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}(\boldsymbol{y}_L - \boldsymbol{K}_{LL}\widehat{\widetilde{\boldsymbol{\alpha}}}) + \lambda \boldsymbol{K}_{LL}\widehat{\widetilde{\boldsymbol{\alpha}}} &= \vec{0} \qquad (4.22) \\ (\boldsymbol{K}_{LL}\boldsymbol{w}_{LL} + \lambda \boldsymbol{I})\boldsymbol{K}_{LL}\widehat{\widetilde{\boldsymbol{\alpha}}} &= \boldsymbol{K}_{LL}\boldsymbol{w}_{LL}\boldsymbol{y}_L \\ \boldsymbol{K}_{LL}\widehat{\widetilde{\boldsymbol{\alpha}}} &= (\boldsymbol{K}_{LL}\boldsymbol{w}_{LL} + \lambda \boldsymbol{I})^{-1}\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}\boldsymbol{y}_L \\ \boldsymbol{K}_{LL}\widehat{\widetilde{\boldsymbol{\alpha}}} &= \boldsymbol{K}_{LL}\boldsymbol{w}_{LL}(\boldsymbol{K}_{LL}\boldsymbol{w}_{LL} + \lambda \boldsymbol{I})^{-1}\boldsymbol{y}_L. \qquad (4.23) \end{aligned}$$

A matrix multiplication in Equation (4.23) is commutative because

$$\begin{aligned}
(\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}+\lambda\boldsymbol{I})^{-1}\boldsymbol{K}_{LL}\boldsymbol{w}_{LL} &= \boldsymbol{w}_{LL}^{-1/2}(\boldsymbol{w}_{LL}^{1/2}\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}^{1/2}+\lambda\boldsymbol{I})^{-1}\boldsymbol{w}_{LL}^{1/2}\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}^{1/2}\boldsymbol{w}_{LL}^{1/2} \\
&= \boldsymbol{w}_{LL}^{-1/2}\left[(\boldsymbol{w}_{LL}^{1/2}\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}^{1/2}+\lambda\boldsymbol{I})^{-1}\right]\left[\boldsymbol{w}_{LL}^{1/2}\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}^{1/2}\right]\boldsymbol{w}_{LL}^{1/2} \\
&= \boldsymbol{w}_{LL}^{-1/2}\left[\boldsymbol{w}_{LL}^{1/2}\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}^{1/2}\right]\left[(\boldsymbol{w}_{LL}^{1/2}\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}^{1/2}+\lambda\boldsymbol{I})^{-1}\right]\boldsymbol{w}_{LL}^{1/2} \\
&= \boldsymbol{K}_{LL}\boldsymbol{w}_{LL}(\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}+\lambda\boldsymbol{I})^{-1}
\end{aligned}$$

since the pair of symmetric matrices in the square brackets have the same eigenvectors. With the labeled fits of $\widehat{\widetilde{\boldsymbol{\eta}}}_L = \boldsymbol{K}_{LL}\widehat{\widetilde{\boldsymbol{\alpha}}}$ in Equation (4.23), the unlabeled fits of

$$\begin{aligned}
\widehat{\widetilde{\boldsymbol{\eta}}}_U &= \boldsymbol{K}_{UL}\widehat{\widetilde{\boldsymbol{\alpha}}} \\
&= \boldsymbol{K}_{UL}\boldsymbol{K}_{LL}^{-}\boldsymbol{K}_{LL}\widehat{\widetilde{\boldsymbol{\alpha}}} \\
&= \boldsymbol{K}_{UL}\boldsymbol{K}_{LL}^{-}\widehat{\widetilde{\boldsymbol{\eta}}}_L
\end{aligned}$$

follow by Lemma 1, so Fits (4.21) are established on the labeled and unlabeled sets.

□

Theorem 2. *Setting $\gamma = 0$ in Optimization (4.15) results in a supervised approach. (In particular, Optimization (4.15) with $\gamma = 0$ results in Optimization (4.17), and the Fits (4.18) of Optimization (4.17) equal the Supervised Fits (4.20) of Optimization (4.19), i.e., $\widehat{\widetilde{\boldsymbol{\eta}}} = \widehat{\boldsymbol{\eta}}_0$.)*

Proof. In the theorem statement, the first sentence is a logical consequence of the second sentence. With this in mind, the proof is similar to that of Lemma 2, and we start by solving the $\boldsymbol{\alpha}_L$-score of the objective from Optimization (4.17) for $\widehat{\boldsymbol{\eta}}_{0L} = \boldsymbol{K}_{LL}\widehat{\boldsymbol{\alpha}}_{0L} + \boldsymbol{K}_{LU}\widehat{\boldsymbol{\alpha}}_{0U}$, i.e.,

$$\begin{aligned}
-\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}(\boldsymbol{y}_L - \widehat{\boldsymbol{\eta}}_{0L}) + \lambda\widehat{\boldsymbol{\eta}}_{0L} &= \vec{0} \\
(\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}+\lambda\boldsymbol{I})\widehat{\boldsymbol{\eta}}_{0L} &= \boldsymbol{K}_{LL}\boldsymbol{w}_{LL}\boldsymbol{y}_L \\
\widehat{\boldsymbol{\eta}}_{0L} &= (\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}+\lambda\boldsymbol{I})^{-1}\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}\boldsymbol{y}_L \\
\widehat{\boldsymbol{\eta}}_{0L} &= \boldsymbol{K}_{LL}\boldsymbol{w}_{LL}(\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}+\lambda\boldsymbol{I})^{-1}\boldsymbol{y}_L \\
\widehat{\boldsymbol{\eta}}_{0L} &= \widehat{\widetilde{\boldsymbol{\eta}}}_L. \qquad (4.24)
\end{aligned}$$

As for the unlabeled fits, these are obtained by solving the $\boldsymbol{\alpha}_U$-score of the objective from Optimization (4.17) for $\widehat{\boldsymbol{\eta}}_U = \boldsymbol{K}_{UL}\widehat{\boldsymbol{\alpha}}_L + \boldsymbol{K}_{UU}\widehat{\boldsymbol{\alpha}}_U$, i.e.,

$$\begin{aligned}
-\boldsymbol{K}_{UL}\boldsymbol{w}_{LL}(\boldsymbol{y}_L - \widehat{\boldsymbol{\eta}}_{0L}) + \lambda\widehat{\boldsymbol{\eta}}_{0U} &= \vec{0} \\
\widehat{\boldsymbol{\eta}}_{0U} &= \boldsymbol{K}_{UL}\left(\boldsymbol{w}_{LL} - \boldsymbol{w}_{LL}(\boldsymbol{K}_{LL}\boldsymbol{w}_{LL} + \lambda\boldsymbol{I})^{-1}\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}\right)\boldsymbol{y}_L/\lambda \\
\widehat{\boldsymbol{\eta}}_{0U} &= \boldsymbol{K}_{UL}\left[\boldsymbol{w}_{LL} - \left(\boldsymbol{K}_{LL} + \lambda\boldsymbol{w}_{LL}^{-1}\right)^{-1}\boldsymbol{K}_{LL}\boldsymbol{w}_{LL}\right]\boldsymbol{y}_L/\lambda \\
\widehat{\boldsymbol{\eta}}_{0U} &= \boldsymbol{K}_{UL}\left(\boldsymbol{K}_{LL} + \lambda\boldsymbol{w}_{LL}^{-1}\right)^{-1}\left[\lambda\boldsymbol{w}_{LL}^{-1}\boldsymbol{w}_{LL}\right]\boldsymbol{y}_L/\lambda \\
\widehat{\boldsymbol{\eta}}_{0U} &= \widehat{\widetilde{\boldsymbol{\eta}}}_U. \qquad (4.25)
\end{aligned}$$

Equations (4.24) and (4.25) complete the proof of $\widehat{\widetilde{\boldsymbol{\eta}}} = \widehat{\boldsymbol{\eta}}_0$.

□

Theorem 2 establishes the effect of setting $\gamma = 0$ as being supervised, and this provides a built in safety of the S3KM in the following sense. Data analysis can as needed default to a near supervised approach if during CV parameter estimates of $\widehat{\lambda_2} = \infty$ and $\widehat{\gamma} = 0$ are obtained.

Connections to Generalized Ridge Regression: The γ parameter of Optimization (4.8) shrinks the latent unlabeled response estimates $\widehat{\boldsymbol{y}}_U$. For example, in the positive extreme of $\gamma = \infty$, Optimization (4.15) reduces to

$$\widehat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \left\| \boldsymbol{w}^{1/2}\left(\boldsymbol{y}(\vec{0}) - \boldsymbol{K}\boldsymbol{\alpha}\right)\right\|_2^2 + \lambda_1\boldsymbol{\alpha}^\top\boldsymbol{K}\boldsymbol{\alpha}. \qquad (4.26)$$

As before, use the spectral decomposition $\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^\top$ to get an $n \times r$ matrix $\boldsymbol{A}$ such that $\boldsymbol{K} = \boldsymbol{A}\boldsymbol{A}^\top$. Then the induced ridge regression problem

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\| \boldsymbol{w}^{1/2}\left(\boldsymbol{y}(\vec{0}) - \boldsymbol{A}\boldsymbol{\beta}\right)\right\|_2^2 + \lambda_1\boldsymbol{\beta}^\top\boldsymbol{\beta}$$

results in the same fits

$$\widehat{\boldsymbol{\eta}} = \boldsymbol{K}\widehat{\boldsymbol{\alpha}} = \boldsymbol{A}\widehat{\boldsymbol{\beta}}$$

as the Kernel-Based Optimization (4.26). A further equivalent simplification of

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\| \boldsymbol{w}_{LL}^{1/2} \left(\boldsymbol{y}_L - \boldsymbol{A}_L \boldsymbol{\beta} \right) \right\|_2^2 + \boldsymbol{\beta}^\top \left(\boldsymbol{A}_U^T \boldsymbol{w}_{UU} \boldsymbol{A}_U + \lambda_1 \boldsymbol{I} \right) \boldsymbol{\beta} \tag{4.27}$$

follows. This is a generalized ridge regression problem, but in this case, the form of the penalty depends on the unlabeled data.

Deeper insight into the unlabeled influence of this penalty follows for a special setting of the case weights. Let $\gamma_1 > 0$ be a new tuning parameter and suppose that $\boldsymbol{w}_i = 1 + (\gamma_1 - 1)\, \mathbf{1}_{\{i \in U\}}$ for $i = 1, \ldots, n$. Then the solution to Optimization (4.27) is

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{I} + \gamma_1 \boldsymbol{M})^{-1} \widehat{\boldsymbol{\beta}}_{\boldsymbol{A}_L},$$

where $\boldsymbol{M} = \left(\boldsymbol{A}_L^\top \boldsymbol{A}_L + \lambda_1 \boldsymbol{I} \right)^{-1} \boldsymbol{A}_U^\top \boldsymbol{A}_U$, and $\widehat{\boldsymbol{\beta}}_{\boldsymbol{A}_L} = \left(\boldsymbol{A}_L^T \boldsymbol{A}_L + \lambda_1 \boldsymbol{I} \right)^{-1} \boldsymbol{A}_L^\top \boldsymbol{y}_L$. This results in a kernel-based generalization of the semi-supervised extreme for ridge regression as defined in Ryan and Culp (2015). Specifically, projecting $\widehat{\boldsymbol{\beta}}_{\boldsymbol{A}_L}$ onto the eigen-decomposition $\{\tau_i, \boldsymbol{\phi}_i\}_{i=1}^r$ of matrix $\boldsymbol{M}$ yields $\widehat{\boldsymbol{\beta}}_{\boldsymbol{A}_L} = \sum_{i=1}^r c_i \boldsymbol{\phi}_i$ which in-turn yields

$$\widehat{\boldsymbol{\beta}} = \left(\frac{c_1}{1 + \gamma_1 \tau_1} \right) \boldsymbol{\phi}_1 + \cdots + \left(\frac{c_r}{1 + \gamma_1 \tau_r} \right) \boldsymbol{\phi}_r.$$

Vector $\boldsymbol{A}_U \boldsymbol{\phi}_1$ will receive the most shrinking for larger γ_1 while $\boldsymbol{A}_U \boldsymbol{\phi}_r$ receives the least amount of shrinking. Ryan and Culp (2015) called $\boldsymbol{A}_U \boldsymbol{\phi}_1$ the direction of largest unlabeled extrapolation. Finite positive γ in Optimization (4.15) has the effect of forcing $\boldsymbol{M} \longrightarrow \left(\boldsymbol{A}_L^\top \boldsymbol{A}_L + \lambda_1 \boldsymbol{I} \right)^{-1}$. This leads to the corresponding $\boldsymbol{\alpha}$ solutions $\widehat{\boldsymbol{\alpha}}$ to this optimization to behave more like the supervised kernel regression estimate, because the $\gamma = 0$ case of Optimization (4.15) is a supervised kernel regression by Theorem 2.

4.5 S3KM Extensions to Classification

The S3KM approach is extended to classification problems. For classification, now assume that $\boldsymbol{y}_i \in \{\pm 1\}$ for each $i \in L$. The goal is to define a prediction rule to obtain class probability estimates for any new observation $\boldsymbol{x}_0$, i.e., estimate the probability $\boldsymbol{y}_0 = 1$ given $\boldsymbol{x}_0$. The proposed Optimization (4.8) is extended to logistic regression for this purpose.

The S3KM under a logistic loss function extends Optimization (4.8) to

$$\left(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{f}}, \widehat{\boldsymbol{y}}_U\right) = \underset{\boldsymbol{\alpha}, \boldsymbol{f}, \boldsymbol{y}_U}{\arg\min} \sum_{i \in L} \log\left(1 + e^{-2\boldsymbol{y}_i \boldsymbol{\eta}_i}\right) + \|(\boldsymbol{y}_U - \boldsymbol{\eta}_U)\|_2^2 + \lambda_1 \boldsymbol{\alpha}^\top \boldsymbol{K} \boldsymbol{\alpha} + \lambda_2 \boldsymbol{f}^\top \boldsymbol{\Delta} \boldsymbol{f} + \gamma \boldsymbol{y}_U^\top \boldsymbol{y}_U \tag{4.28}$$

with $\boldsymbol{\eta} = \boldsymbol{f} + \boldsymbol{K}\boldsymbol{\alpha}$. As established above in Section 4.4.1, the positive semi-definite kernel matrix $\boldsymbol{K}$ has the representation $\boldsymbol{K} = \boldsymbol{A}\boldsymbol{A}^\top$ for some $n \times r$ matrix $\boldsymbol{A}$ with $r \leq n$. From this, Optimization (4.8) with $\boldsymbol{\beta} = \boldsymbol{A}^\top \boldsymbol{\alpha}$ reduces to

$$\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{f}}, \widehat{\boldsymbol{y}}_U\right) = \underset{\boldsymbol{\beta}, \boldsymbol{f}, \boldsymbol{y}_U}{\arg\min} \sum_{i \in L} \log\left(1 + e^{-2\boldsymbol{y}_i \boldsymbol{\eta}_i}\right) + \|(\boldsymbol{y}_U - \boldsymbol{\eta}_U)\|_2^2 + \lambda_1 \boldsymbol{\beta}^\top \boldsymbol{\beta} + \lambda_2 \boldsymbol{f}^\top \boldsymbol{\Delta} \boldsymbol{f} + \gamma \boldsymbol{y}_U^\top \boldsymbol{y}_U \tag{4.29}$$

with $\boldsymbol{\eta} = \boldsymbol{f} + \boldsymbol{A}\boldsymbol{\beta}$. Theorem 3 simplifies the joint optimization problem to an equivalent problem in decision variables $\boldsymbol{f}_L$ and $\boldsymbol{\beta}$ by profiling out decision variables $\boldsymbol{f}_U$ and $\boldsymbol{y}_U$.

Theorem 3. *There exists a* $(r + |L|) \times (r + |L|)$ *positive semi-definite matrix*

$$\boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{\Gamma}_{11} & \boldsymbol{\Gamma}_{12} \\ \boldsymbol{\Gamma}_{12}^\top & \boldsymbol{\Gamma}_{22} \end{pmatrix}$$

such that

$$\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{f}}_L\right) = \underset{\boldsymbol{\beta}, \boldsymbol{f}_L}{\arg\min} \sum_{i \in L} \log\left(1 + e^{-2\boldsymbol{y}_i \boldsymbol{\eta}_i}\right) + \begin{pmatrix} \boldsymbol{f}_L^\top & \boldsymbol{\beta}^\top \end{pmatrix} \boldsymbol{\Gamma} \begin{pmatrix} \boldsymbol{f}_L \\ \boldsymbol{\beta} \end{pmatrix} \tag{4.30}$$

if and only if $\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{f}}, \widehat{\boldsymbol{y}}_U\right)$ *solves (4.29) for any* $\left(\widehat{\boldsymbol{y}}_U, \widehat{\boldsymbol{f}}_U\right)$ *satisfying*

$$\begin{aligned}\left(\lambda_2 \boldsymbol{\Delta}_{UU} + \frac{\gamma}{1+\gamma}\boldsymbol{I}\right)\widehat{\boldsymbol{f}}_U &= -\left(\boldsymbol{\Delta}_{UL}\boldsymbol{f}_L + \tfrac{\gamma}{1+\gamma}\boldsymbol{A}_U\boldsymbol{\beta}\right) \\ \widehat{\boldsymbol{y}}_U &= \frac{\gamma}{1+\gamma}\widehat{\boldsymbol{\eta}}_U.\end{aligned}$$

Proof. To begin, differentiating Objective (4.29) with respect to $\boldsymbol{y}_U$ yields

$$\boldsymbol{y}_U = \frac{1}{1+\gamma}\boldsymbol{\eta}_U. \tag{4.31}$$

Plugging Constraint (4.31) into the terms involving $\boldsymbol{y}_U$ reduces Objective (4.29) to

$$\|(\boldsymbol{y}_U - \boldsymbol{\eta}_U)\|_2^2 + \gamma \boldsymbol{y}_U^\top \boldsymbol{y}_U = \left(\left(1 - \tfrac{\gamma}{1+\gamma}\right)^2 + \tfrac{\gamma}{(1+\gamma)^2}\right)\boldsymbol{\eta}_U^\top\boldsymbol{\eta}_U = \tfrac{\gamma}{1+\gamma}\boldsymbol{\eta}_U^\top\boldsymbol{\eta}_U.$$

Parameter $\boldsymbol{y}_U$ is then profiled out of Objective (4.29) leading to optimization

$$\left(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{f}}\right) = \underset{\boldsymbol{\alpha}, \boldsymbol{f}}{\arg\min} \sum_{i \in L} \log\left(1 + e^{-\boldsymbol{y}_i \boldsymbol{\eta}_i}\right) + \lambda_1 \boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} + \lambda_2 \boldsymbol{f}^\top \boldsymbol{\Delta}\boldsymbol{f} + \tfrac{\gamma}{1+\gamma}\boldsymbol{\eta}_U^\top\boldsymbol{\eta}_U. \tag{4.32}$$

Taking the gradient of Objective (4.32) with respect to $\boldsymbol{f}_U$ results in score

$$\lambda_2 \boldsymbol{\Delta}_{UU}\boldsymbol{f}_U + \lambda_2 \boldsymbol{\Delta}_{UL}\boldsymbol{f}_L + \tfrac{\gamma}{1+\gamma}\boldsymbol{f}_U + \tfrac{\gamma}{1+\gamma}\boldsymbol{A}_U\boldsymbol{\beta} = \vec{0},$$

solving for $\boldsymbol{f}_U$ produces

$$\boldsymbol{f}_U = -\left(\lambda_2 \boldsymbol{\Delta}_{UU} + \tfrac{\gamma}{1+\gamma}\boldsymbol{I}\right)^{-1}\left(\boldsymbol{\Delta}_{UL}\boldsymbol{f}_L + \tfrac{\gamma}{1+\gamma}\boldsymbol{A}_U\boldsymbol{\beta}\right).$$

Algorithm 1 Logistic Version of the S3KM

1: **Input** $\boldsymbol{y}_L \in \{-1,1\}^{|L|}$, $\boldsymbol{A}$, $\boldsymbol{\Delta}$, and $(\lambda_1, \lambda_2, \gamma)$.
2: **Initialize** $\widehat{\boldsymbol{\eta}} = \vec{0}$.
3: **Repeat**
4: **Set** the weights vector with components

$$\boldsymbol{\mu}_i = \frac{\exp\left(2\widehat{\boldsymbol{\eta}}_i\right)}{\left(1+\exp\left(2\widehat{\boldsymbol{\eta}}_i\right)\right)}.$$

5: **Compute** the linearized response

$$z_i = \boldsymbol{\eta}_i + \frac{\frac{\boldsymbol{y}_i+1}{2} - \boldsymbol{\mu}_i}{\boldsymbol{\mu}_i(1-\boldsymbol{\mu}_i)}.$$

6: **Solve**

$$\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{f}}_L\right) = \underset{\boldsymbol{\beta}, \boldsymbol{f}_L}{\arg\min} \sum_{i \in L} \boldsymbol{\mu}_i(1-\boldsymbol{\mu}_i)(z_i - \boldsymbol{\eta}_i)^2 + \begin{pmatrix} \boldsymbol{f}_L^\top & \boldsymbol{\beta}^\top \end{pmatrix} \boldsymbol{\Gamma} \begin{pmatrix} \boldsymbol{f}_L \\ \boldsymbol{\beta} \end{pmatrix}.$$

7: **Update** $\widehat{\boldsymbol{\eta}}_L = \boldsymbol{A}_L \widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{f}}_L$.
8: **Until** $\widehat{\boldsymbol{\eta}}$ converges.
9: **Compute** $\widehat{\boldsymbol{\eta}}_U$ and $\widehat{\boldsymbol{y}}_U$ as described in Theorem 3.

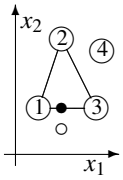
Plugging this $\boldsymbol{f}_U$ into Equation (4.32) identifies the partitions of $\boldsymbol{\Gamma}$ as

$$\begin{aligned}
\boldsymbol{\Gamma}_{11} &= \lambda_1 \boldsymbol{\Delta}_{LL} - \lambda_1^2 \boldsymbol{\Delta}_{LU} \left(\lambda_2 \boldsymbol{\Delta}_{UU} + \tfrac{\gamma}{1+\gamma}\right)^{-1} \boldsymbol{\Delta}_{UL} \\
\boldsymbol{\Gamma}_{12} &= -\tfrac{\gamma}{1+\gamma} \lambda_1 \boldsymbol{\Delta}_{LU} \left(\lambda_2 \boldsymbol{\Delta}_{UU} + \tfrac{\gamma}{1+\gamma}\right)^{-1} \boldsymbol{A}_U \\
\boldsymbol{\Gamma}_{22} &= \lambda_2 \boldsymbol{I} + \tfrac{\gamma}{1+\gamma} \boldsymbol{A}_U^\top \boldsymbol{A}_U - \left(\tfrac{\gamma}{1+\gamma}\right)^2 \boldsymbol{A}_U^\top \left(\lambda_2 \boldsymbol{\Delta}_{UU} + \tfrac{\gamma}{1+\gamma} \boldsymbol{I}\right)^{-1} \boldsymbol{A}_U.
\end{aligned}$$

□

Theorem 3 establishes that Joint Optimization (4.28) can be re-expressed as a penalized semi-parametric model (Hastie et al., 2009). As such, Logistic Regression Algorithm 1 provides the solution to this optimization problem.

4.6 S3KM Extensions to Anchor Graphs



A computationally-efficient, sparse version of Optimization (4.8) is sought in order to extend the viability of the proposed method to practical big data problems. The anchor point approximation is ideally suited for this. Let $\boldsymbol{Q}$ denote an $a \times p$ matrix of *a anchor points* in $\mathbb{R}^p$. The initial objective is to find a matrix $\boldsymbol{Z}$ so that $\boldsymbol{X}$ is close to $\boldsymbol{ZQ}$. Precisely, each row of $\boldsymbol{Z}$ is restricted to be on a simplex, so the ith row of $\boldsymbol{ZQ}$ is constrained to the convex polytope of the s closest anchor points to the ith row of $\boldsymbol{X}$. Refer to the example on right with $p = 2$, $k = 4$, $s = 3$, $\circ$ is an arbitrary vector $\boldsymbol{x} \in \mathbb{R}^p$, and $\bullet$ is the corresponding projection $\sum_{i=1}^{a} \boldsymbol{Z}_i \boldsymbol{Q}_i$. The vector $\boldsymbol{z}$ is the simplex projection of $\boldsymbol{x}$ onto the convex polygon consisting of the closest $s = 3$ of $k = 4$ anchor points. The Local Anchor Embedding algorithm of Liu et al. (2010) solves for each row of $\boldsymbol{Z}$ by simplex projecting the corresponding row of $\boldsymbol{X}$ in this manner. From this, adjacency matrix $\boldsymbol{Z}\text{diag}\left(\boldsymbol{Z}^T\vec{1}\right)\boldsymbol{Z}^\top$ is the *anchor graph* with Laplacian $\boldsymbol{\Delta} = \boldsymbol{I} - \boldsymbol{Z}\text{diag}\left(\boldsymbol{Z}^T\vec{1}\right)\boldsymbol{Z}^\top$ and *reduced* Laplacian $\widetilde{\boldsymbol{\Delta}} = \boldsymbol{Z}^\top\boldsymbol{\Delta}\boldsymbol{Z}$.

Given an $n \times n$ kernel matrix $\boldsymbol{K} = \boldsymbol{A}\boldsymbol{A}^\top$ of rank r, substituting anchor graphs and linearized functions $\boldsymbol{f} = \boldsymbol{Z}\boldsymbol{\alpha}$ into Optimization (4.8) results in

$$\left(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{y}}_U\right) = \underset{\boldsymbol{\theta}, \boldsymbol{y}_U}{\arg\min} \left\| \boldsymbol{w}^{1/2}\left(\boldsymbol{y}(\boldsymbol{y}_U) - \widetilde{\boldsymbol{X}}\boldsymbol{\theta}\right)\right\|_2^2 + \lambda_1 \boldsymbol{\theta}^\top \boldsymbol{P}\boldsymbol{\theta} + \gamma \boldsymbol{y}_U^\top \boldsymbol{y}_U, \tag{4.33}$$

where decision variables $\boldsymbol{\theta} = \left(\boldsymbol{\beta}^\top, \boldsymbol{\alpha}^\top\right)^\top$ such that $\boldsymbol{\beta} \in \mathbb{R}^r$ and $\boldsymbol{\alpha} \in \mathbb{R}^a$, induced model matrix $\widetilde{\boldsymbol{X}} = [\boldsymbol{A}|\boldsymbol{Z}]$ is an $n \times (r+a)$ columnwise concatenation, and penalty matrix

$$\boldsymbol{P} = \begin{pmatrix} \frac{\lambda_2}{\lambda_1}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \widetilde{\boldsymbol{\Delta}} \end{pmatrix}.$$

Optimization (4.33) has linear fits

$$\widehat{\boldsymbol{\eta}} = \widetilde{\boldsymbol{X}}\widehat{\boldsymbol{\theta}} = \boldsymbol{A}\widehat{\boldsymbol{\beta}} + \boldsymbol{Z}\widehat{\boldsymbol{\alpha}}.$$

We call the solution to Optimization (4.33) the Anchor Safe Semi-Supervised Kernel Model

(AS3KM). The logistic version for classification can be solved in a similar manner.

The complexity of solving the Optimization (4.8) is overshadowed by the need to first carry out a quadratic in n graph construction phase. On the other hand, solving (4.33) only requires an initial linear in n anchor graph construction phase. Computing matrix $\boldsymbol{A}$ from kernel $\boldsymbol{K}$ is an n^3 operation that both techniques require. The anchor graph simplification requires one $a+r$ inverse which is of order $(a+r)^3$. Moreover, local kernel parameter σ^2 is eliminated from the anchor graph method, which leads to fewer parameters for CV. A comparison of Optimizations (4.8) and (4.33) brings into focus a familiar performance versus speed tradeoff: (a) get the best performance by optimizing a computationally intense problem versus (b) get (hopefully) comparable performance results faster by optimizing a problem requiring substantially less computation. This tradeoff is investigated empirically in Section 4.8

4.7 Manifold Regularization: An S3KM Competitor

Many supervised approaches such as those discussed in Chapter 3 including ridge regression, smoothing splines, and SVMs are penalized regression problems. The main competitor for the proposed S3KM is manifold regularization (Belkin et al., 2006). Manifold regularization works in a similar manner, but the regularizer is more complex than the ones discussed in Chapter 3. Manifold regularization is semi-supervised RKHS approach. The RKHS space paradigm is the same as previously established, i.e., denote $\mathscr{H}_K$ as the RKHS with corresponding inner product norm $||f||^2_{\mathscr{H}_K} = \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}$ where $\boldsymbol{K}$ is the kernel Gram matrix constructed on all observations in $L \cup U$. The optimization problem posits a dual functional that penalizes in both an intrinsic and ambient fashion. The geometric penalty uses the intrinsic information in the marginal density of $\boldsymbol{X}$ denoted as $P_{\boldsymbol{X}}$. This is the semi-supervised component of the optimization since the usage of this information relies on the cluster assumption. The authors of Belkin et al. (2006) offer insight from a physics perspective into the geometric penalty for the case when the marginal density of $\boldsymbol{X}$ is unknown. The main result is that $\int_{\boldsymbol{x} \in \mathscr{M}} \langle \nabla_{\mathscr{M}} f, \nabla_{\mathscr{M}} f \rangle dP_{\boldsymbol{X}}(\boldsymbol{x})$ approximates $||f||^2_I$, where $\mathscr{M}$ is a compact sub-manifold

of $\boldsymbol{X}$ and $\nabla_{\mathscr{M}} f$ is the gradient of f with respect to $\mathscr{M}$. The ambient penalty assumes that the desired function should be sufficiently smooth with respect to the RKHS norm $||f||_{\mathscr{H}_K}$. Putting these ideas together, Manifold Regularization proposes

$$f^{\star} = \underset{f \in \mathscr{H}_K}{\arg\min} \sum_{i \in L} V(\boldsymbol{x}_i, \boldsymbol{y}_i, f) + \lambda_1 ||f||^2_{\mathscr{H}_K} + \lambda_2 f^T \boldsymbol{\Delta} f, \tag{4.34}$$

where V is a loss function in their notation. Proceeding as before, the minimizer was proven to be of the form

$$f^{\star}(\boldsymbol{x}) = \sum_{i \in L \cup U} \boldsymbol{\alpha}_i K(\boldsymbol{x}, \boldsymbol{x}_i).$$

The $\widehat{\boldsymbol{\alpha}}$ can be estimated using the dual form of this optimization problem. The manifold regularization method is defined as the solution to Optimization (4.34) with $\lambda_1, \lambda_2 \geq 0$. Special cases are of note. In particular $\lambda_1 \geq 0, \lambda_2 = 0$ results in supervised kernel ridge regression and the SVM depending on the loss. Labeled loss graph regularization results in the case of $\lambda_1 = 0, \lambda_2 \geq 0$ (Culp and Ryan, 2013). It is of note that the graph term only influences the $\boldsymbol{\alpha}$ coefficient, and thus if a prediction $\boldsymbol{x}_0$ is to be performed then the prediction is independent of the proximity graph associated with $\boldsymbol{x}_0$.

The proposed S3KM (4.8) offers more flexibility than manifold regularization. In this case, a model $\boldsymbol{\eta} = \boldsymbol{f} + \boldsymbol{K}\boldsymbol{\alpha}$ is estimated, but the form of the optimization has similar penalty terms. The $\boldsymbol{f}$ component is optimized to account for the intrinsic geometry using the graph Laplacian. The ambient smoothness penalty associated with the Hilbert norm is accounted for separately by $\boldsymbol{\alpha}$. The prediction function differs in that the residual from the graph term is fit within the kernel framework. An interpolation routine over the graph is incorporated to fit this structure for a new point $\boldsymbol{x}_0$ and hence prediction depends on both the intrinsic and ambient information associated with $\boldsymbol{x}_0$. The γ parameter adds a new novelty over the manifold regularization framework in general allowing for the approach to adapt to extrapolations within the manifolds. These flexibilities although subtle in presentation offer a significant difference between the two frameworks. The empirical results in Section 4.8 favor the proposed S3KM over its more rigid manifold regularization competitor.

Table 4.1: Benchmark Data Sets.

Data Set	(n, p)	Type	Response	Reference
Meat Spec	$(215, 100)$	Regress	Fat	Faraway (2016)
Thyroid	$(215, 5)$	Class	Disease	Lichman (2013)
Ionosphere	$(351, 33)$	Class	Radar	Lichman (2013)
Navy	$(11933, 16)$	Regress	GT Decay	Lichman (2013)
Image	$(2310, 18)$	Class	Type	Lichman (2013)
Solubility	$(5631, 72)$	Class	Solubility	Culp et al. (2006)

4.8 Empirical Demonstrations

Semi-supervised performance tests on are performed to assess the main contributions of this Chapter, i.e., the proposed AS3KM and S3KM are compared against manifold regularization (MREG) and a supervised SVM. The results were fit in R (R Core Team, 2016) for the six data sets summarized in Table 4.1.

For this experiment, both the Gaussian kernel (RBF) and Linear kernel were fit. Three-fold CV was used to estimate the parameters on a finite grid. For the AS3KM, parameter settings $s = 5$, $c_n = 4$ and $a = \lceil 0.15 \times n \rceil \mathbf{1}_{\{n \leq 1000\}} + 200 \mathbf{1}_{\{n > 1000\}}$ were used, where k-means centroids were defined as the anchor points. For the S3KM, a k-NN graph with $k = 6$ was fit. The associated σ^2 parameter was estimated using the 0.05,0.50, and 0.95 quantiles on a random sample of 50% of the distances between labeled observations (Karatzoglou et al., 2004), and the value 0.12 was also included in each grid search for σ^2. The tuning parameter for the RBF kernel was estimated using the *sig.est* function from the `kernlab` package (Karatzoglou et al., 2004). MREG used the same k-NN graph, σ^2, and RBF tuning parameter estimation approach. For all three of these techniques, the grid

$$(\lambda_1, \lambda_2) \in \{0.1, 1.0, 2.0, 10.0\} \times \{0.01, 0.1, 1.0, 2.0, 10.0\}$$

was used. The γ parameter was estimated over gird $\{0.0, 0.001, 0.01\}$ for the AS3KM and S3KM, whereas parameter $\gamma = 0$ for MREG. For the SVM, the RBF tuning parameter was estimated in same fashion as above, and λ_1 was optimized over the grid as above. However,

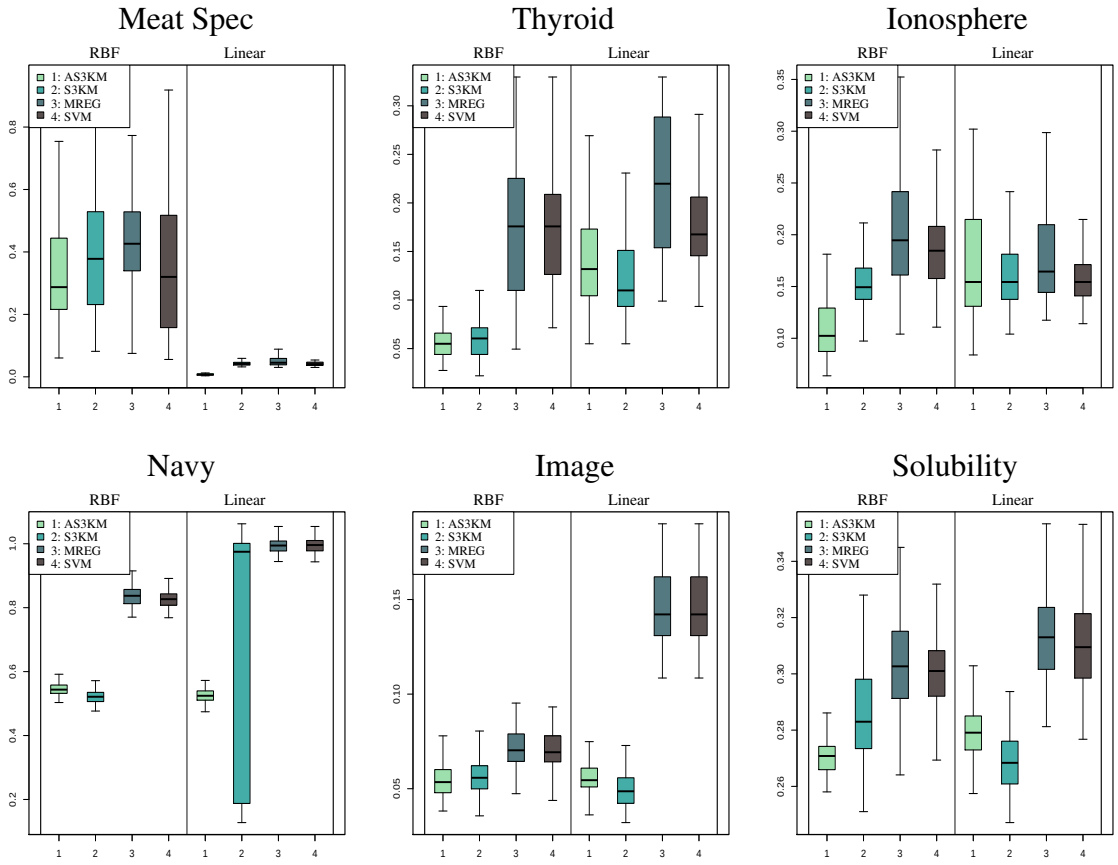


Figure 4.1: Unlabeled Performance on Real Data Sets.

constraints $\lambda_2 = 0$ and $\gamma = 0$ were used to fit a supervised SVM. Fifteen percent was used for labeled training percentage. The process was repeated 100 times per kernel, and the unlabeled error was recorded.

The results in Figure 4.1 come out very strongly in favor of the proposed approaches: S3KM and AS3KM. In all cases, they were as good or better than MREG and a supervised SVM. The proposed kernel based approaches optimize two functions separately for the graph and kernel part of Optimization (4.8), while MREG offers one function with two regularizers. This idea is less flexible in practice and does not perform as strongly on bottom-line metrics as presented here.

CHAPTER 5

DISCUSSION AND FUTURE DIRECTIONS

5.1 General Discussion

The need to assess ordinal measurement systems motivated the work in Chapter 2. With this in mind, a random effects model was developed in Section 2.3, and the surprisingly simple Bayesian `jags` program in Section 2.B made this all work. The portion of this effort in Section 2.4 necessarily concentrated on defining parametric functions that adequately measured repeatability and reproducibility (R&R), and the approach used leaned on the definition of R&R traditionally used for gauge R&R with a continuous response. In this sense, the proposed modeling and terminology extended the literature in a logical fashion for practical use.

A Reproducing Kernel Hilbert Space (RKHS) framework for machine learning problems was carefully outlined and motivated in Chapter 3. Classical Euclidean spaced prediction approaches were initially motivated for machine learning problems including ridge regression and smoothing splines in Chapter 3.1. The smoothing spline approach was directly

extended to an RKHS in Section 3.2 as a first step. Practical shortcomings of this technique were noted, so a more useful Hilbert space construction using Mercer kernels was presented in Section 3.2.1. This led to the main framework used for the machine learning contributions in this dissertation. However, before we proceeded, a negative result was presented regarding the practical use of complex loss functions. As noted, the use of kernel techniques in machine learning has led to these more complex loss functions as potentially powerful learning methods. Two such popular approaches were presented in Section 3.3, but they did not facilitate improvements. This informed the direction of the more substantial contribution in Chapter 4 and justified our usage of squared error loss and logistic loss.

In Chapter 4, a kernel based semi-supervised method was developed for real-data prediction problems. It was established that most semi-supervised techniques are motivated with the requirement of strong smoothness assumptions holding for real data sets. Practically, this is not feasible or likely, and as such, many techniques are known to subsequently fail. To improve upon this, the proposed work optimized for an additive function with two smooth terms each accounting for different components of smoothness. The first smooth term accounts for the intrinsic geometry by taking advantage of manifolds within the data. The second accounts for smoothness with respect to the Hilbert norm. The problem was carefully setup in Section 4.2, and a detailed, yet informative, presentation of kernel ridge regression was given in Section 4.3. The main result was presented Section 4.4, and important connections to special cases was also provided. This included a novel connection to semi-supervised shrinking involving directions of unlabeled extrapolation in Section 4.4.2. A classification extension was provided in Section 4.5, and a computationally efficient anchor graph version was provided in Section 4.6. The connection to manifold regularization, our closest competitor, was provided in Section 4.7, and it is clearly stated how the proposed approach is designed to be more flexible than this prior work. Empirically, the proposed approach dominated the prior work in Section 4.8 and thus extended the literature with a novel contribution.

5.2 Future Research Directions

The work on ordinal R&R in Chapter 2 is not the final word. Future directions in this area should stem from the practical use of our methods, although we provide two possible extensions here that complicate the context entertained in Chapter 2. A first extension might look at how ordinal R&R changes if the quality of the part distribution shifts. The parts in the actual R&R experiment might be easier or harder to consistently place in the same category if compared to the parts coming off of an assembly line. An example comes from grading papers as a teacher. It may be really easy to rate papers as A's or F's, but much harder to consistently rate papers in the B versus C categories of an ordinal grading scale. As a second extension, one could imagine a gold standard, i.e., the existence of a super operator or trainer who can always place an item in its true ordinal class (by some agreed upon standard). This second extension might look at incorporating such information into the assessment of operators in training.

In the case of the machine learning work presented in Chapters 3 and 4, GPU processing has become ever more relevant in this field, and the proposed approach could take advantage of such massively parallel systems. Tools such as `snow`, `snowfall`, `foreach` , Hadoop, and Apache Spark is to be incorporated to improve computational speed. Also, the cross-validation (CV) search is less than ideal for practical problems and improvements in this directions are always under examination. An R package for general wide-spread use is to be developed as part of this future work.

In Chapter 3, a negative result involving complex loss functions was presented. This research is not done. The mechanics for these optimization problems makes sense, and there may be some justification for them. Simply stated, the burden of optimizing loss function parameters along with penalty parameters is too much to make these practically useful. A middle ground involves better CV methods. One fruitful idea is to break the CV method into stages. The penalty tuning parameter are optimized using a simple loss function. Then a more complex loss function is fit with those parameters fixed to optimize the loss functions parameters separately. This type of estimation approach may have some real promise for

improving the practical usage of these more complex functions.

The work in Chapter 4 establishes a state-of-the-art kernel-based semi-supervised tool. This idea has many practical extensions involving multi-view learning, active learning, and also applications in reinforcement learning. These extensions will offer their own challenges for this work to progress.

BIBLIOGRAPHY

M Azizyan, A Singh, and L Wasserman. Density-sensitive semisupervised inference. *The Annals of Statistics*, 41(2):751–771, 2013.

M Belkin and P Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Journal of Neural Computation*, 15(6):1373–1396, 2003.

M Belkin, P Niyogi, and V Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7: 2399–2434, 2006.

B E Boser, I M Guyon, and V N Vapnik. A training algorithm for optimal margin classifiers. pages 144–152, 1992.

R K Burdick, C M Borror, and D C Montgomery. *Design and analysis of gauge R&R studies: Making decisions with confidence intervals in random and mixed ANOVA models*, volume 17. SIAM, 2005.

G Casella and E I George. Explaining the Gibbs sampler. *The American Statistician*, 46(3): 167–174, 1992.

O Chapelle, M Chi, and A Zien. A continuation method for semi-supervised SVMs. In

Proceedings of the 23rd International Conference on Machine Learning, pages 185–192, New York, NY, USA, 2006a. ACM.

O Chapelle, B Schölkopf, and A Zien, editors. *Semi-supervised learning*. MIT Press, Cambridge, MA, 2006b. URL http://www.kyb.tuebingen.mpg.de/ssl-book.

O Chapelle, V Sindhwani, and S Keerthi. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9:203–233, 2008.

S Chib and E Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

C Cortes and V Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

M Culp, K Johnson, and G Michailides. ada: An r package for stochastic boosting. *Journal of Statistical Software*, 17(1):1–27, 2006.

M Culp, G Michailidis, and K Johnson. On multi-view learning with additive models. *Annals of Applied Statistics*, 3(1):292–318, 2009.

M V Culp and K J Ryan. Joint harmonic functions and their supervised connections. *Journal of Machine Learning Research*, 14:3721–3752, 2013.

J de Mast and W N van Wieringen. Modeling and evaluating repeatability and reproducibility of ordinal classifications. *Technometrics*, 52(1):94–106, 2010.

J de Mast, T Akkerhuis, and T Erdmann. The statistical evaluation of categorical measurements: Simple scales, but treacherous complexity underneath. *Quality Engineering*, 26 (1):16–32, 2014.

L Deldossi and D Zappa. A novel approach to evaluate repeatability and reproducibility for ordinal data. *Communications in Statistics-Theory and Methods*, 43(4):851–866, 2014.

J Faraway. *faraway: Functions and Datasets for Books by Julian Faraway*, 2016. URL https://CRAN.R-project.org/package=faraway. R Package Version 1.0.7.

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

A Gelman, J B Carlin, H S Stern, D B Dunson, A Vehtari, and D B Rubin. *Bayesian data analysis*. Chapman & Hall/CRC Boca Raton, FL, USA, third edition, 2013.

T Hastie, R Tibshirani, and J Friedman, editors. *The Elements of Statistical Learning (Data Mining, Inference and Prediction, Second Edition)*. Springer, New York, NY, 2009.

N Heckman. The theory and application of penalized methods or reproducing kernel hilbert spaces made easy. *Statistics Surveys*, 6:113–141, 2012.

M Hein, J Audibert, and U von Luxburg. From graphs to manifolds–weak and strong pointwise consistency of graph Laplacians. In *Proceedings of the 18th Annual Conference on Learning Theory*, pages 470–485, New York, NY, USA, 2005. Springer.

S I Hill and A Doucet. A framework for kernel-based multi-category classification. *Journal of Artificial Intelligence Research*, 30(1):525–564, 2007.

A Karatzoglou, A Smola, K Hornik, and A Zeileis. kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20, 2004.

M Kawakita and T Jun'ichi. Safe semi-supervised learning based on weighted likelihood. *Neural Networks*, 53(1):146–164, 2014.

M Kuhn. Building predictive models in R using the `caret` package. *Journal of Statistical Software*, 28(5):1–26, 2014.

S Y Kung. *Kernel Methods and Machine Learning*. Cambridge, 2014.

J Lafferty and L Wasserman. Statistical analysis of semi-supervised regression. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 801–808. Curran Associates, Inc., 2008.

Y Li and Z Zhou. Towards making unlabeled data never hurt. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1081–1088, New York, NY, USA, 2011. ACM.

M Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Y Lin, G Wahba, H Zhang, and Y Lee. Statistical properties and adaptive tuning of support vector machines. *Machine Learning*, 48(1):115–136, 2002.

W Liu, J He, and S Chang. Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27rd International Conference on Machine Learning*, pages 679–687, Haifa, Israel, 2010. ACM.

J Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, Microsoft Research Technical Report, 1998.

M Plummer. *JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling*, 2015. Version 4.0.0.

M Plummer. *rjags: Bayesian Graphical Models using MCMC*, 2016. URL https://CRAN.R-project.org/package=rjags. R package version 4-6.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL https://www.R-project.org/.

K J Ryan and M V Culp. On semi-supervised linear regression in covariate shift problems. *Journal of Machine Learning Research, In Press.*, 2015.

T Scheetz, K Kim, R Swiderski, A Philp, T Braun, K Knudtson, A Dorrance, G DiBona, J Huang, T Casavant, V Sheffield, and E Stone. Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103(39):14429–14434, 2006.

A Shashua and A Levin. Ranking with large margin principle: Two approaches. In *NIPs*, pages 961–968, 2002.

A Singh, R Nowak, and X Zhu. Unlabeled data: Now it helps, now it doesn't. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1513–1520. Curran Associates, Inc., 2009.

S B Vardeman and E S VanValkenburg. Two-way random-effects analyses and gauge R&R studies. *Technometrics*, 41(3):202–211, 1999.

J Wang, T Jebara, and S Chang. Semi-supervised learning using greedy max-cut. *Journal of Machine Learning Research*, 14:771–800, 2013.

D Zhou, O Bousquet, T N Lal, J Weston, and B Schölkopf. Learning with local and global consistency. In S. Thrun, L.K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 321–328. MIT Press, 2004.