

Methods for Improving Performance of Precision Health Prediction Models

by

Tatiana Krikella

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Statistics (Biostatistics)

Waterloo, Ontario, Canada, 2024

© Tatiana Krikella 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Daniel J. McDonald
Associate Professor, Dept. of Statistics,
University of British Columbia

Supervisor(s): Joel A. Dubin
Professor, Dept. of Statistics and Actuarial Science,
University of Waterloo

Internal Member: Glen McGee
Assistant Professor, Dept. of Statistics and Actuarial Science,
University of Waterloo

Internal Member: Yeying Zhu
Associate Professor, Dept. of Statistics and Actuarial Science,
University of Waterloo

Internal-External Member: Abel Torres Espin
Assistant Professor, School of Public Health Sciences,
University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Tatiana Krikella was the sole author for Chapters 1, 3, 4, and 5 which were written under the supervision of Dr. Joel A. Dubin. Chapters 3 and 4 are the second and third project chapters (Chapter 2 being the first), but have not yet been submitted for publication. The research presented in Chapter 2 was conducted at the University of Waterloo by Tatiana Krikella under the supervision of Dr. Joel A. Dubin. Tatiana Krikella derived the methods, conducted the simulation studies and the data analysis, and drafted the manuscript. Dr. Joel A. Dubin provided intellectual input to this chapter, including on manuscript drafts, and was involved in the editorial process. This work has been submitted to *Statistics in Medicine*, but is currently available on ArXiv, the citation of which is included below.

Krikella, T., & Dubin, J. A. (2024). A Personalized Predictive Model that Jointly Optimizes Discrimination and Calibration. arXiv preprint arXiv:2403.17132.

Abstract

Prediction models for a specific index patient which are developed on a similar subpopulation have been shown to perform better than one-size-fits-all models. These models are often called *personalized predictive models* (PPMs) as they are tailored to specific individuals with unique characteristics. In this thesis, through a comprehensive set of simulation studies and data analyses, we investigate the relationship between the size of similar subpopulation used to develop the PPMs and model performance.

We propose an algorithm which fits a PPM using the size of a similar subpopulation that optimizes both model discrimination and calibration, as it is criticized that calibration is not assessed as often as discrimination in predictive modelling. We do this by proposing a loss function to use when tuning the size of subpopulation which is an extension of a Brier Score decomposition, which consists of separate terms corresponding to model discrimination and calibration, respectively. We allow flexibility through use of a mixture loss term to emphasize one performance measure over another. Through simulation study, we confirm previously investigated results and show that the relationship between the size of subpopulation and discrimination is, in general, negatively proportional: as the size of subpopulation increases, the discrimination of the model deteriorates. Further, we show that the relationship between the size of subpopulation and calibration is quadratic in nature, thus small and large sizes of subpopulation result in relatively well-calibrated models. We investigate the effect of patient weighting on performance, and conclude, as expected, that the choice of the size of subpopulation has a larger effect on the PPM's performance compared to the weight function applied. We apply these methods to a dataset from the eICU database to predict the mortality of patients with diseases of the circulatory system.

We then extend the algorithm by proposing a more general loss function which allows further flexibility of choosing the measures of model discrimination and calibration to include in the function used to tune the size of subpopulation. We also recommend bounds on the grid of values used in tuning to reduce the computational burden of the algorithm. Prior to recommending bounds, we further investigate the relationship between the size of subpopulation and discrimination, as well as the size of subpopulation and calibration, under 12 different simulated datasets, to determine if the results from the previous investigation were robust. We find that the relationship between the size of subpopulation and discrimination is always negatively proportional, and the relationship between the size of subpopulation and calibration, although not entirely consistent among the 12 cases, shows that a low size of subpopulation is good, if not optimal, in many cases we considered. Based on this study, we recommend a lower bound on the grid of values to be 20% of

the entire training dataset, and the upper bound to be either 50% or 70% of the training dataset, depending on the interests of the study. We apply the methods proposed to both simulated and real data, specifically, the same dataset from the eICU database, and show that the results previously seen are robust, and that the choice of measures for the general loss function have an effect on the optimal size of subpopulation chosen.

Finally, we extend the algorithm to predict the longitudinal, continuous outcome trajectory of an index patient, rather than predicting a binary outcome. We investigate the relationship between the size of subpopulation and the mean absolute error, and find that the performance drastically improves up to a point, where it then stabilizes, leading to the model fit to the full training data as the optimal model, just tending to be slightly better than a model fit to 60% of the subpopulation. As these results are counter-intuitive, we present three other simulation studies which show that these results stem from predicting the trajectory of a patient, rather than from predicting a continuous outcome. Although the answer to why this is the case is an open research question, we speculate that, since a personalized approach still leads to comparable performance to the full model, these results can be attributed to testing these methods on a small sample size. Due to the computational intensity of the methods, however, testing on a larger sample size to generalize these results is currently impractical.

Areas of future work include improving the computational efficiency of these methods, which can lead to investigating these same relationships under more complex models, such as random forest or gradient boosting. Further investigation of personalized predictive model performance when predicting a trajectory should also be considered. The methods presented in this thesis will be summarized into an R package to allow for greater usability.

Acknowledgements

I would first like to thank my supervisor, Joel Dubin, for his support over the course of this degree. I am grateful that he took a chance on me as his PhD student, despite me having taken two statistics courses total up to that point and not knowing how to code in R. I appreciate the time and effort he has put into mentoring me, both on this thesis and otherwise. I would also like to thank the members of my committee, Glen McGee, Yeying Zhu, Abel Torres Espin, and Daniel J. McDonald, for reading my thesis and giving helpful suggestions. To Alex Stringer and Chelsea Ugenti, thank you for providing both mentorship and friendship to me outside of this thesis. And thank you to Mary Lou Dufton, for the guidance and support that she has given me throughout my studies.

I've had the absolute pleasure of meeting so many incredible people from this department who I am lucky to call my friends. To Minzee, Agassi, Chris, Marcus, Megan, Chi-Kuang, Alexandra, and Luke, thank you for many lunches and coffees, and for making me want to come into M3. To Augustine and Grace, thank you for making Waterloo feel like home: Amos girls forever. Somehow I also managed to make friends outside of the department in Waterloo, and they helped me step outside the academia bubble. Thank you to Ioana for girls nights at the Civil, and to Riley for having countless girl coffees with me. Although Kate is an old friend, her moving to Waterloo at the same time was meant to be - thank you for Wednesday trivia and Dutch blitz.

I may have left Windsor but my friends never gave me a chance to feel homesick. To Issraa, Mackenzie, Natalie, Tristan, Ben, and Gillian, thank you for bringing a little piece of home with you here with every visit. Angela, thank you for being my unofficial roommate - we went from living five minutes apart, to three hours, and finally settled at one hour. Even when you weren't physically here, you never felt too far away.

I'd like to thank my boyfriend, Blake, for supporting me, whether that was through listening to me complain about Sharcnet, watching me practice a presentation ten times, or taking me out for ice cream when I had a bad day. Through the roadblocks and celebrating the wins (big and small), I don't know how I would have gotten through this without you.

To my family all over the world, especially my godmother, Helen, my Yiayia, and my cousin, Elena, thank you for making the distance feel negligible. The biggest thanks goes to my sister and parents. Gia, thanks for transferring to UWaterloo one year after I moved here, living in this city without my sister wasn't any fun at all. To my dad, thank you for all the support over these past four years and beyond - I always loved our long drives between Windsor and Waterloo. To my mom, thank you for always being just a phone call away - leaving Windsor was hard, but you have always been my rock, and the three hour distance did not change that even one bit. Thank you for everything.

Dedication

This is dedicated to my parents.

Table of Contents

Examining Committee	ii
Author's Declaration	iii
Statement of Contributions	iv
Abstract	v
Acknowledgements	vii
Dedication	viii
List of Figures	xiii
List of Tables	xvi
1 Introduction	1
1.1 eICU Data	5
2 Personalized Predictive Model that Jointly Optimizes Discrimination and Calibration	6
2.1 Introduction	6
2.2 Methods	8
2.2.1 Notation	8

2.2.2	Patient Similarity Metric	8
2.2.3	Performance Measures	9
2.2.4	Mixture Loss Function	11
2.2.5	Weighting	12
2.2.6	Personalized Predictive Modelling Algorithm	13
2.3	Simulation Studies	16
2.3.1	Setup	16
2.3.2	Relationship Between the Size of Subpopulation and the Performance Measures	17
2.3.3	Effect of Patient Weighting on Performance	18
2.3.4	Applying the Algorithm	19
2.4	Data Analysis	23
2.5	Discussion	29
3	A Mixture Loss Function that Considers Discrimination and Moderate Calibration	32
3.1	Introduction	32
3.2	Methods	34
3.2.1	Notation	34
3.2.2	Performance Measures	34
3.2.3	Tuning the Size of Subpopulation	36
3.3	Simulation Studies	39
3.3.1	Setup	39
3.3.2	Relationships between the Size of Subpopulation and the Performance Measures	41
3.3.3	Empirically Showing the Brier Score Measures Mean Calibration and Implementing Bound Recommendations	45
3.3.4	Relationship Between the Size of Subpopulation and the Mixture Term	46
3.4	Data Analysis	49
3.5	Discussion	56

4	A Personalized Approach to Flexible Modelling of Longitudinal Data	61
4.1	Introduction	61
4.2	Methods	63
4.2.1	From Linear Mixed Effects Models to Additive Mixed Effects Models	63
4.2.2	A Personalized Approach to Additive Mixed Models using Penalized Splines	64
4.3	Simulation Studies	67
4.3.1	Simulation 4.1: Predicting a Continuous, Longitudinal Outcome: A Complex Case	67
4.3.2	Simulation 4.2: Predicting a Continuous, Longitudinal Outcome: A Simple Case	71
4.3.3	Simulation 4.3: Predicting a Continuous Outcome: A Non-Time-Varying Scenario	75
4.3.4	Simulation 4.4: Predicting the Difference of Two Time Points	76
4.4	Discussion	79
5	Discussion	82
5.1	Summary	82
5.2	Future Work	85
	References	88
	APPENDICES	94
A	Additional Plots - Chapter 3	95
A.1	Linear, 10 Association, Balanced Prevalence	96
A.2	Linear, 20 Association, Balanced Prevalence	97
A.3	Linear, 10 Association, Moderate Prevalence	98
A.4	Linear, 10 Association, Low Prevalence	99
A.5	Linear, 20 Association, Low Prevalence	100

A.6	Non-Linear, 10 Association, Balanced Prevalence	101
A.7	Non-Linear, 20 Association, Balanced Prevalence	102
A.8	Non-Linear, 10 Association, Moderate Prevalence	103
A.9	Non-Linear, 20 Association, Moderate Prevalence	104
A.10	Non-Linear, 10 Association, Low Prevalence	105
B	Proposed R Package	106
B.1	Tuning the Size of Subpopulation	106
B.2	Fitting a PPM	108
B.3	Datasets	109

List of Figures

1.1	Schematic diagram showing the general algorithm for training a PPM. The goal of this model is to make a prediction for an index patient. A similarity metric is used to identify patients that are similar to this index patient from the full population of patients in the training set. The similar patients define the subpopulation, which is used as the training data for the model. This model outputs a prediction for the index patient.	2
2.1	Comparison of the three weight functions applied to each vector $\mathbf{X}_k$ representing patient k in the training dataset.	14
2.2	Performance measures as a function of M , where the patients in the similar subpopulation are weighted uniformly.	18
2.3	Performance measures as a function of M , where the patients in the similar subpopulation are weighted using the tri-cube weighting structure. In this weighting scheme, more weight is given to patients that are more similar to the index patient.	20
2.4	Performance measures as a function of M , where the patients in the similar subpopulation are weighted counter-intuitively using the decreasing weighting structure. In this weighting scheme, more weight is given to patients that are less similar to the index patient.	21
2.5	Dot plot showing the resulting optimal M proportions found in the training step of the algorithm across five randomly generated datasets.	22
2.6	Performance measures as a function of α , using the optimal M proportions found in the training/testing stage. The bands shown in each plot represent the 95% bias-corrected and accelerated (BCa) bootstrap confidence intervals.	23
2.7	Flow chart of cohort selection from the eICU database.	25

2.8	Dot plot showing the resulting optimal M proportions found in the training step when fitting PPMs to the eICU data.	27
3.1	Performance measures as a function of M for Dataset 1 from Table 3.1. . .	42
3.2	Performance measures as a function of M for Dataset 2 from Table 3.1. . .	44
3.3	Dot plot showing the resulting optimal M proportions found in the training step of the algorithm across five different randomly generated datasets. . .	47
3.4	Resulting optimal M proportions for Dataset 1 from Table 3.1 under seven values of α and five different randomly generated datasets.	48
3.5	Resulting optimal M proportions for Dataset 2 from Table 3.1 under seven values of α and five different randomly generated datasets.	49
3.6	Performance measures as a function of M for the cardiac dataset from the eICU database	51
3.7	Optimal M proportions found using L^* from Equation (3.4) for varying values of α found when tuning M during the training/testing step of applying the proposed algorithm to the eICU dataset.	53
3.8	Optimal M proportions found using L^{**} from Equation (3.7) for varying values of α found when tuning M during the training/testing step of applying the proposed algorithm to the eICU dataset.	55
4.1	Individual trajectories under two different time-outcome relationships in Simulation 4.1.	70
4.2	The mean absolute error of models fit to subpopulations of varying sizes under five different seeds in Simulation 4.1. The mean across the seeds at each size of subpopulation is represented by the black triangle.	72
4.3	Individual trajectories of the outcome for Simulation 4.2.	73
4.4	The mean absolute error of models fit to subpopulations of varying sizes under five different seeds in Simulation 4.2. The mean across the seeds at each size of subpopulation is represented by the black triangle.	74
4.5	Distribution of responses for a single continuous outcome in Simulation 4.3.	76
4.6	Relationship between M and the MAE under four separate seeds when fitting a PPM in Simulation 4.3.	77
4.7	Relationship between M and the MAE when fitting a PPM in Simulation 4.4.	79

A.1	Performance measures as a function of M when the outcome is simulated by a model which is linear, has 10 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.50. .	96
A.2	Performance measures as a function of M when the outcome is simulated by a model which is linear, has 20 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.50. .	97
A.3	Performance measures as a function of M when the outcome is simulated by a model which is linear, has 10 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.15. .	98
A.4	Performance measures as a function of M when the outcome is simulated by a model which is linear, has 10 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.05. .	99
A.5	Performance measures as a function of M when the outcome is simulated by a model which is linear, has 20 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.05. .	100
A.6	Performance measures as a function of M when the outcome is simulated by a model which is non-linear, has 10 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.50. .	101
A.7	Performance measures as a function of M when the outcome is simulated by a model which is non-linear, has 20 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.50. .	102
A.8	Performance measures as a function of M when the outcome is simulated by a model which is non-linear, has 10 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.15. .	103
A.9	Performance measures as a function of M when the outcome is simulated by a model which is non-linear, has 20 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.15. .	104
A.10	Performance measures as a function of M when the outcome is simulated by a model which is non-linear, has 10 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.05. .	105

List of Tables

2.1	Proportion is optimal proportion, M , found in training step of algorithm. All other results comes from evaluation on the hold-out validation set. Point estimates are given for each of the performance measures. First set of parentheses contain the bootstrap standard error for the estimates. Second set of parentheses contain the BCa confidence interval.	24
2.2	Table of results from validating the optimal M proportion found in the training/testing stage of the data analysis. Proportion is optimal proportion found in training step of algorithm. Point estimates are given for each of the performance measures and are in bold. First set of parentheses contain the bootstrap standard error for the estimates. Second set of parentheses contain the BCa bootstrap confidence intervals.	28
3.1	Characteristics of two of the twelve datasets simulated.	41
3.2	Cardiac Validation Under L^*	54
3.3	Cardiac Validation Under L^{**}	57
4.1	Details of the eight simulated baseline predictors for the two clusters of patients in Simulation 4.1. The table on the left corresponds to the first patient cluster, while the table on the right corresponds to second patient cluster.	68
4.2	Parameter values used to generate the two longitudinal predictors in our simulation study. Each table corresponds to a different longitudinal predictor, the table on the left corresponding to x_9 and the table on the right corresponding to x_{10} . Within each table, the superscript corresponds to the cluster of patients for which that parameter was used.	69

4.3	Parameter values used to generate the longitudinal outcome in our simulation study. The table on the left were the values when the function $g(t_j)$ is linear, and the one on the right corresponds to the curved function $g(t_j)$. .	70
4.4	Parameter values used to generate the longitudinal outcome in Simulation 4.2.	73
4.5	Parameter values used to generate the longitudinal outcome in Simulation 4.4.	78

Chapter 1

Introduction

Models developed in health prediction, including for mortality, have been shown to perform better when the model for a specific index patient is developed on a similar subpopulation. These models are often called *personalized predictive models* (PPMs) as they are tailored to specific individuals with unique characteristics. This area of research is especially valuable in the context of intensive care unit (ICU) prediction models, where it has been suggested that due to the complexity and ambiguity of illnesses in the ICU, a personalized approach should be adopted to both research and practice [Maslove et al., 2017].

It has been shown that PPMs can lead to better prediction compared to one-size-fits-all models [Celi et al., 2012, Lee et al., 2015]. Unlike in a regression setting where heterogeneity among the sample is preferred, in a prediction setting, the goal is to make the most accurate prediction possible for an index patient, which involves a more targeted approach that narrows the heterogeneity to patients that are similar to the index patient. Consequently, many researchers have proposed algorithms that define a subpopulation of patients that are similar to an index patient I and train a model using only this subpopulation, ultimately to make a prediction for patient I [Sharafoddini et al., 2017]. Algorithms of this type are called *neighbourhood-based algorithms*, and the most significant differences in these various algorithms are the similarity criterion that is used and the performance measure applied to evaluate the final models. A schematic diagram of this method is shown in Figure 1.1.

An example of a neighbourhood-based algorithm for fitting a PPM is one proposed by Hielscher et al. [2014] for the classification of epidemiological data on the hepatic steatosis disorder. They use a weighted k -nearest neighbour (k NN) classifier with Euclidean distance to measure similarity, and choose k to be the value that results in the best average sensitivity, specificity, and accuracy after a two-fold validation. Another neighbourhood-

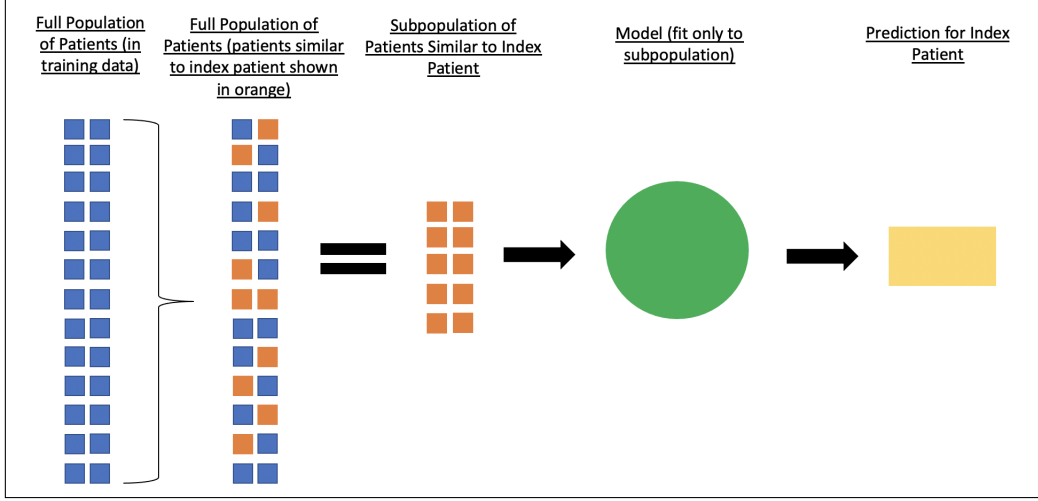


Figure 1.1: Schematic diagram showing the general algorithm for training a PPM. The goal of this model is to make a prediction for an index patient. A similarity metric is used to identify patients that are similar to this index patient from the full population of patients in the training set. The similar patients define the subpopulation, which is used as the training data for the model. This model outputs a prediction for the index patient.

based algorithm, proposed by [Ng et al. \[2015\]](#), uses a Locally Supervised Metric Learning (LSML) similarity metric, which is based on the Mahalanobis distance measure. To assess the performance of their resulting PPM, they use the area under the receiver operating characteristic curve (AUROC), a commonly calculated measure of model discrimination. A more recent paper by [Wang et al. \[2021\]](#) also uses Mahalanobis distance to define similarity between patients in their neighbourhood-based algorithm. They choose the optimal size of subpopulation based on the AUROC, cross-entropy loss, and the F1-score, which is a measure of the model’s accuracy.

Non-distance-based similarity metrics used in neighbourhood-based algorithms include a similarity measure proposed by [Campillo-Gimenez et al. \[2013\]](#) which uses the logical OR-operator. This measure incorporates weights that need to be learned before they are applied to the predictors. Then, the similarity measure is used to define a subpopulation consisting of the top M similar patients to an index patient, where M is chosen to maximize the AUROC. Another example is the cosine-similarity measure used in an algorithm proposed by [Lee et al. \[2015\]](#), where the measure is used to define a subpopulation consisting of the top M similar patients to the index patient, where, again, M is chosen to maximize the AUROC.

Although a heavily investigated area in time-series literature [Fang et al., 2015, Yang and Shahabi, 2004], not much work has been done in patient similarity in defining a similarity metric to use when data is longitudinal in nature, or when there are both longitudinal and baseline predictors to consider when measuring similarity. In some cases, the longitudinal information for a given patient is summarized in the form of summary statistics, such as the mean and median of the predictor over time, and this statistic is used to measure similarity along with other baseline predictors [Lee et al., 2015, Ng et al., 2015]. However, by aggregating a patient’s longitudinal information into a single summary statistic, the temporal aspect of these time-varying predictors is lost. Another method, which can identify the dominant modes of variation in the longitudinal predictors, is to apply functional principal component analysis (fPCA) [Yao et al., 2005] to extract the functional principal components (FPCs) and ultimately use the FPC scores, along with the baseline predictors, as input into a similarity metric, as was done by Sharafoddini et al. [2021]. This way, the temporal information is not lost, or at least not oversmoothed, and one can control the amount of variation explained by the FPC scores by choosing the number of FPC scores to include in the similarity calculation.

Lee [2017] suggests that choosing the optimal size of subpopulation, M , will become important when taking these similarity methods into practice. Proposed algorithms have included a method of tuning M , as mentioned above, where the optimal M is one that maximizes a common measure of discrimination, specifically the AUROC. *Discrimination* is the model’s ability to distinguish between outcome classes. It has been criticized by Van Calster et al. [2019] that model calibration is not assessed nearly as often as discrimination in prediction models, which is problematic because poorly calibrated models can be potentially harmful when making decisions. *Calibration* is the agreement between the observed outcomes and predictions. Van Calster *et al.* also argue that when calibration is assessed, it is more often than not done so via external validation after the model has already been developed.

In this thesis, we propose three algorithms to fitting PPMs with the intention of improving the performance of precision health prediction models. Our first project is motivated by the criticisms of Van Calster et al. [2019], and we propose an algorithm inspired by Lee et al. [2015] to develop a PPM that jointly optimizes both model discrimination and calibration when predicting a binary outcome. To do this, we propose a mixture loss function, with terms corresponding to both discrimination and calibration, which also allows flexibility of emphasizing one performance measure over another, and is used to tune M . We present a set of simulation studies, including the investigation of relationships between M and measures of discrimination and calibration, the effect of patient weighting on performance, and the application of our algorithm to simulated data. We also conduct a data

analysis, where we apply our algorithm to data from the eICU database, which is discussed in detail in Section 1.1.

Our second project extends the algorithm proposed in the first project by improving the loss function to be further flexible by allowing the user to choose which measures of discrimination and calibration to include in the loss function, while also addressing computational concerns by proposing bounds on the grid of values on which to tune M . We present another set of simulation studies, the first of which we investigate the relationship between M and various performance measures under 12 different datasets which differ in their true outcome model to test whether the relationships presented in the first project are robust. We also test the updated algorithm on simulated data and compare the results to what was seen in the previous chapter. The data analysis included in this chapter is based on the same data set used in the first project, and we apply our proposed algorithm to this data using two loss functions which differ in their term corresponding to discrimination.

In the final project, we further extend the algorithm presented in the first and second projects to predict a longitudinal, continuous outcome trajectory of an index patient in the case where there are two distinct clusters of patients in a large dataset, as could be the case in electronic health record (EHR) data. We propose an algorithm to predict the trajectory of an index patient on a similar subpopulation of patients rather than on the entire population of patients in the training data which, to our knowledge, has not yet been investigated. We apply our proposed algorithm to simulated data to determine whether a personalized approach to predicting the trajectory of an index patient results in improved performance compared to a one-size-fits-all model. Based on these results, we investigate the relationship between the size of subpopulation and performance through two additional simulation studies.

The remainder of this thesis is structured as follows. In Chapter 2, we present the first project of this thesis, including the proposed algorithm, a set of simulation studies, and a data analysis. In Chapter 3, we show the extension of this algorithm, as mentioned above, and discuss the simulation and data analysis results of the second project of this thesis. Chapter 4, ie. the third project of this thesis, includes the final discussed extension of our proposed algorithm, along with a set of simulation studies. We summarize the results across the entire thesis in Chapter 5, where we also include a subsection outlining current open research problems pertaining to this work.

1.1 eICU Data

We apply the methods discussed in Chapters 2 and 3 to the eICU Collaborative Research Database: a large multi-center critical care database from the United States [Pollard et al., 2018]. In this thesis, we will refer to this database as the eICU database for simplicity. The eICU database is sourced from the eICU Telehealth Program, which is a model of care where remote providers monitor patients continuously.

The database consists of 200,859 unit encounters for 139,367 unique patients. These patients were admitted between 2014 and 2015 at 208 hospitals located throughout the USA. The database is deidentified, and includes tables with vital sign measurements, nurse’s notes, severity of illness scoring results, diagnoses, treatment information, care plan documentation, and more. One specific table in the eICU database which we will be referencing frequently, *vitalPeriodic*, includes data which is interfaced from a bedside monitor at one-minute averages, but is archived in the *vitalPeriodic* table as five-minute median values.

Data is publicly available via a PhysioNet repository. However, to access, one must complete a training course in research with human subjects.

Chapter 2

Personalized Predictive Model that Jointly Optimizes Discrimination and Calibration

2.1 Introduction

In a clinical setting, personalized predictive models (PPMs) generally lead to better prediction compared to one-size-fits all models [Celi et al., 2012, Lee et al., 2015]. As a result, many researchers have proposed algorithms that define a subpopulation of patients that are similar to an index patient I and train a model using only this subpopulation, ultimately to make a prediction for patient I [Sharafoddini et al., 2017]. Algorithms of this type are often called *neighbourhood-based algorithms*. Many neighbourhood-based algorithms for fitting a PPM have been proposed in the literature [Hielscher et al., 2014, Ng et al., 2015, Wang et al., 2021, Lee et al., 2015], and key questions that must be addressed when designing these algorithms are (1) How do we define similarity, (2) How many individuals do we include in the similar subpopulation, and (3) How do we measure the performance of the resulting PPM.

There are many ways of measuring similarity. In their proposed algorithm, Hielscher et al. [2014] use a weighted k -nearest neighbour (k NN) classifier with Euclidean distance to measure similarity, where k is chosen as the value that results in the best average sensitivity, specificity, and accuracy after a two-fold validation. Both Ng et al. [2015] and Wang et al. [2021] use a metric based on Mahalanobis distance to define similarity between patients in their neighbourhood-based algorithms. An example of a non-distance-based similarity

metric used in neighbourhood-based algorithms is the cosine similarity metric used by Lee et al. [2015], which is bounded, interpretable and able to handle non-continuous predictors. Further, it is hypothesized that the cosine metric may be more sensitive to details in patient differences when measuring similarities compared to other similarity metrics [Sharafoddini et al., 2021].

A common method of choosing how many individuals to include in the similar subpopulation is to determine which size of subpopulation, M , optimizes the area under the receiver operating characteristic curve (AUROC): a popular measure of discrimination [Ng et al., 2015, Wang et al., 2021, Campillo-Gimenez et al., 2013, Lee et al., 2015]. In these cases, a sparse grid of values on which to tune M is defined, a PPM is fit using these varying M values, performance is measured by the AUROC, and the optimal M is then chosen as the one corresponding to the PPM with the greatest AUROC. In later work, Lee [2017] suggests that choosing the optimal M value will become important when taking these methods into practice.

As previously mentioned, for prediction models in general, it has been criticized that calibration is not assessed nearly as often as discrimination [Van Calster et al., 2019]. Van Calster *et al.* also argue that even when calibration is assessed, it is usually done via external validation after the model has been developed via internal validation. Motivated by this issue, Jiang et al. [2012] propose a prediction algorithm that jointly optimizes the discrimination and calibration of a support vector machine by manipulating the decomposition of an overall performance measure, the *Brier Score*. The Brier Score is a summary measure that is optimized when the correct probabilities are used, making it a *proper scoring rule* [Gneiting and Raftery, 2007]. In other words, the Brier Score will give a lower score to a model if the resulting predicted distribution matches the data distribution. For models that have similar discrimination, a well-calibrated model has a better Brier Score than a miscalibrated model. However, in fairly recent work by Assel et al. [2017], they claim that the Brier Score does not appropriately evaluate the clinical utility of prediction models.

Through our literature review, we notice that, in general, calibration is not considered when choosing an optimal size of the subpopulation for PPMs. We use this gap in the literature to motivate our work.

The remainder of the chapter will be structured as follows. In Section 2.2, we propose an algorithm inspired by Lee et al. [2015] to develop a PPM that jointly optimizes both discrimination and calibration. We do this by finding the optimal size of the subpopulation, M , that jointly optimizes these performance measures through the use of a proposed loss function, which is an extension of the Brier Score. We use a two-term decomposition of

the Brier Score, along with a mixture term, to allow for the flexibility of emphasizing one performance measure over another when training the PPM to find the optimal M . Section 2.3 contains results from a set of simulation studies which show the relationship between M and various performance measures in addition to a sensitivity analysis on the mixture term of our defined loss function. In Section 2.4, our proposed algorithm is tested on data from the eICU database to predict mortality of patients with diseases of the circulatory system.

2.2 Methods

2.2.1 Notation

Let there be n_{train} patients in the training data, where each patient k is represented by a Euclidean vector, $\mathbf{X}_k$, $k = 1, \dots, n_{train}$, in the multi-dimension predictor space defined by p predictors. We let x_h , $h = 1, \dots, p$, represent the $n_{train} \times 1$ vector consisting of each patient's value of a given predictor. Further, let $\mathbf{X}_I$, $I = 1, \dots, n_{test}$, be the Euclidean vector representation of the index patient, for whom we would like to make a prediction. Note that the index patient is not included in the training dataset.

The focus of the prediction task in this paper is on binary classification. Thus, we consider a binary outcome. The actual outcome for a patient I in the testing dataset is represented by $y_I \in \{0, 1\}$. Let M be a scalar representing the size of the subpopulation in the training data consisting of similar patients to the index patient. Finally, let $p_I^{(M)} \in [0, 1]$ represent the predicted probability of the outcome for a patient I found using a model fit on a similar subpopulation of size M ; we would add further specificity to $p_I^{(M)}$ in the case where there are competing models of the same and/or different type(s), such as $p_{I(logis1)}^{(M)}$, $p_{I(logis2)}^{(M)}$ or $p_{I(RF)}^{(M)}$ for two competing logistic regression models and a random forest model, respectively.

2.2.2 Patient Similarity Metric

When developing a PPM, one must choose a criterion that will be used to measure similarity between individuals. In this chapter, we focus on an unsupervised, one-stage process, where the predictors being considered for identifying the subpopulation will ideally be identified by subject-matter expert(s), or short of that possibility, through a study of the relevant subject-matter literature. We take this unsupervised approach because in a prediction

(future) context, the index patient does not have an outcome label. The similarity between patients I and k is calculated prior to determining a subpopulation which will be used as training data for a PPM to predict the outcome of patient I . Another way to think about this is that everyone in the training data has a (similarity) score calculated to measure similarity to the index patient, but only those in the resulting subpopulation, based on M , will receive a non-zero weight in the PPM.

We use the cosine similarity metric as it is bounded, interpretable, and able to handle non-continuous data. Further, it is hypothesized that the cosine metric may be more sensitive to details in patient differences when measuring similarities [Sharafoddini et al., 2021]. The cosine similarity between patients I and k is shown in Equation (2.1) where $\bullet$ and $\|\cdot\|$ represent the dot product and the Euclidean vector magnitude, respectively. We label this metric as $CSM(\mathbf{X}_I, \mathbf{X}_k)$.

$$CSM(\mathbf{X}_I, \mathbf{X}_k) = \frac{\mathbf{X}_I \bullet \mathbf{X}_k}{\|\mathbf{X}_I\| \|\mathbf{X}_k\|} \quad (2.1)$$

Equation (2.1) is a bounded measure of similarity, with $-1 \leq CSM(\mathbf{X}_I, \mathbf{X}_k) \leq 1$. $CSM(\mathbf{X}_I, \mathbf{X}_k) = -1$ corresponds to minimum (opposite) similarity, $CSM(\mathbf{X}_I, \mathbf{X}_k) = 1$ corresponds to maximum similarity, and a value of 0 refers to orthogonal similarity.

2.2.3 Performance Measures

To measure the performance of a PPM, one should measure both the discrimination and calibration of the model [Steyerberg, 2019]. Often, overall performance measures such as the Brier Score, shown in Equation (2.2), are used, as the score is affected by both model discrimination and calibration.

$$BrS = \frac{1}{n_{test}} \sum_{I=1}^{n_{test}} (y_I - p_I^{(M)})^2 \quad (2.2)$$

In a binary setting, such as the one we consider in this project, the Brier Score can range from 0 for a perfect model to 0.25 for a non-informative model when the outcome proportion is 50%. However, the range of the Brier Score changes depending on the outcome proportion, thus making the interpretation of the Brier Score a function of that proportion. A solution to this problem is to scale the Brier Score by its maximum possible score for a realized outcome proportion under a non-informative model [Steyerberg et al., 2010].

However, in this work, we do not consider the scaled Brier Score as the interpretation of the Brier Score is unnecessary for our purposes, as will be demonstrated below.

Even when predictive models have high model discrimination, the calibration may still be poor due to statistical overfitting [Van Calster et al., 2019]. One limitation of the Brier Score is that it may select a model that is miscalibrated with good discrimination over a model that is better calibrated with average discrimination [Assel et al., 2017]. Thus, it may be advantageous to evaluate the discrimination and calibration separately to obtain a more comprehensive assessment of the predictive performance of the model.

A common measure of discrimination is the area under the receiver operating characteristic curve (AUROC), which plots the sensitivity (true positive rate) against $1 - \text{specificity}$ (false positive rate) for consecutive cutoffs for the probability of an outcome. A close-to-perfect discriminated model will have an AUROC close to 1, while a model that cannot distinguish between classes will have an AUROC close to 0.5.

Van Calster et al. [2016] have defined a hierarchy of four increasingly strict levels of calibration, referred to as mean, weak, moderate, and strong calibration. We will be using mean, weak, and moderate measures of calibration, as strong calibration is the utopic goal but unrealistic in practice.

Calibration-in-the-large (CITL) is a measure of mean calibration and evaluates whether the observed event rate is equal to the average predicted probabilities, shown in Equation (2.3). A CITL value of 0 indicates perfect mean calibration.

$$CITL = \frac{1}{n_{test}} \sum_{I=1}^{n_{test}} |y_I - p_I^{(M)}| \quad (2.3)$$

Consider the logistic recalibration framework [Cox, 1958, Miller et al., 1993], where $\text{logit}(y) = a + b_B B$. Here, $B = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$ is the linear predictor, where $\beta = (\alpha, \beta_1, \dots, \beta_p)$ are the regression coefficients corresponding to the predictors. In the logistic calibration framework, the value b_B is called the *calibration slope*, which is a measure of weak calibration and is simply the slope of the line that results from plotting the predicted probabilities on the x -axis and the observed outcomes on the y -axis. A slope of 1 indicates perfect weak calibration. Values of the calibration slope below 1 imply that the predicted probabilities are too extreme, while values of the calibration slope greater than one imply that the predicted probabilities are too modest. We further note that the intercept in the logistic calibration framework, a , is a measure of mean calibration, related to the calibration-in-the-large discussed in the paragraph prior.

The *Integrated Calibration Index* (ICI), valued between 0 and 1, is a measure of moderate calibration [Austin and Steyerberg, 2019], and is the weighted difference between observed and predicted probabilities, where the observations are weighted by the empirical density function of the predicted probabilities. This measure is used to compare the relative calibration of different prediction models, where a lower value of the ICI corresponds to better calibration. An estimate of the ICI is shown in Equation 2.4, where $\hat{p}_I^{(M)}$ are the smoothed predicted probabilities based on the LOESS calibration curve.

$$ICI = \frac{1}{n_{test}} \sum_{I=1}^{n_{test}} \left| \hat{p}_I^{(M)} - p_I^{(M)} \right| \quad (2.4)$$

2.2.4 Mixture Loss Function

The optimal size of subpopulation, M , is the value which results in a PPM that minimizes some specified loss function. The primary goal of this work is to find the M that results in a PPM with good model discrimination and calibration, while allowing the potential to emphasize one measure over another, if of interest. We capitalize on the fact that the Brier Score can be decomposed into its discrimination and calibration subparts, and invoke a mixture loss term, $\alpha \in [0, 1]$, which allows flexibility in emphasizing one performance measure over the other as shown in Equation (2.5). We use the decomposition discussed by Rufibach [2010], the details of which are shown in Equation (2.6).

$$L^{(M)} = \frac{\alpha}{n_{test}} \sum_{I=1}^{n_{test}} (y_I - p_I^{(M)})(1 - 2p_I^{(M)}) + \frac{1 - \alpha}{n_{test}} \sum_{I=1}^{n_{test}} p_I^{(M)}(1 - p_I^{(M)}). \quad (2.5)$$

$$\begin{aligned} BrS &= \frac{1}{n_{test}} \sum_{I=1}^{n_{test}} [(y_I - p_I^{(M)})^2] \\ &= \frac{1}{n_{test}} \sum_{I=1}^{n_{test}} [y_I^2 - 2y_I p_I^{(M)} + (p_I^{(M)})^2] \\ &= \frac{1}{n_{test}} \sum_{I=1}^{n_{test}} [y_I - 2y_I p_I^{(M)} - p_I^{(M)} + 2(p_I^{(M)})^2 + p_I^{(M)} - (p_I^{(M)})^2] \quad \text{since } y_I \in \{0, 1\} \\ &= \frac{1}{n_{test}} \sum_{I=1}^{n_{test}} [y_I(1 - 2p_I^{(M)}) - p_I^{(M)}(1 - 2p_I^{(M)}) + p_I^{(M)}(1 - p_I^{(M)})] \\ &= \frac{1}{n_{test}} \sum_{I=1}^{n_{test}} [(y_I - p_I^{(M)})(1 - 2p_I^{(M)}) + p_I^{(M)}(1 - p_I^{(M)})] \\ &= \frac{1}{n_{test}} \sum_{I=1}^{n_{test}} [(y_I - p_I^{(M)})(1 - 2p_I^{(M)})] + \frac{1}{n_{test}} \sum_{I=1}^{n_{test}} [p_I^{(M)}(1 - p_I^{(M)})] \end{aligned} \quad (2.6)$$

The first and second terms after the equal sign are related to calibration and discrimination, respectively. Specifically, the first term measures the lack of calibration, with an

expectation of 0 under perfect calibration. The second term measures the lack of spread of the predictions. For larger values of α , greater emphasis is put on calibration, while for smaller values of α , greater emphasis is placed on discrimination. When $\alpha = 0.5$, the loss function is proportional to the Brier Score.

2.2.5 Weighting

The idea of giving heavier weights to units that are more similar to an index unit has been used in areas such as k NN [Dudani, 1976, Paredes and Vidal, 2006, Zuo et al., 2008] and smoothing [Cleveland and Devlin, 1988, Ramsay, 1991]. This idea has also been applied in the context of patient similarity, specifically in deep learning [Du and Wu, 2021], but not from a statistical context. We will apply this idea to our proposed method. Once the similar subpopulation for index patient I is chosen, this subpopulation will be weighted using one of three structures. The first weighting structure will assign each patient a uniform weight. The second weighting structure will give heavier weights to patients who are more similar to the index patient. The final weighting structure will give heavier weights to patients who are the least similar to the index patient. This third (naive) weighting structure is applied to see how much of an effect purposefully poor weighting has on the predictive accuracy of the resulting model. We will then compare the predictive accuracy of the resulting models built on these weighted subpopulations. We speculate that, akin to smoothing, the subpopulation (ie. neighbourhood) size will have a greater influence on predictive accuracy than will the weighting structure.

In the uniform weighting structure, each patient will receive a weight of $1/M$.

To give heavier weights to patients who are most similar to the index patient, we will employ the tri-cube weight function. This weighting function, which is used in the LOESS smoothing algorithm [Cleveland, 1979], is shown in Equation (2.7). In Equation (2.7), d is the distance from the index patient, $\mathbf{X}_I$ to another patient in the population, $\mathbf{X}_k$. The parameter h is the half-window width, also referred to as the *bandwidth*. To calculate h , we first need to define how many patients we will consider as the neighbourhood. In this work, we consider M patients. The bandwidth h is the distance $d(\mathbf{X}_I, \mathbf{X}_M)$, where $\mathbf{X}_M$ is the M^{th} most similar patient to the index patient.

$$W_T(d) = \begin{cases} \left(1 - \left|\left(\frac{d}{h}\right)^3\right|\right)^3, & \left|\frac{d}{h}\right| < 1 \\ 0, & \left|\frac{d}{h}\right| \geq 1 \end{cases} \quad (2.7)$$

The weight that is given to patient k in the training dataset based on the tri-cube weight function is shown in Equation (2.8). We consider the distance function $d(\mathbf{X}_I, \mathbf{X}_k) = (1 - CSM(\mathbf{X}_I, \mathbf{X}_k))$, which will output values between 0 and 2, with a value of 0 corresponding to perfect similarity. We do this since the inputs for the tri-weight function are distances, thus lower distances imply higher levels of similarity. As higher values of the cosine similarity measure imply similarity, we subtract the cosine similarity value from 1 to ensure lower values (in this case, close to 0) imply similarity. We normalize the weights applied to the population by dividing by the sum of the weights for all patients in the subpopulation, $j = 1, \dots, M$. This leads to legitimate weights since, by definition, tri-cube weights are non-negative.

$$w_T(\mathbf{X}_I, \mathbf{X}_k) = \frac{W_T(1 - CSM(\mathbf{X}_I, \mathbf{X}_k))}{\sum_{j=1}^M W_T(1 - CSM(\mathbf{X}_I, \mathbf{X}_j))} \quad (2.8)$$

Equation (2.9) shows the weight that is given to patient k to give the least similar patients heavier weights. We call these weights *decreasing weights*. The value of 1.1 in the denominator was an arbitrary value chosen to avoid dividing by 0, when $CSM(\mathbf{X}_I, \mathbf{X}_j)$ is equal to its minimum, -1.

$$w_I(\mathbf{X}_I, \mathbf{X}_k) = \frac{\frac{1}{1.1 + CSM(\mathbf{X}_I, \mathbf{X}_k)}}{\sum_{j=1}^M \frac{1}{1.1 + CSM(\mathbf{X}_I, \mathbf{X}_j)}} \quad (2.9)$$

Figure 2.1 shows all three weight functions applied to each vector $\mathbf{X}_k$ representing patient k in the training dataset.

2.2.6 Personalized Predictive Modelling Algorithm

We propose an algorithm which sets up a validation procedure to tune the subpopulation size, M , for a given value of α . We include an external validation step in the algorithm where we estimate the performance of the resulting PPM along with quantifying the uncertainty of these estimates. Step-by-step details of the algorithm are given below.

1. Split the entire population into a $(100q)\%$ hold-out validation set and $(100[1 - q])\%$ training-testing (TrTe) set, where $q \in [0, 1]$. We often use $q = 0.2$.
2. Tune M by following the steps below:

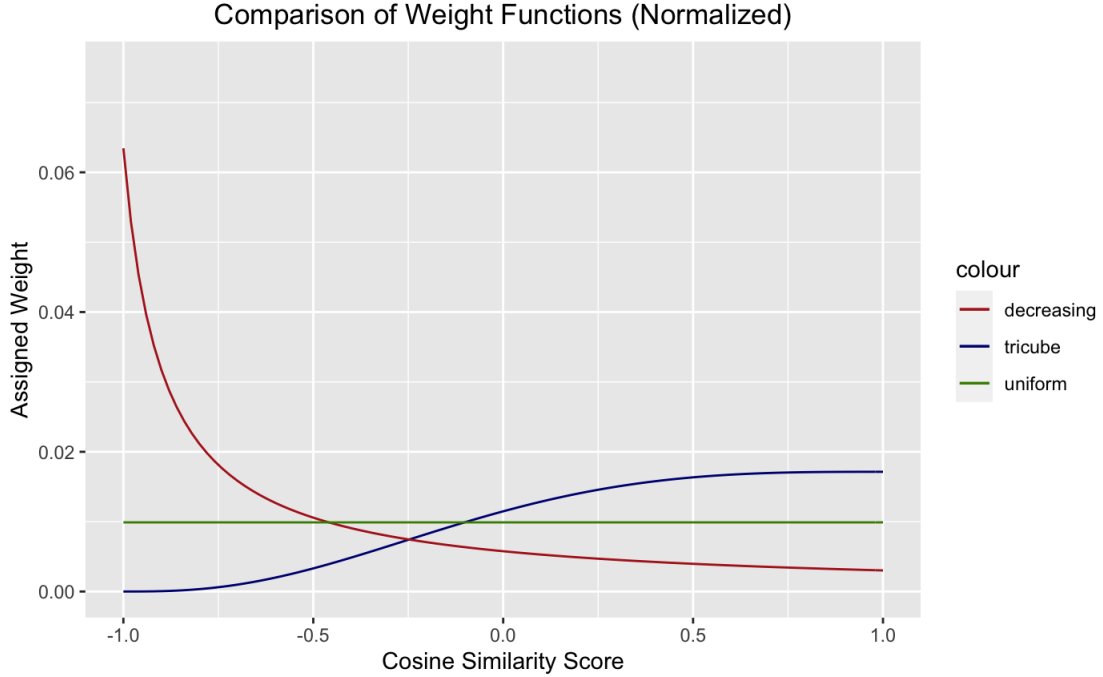


Figure 2.1: Comparison of the three weight functions applied to each vector $\mathbf{X}_k$ representing patient k in the training dataset.

- (a) Randomly split the TrTe set into distinct training and testing sets using K -fold cross-validation (CV).
 - i. For index patient I in the test set, calculate $d(\mathbf{X}_I, \mathbf{X}_k)$, for $k = 1, \dots, n_{train}$ in the training set, where $d(\mathbf{X}_I, \mathbf{X}_k)$ is the similarity metric between the index patient and the k^{th} training set patient.
 - ii. Consider a grid of M values and tune M through a grid search.
 - A. Sort all $d(\mathbf{X}_I, \mathbf{X}_k)$ in descending order, and the patients corresponding to the top M scores form the subpopulation.
 - B. Apply a pre-determined weighting function (eg. uniform, (half-)tricube) to each patient in the subpopulation. Note that the (half-)tricube weights give those more similar to the index patients a greater weight.
 - C. Fit the model of interest, such as logistic regression, random forest, boosted trees, etc., on the subpopulation.
 - D. Average the value of the loss function over the K -folds.

- iii. Repeat the K -fold CV v times, where vK is at least 200. Then, the optimal M is that which has the smallest loss function averaged over the v repeated K -fold CV runs.
 - iv. The optimal M proportion is obtained by dividing the optimal M by the number of patients in the training set, ie. $P_{optimal} = M/n_{train}$.
- 3. Externally validate the model using the hold-out validation sample to evaluate the performance of all model characteristics (including optimal M proportion and weighting function used to weight patients in the training set).
 - (a) Draw a bootstrap sample from the validation sample (each the same size as the validation sample).
 - i. Randomly split the bootstrap sample into an 80% training sample and 20% testing sample.
 - ii. Set the integer $M = \text{ceiling}(n_{val.train}P_{optimal})$ where $n_{val.train}$ is the size of the training set.
 - iii. For index patient I in the test set, calculate $d(\mathbf{X}_I, \mathbf{X}_k)$, for $k = 1, \dots, n_{train}$ in the training set, where $d(\mathbf{X}_I, \mathbf{X}_k)$ is the similarity metric between the index patient and the k^{th} patient.
 - iv. Use the M obtained in step (2) to form the subpopulation
 - v. Apply the chosen weighting function to each patient in the subpopulation.
 - vi. Fit the model of interest on the subpopulation.
 - vii. Measure the performance of the model using, for example, the discrimination and calibration measures discussed in Section 2.2.3.
 - (b) Repeat the above algorithm with B different bootstrap samples.
 - (c) In order to quantify uncertainty around the performance measures, fit bias-corrected and accelerated (BCa) bootstrap confidence intervals for each of the performance measures using the B bootstrap estimates.
- 4. Repeat the entire process Z times, ie. repeat the process with Z different validation holdout samples (and, thus, TrTe samples) to view the consistency of the tuning of M (step (1) above) as well as the model performance measures in different hold-out validation samples. We suggest Z to be approximately 10.

The proposed algorithm above is done for a given α , perhaps one that is determined ahead of time, such as $\alpha = 0.5$, where both discrimination and calibration are equally

balanced in Equation (2.5). However, there may be some situations where we want to investigate the performance and resulting M for different scenarios, say a bit more emphasis being placed on calibration over discrimination. We could then implement the above algorithm for a few different α values, perhaps to see how the performance will do as a function of both α and M . We will see some of this work in Sections 2.3 and 2.4 below.

2.3 Simulation Studies

We conducted four simulation studies using R [R Core Team, 2020]. We first investigated the relationship between the size of the subpopulation, M , and the model discrimination measure, AUROC, to confirm the results of Lee et al. [2015] that as M increases, the AUROC deteriorates. We then investigated the relationship between calibration and the size of subpopulation, M . We also explored the effects of applying different weighting functions to the similar subpopulation prior to fitting the PPMs. Finally, we applied our PPM algorithm to randomly generated data under a number of α values to determine how the value of α affects the optimal M proportion found in the training/testing step of the algorithm.

2.3.1 Setup

We considered a total sample size of $n = 16000$. We simulated 20 predictors, where 10 were binary, 10 were continuous, and the correlation between any two predictors was $r = 0.2$. We simulated an outcome with a non-linear true outcome model, shown in Equation (2.10). This model was passed through an inverse logit function to obtain a probability, π , as shown in Equation (2.11). This probability was then used to simulate a Bernoulli outcome, y_k , where $Y_k \sim \text{Bernoulli}(\pi)$. Note that the simulated outcome of each index patient in the testing dataset, y_I , was generated in the same way. The prevalence of the outcome was moderate, with values between 0.3 and 0.5 for all simulated datasets.

$$z = -4x_1 + x_6 - 2x_1x_3 + 3\exp(x_4) - 5\exp(x_2x_8) + \epsilon \quad (2.10)$$

$$\pi = 1/(1 + \exp(-z)). \quad (2.11)$$

We randomly generated 10 datasets to test the algorithm, and, within each dataset, we conducted 20 repeated 10-fold CV. We tested the algorithm under the following values of α :

$\{0.475, 0.49, 0.5, 0.58, 0.6, 0.62, 0.75, 0.85, 0.9, 0.99\}$. We chose a number of α values between 0.475 and 0.75 to see the effect of weighting one performance measure slightly over another. We also chose values of α which put a much heavier emphasis on the calibration term in the loss function, specifically $\alpha = \{0.85, 0.9, 0.99\}$, to further investigate the sensitivity of the loss function. We did not consider values lower than $\alpha = 0.475$, which would have put a heavier emphasis on the discrimination term in the loss function, because the resulting optimal size of subpopulation was too small to apply to the validation set. We discuss this further in Section 2.5.

In the validation step, we simulated $B = 1000$ bootstrap samples. We calculated 95% bias-corrected and accelerated (BCa) bootstrap confidence intervals for four performance measures: AUROC, CITL, calibration slope, and the ICI.

2.3.2 Relationship Between the Size of Subpopulation and the Performance Measures

We investigated the relationship between M and the four performance measures. In this set of simulation studies, the patients were weighted uniformly. In other words, the top M similar patients each receive a weight of $1/M$ and the remainder of patients a weight of 0.

The results obtained are shown in Figure 2.2. There is a negatively proportional relationship between M and the AUROC in Figure 2.2(a). As the size of subpopulation increases, the AUROC, hence the discrimination, first has a slight improvement then quickly begins to deteriorate. This confirms the findings of the application data analysis in Lee et al. [2015].

The remaining three plots in Figure 2.2 show the relationship between M and the three calibration measures. In all three plots, the relationship between M and calibration is quadratic: the calibration of the PPM when M is low is good, then as we increase M the calibration deteriorates up to a point, where it then begins to improve. Under all three performance measures, the best calibrated PPM is the one which uses all of the available data in the training set. Although, the PPMs using low values of M still are well calibrated, the rate of change of calibration between low M values is much higher than that of higher M values. For instance, Figure 2.2(d) shows that the difference between the ICI of the PPMs when $M = 450$ and $M = 550$ is 0.0057, while the difference in the ICI of the PPMs when $M = 11350$ and $M = 11450$ is only 0.0014. The calibration is more sensitive to changes in M when M is low. Under both the calibration slope and the ICI, the calibration of the PPM begins to deteriorate at a much smaller M value compared

to the CITL. However, the CITL has a much faster rate of deterioration compared to the other two calibration measures.

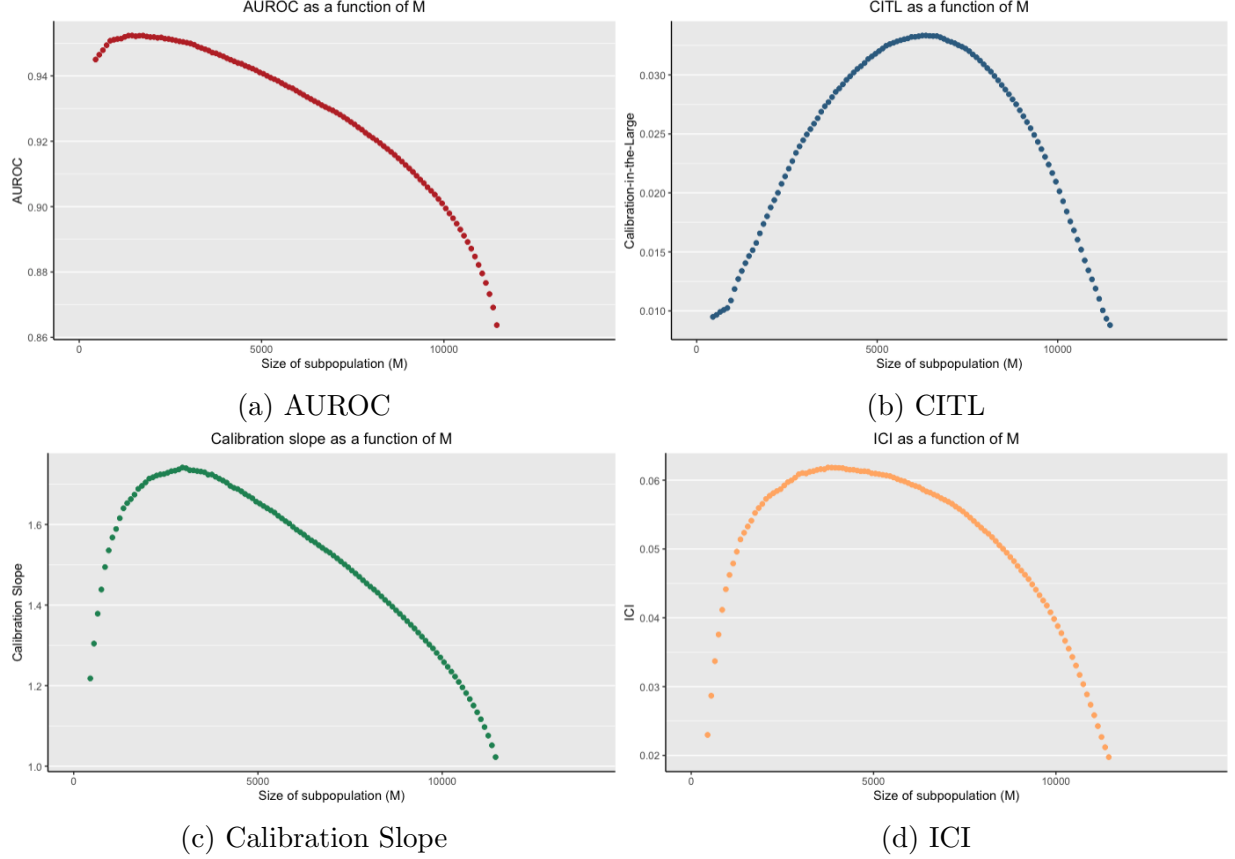


Figure 2.2: Performance measures as a function of M , where the patients in the similar subpopulation are weighted uniformly.

2.3.3 Effect of Patient Weighting on Performance

We also tested two other weighting schemes to investigate the effect of weighting the similar subpopulation on the performance of the PPMs. In addition to the uniform weights considered in Section 2.3.2, we considered (half-)tri-cube weights [Cleveland, 1979], and a purposely counter-intuitive weighting scheme which gives higher weights to individuals that are the least similar to the index patient; these additional weighting methods were described in Section 2.2.5.

Specifically, we implemented the following two weighting schemes:

1. top M similar patients are assigned tri-cube weights and the remainder of patients a weight of 0
2. top M similar patients are assigned decreasing weights and the remainder of patients a weight of 0

We looked at the relationship between M and discrimination, as well as M and calibration, under the different weighting structures to see how the predictive accuracy of the model was affected based on how the data was weighted. Figures 2.3 and 2.4 show the results from applying the tri-cube and decreasing weights to the similar patients, respectively. We compare these results to Figure 2.2, where we applied uniform weights to the similar patients.

Under all three weighting schemes, the results are almost identical, with very small differences between the three that are not distinguishable in the presented figures. We do not see a large improvement in the performance of the PPMs when we give higher weights to patients that are more similar to the index patient using the tri-cube weighting scheme. Further, we do not see a notable deterioration of performance when we use the counter-intuitive decreasing weights and give patients who are less similar to the index patient higher weights. We conclude that the size of subpopulation has a much larger effect on the predictive performance of the PPM compared to the choice of weight function. This is a consistent finding with that in the curve estimation literature, where bandwidth size is much more important than neighbourhood weighting.

2.3.4 Applying the Algorithm

In the last simulation study, we applied the algorithm to randomly generated data and show the results in a variety of plots.

Figure 2.5 shows the relationship between the mixture term, α , and the resulting optimal M proportion found in the training/testing set of the algorithm. As we increase α , the optimal M proportion also increases. The rate of increase is relatively small until $\alpha = 0.75$, where we begin to see a steep slope, especially between $\alpha = 0.75$ and $\alpha = 0.85$. When $\alpha = 0.99$, that is, when there is a very high emphasis on calibration in the loss function, the optimal M proportion is nearly the entire available training population. This algorithm was tested on 10 randomly generated datasets and from the plot, which displays results from five of these datasets, we see that the results are robust.

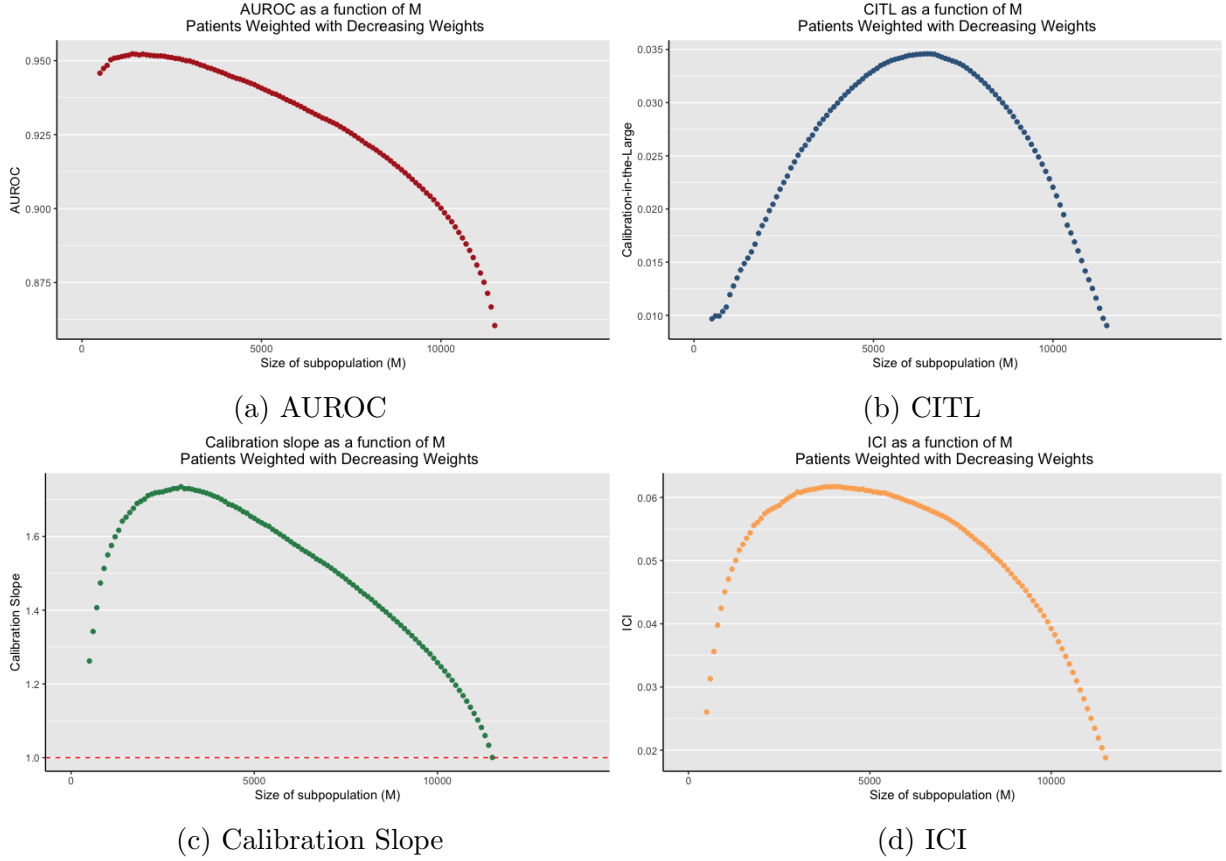


Figure 2.3: Performance measures as a function of M , where the patients in the similar subpopulation are weighted using the tri-cube weighting structure. In this weighting scheme, more weight is given to patients that are more similar to the index patient.

We then applied the optimal M proportions shown in Figure 2.5 to the validation step where we assessed the performance of the resulting PPMs. All results are shown via plots in Figure 2.6. A tabular representation of the results are shown in Table 2.1.

Figure 2.6(a) shows the AUROC as a function of α , with the associated pointwise confidence bands. The peak AUROC of 0.944 (0.919, 0.958) was achieved when $\alpha = 0.75$, which corresponds to an optimal M proportion of 0.156. The worst performing PPM is seen under $\alpha = 0.99$, which corresponds to an optimal M proportion of approximately one. The value of the AUROC under $\alpha = 0.99$ is 0.861 (0.840, 0.894) which is a significant difference. This plot follows the same pattern as shown in Figure 2.2(a), where we initially see an improvement in discrimination up until a certain point (in this case, $\alpha = 0.75$)

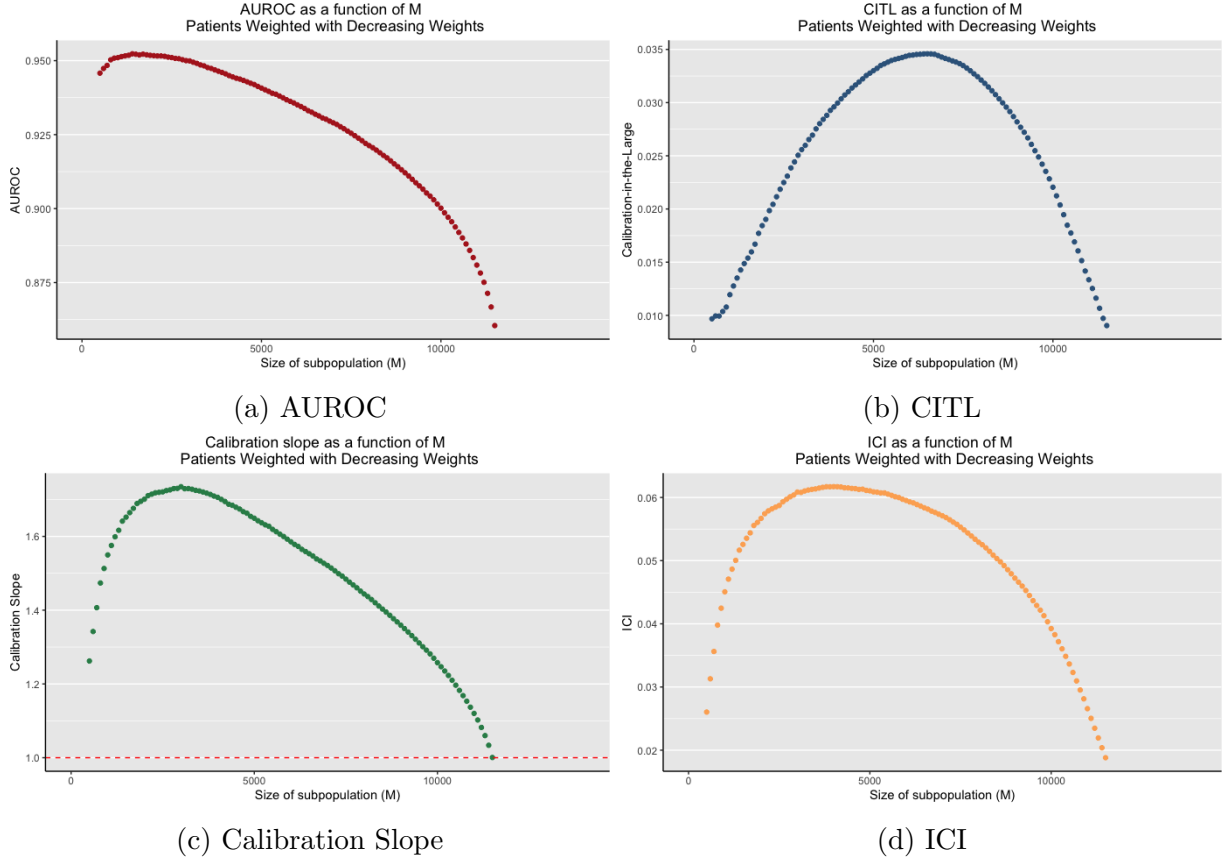


Figure 2.4: Performance measures as a function of M , where the patients in the similar subpopulation are weighted counter-intuitively using the decreasing weighting structure. In this weighting scheme, more weight is given to patients that are less similar to the index patient.

where it then begins to deteriorate rapidly.

The calibration as a function of α is shown in Figures 2.6(b)-(d). Under the CITL and ICI measures, the best performing PPM is a result of $\alpha = 0.99$, with values of 0.019 (0.004, 0.052) and 0.023 (0.009, 0.041), respectively. Under the calibration slope, the best performing PPM is when $\alpha = 0.5$, which corresponds to an optimal M proportion of 0.078. The value of the calibration slope at $\alpha = 0.5$ is 0.947 (0.829, 1.179). Recall that a calibration slope of 1 corresponds to perfect weak calibration. Under all three measures of calibration, the worst performing PPM occurs at $\alpha = 0.85$, which corresponds to an optimal M proportion of 0.642. For small and large values of α , the PPMs are well-

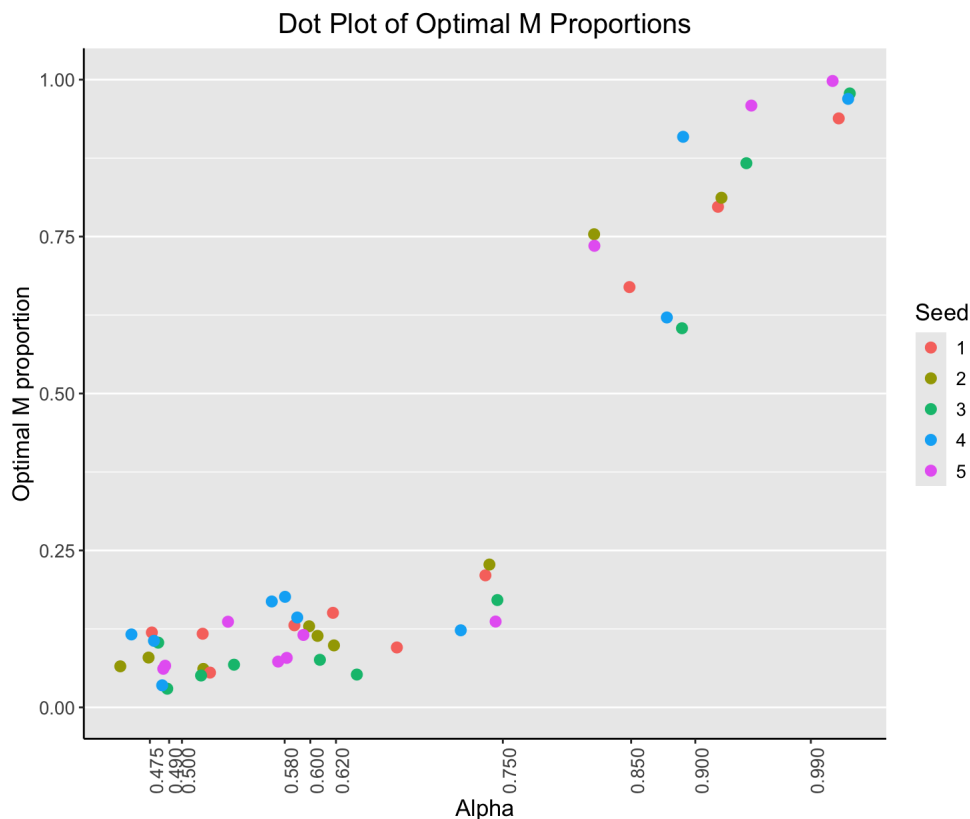


Figure 2.5: Dot plot showing the resulting optimal M proportions found in the training step of the algorithm across five randomly generated datasets.

calibrated. However, when α is 0.75 and 0.85, these PPMs have poor calibration. This is consistent with the quadratic relationship between M and calibration we showed in Figure 2.2(b)-(d). Under the CITL, the difference between the best and worst calibrated PPMs is 0.029. Under the calibration slope, that difference is 0.586. Finally, under the ICI the difference is 0.035. The PPMs fit in this simulation study are, in general, well-calibrated, so, although appearing small, these differences are particularly meaningful. For instance, a difference of 0.029 in the CITL between two PPMs indicates that the PPM with the worse calibration has an additional 2.9% of the predicted outcome points that are incorrect compared to the true outcome. In a health setting specifically, when dealing with, say, the mortality of patients, this is not ideal.

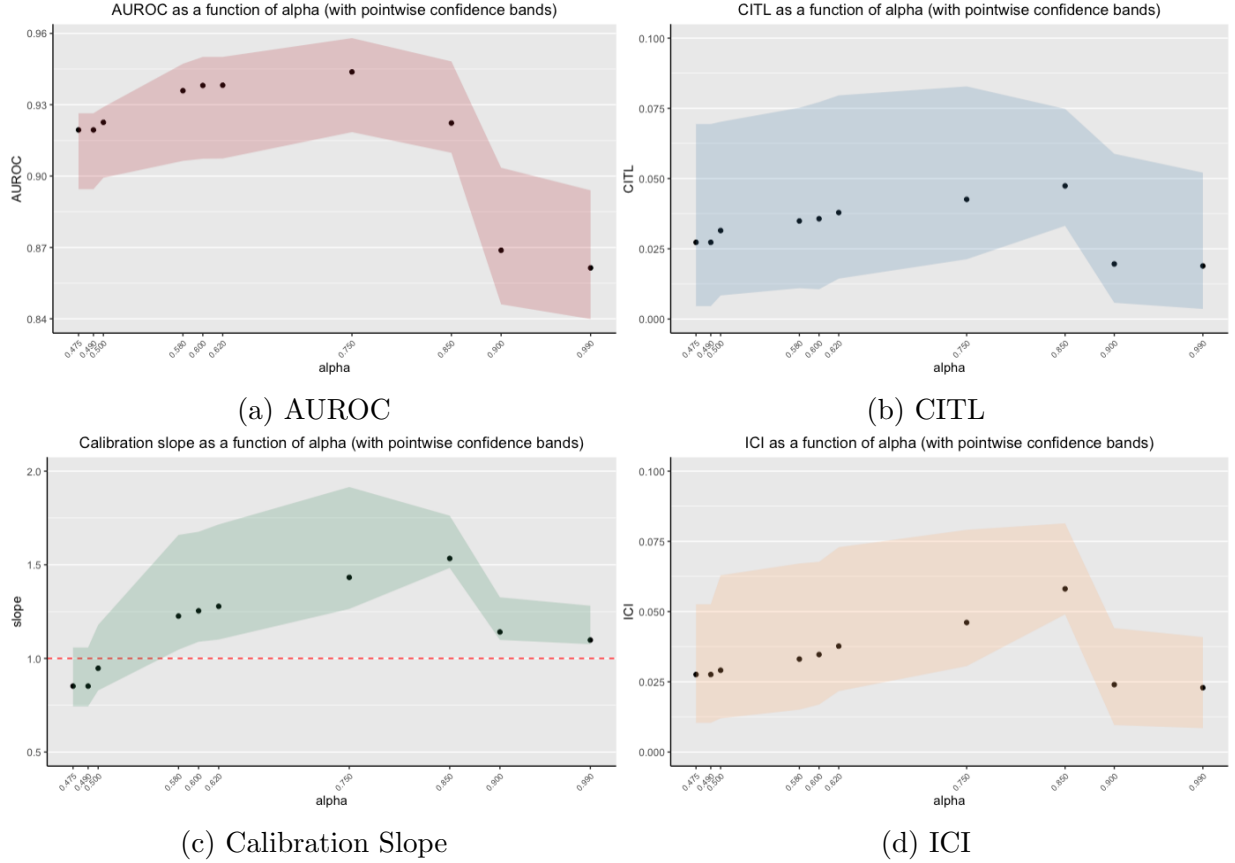


Figure 2.6: Performance measures as a function of α , using the optimal M proportions found in the training/testing stage. The bands shown in each plot represent the 95% bias-corrected and accelerated (BCa) bootstrap confidence intervals.

2.4 Data Analysis

We applied our proposed algorithm to a dataset from the eICU Database [Pollard et al., 2018] to predict mortality of patients with diseases of the circulatory system.

We only considered patients who were in their first ICU stay of their first hospital visit and had a disease of the circulatory system, identified by their ICD9 code: an official system in the US which assigns codes to various diseases and procedures. We considered only those patients for whom the first hospital stay was identifiable because, although the ordered years of the patient were kept intact, different hospital stays within each year were not ordered chronologically. They must have survived in the ICU for at least 24 hours, but

Table 2.1: Proportion is optimal proportion, M , found in training step of algorithm. All other results comes from evaluation on the hold-out validation set. Point estimates are given for each of the performance measures. First set of parentheses contain the bootstrap standard error for the estimates. Second set of parentheses contain the BCa confidence interval.

α	Proportion	AUROC	CITL	Slope	ICI
0.475	0.0694	0.9194 (0.011) (0.8945, 0.9264)	0.0273 (0.015) (0.0046, 0.0694)	0.8519 (0.141) (0.7437, 1.058)	0.0276 (0.011) (0.0104, 0.0526)
0.49	0.0694	0.9194 (0.011) (0.8945, 0.9264)	0.0273 (0.015) (0.0046, 0.0694)	0.8519 (0.141) (0.7437, 1.058)	0.0276 (0.011) (0.0104, 0.0526)
0.5	0.0781	0.9226 (0.010) (0.8993, 0.9290)	0.0315 (0.015) (0.0084, 0.0702)	0.9473 (0.131) (0.8291, 1.1786)	0.0291 (0.012) (0.0120, 0.0629)
0.58	0.1172	0.9359 (0.009) (0.9064, 0.9472)	0.0349 (0.016) (0.0110, 0.0752)	1.226 (0.131) (1.0472, 1.6587)	0.0331 (0.012) (0.0151, 0.0671)
0.6	0.1215	0.9381 (0.009) (0.9073, 0.9501)	0.0357 (0.017) (0.0106, 0.0772)	1.254 (0.134) (1.089, 1.676)	0.0347 (0.0121) (0.0169, 0.0677)
0.62	0.1259	0.9382 (0.009) (0.9074, 0.9501)	0.0379 (0.016) (0.0144, 0.0796)	1.278 (0.134) (1.1013, 1.7151)	0.0377 (0.012) (0.0217, 0.0729)
0.750	0.1563	0.9438 (0.009) (0.9185, 0.9580)	0.0426 (0.017) (0.0213, 0.0828)	1.4325 (0.146) (1.2643, 1.9152)	0.0461 (0.012) (0.0306, 0.0791)
0.85	0.6424	0.9223 (0.011) (0.9098, 0.9482)	0.0474 (0.015) (0.0332, 0.0748)	1.5337 (0.131) (1.4832, 1.7624)	0.0581 (0.012) (0.0490, 0.0814)
0.9	0.8594	0.8688 (0.015) (0.8461, 0.9035)	0.0196 (0.010) (0.0058, 0.0588)	1.141 (0.097) (1.099, 1.326)	0.0240 (0.009) (0.0096, 0.0441)
0.99	0.9983	0.8614 (0.015) (0.8399, 0.8940)	0.0189 (0.10) (0.0036, 0.0521)	1.098 (0.097) (1.076, 1.281)	0.0229 (0.009) (0.0085, 0.0409)

their stay could not exceed 14 days as these patients are outliers, with only 2% of unique patients having ICU stays lasting more than two weeks in this dataset. An investigation of this group of patients who both survived and stayed in the ICU for over two weeks could be the focus of a future investigation. We removed any patient with missing values in any of the predictors considered. Further, we removed patients who did not have any vital sign measures in the first 24 hours of their ICU stay. We talk more about handling missing data for this analysis in the Discussion. Figure 2.7 illustrates the cohort selection from the eICU database.

For each patient, we extracted the following predictors:

- Admission and demographic data: age, gender (male/female), hospital teaching status, and hospital discharge status (Expired/Alive).
- Vital signs from the first 24 hours after the start time of ICU admission: oxygen saturation (sao2), respiratory rate (respiration), and heart rate.

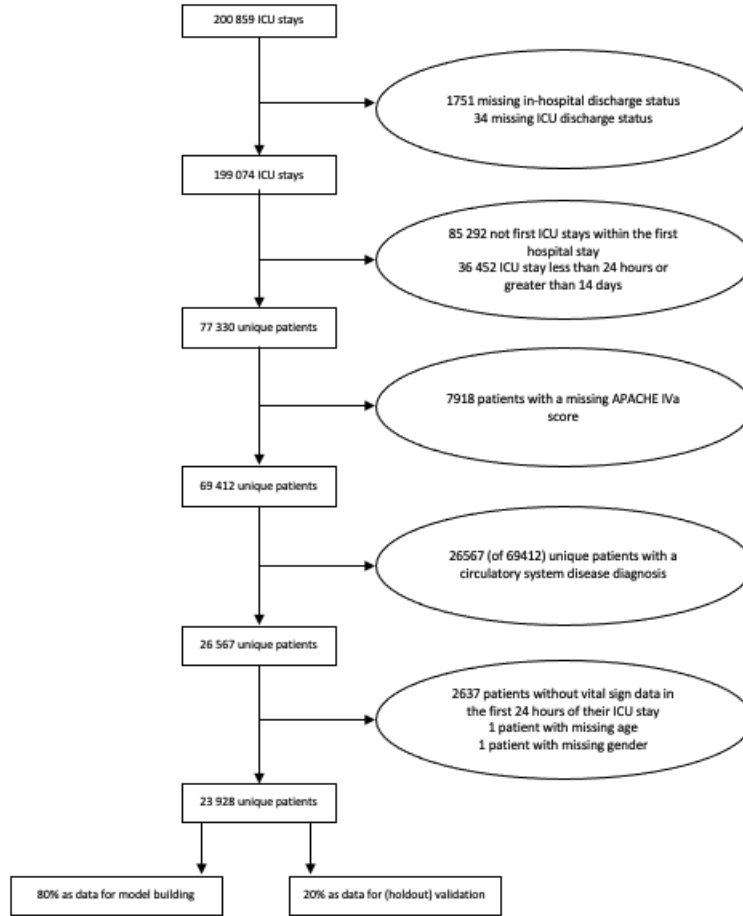


Figure 2.7: Flow chart of cohort selection from the eICU database.

- APACHE IVa score: a severity-of-illness score utilizing demographic information as well information collected within the first 24 hours of a patient’s ICU stay.

Summary statistics are often used to summarize longitudinal data in analyses of ICU data, such as the vital signs in the case of this data analysis. Such summary statistics might include the mean or median of the repeated measures for a given vital sign, as well as a minimum and maximum value. However, just including summary statistics does not consider the temporal aspect of vital signs. Thus, due to the dense curve-like nature of the vital signs, we use functional principal components (FPCs) [Yao et al., 2005] to

identify each of their dominant modes of variation, specifically for the vital signs in the first 24-hours of a patient’s stay in the ICU. We applied fPCA to the vital sign data using the Principal Analysis by Conditional Estimation (PACE) algorithm provided in the R package *fdapace* [Carroll et al., 2020]. The PACE algorithm uses eigenfunctions to expand the trajectory of a longitudinal predictor for a given patient, as shown in Equation (2.12). The first r eigenfunctions ($\hat{\phi}_q(t)$, $q = 1, \dots, r$) are used in this expansion, where $\hat{\mu}(t)$ is the estimated mean function of the predictor and $\hat{\xi}_{kq}$ represent the FPCs.

$$\hat{X}_k^{(r)}(t) = \hat{\mu}(t) + \sum_{q=1}^r \hat{\xi}_{kq} \hat{\phi}_q(t). \quad (2.12)$$

We extracted the first three FPCs for each of the vital signs, which cumulatively explained at least 95% of the total variation in the trajectory of each given vital sign; a different dataset for these vital signs, or even different vital signs in this dataset, may have required a different number of FPCs to account for at least 95% variation. Our resulting dataset thus consisted of 13 predictors, including the first three FPC scores for each of the three vital signs.

We considered $\alpha = \{0.475, 0.49, 0.5, 0.51, 0.55, 0.6, 0.750, 0.9, 0.95\}$. We conducted 20 repeated 10-fold CV in the training/testing step of the algorithm. Similar to the simulation study, our outcome in this case is binary, and we fit a logistic regression model to the data. We weighted the subpopulation uniformly. The prevalence of outcome in this dataset is approximately 0.107, which is much lower than that of our simulation study (approximately 0.4). Because of this, we calculated the area under the precision recall curve (AUPRC) as a measure of discrimination in addition to the AUROC, as the latter can be misleading when the prevalence of outcome is highly skewed [Davis and Goadrich, 2006, Ozenne et al., 2015]. Specifically, in these cases, the AUROC tends to be too optimistic.

Figure 2.8 show the relationship between the mixture term, α , and the resulting optimal M proportion, and these results match those seen in Figure 2.5 from the simulation study. As α increases, which in turn puts more emphasis on the calibration term in the loss function, the optimal M proportion also increases. We found that with this dataset, the optimal M proportion was very sensitive to changes in α . The optimal M proportion increases by 0.329 when we increase α from 0.5 to 0.51. Then, it increases by 0.337 when we increase α from 0.51 to 0.55. Further, unlike the results from the simulation study, from $\alpha = 0.55$ onwards, the optimal M proportion stays constant at 0.871.

The last row of this table displays the model performance when the entire available training data is used to train the PPM, for comparison purposes. The optimal M proportion, in general, increases as α increases. In other words: as we put more emphasis

on calibration in our loss function, the size of subpopulation that optimizes discrimination and calibration increases. The one exception to this pattern is $\alpha = 0.475$, which results in a higher optimal M proportion than $\alpha = 0.49$. We found that the differences in the value of the loss function under the first three values of α considered were very close in magnitude, differing by values in the ten thousandths. Thus, for $\alpha = 0.475, \alpha = 0.49$ and $\alpha = 0.5$, any proportion between 0.205 and 0.261 would be acceptable.

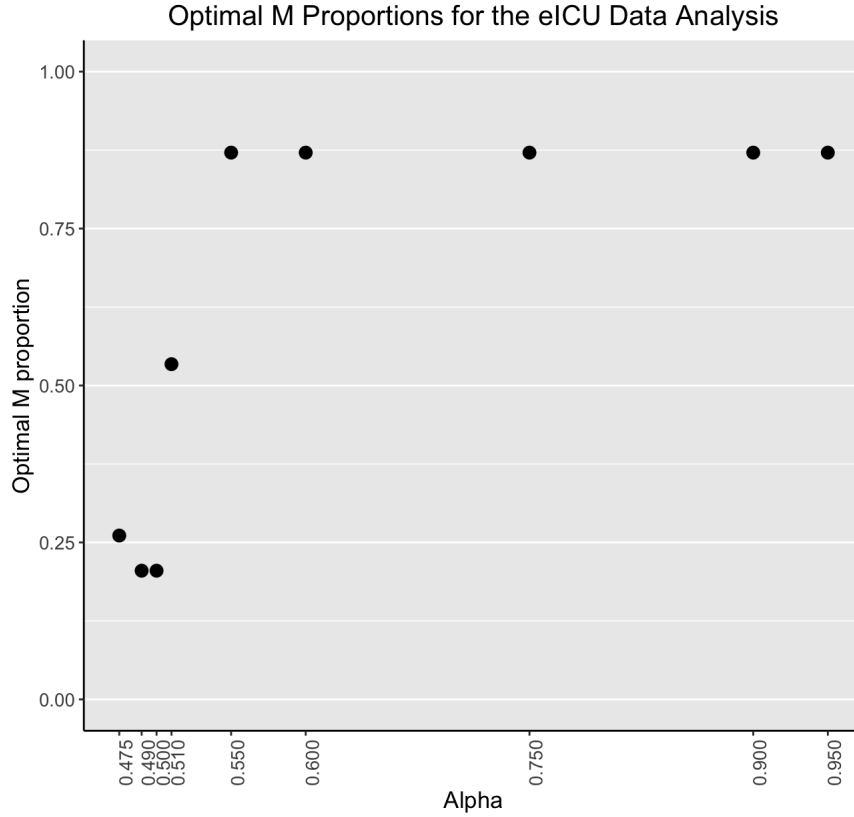


Figure 2.8: Dot plot showing the resulting optimal M proportions found in the training step when fitting PPMs to the eICU data.

As we saw in the simulation studies, we notice that in general, as we increase α , the discrimination deteriorates and the calibration improves. We do not see this decreasing pattern under the AUROC measure, however, this measure is not reliable when the prevalence of outcome is skewed. Under all α values, the AUROC estimate is greater than the upper bound of its confidence interval, which further supports that the AUPRC is a better measure in this case. We also do not see an increasing pattern under the CITL, but it is

Table 2.2: Table of results from validating the optimal M proportion found in the training/testing stage of the data analysis. Proportion is optimal proportion found in training step of algorithm. Point estimates are given for each of the performance measures and are in bold. First set of parentheses contain the bootstrap standard error for the estimates. Second set of parentheses contain the BCa bootstrap confidence intervals.

α	Proportion	AUROC	AUPRC	CITL	Slope	ICI
0.475	0.261	0.7737 (0.023) (0.7466, 0.7588)	0.5106 (0.054) (0.4459, 0.6248)	0.0211 (0.006) (0.0204, 0.0296)	0.8079 (0.097) (0.6456, 0.8985)	0.0143 (0.005) (0.0051, 0.0222)
0.49	0.205	0.7724 (0.023) (0.7461, 0.7512)	0.5200 (0.054) (0.4704, 0.6274)	0.0212 (0.006) (0.0206, 0.0304)	0.8014 (0.095) (0.5987, 0.9243)	0.0200 (0.005) (0.0111, 0.0327)
0.5	0.205	0.7724 (0.023) (0.7461, 0.7512)	0.5200 (0.054) (0.4704, 0.6274)	0.0212 (0.006) (0.0206, 0.0304)	0.8014 (0.095) (0.5987, 0.9243)	0.0200 (0.005) (0.0111, 0.0327)
0.51	0.534	0.7830 (0.023) (0.7567, 0.7796)	0.5175 (0.054) (0.4551, 0.6258)	0.0219 (0.006) (0.0217, 0.0296)	0.8763 (0.101) (0.6435, 0.9911)	0.0135 (0.005) (0.0051, 0.0203)
0.55	0.871	0.7873 (0.022) (0.7637, 0.7864)	0.5030 (0.053) (0.4311, 0.6194)	0.023 (0.006) (0.0219, 0.0304)	0.8816 (0.103) (0.6919, 0.9802)	0.0126 (0.005) (0.0033, 0.0184)
0.6	0.871	0.7873 (0.022) (0.7637, 0.7864)	0.5030 (0.053) (0.4311, 0.6194)	0.023 (0.006) (0.0219, 0.0304)	0.8816 (0.103) (0.6919, 0.9802)	0.0126 (0.005) (0.0033, 0.0184)
0.750	0.871	0.7873 (0.022) (0.7637, 0.7864)	0.5030 (0.053) (0.4311, 0.6194)	0.023 (0.006) (0.0219, 0.0304)	0.8816 (0.103) (0.6919, 0.9802)	0.0126 (0.005) (0.0033, 0.0184)
0.9	0.871	0.7873 (0.022) (0.7637, 0.7864)	0.5030 (0.053) (0.4311, 0.6194)	0.023 (0.006) (0.0219, 0.0304)	0.8816 (0.103) (0.6919, 0.9802)	0.0126 (0.005) (0.0033, 0.0184)
0.95	0.871	0.7873 (0.022) (0.7637, 0.7864)	0.5030 (0.053) (0.4311, 0.6194)	0.023 (0.006) (0.0219, 0.0304)	0.8816 (0.103) (0.6919, 0.9802)	0.0126 (0.005) (0.0033, 0.0184)
	1.000	0.7892 (0.022) (0.7625, 0.7857)	0.5085 (0.053) (0.4392, 0.6260)	0.0233 (0.006) (0.0255, 0.0307)	0.9065 (0.103) (0.6961, 1.016)	0.0120 (0.005) (0.0039, 0.0164)

the weakest measure of calibration which also may make it unreliable. An example of its unreliability is shown in the performance of the model that uses the full training data: the point estimate of CITL is not included in the confidence interval. This is why we include measures of weak and moderate calibration.

The best performing model is seen when $\alpha = 0.51$. Compared to the model with the best discrimination ($\alpha = 0.5$) we sacrifice some discrimination (-0.0025 AUPRC), but we see a large increase in the value of the calibration slope (+0.07), and we also see that this model has better moderate calibration. Further, we see that although the model which uses the full training data has the best calibration, we lose some predictive accuracy through discrimination. This supports what we found in the simulation study. Again, we show that even though researchers can use the Brier Score as the loss function, it may not result in the best performing model.

2.5 Discussion

While there have been algorithms proposed to fit personalized predictive models (PPMs), there is lack of emphasis in the literature on models that are well-calibrated. We thus proposed an algorithm which fits a PPM using the optimal size of subpopulation that considers both discrimination and calibration. This is done through a proposed loss function that allows for flexibility in emphasizing one performance measure over another, as desired.

We investigated the relationship between the size of subpopulation and calibration and found that the two have a quadratic relationship, with the best calibrated PPMs using the entire population of patients in the training data. However, if one can live with the slightly worse calibration for a smaller M while also wanting better discrimination, then the smaller M will be a better choice that still results in well-calibrated PPMs; alternatively, if calibration is considered much more important than discrimination for a given problem, then the larger M may be the better choice here. Also, the rate of change in calibration between the low values of M is greater than that of higher values of M .

The relationship between M and calibration is different from what has been found in the literature regarding discrimination, where, as the size of subpopulation increases, the discrimination deteriorates. Thus, PPMs fit using a low M value will have better model discrimination compared to a PPM fit with a higher M value. We confirmed these results in our simulation study. There were much larger differences in the values of the AUROC between the worst and best performing PPMs compared to what we saw under calibration for both the simulation studies and the data analysis. Further, due to the low

prevalence of outcome in the data analysis, the preferred measure of discrimination was the AUPRC which demonstrated the same negatively proportional relationship with the size of subpopulation as we saw with the AUROC.

The trade-off between discrimination and calibration when choosing an M value was the motivation behind our proposed algorithm which fits a PPM using an M that considers both performance measures. Our proposed loss function, an extension of the Brier Score, uses the mixture term α to emphasize either performance measure. When the mixture term, α , is set to 0.5, then the proposed mixture loss function is proportional to Brier Score. Thus, they attain their minimum at the same value of M . Although the Brier Score is a proper scoring rule, we do not know whether our proposed loss function shares this property. Future work is necessary to verify that the mixture term of the Brier Score gives the lowest score when the predicted distribution matches the data distribution. Through simulation studies, we found that as we increased α , the optimal M proportion found in the training/testing step of the algorithm also increased. In other words, as we put more emphasis on the calibration term in our proposed mixture loss function, the optimal proportion of subpopulation to fit the PPM also increased. However, this rate of increase was not consistent. The largest rate of change we see is between $\alpha = 0.75$ and $\alpha = 0.85$. Before this, the changes in the optimal M proportion between subsequent α values in very gradual. This observation requires further investigation, and to do so, a surface plot looking at α against M could be informative. These results were robust across the randomly generated datasets. In our study, the best performing model was when $\alpha = 0.58$, which means slightly more emphasis was put on calibration compared to discrimination in the mixture loss function. The main purpose of this study was to show that researchers can use the Brier Score as the loss function, but the resulting model will not necessarily be optimal in a general setting. What is optimal will depend on the user and what they care about, whether that is a model that has better discrimination over calibration or vice versa (or a balance between the two), and our next step is to suggest criterion in choosing this α value so the user can obtain the model performance desired. We were limited in the number of α values below 0.5 we could test, as these α values resulted in very small optimal M proportions. When we applied the optimal M proportion to the validation step, the subpopulation used for training the PPM was not a sufficient size.

To our knowledge, weighting the subpopulation using non-uniform weights had not been investigated, so we tested our algorithm under three different weighting schemes. We found that the choice of M was much more influential on the performance of the resulting model compared to the weighting function used. As stated earlier, this is consistent with smoothing literature, where the choice of bandwidth is more influential than the choice of neighbourhood weighting function. It might be worth investigating this area further to

determine whether there are scenarios where weighting might play at least somewhat of a larger role than it seems to be in the investigation already done. For instance, perhaps looking into applying a sigmoidal weighting scheme to the similar subpopulation.

In our data analysis, we use a dataset from the eICU database to predict mortality of patients with diseases of the circulatory system. Instead of using summary statistics to encapsulate the longitudinal vital sign data, we implemented fPCA so that the temporal aspect of these data was not lost. We applied our proposed algorithm to the data and obtained results consistent with what was found in the simulation study. The largest rate of change in our findings occurred between $\alpha = 0.51$ and $\alpha = 0.55$, however this change is comparable to what we see between $\alpha = 0.5$ and $\alpha = 0.51$. Something interesting to note is that from $\alpha = 0.55$ onwards, the optimal M proportion does not change from 0.871. One reason why the optimal M proportion is not the entire population of patients in the training set for the largest values of α may be due to too much noise in the data, which is common with EHR data. However, this will need to be studied further. The best performing model found in the data analysis corresponded to $\alpha = 0.51$, which used 53.4% of the full training data as the similar subpopulation. This further supports that using the Brier Score as the loss function may not result in the best performing model. It is important to note that no imputation was performed for variables, such as APACHE IVa, when preparing the data for this analysis. This would be something to focus on in further analyses of these data.

Our study has a few limitations. The loss function we propose is an extension of the Brier Score, however we did not conduct any studies to ensure that our loss function is in line with the recommendations of [Van Calster et al. \[2016\]](#) to strive for moderately calibrated models. Further, our algorithm is very computationally intensive. We employed the *Rcpp* package [[Eddelbuettel and François, 2011](#)], along with parallelization, which sped up the running time significantly, but there is still much room for improvement. If the algorithm can be adjusted to become computationally efficient, other statistical models such as random forest or gradient boosting machines can be tested to see if our results translate to other models. Finally, we do not know if the relationships we have shown are specific to the data we considered. Eventually, we want to provide researchers with criterion on choosing an α value, but more work needs to be put into investigating various data scenarios before we can provide more informed advice.

Chapter 3

A Mixture Loss Function that Considers Discrimination and Moderate Calibration

3.1 Introduction

When fitting a personalized predictive model (PPM), one needs to decide how to define similarity, how many patients to include in the similar subpopulation and how to measure performance. Motivated by the fact that it has been criticized that calibration is not assessed nearly as often as discrimination [Van Calster et al., 2019], in Chapter 2, we proposed an algorithm to fit a PPM which jointly optimizes discrimination and calibration.

In our proposed algorithm, the size of subpopulation is tuned to minimize a loss function, shown in Equation (3.1), which is an extension of a decomposition of the Brier Score, an overall performance measure, that consists of terms related to both discrimination and calibration. In Equation (3.1), n_{test} denotes the size of the testing set (ie. the number of index patients to which we are fitting a PPM), y_I corresponds to the observed binary outcome of the index patient, I , and $p_I^{(M)}$ corresponds to the predicted probability for individual I experiencing the event, obtained from fitting a PPM on a similar subpopulation of size M .

$$L^{(M)} = \frac{\alpha}{n_{test}} \sum_{I=1}^{n_{test}} (y_I - p_I^{(M)})(1 - 2p_I^{(M)}) + \frac{1 - \alpha}{n_{test}} \sum_{I=1}^{n_{test}} p_I^{(M)}(1 - p_I^{(M)}). \quad (3.1)$$

The first term after the equal sign of this loss function is a measure of calibration, related to Spiegelhalter’s z-statistic, and the second term after the equal sign is a measure of discrimination, measuring the lack of spread in the predictions [Spiegelhalter, 1986]. Both of these terms have an expected value of 0 under perfect performance, and we include the mixture term $\alpha \in [0, 1]$ with the purpose of allowing the emphasis of one performance measure over another based on the needs of the researcher. For larger values of α , greater emphasis is put on calibration, while for smaller values of α , greater emphasis is placed on discrimination. When $\alpha = 0.5$, the loss function is proportional to the Brier Score.

Van Calster et al. [2016] propose a calibration hierarchy consisting of four increasingly strict levels of calibration: mean, weak, moderate and strong, which we discussed in Chapter 2. They say that, although strong calibration is the utopic goal, it is unrealistic in practice, thus we should strive for models that are moderately calibrated. Many calibration measures reported in the literature, such as the calibration-in-the-large and calibration slope, do not reach the level of moderate calibration. Thus, we want to attempt to use a moderate measure of calibration when possible, such as the Integrated Calibration Index [Austin and Steyerberg, 2019].

Many algorithms proposed to fit PPMs have been based on optimizing the size of subpopulation using a popular measure of discrimination, the area under the receiver operating characteristic curve (AUROC) [Ng et al., 2015, Wang et al., 2021, Campillo-Gimenez et al., 2013, Lee et al., 2015]. The relationship between the size of subpopulation, M , and the AUROC has thus been earlier investigated and, in Chapter 2, we confirmed the results of Lee et al. [2015] and demonstrated that the relationship between the AUROC and M is negatively proportional, aside from the lowest values of M . We do not know whether this relationship is specific to the data considered: in Lee et al. [2015], only one dataset was used, and in Chapter 2, we only considered one simulated dataset and one dataset from the eICU database [Pollard et al., 2018]. Further, in Chapter 2, we showed that although the best calibrated models use the entire training data, a smaller M may be the better choice that still results in a well-calibrated PPM. We want to further explore this relationship between M and calibration to determine whether this is always the case.

The purpose of this chapter is to address the limitations of the algorithm proposed in Chapter 2. Specifically, we propose an updated loss function to tune M that is more generalized and allows the user to choose the measures of discrimination and calibration in the function. We also test the proposed algorithm under a variety of datasets to further investigate the relationship between M and a number of performance measures, the results of which we then use to define lower and upper bounds on the grid of values on which to tune M to reduce the computational burden of the algorithm.

This chapter is organized as follows. We propose the updated loss function, along with an updated algorithm, in Section 3.2. This algorithm is then tested on 12 simulated datasets to determine the robustness of the relationship between M and various performance measures, the results of which are included in Section 3.3. Based on these results, we also provide recommendations for obtaining lower and upper bounds for the grid of values on which to tune M . In Section 3.4, our updated proposed algorithm is tested on data from the eICU Collaborative Research Database using two different loss functions, each differing in the measure of discrimination chosen.

3.2 Methods

3.2.1 Notation

Let there be n_{train} patients in the training data, where each patient k is represented by a Euclidean vector, $\mathbf{X}_k$, $k = 1, \dots, n_{train}$, in the multi-dimension predictor space defined by p predictors. We let x_h , $h = 1, \dots, p$, represent the $n_{train} \times 1$ vector consisting of each patient's value of a given predictor. Further, let $\mathbf{X}_I$, $I = 1, \dots, n_{test}$ be the Euclidean vector representation of the index patient, for whom we would like to make a prediction. Note that the index patient is not included in the training dataset.

The focus of the prediction task in this paper is on binary classification. Thus, we consider a binary outcome. The actual outcome for an index patient I is represented by $y_I \in \{0, 1\}$. Let M be a scalar representing the size of the subpopulation in the training data consisting of similar patients to the index patient. Finally, let $p_I^{(M)} \in [0, 1]$ represent the predicted probability of the outcome for a patient I found using a model fit on a similar subpopulation of size M .

3.2.2 Performance Measures

To measure the performance of a PPM, one should measure both the discrimination and calibration of the model [Steyerberg, 2019]. A common measure of discrimination is area under the receiver operating characteristic curve (AUROC), which plots the sensitivity (true positive rate) against $1 - \text{specificity}$ (false positive rate) for consecutive cutoffs for the probability of an outcome. A model with excellent discrimination will have an AUROC value close to 1, while an AUROC value close to 0.5 means that the model has hardly any class separation capacity. For a binary outcome, the AUROC is equivalent to the

concordance, or c , statistic: the estimated probability that predictions made by the model and true outcomes are concordant [Steyerberg et al., 2010].

In cases where the outcome we wish to predict is rare, ie. has a low prevalence, then a better measure of discrimination is the area under the precision recall curve (AUPRC) [Davis and Goadrich, 2006, Ozenne et al., 2015], which was used in the data analysis in Chapter 2. The precision recall curve plots the trade-off between precision, which is $\frac{\text{true positive}}{\text{true positive} + \text{false positive}}$, often referred to as the *positive predictive value* (or PPV), and sensitivity, for all possible thresholds. The AUPRC has a value between 0 and 1, but unlike the AUROC which will be 0.5 for an uninformative model, the AUPRC of an uninformative model is dependent on the prevalence, and approaches 0 as the prevalence decreases.

A third measure of discrimination, not used as often in practice, is the *lack of spread* in predictions [Spiegelhalter, 1986] which is the measure corresponding to discrimination in the Brier Score decomposition discussed by Rufibach [2010]. This term is 0 when the discrimination is perfect (ie. all predictions are either 1 or 0), and under the worst case scenario (ie. all $p_I^{(M)} = 0.5$), the value of this term is 0.25.

In Chapter 2, we introduced Van Calster et al. [2016]’s hierarchy of four increasingly strict levels of calibration, referred to as mean, weak, moderate, and strong calibration. Mean calibration measures the overall calibration between observed and predicted risks, and can be determined through measures such as the *calibration-in-the-large* (CITL) discussed in the previous chapter. Weak calibration is represented by the *calibration slope*, and is an indication as to whether there is overfitting or underfitting, and if there is any systematic overestimation or underestimation. A model is moderately calibrated if, among patients with the same predicted risk, the observed event rate equals the predicted risk. For example, among patients with an estimated risk of 40%, 4 out of 10 have experienced the event. Measures of moderate calibration are based on flexible calibration curves, which plot the predicted probabilities on the x -axis and the observed outcomes on the y -axis using, for example, loess or spline functions. These curves can reveal miscalibration that cannot be described by the two-parameter calibration line described above in weak calibration. To numerically assess moderate calibration, we can use the measure described in Chapter 2: the *Integrated Calibration Index* (ICI).

Recall one of the decompositions of the Brier Score [Rufibach, 2010], shown in Equation (3.2) below.

$$\frac{1}{n_{test}} \sum_{I=1}^{n_{test}} (y_I - p_I^{(M)})^2 = \frac{1}{n_{test}} \sum_{I=1}^{n_{test}} [(y_I - p_I^{(M)})(1 - 2p_I^{(M)})] + \frac{1}{n_{test}} \sum_{I=1}^{n_{test}} [p_I^{(M)}(1 - p_I^{(M)})] \quad (3.2)$$

The first term after the equal sign measures the lack of calibration, with an expectation of 0 under perfect calibration. Specifically, if we define the calibration in terms of the calibration hierarchy, we notice that since this term measures the overall agreement between the predicted probabilities to the actual outcomes, this term is actually a measure of mean calibration, the weakest measure. The second term after the equal sign, as mentioned above, is a measure of discrimination, specifically, it is the lack of spread of the predictions.

Van Calster et al. [2016] recommend that when fitting clinical predictions models, we strive for moderate calibration, however the Brier Score assesses discrimination and mean calibration. Further, it has been shown that the Brier Score may select a model that is miscalibrated with good discrimination over a model that is better calibrated with average discrimination [Assel et al., 2017]. These facts support that we should be assessing discrimination and calibration separately to obtain a better representation of a model’s performance, and justification of this point for PPMs was shown in Chapter 2.

3.2.3 Tuning the Size of Subpopulation

When fitting a PPM, Lee [2017] suggests that choosing the optimal size of subpopulation, M , will become important when taking these methods into practice. Motivated by this idea, in Chapter 2 we proposed an algorithm that tunes M to find the value that jointly optimizes discrimination and calibration using a loss function, shown in Equation (3.1) that is inspired by the decomposition of the Brier Score.

As we increase α , more emphasis is put on the calibration term in the loss function, which is a measure of mean calibration. Using this loss function is acceptable, especially if mean, weak and moderate calibration will always result in the same optimal M value. However, because we do not know this for sure, we propose another method of tuning the size of subpopulation. Motivated by the loss function in Equation (3.1), we present a more general loss function that allows the user to choose their preferred measures of calibration and discrimination, represented by C and D , respectively. This general loss function is shown in Equation (3.3), and note that we keep the mixture term $\alpha \in [0, 1]$, which allows flexibility in weighting one performance measure over another.

$$L^{(M)} = \frac{\alpha}{n_{test}} \sum_{I=1}^{n_{test}} C_k^{(M)} + \frac{1 - \alpha}{n_{test}} \sum_{I=1}^{n_{test}} D_k^{(M)} \quad (3.3)$$

We can, for example, use the ICI as the calibration measure, and lack of spread as the discrimination measure, to obtain the loss function shown in Equation (3.4), which

optimizes M in terms of discrimination and moderate calibration. In Equation (3.4), $\hat{p}_I^{(M)}$ are the smoothed predicted probabilities based on the LOESS calibration curve:

$$L^{*(M)} = \frac{\alpha}{n_{test}} \sum_{I=1}^{n_{test}} \left| \hat{p}_I^{(M)} - p_I^{(M)} \right| + \frac{1 - \alpha}{n_{test}} \sum_{I=1}^{n_{test}} p_I^{(M)} (1 - p_I^{(M)}). \quad (3.4)$$

The original iteration of this algorithm proposed in Chapter 2 also suffered from a large computational burden. One way of remedying this computational burden, that we are proposing to add in this chapter, is a way of defining a lower and upper bound on the grid of values used to tune M . Specifying a lower bound would also rectify the limitation of the algorithm where an optimal M proportion could be chosen in the training/testing step that is not a sufficient sample size in the validation step.

As mentioned before, we do not know the relationships between M and discrimination and calibration under different datasets. We will look at these relationships in Section 3.3 and then make a recommendation on bounds for this grid of values.

An updated algorithm is proposed below, with the purpose of (1) offering an improved way of finding the optimal size of subpopulation and (2) lessening the computational burden of tuning the size of subpopulation.

1. Split the entire population into a $(100q)\%$ hold-out validation set (n_{val}) and $(100[1 - q])\%$ training-testing (TrTe) set (n_{TrTe}), where $q \in [0, 1]$. We often use $q = 0.2$.
2. Determine a measure of discrimination and a measure of calibration to be included in the generalized loss function, $L^{(M)}$, displayed in Equation (3.3), along with a desired α value.
3. Tune M by following the steps below:
 - (a) Randomly split the TrTe set into distinct training and testing sets using K -fold cross validation.
 - i. For index patient I in the test set, calculate $d(\mathbf{X}_I, \mathbf{X}_k)$, for $k = 1, \dots, n_{train}$ in the training set, where $d(\mathbf{X}_I, \mathbf{X}_k)$ is the similarity metric between the index patient and the k^{th} training set patient.
 - ii. Using our recommendations for an upper and lower bound described in Section 3.3.2, define a grid of M values and tune M through random grid search.

- A. Sort all $d(\mathbf{X}_I, \mathbf{X}_k)$ in descending order, and the patients corresponding to the top M scores form the subpopulation.
 - B. Fit the model of interest, such as logistic regression, random forest, etc. on the subpopulation.
 - C. Average the values of the specified loss function over the K -folds.
 - iii. Repeat the K -fold cross validation v times, where $vK \geq 200$.
 - iv. Optimal M is chosen as the one which minimizes the updated loss function over the v repeated K -folds.
 - v. Optimal M is divided by the number of patients in the training set to obtain the optimal M proportion, $M_{optimal} = M/n_{train}$.
4. Draw B bootstrap samples from the validation sample (each the same size as the validation sample). Do the following steps for each bootstrap sample:
- (a) Randomly split the bootstrap sample into an 80% training sample and 20% testing sample.
 - (b) Set the integer $M = \text{ceiling}(n_{val.train}M_{optimal})$ where $n_{val.train}$ is the size of the training set.
 - (c) For index patient I in the test set, calculate $d(\mathbf{X}_I, \mathbf{X}_k)$, for $k = 1, \dots, n_{val.train}$ in the training set, where $d(\mathbf{X}_I, \mathbf{X}_k)$ is the similarity metric between the index patient and the k^{th} patient.
 - (d) Use the M obtained in step 4(b) to form the subpopulation.
 - (e) Fit the model of interest on the subpopulation.
 - (f) Measure the performance of the model using, for example, the discrimination and calibration measures discussed in Section 3.2.2.
5. To quantify the uncertainty of the performance measures, fit bias-corrected and accelerated (BCa) bootstrap confidence intervals for each of the performance measures using the B bootstrap estimates.
6. Repeat Step 1-6 Z times, ie. repeat the process with Z different validation holdout samples (and, thus, TrTe samples) to view consistency of the tuning of M (step (3) above) as well as the model performance measures in different hold-out validation samples. We suggest Z to be approximately ten.

Although seemingly very similar, there are two important distinctions between this algorithm and the one proposed in Chapter 2. Firstly, we updated the previous loss

function to be more general, shown in Equation (3.3), in the sense that we do not specify which measures of discrimination and calibration to be included in the function, they are user-dependent. Thus, in this algorithm, we have included Step 2 which involves choosing measures of discrimination and calibration to include in the defined loss function. Secondly, this algorithm contains specific instructions on how to define the grid of values on which to tune M with the intention of lowering the computation time of running this algorithm. We specifically motivate and state our recommendations for these bounds in Section 3.3.2.

3.3 Simulation Studies

We conducted two sets of simulation studies using R [R Core Team, 2020]. The first involves investigating the relationship between the size of subpopulation and six different performance measures under various outcome models. The goal of this simulation study was to determine if the relationships found in our previous chapter are robust against the true underlying outcome model. In the second simulation study, we demonstrate our previously proposed algorithm but with bounds put on the grid of values on which M is tuned to lessen the computational burden, and using the updated loss function with the lack of spread as the discrimination measure and the ICI as the calibration measure.

3.3.1 Setup

We considered a total sample size of $n = 10000$ and 40 predictors.

We replicated a scenario in which we have two clusters of patients of size $n_1 = 5000$ and $n_2 = 5000$, each cluster differing in the distributions of their respective predictors, as well as the strength of their association to the binary outcome of interest. Both patient clusters have the same predictors relating to the outcome, but the coefficients may differ. In the first cluster of patients, of the 40 predictors generated, 20 predictors were continuous with eight predictors simulated from a $N(30, 4)$ distribution, five predictors simulated from a $N(5, 1)$ distribution and seven predictors simulated from a $N(89, 29)$ distribution. We then simulated seven predictors from a $Bernoulli(0.3)$ distribution, nine predictors from a $Bernoulli(0.5)$ distribution and four predictors from a $Bernoulli(0.7)$ distribution. In the second cluster of patients, 20 predictors were continuous with eight predictors simulated from a $N(20, 8)$ distribution, five predictors simulated from a $N(15, 8)$ distribution and seven predictors simulated from a $N(50, 9)$ distribution. We then simulated seven predictors from a $Bernoulli(0.5)$ distribution, nine predictors from a $Bernoulli(0.15)$ distribution and four predictors from a $Bernoulli(0.45)$ distribution. These two clusters of

patients were included in one dataset of size $n_1 + n_2 = n = 10000$. We standardized this dataset so that all values of the predictors were between -1 and 1.

We defined 12 different true outcome models using the dataset described above. We simulated six outcomes from a linear model which differed by the number of predictors associated with the outcome, which could be either 10/40 or 20/40, or by prevalence of the outcome, which were low (~ 0.05), moderate (~ 0.15) or balanced (~ 0.5). We also simulated six outcomes from a non-linear model which differed by the number of predictors associated with the outcome, or by prevalence, using the same specifications described above.

We provide an example of one of these outcome models in Equation (3.5), specifically, the linear model which results in an outcome with low prevalence, and has ten predictors associated with the outcome. In this model, z_1 and z_2 are the linear predictors of the first and second cluster of patients, respectively. Each cluster of patients has the same predictors associated with the outcome, but these relationships generally differ in direction and/or strength of the association.

$$\begin{aligned} z_1 &= -3 - 3x_1 + x_2 + 0.5x_9 + x_{16} + x_{19} + x_{22} + x_{30} - x_{34} - 4x_{38} - 5x_{39} + \epsilon \\ z_2 &= -3.5 + 3x_1 + x_2 - x_9 + 2x_{16} + x_{19} + x_{22} + 4x_{30} + 5x_{34} + 3x_{38} + 3x_{39} + \epsilon \end{aligned} \quad (3.5)$$

After the outcome was simulated from one of the 12 different models, both z_i , $i = 1, 2$ were passed through an inverse logit function to obtain two probabilities, π_i , $i = 1, 2$, as shown in Equation (3.6). These probabilities were then used to simulate Bernoulli outcomes, y_{ki} , where $Y_{ki} \sim \text{Bernoulli}(\pi_i)$. Note that the simulated outcome of each index patient in the testing dataset, y_{Ii} , was generated in the same way.

$$\pi_i = 1/(1 + \exp(-z_i)). \quad (3.6)$$

The final dataset includes the 40 predictors and the binary outcome of all $n = 10000$ patients. We stress again that, as we considered 12 different outcome models, we ultimately simulated 12 different datasets.

To measure similarity between patients, we again define the similarity metric to be cosine similarity, ie. $d(\mathbf{X}_I, \mathbf{X}_k) = CSM(\mathbf{X}_I, \mathbf{X}_k)$.

In all simulation studies, we simulate 10 randomly generated datasets and, within each dataset, we conducted 20-repeated 10-fold cross validation.

In the validation step, we calculated a BCa bootstrap confidence interval, where we simulated $B = 1000$ bootstrap samples, for each performance measure.

3.3.2 Relationships between the Size of Subpopulation and the Performance Measures

In this section, we display the results from two of the 12 datasets. Dataset 1 has a linear true outcome model with 20 predictors associated with the outcome and a resulting prevalence that is approximately 0.15. Dataset 2 has a non-linear true outcome model, also with 20 predictors associated with the outcome and a resulting prevalence that is approximately 0.05. These descriptions are displayed in Table 3.1, and we will refer back to these datasets throughout this section. Results from the remaining ten datasets are found in Appendix A.

Table 3.1: Characteristics of two of the twelve datasets simulated.

	Dataset 1	Dataset 2
Linearity of outcome	Linear	Non-Linear
Number of predictors associated with outcome	20	20
Outcome prevalence	0.15	0.05

We analyze the relationships between M and six performance measures (AUROC, AUPRC, lack of spread, CITL, calibration slope and ICI) for Dataset 1 from Table 3.1, shown in Figure 3.1. The relationship between M and the AUROC and AUPRC, as seen in Figure 3.1(a) and (b), is negatively proportional after $M = 3000$, which matches the results seen in Chapter 2, ie. the discrimination improves up to a point, where it then begins to deteriorate. Under the lack of spread measure, displayed in Figure 3.1(c), the relationship is negatively proportional, even for the lowest values of M . As M increases, the discrimination deteriorates, meaning that a low value of M will result in a PPM with relatively better discriminative ability than a higher value of M . Figure 3.1(d)-(f) displays the relationship between M and calibration using the three increasingly strict measures of calibration: the CITL, calibration slope and the ICI. The calibration slope and the ICI both have a concave quadratic relationship with M , with both low and high value of M resulting in models with very good calibration. Under calibration slope, the best performing model is when $M = 1250$, which is the second best performing model under the ICI, the best, only by a difference of 0.003, being under $M = 9000$ for the latter measure of calibration. The CITL has a convex relationship with M , and is minimized at $M = 4000$. We do note that the range of CITL values, which has differences in the thousandths, specifically 0.003, is much smaller compared to the calibration slope and the ICI, which have a ranges of 0.631 and 0.017, respectively.

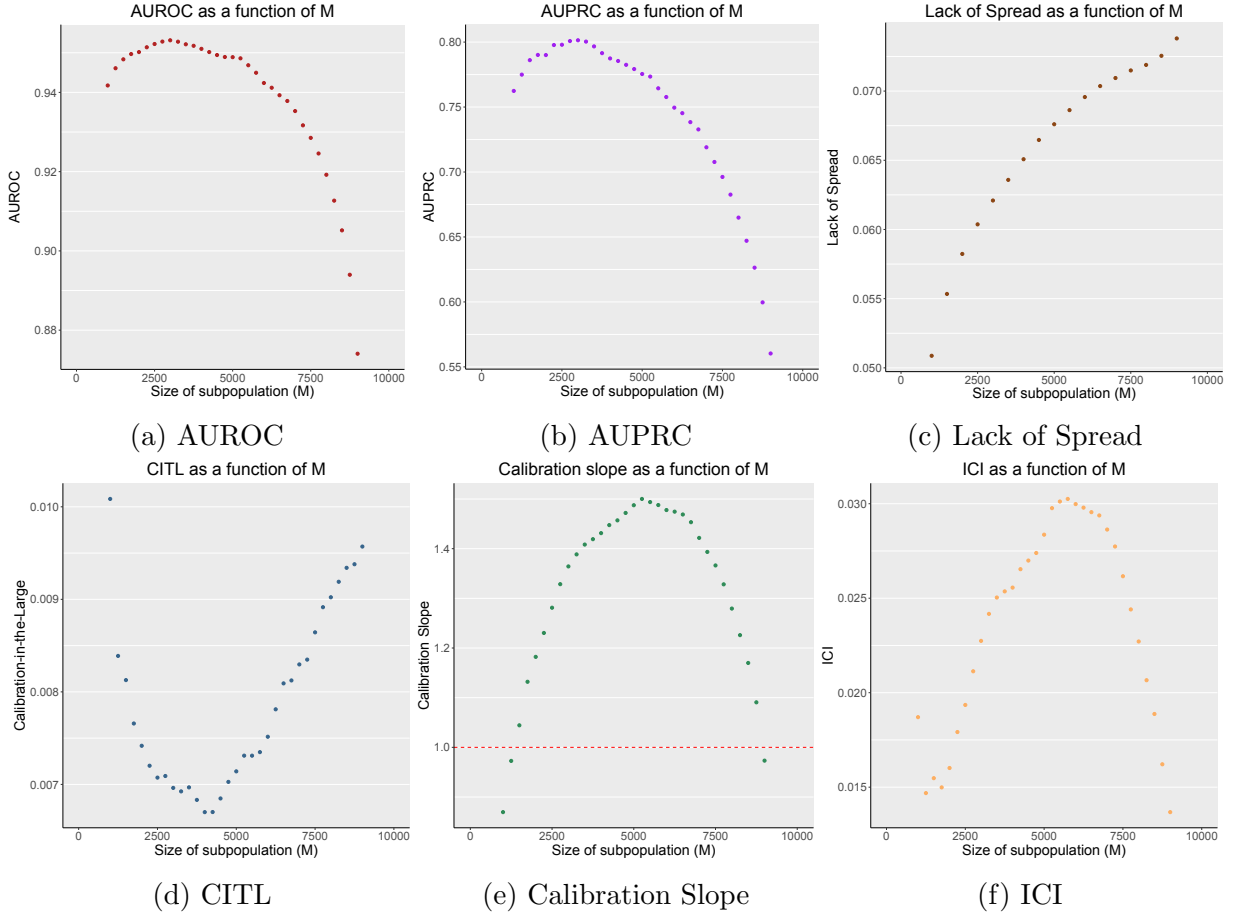


Figure 3.1: Performance measures as a function of M for Dataset 1 from Table 3.1.

The relationship between M and the same six performance measures for Dataset 2 from Table 3.1 is shown in Figure 3.2. As expected, the AUROC and the AUPRC improve up to a point, specifically at $M = 2500$, where they then begin to deteriorate as M increases. The relationship between lack of spread and M , shown in Figure 3.2(c), is also generally negatively proportional, however, at $M = 3500$, the discrimination under this measure begins to improve slightly before deteriorating again. Also, there is an outlier at $M = 9000$ that indicates improvement in the lack of spread from the previous M value of $M = 8500$. We would like to note that out of the 12 datasets considered, this case is the only one which did not result in a true negative proportional relationship between M and the lack of spread. The complexity of the outcome model (ie. non-linear, low prevalence and 20 predictors associated with the outcome) may have had an effect on this relationship.

Both the CITL and the ICI, as shown in Figure 3.2(d) and (f), are optimal under relatively larger values of M , the former being minimized at $M = 7250$ and the latter at $M = 6500$. The relationship between M and the calibration slope is, once again, concave, seen in Figure 3.2(e), and is optimal, or a negligible distance away from optimal, at both $M = 2250$ and $M = 8500$. Similar to what was seen in the previous case, the second best choice of M under the ICI is the same value which optimizes the calibration slope: $M = 2250$. All three measures are in agreement in terms of the worst calibrated model, which in the case of this dataset, is the model which is fit on the smallest size of subpopulation considered: $M = 1000$. The difference between the best calibrated PPM and worst calibrated PPM, in terms of the CITL, is 0.014, which is a much wider range compared to the range of CITL values in the previous dataset. The calibration slope also has a much wider range compared to what was seen in Figure 3.1, where here, the difference between the best and worst calibrated model is 0.989. The ICI has a similar range to the previous example, valued at 0.016. In this simulation study, higher values of M perform better than lower values of M in terms of calibration under all three measures, however, a lower value of M still results in a model with good weak and moderate calibration.

We would like to point out that the relationships observed in the above two cases are not unique to the specific characteristics of the datasets considered. For instance, Dataset 2 has an outcome that is generated from a non-linear model, and in Figure 3.2, we see that the optimal model under the ICI is one which is trained on approximately 65% of the training dataset. However, there were other datasets we considered, shown in Appendix A, which had outcomes generated from non-linear models and resulted in a low optimal proportion under the ICI. We cannot conclude that any of these results are due to the specific characteristics we varied, however, we can make some overall conclusions that we discuss below.

Under all 12 simulation studies, the relationship between M and discrimination was negatively proportional, aside from the lowest values of M when using the AUROC or the AUPRC as measure of performance. In terms of calibration, the relationships between M and calibration differed under different measures, as well as under different datasets. In many cases, a low M was good, if not the best, under the ICI. The calibration slope, in most cases, had a concave relationship with M , resulting in relatively better weak calibration in models fit on both small and large M values. Due to this, there was usually agreement between the optimal M under the calibration slope and the ICI. The optimal M under the CITL was more sensitive to changes in dataset, with no clear pattern amongst the 12 cases, but in many cases, it was not in agreement with the optimal M value found under the slope or ICI. As measures of moderate calibration ensure that models have good mean, weak and moderate calibration, we emphasize the recommendation by [Van Calster et al.](#)

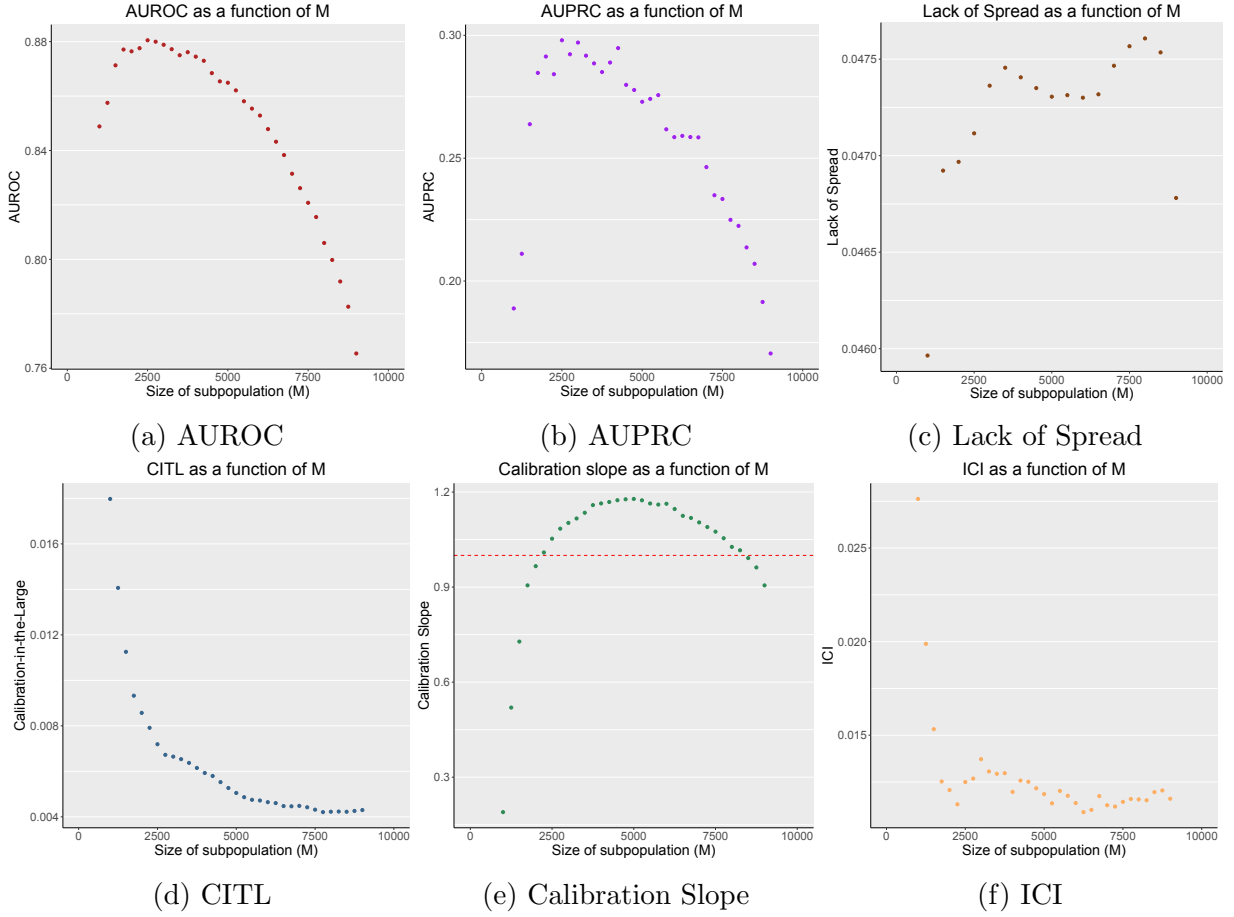


Figure 3.2: Performance measures as a function of M for Dataset 2 from Table 3.1.

[2016] and state that, in the general loss function shown in Equation (3.3), although there is flexibility in which calibration measure to use, we encourage the use of a measure of moderate calibration, such as the ICI.

Based on these results, we recommend that when tuning M , the lower bound on the grid of values should be $0.2n_{train}$, where n_{train} is the size of the training set (in this case, $n_{train} = 0.9 * n_{TrTe}$ to account for the 10-fold cross validation), and the upper bound on the grid of values to be either $0.5n_{train}$ or $0.7n_{train}$, depending on the interests of the study. We recommend the lower bound be 20% of the training data as we see that this proportion gives at the very least, good, if not the best, discriminative ability under the AUROC and AUPRC while still ensuring that the sample size in the validation step is sufficient. The

upper bound on this grid of values will depend on the α value chosen, specifically, how much of an interest there is in the calibration of the model. For $\alpha = 0.5$ or lower, thus if there is equal interest in both discrimination and calibration, or even a greater interest in discrimination, we suggest an upper bound of 50%, which greatly reduces the values on which to tune M while still capturing values which are known to include the optimal size under discrimination and, possibly, the optimal size under calibration as well. For larger values of α , ie. values of α greater than 0.5, when there is greater interest in calibration, we recommended an upper bound of 70% of the training data, which could capture the optimal value of M in cases where that value is large in terms of calibration. We do not recommend a value higher than 70% as an M value greater than this results in a significant deterioration in the discriminative ability of the model.

3.3.3 Empirically Showing the Brier Score Measures Mean Calibration and Implementing Bound Recommendations

Recall the mixture loss function proposed in Chapter 2, shown in Equation (3.1). This loss function is an extension of a decomposition of the Brier Score, consisting of a term corresponding to discrimination and one corresponding to calibration, and when we set $\alpha = 0.5$, this loss function is proportional to the Brier Score. The term corresponding to discrimination in this loss function is the lack of spread, and although we have given reasonable justification as to why the calibration term in this loss function is a measure of mean calibration in Section 3.2.2, here we will show this empirically.

Consider Dataset 1 from Table 3.1. We use the algorithm proposed in Section 3.2.3, where we use the mixture loss function described in Equation (3.1) to tune M , and use four values of α : 0.25, 0.5, 0.75 and 0.9. We chose a value below $\alpha = 0.5$ to test the sensitivity of the loss function when greater emphasis is put on the discrimination term. We also chose two values of α greater than 0.5 to investigate the effects of putting a heavier emphasis on the calibration term in the loss function. To empirically show that the Brier Score measures mean calibration, we expect that when we increase the weight on the calibration term in Equation 3.1, we are increasing the weight on a measure of mean calibration, and the resulting optimal M proportion will correspond to what was seen in Figure 3.1, ie. the optimal M proportion under a measure of mean calibration, the CITL, is approximately 0.5.

We also implement our bound recommendations in this simulation study. For $\alpha = 0.25$ and $\alpha = 0.5$, the grid of values on which M is tuned is bounded below by 20% of the training dataset and bounded above by 50% of the training dataset. For the remaining

two α values, we increase the upper bound to be 70% of the training dataset.

Figure 3.3 shows how the optimal M proportion found in the training/testing step of the algorithm changes as we increase the mixture term, α . We first note that, under all values of α , the optimal M proportion found through tuning is within the bounds we specified. This may not always be the case, since in some examples the optimal M proportion may be the entire training dataset or a very small M , especially when weak or moderate calibration is used as the calibration term. However, we find here that the bounds we recommend are sufficient and, at the very least, a good starting point when tuning M .

From Figure 3.1, recall that under weak calibration, a low value of M is optimal, and under moderate calibration, using the entire training dataset is optimal (although, a low M results in performance that is close to optimal). However, under mean calibration, the optimal model is the one which uses approximately 50% of the training dataset. In this simulation study, Figure 3.3 shows that when we put heavier emphasis on the calibration term in the mixture loss function (ie. $\alpha = 0.75$ and $\alpha = 0.9$), the optimal M proportion is between 0.48 and 0.61 across the randomly simulated datasets. This optimal M proportion is in agreement with Figure 3.1(d), which shows the relationship between M and a measure of mean calibration, thus we conclude that Brier Score is an overall performance measure that measures discrimination and, specifically, mean calibration.

3.3.4 Relationship Between the Size of Subpopulation and the Mixture Term

Now that we have a defined lower bound on the grid of values on which we tune M , we can consider values of α lower than 0.475 (ie. the lowest value of α considered in Chapter 2) since the defined bound ensures there will be a sufficient sample size for the validation step. Thus, in this simulation study, we consider the following values of α : 0.5, 0.5 ± 0.1 , 0.5 ± 0.25 , 0.5 ± 0.4 . We choose 0.5 as the baseline since this weight will put equal value on both the discrimination and calibration terms in the loss function.

We begin by considering Dataset 1 from Table 3.1. Recall in Figure 3.1, a low value M is associated with good discrimination as well as good weak and moderate calibration, but a value of M that is approximately 50% of the whole training data leads to a model with good mean calibration. We applied our proposed algorithm from Section 3.2.3, and we defined the measure of calibration in the loss function to be the ICI and the measure of discrimination to be the lack of spread to, ultimately, obtain the loss function shown in Equation (3.4). Figure 3.4 shows that the resulting optimal M under the seven values of α considered is low. This plot shows the relationship between M and α using five

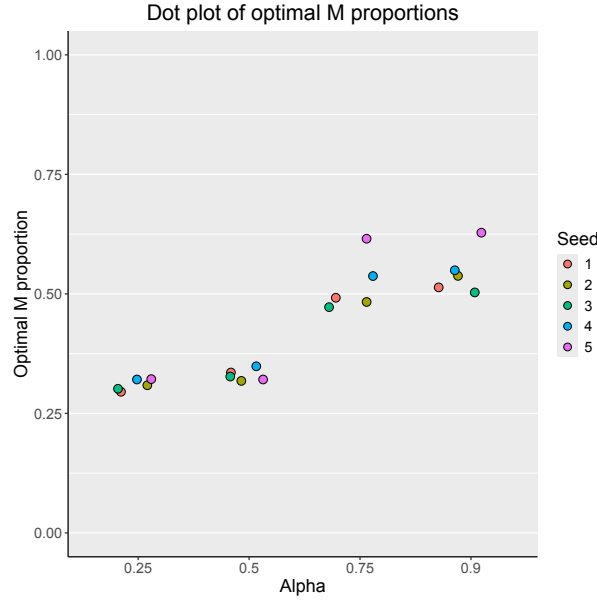


Figure 3.3: Dot plot showing the resulting optimal M proportions found in the training step of the algorithm across five different randomly generated datasets.

different randomly generated datasets, and is in line with the information displayed in Figure 3.1. In Figure 3.1, the optimal value of M , of one seed considered, under the ICI was $M = 9000$, but the difference between this optimal model and the second optimal model was very small, thus in this simulation study where we conducted 20 repeated 10-fold cross validation, across 10 different seeds, the more robust result shows that the smaller value of M was, indeed, the optimal value. For all 10 randomly generated datasets, the optimal M proportion is approximately 0.22 under all values of α , though in the plot we only display five out of 10 seeds. Also, we note that in Section 3.3.3, we applied this algorithm to this same dataset (ie. Dataset 1) using the original loss function shown in Equation (3.1), consisting of a measure of mean calibration as the calibration term, which lead to different results, as seen in Figure 3.3. Thus, the choice of performance measure to include in the loss function has an effect on the optimal M proportion found through tuning.

Next, we considered Dataset 2 described in Table 3.1. As shown in Figure 3.2, high values of M are associated with better mean, weak and moderate calibration, while low values of M are associated with better discrimination. In Figure 3.5, it is shown that as we increase α , the optimal M proportion also increases, under all 10 randomly generated

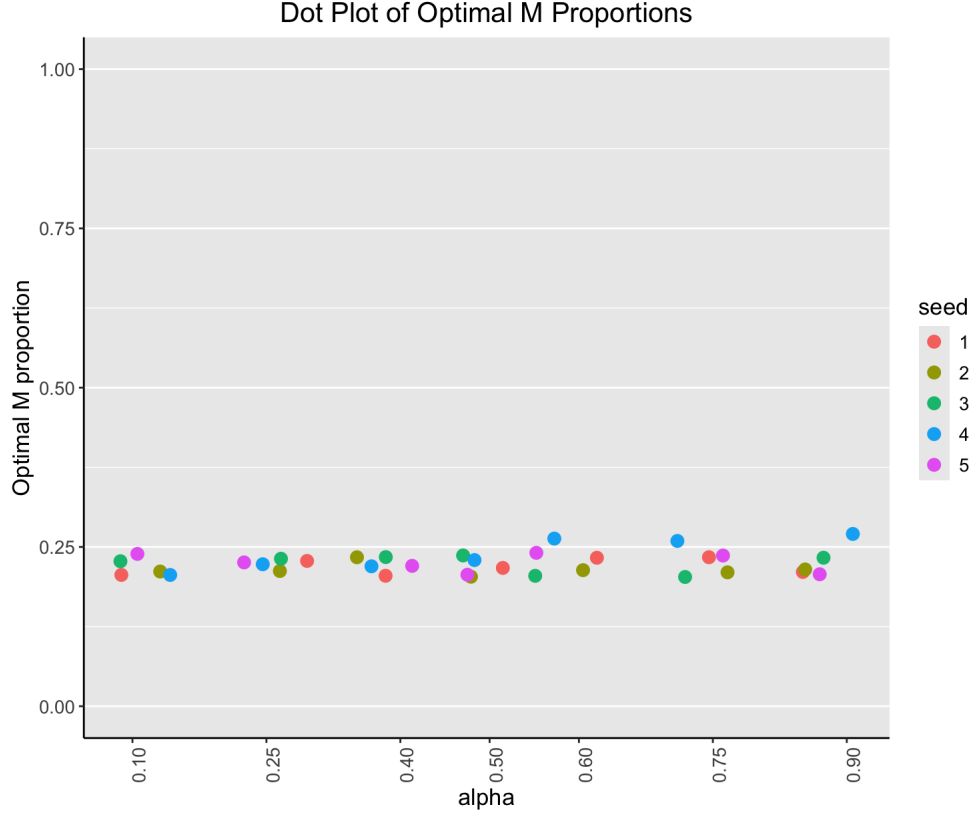


Figure 3.4: Resulting optimal M proportions for Dataset 1 from Table 3.1 under seven values of α and five different randomly generated datasets.

datasets, though only five out of 10 are shown in this plot. For higher values of α , there is more emphasis on the moderate calibration term in our loss function, thus these results coincide with the results seen in Figure 3.2: a higher value of M is associated with good moderate calibration. For low values of α , the optimal M proportion is between 0.2 and 0.4. For high values of α , the optimal M proportion is between 0.6 and 0.7. Unlike in Figure 3.4, there is more variability in the optimal M proportions, especially for values of α that are in the middle of the range (ie. 0.4, 0.5, 0.6). However, the results are still consistent with what we expected as there is still the increasing relationship between M and α , regardless of the outliers.

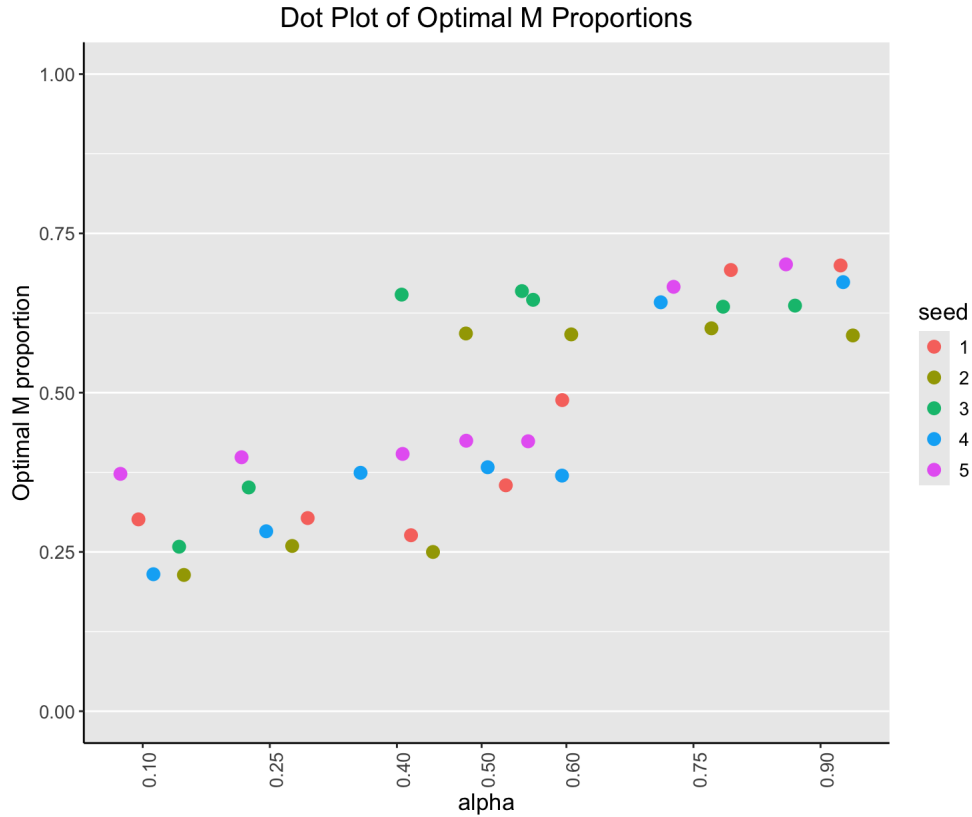


Figure 3.5: Resulting optimal M proportions for Dataset 2 from Table 3.1 under seven values of α and five different randomly generated datasets.

3.4 Data Analysis

We applied our proposed algorithm to a dataset from the eICU Database [Pollard et al., 2018] to predict mortality of patients with diseases of the circulatory system.

We used the same dataset that was used in Chapter 2. Recall, this dataset only considers patients in their first ICU stay of their first hospital visit and had a disease of the circulatory system, identified by their ICD9 code. These patients have survived in the ICU for at least 24 hours, but their stay did not exceed 14 days. Any patient with missing values in any of the predictors, as well as any patient with no vital sign data collected in the first 24 hours of their ICU stay, was removed. Refer to Figure 2.7 for a visualization of the cohort selection from the eICU database.

For each patient, the following predictors were extracted:

- Admission and demographic data: age, gender (male/female), hospital teaching status, and hospital discharge status (Expired/Alive).
- Vital signs from the first 24 hours after the start time of ICU admission: oxygen saturation (sao2), respiratory rate (respiration), and heart rate.
- APACHE IVa score: a severity-of-illness score utilizing demographic information as well as medical information collected within the first 24 hours of a patient’s ICU stay.

To identify the dominant modes of variation, we again used functional principal components (FPCs) of the vitals signs in the first 24-hours of a patient’s stay in the ICU. We extracted the first three FPCs for each of the vital signs, which cumulatively explained at least 95% of the total variation in the trajectory of each given vital sign. Our resulting dataset consists of 13 predictors, including the first three FPC scores for each of the three vital signs.

When testing the algorithm, we considered $\alpha = \{0.1, 0.25, 0.4, 0.5, 0.6, 0.75, 0.9\}$. We conducted 20 repeated 10-fold CV in the training/testing step of the algorithm, then validated the results with a holdout sample. We fit a logistic regression model to the data as the outcome (hospital discharge status) is binary. As the prevalence of outcome in this dataset is approximately 0.107, we use the AUPRC as a measure of performance.

We show the relationships between M and the performance measures when fitting PPMs to this dataset in Figure 3.6. As expected, Figure 3.6(a) shows that the AUPRC of the model increases up to a point, but instead of decreasing as M increases, the value of the measure stays relatively constant after this point, but still decreases slightly. The optimal M value under the AUPRC is $M = 9000$. Figure 3.6(b) shows the relationship between M and lack of spread, and, in general, as the size of subpopulation increases, the lack of spread of the model deteriorates, with the optimal M value being $M = 4500$. Figures 3.6(c) and (d) show that the optimal M under mean and weak calibration is very high, however we do note that the differences in the CITL between models fit using varying values of M is very small. Further, similar to what was seen in the simulation study, there are two values of M that result in optimal models under the calibration slope, with the first optimal value being approximately 50% of the entire training data, and the second being nearly the entire dataset. Under the ICI, the optimal M is the smallest value considered, however, moderate calibration performs very well for high values of M as well, as seen in Figure 3.6(e).

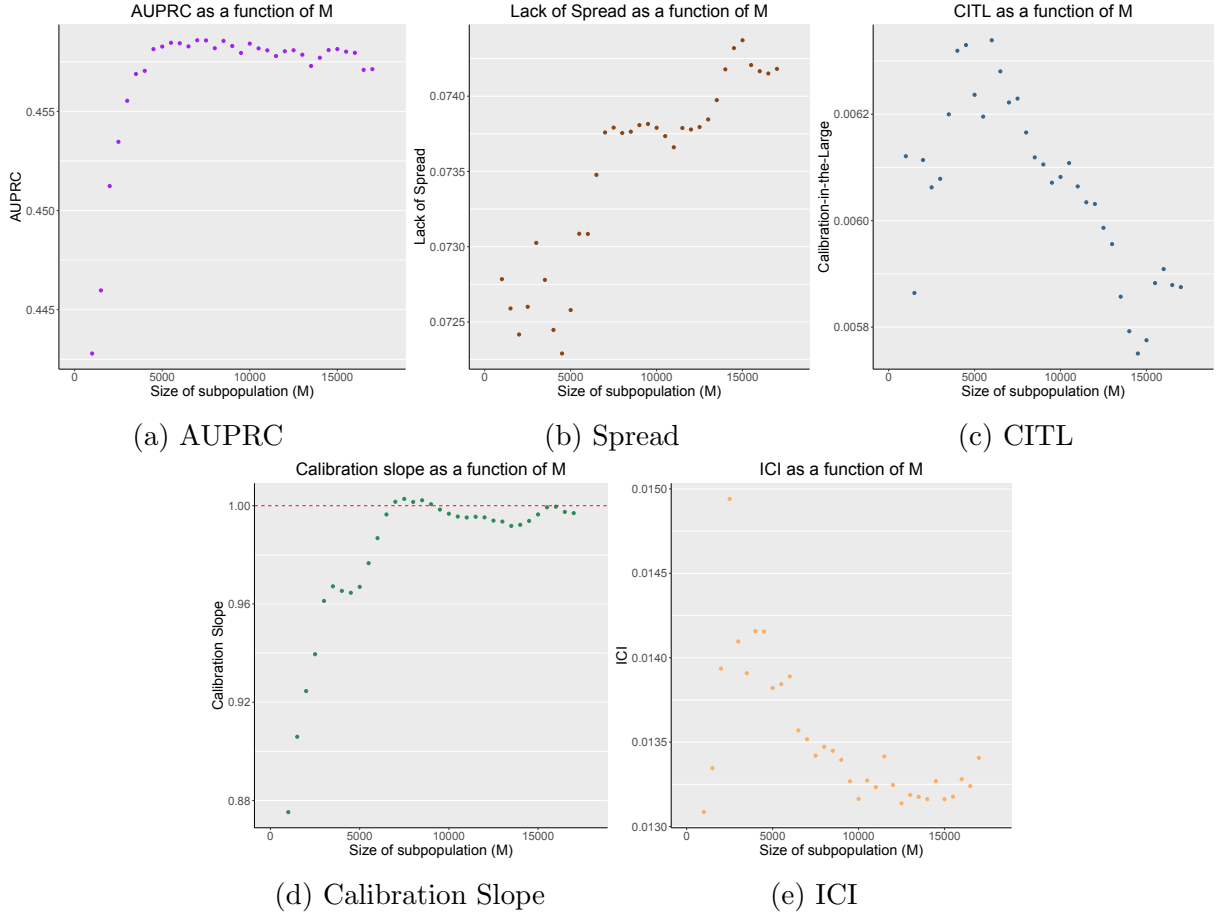


Figure 3.6: Performance measures as a function of M for the cardiac dataset from the eICU database

After investigating the relationships between M and the performance measures, we then applied the proposed algorithm from Section 3.2.3 to the eICU data. We applied this algorithm twice, considering two different loss functions. The first is the same loss function used in the simulation studies, shown in Equation (3.4), which consists of the ICI and the lack of spread measure, as the measures of calibration and discrimination, respectively. The second loss function we will consider consists of the ICI as the calibration measure and the AUPRC as the discrimination measure, as shown below in Equation (3.7). We approximate the AUPRC in the second term after the equal sign and define $P(c)$ as the precision at a cutoff c , $c = 1, \dots, d$, and $\Delta r(c)$ as the change in recall that occurred between cutoff $c - 1$ and cutoff c . We use $(1 - \text{AUPRC})$ as the measure since higher values of the

AUPRC correspond to better discriminative ability, and we wish to minimize L^{**} .

$$L^{**}(M) = \frac{\alpha}{n_{test}} \sum_{I=1}^{n_{test}} \left| \hat{p}_I^{(M)} - p_I^{(M)} \right| + (1 - \alpha) \left[1 - \sum_{c=1}^d P(c) \Delta r(c) \right]. \quad (3.7)$$

In both cases, we consider a grid of values to tune M which has a lower bound of 20% of the entire training dataset, and an upper bound of 70% of the entire training dataset, as per our recommendations given in Section 3.3.2. We choose 70% as the upper bound, which we recommended if the interest lies in calibration, rather than the stricter 50% in this data analysis just as a proof of concept, to check where the optimal M values lie in the grid of values considered under the various values of α .

We begin by displaying the results obtained when using Equation (3.4) as the loss function. A dot plot showing the resulting optimal M proportions is shown in Figure 3.7, and is consistent with what we expected: since both moderate calibration and the lack of spread perform well when M is low, the optimal M value would be low regardless of the choice of α . The optimal M proportion under all values of α is 0.29, except $\alpha = 0.9$ where the optimal M proportion is 0.20.

Following the training/testing step, we used the optimal proportions shown in Figure 3.7 in the validation step and assessed the performance of the resulting models. The performance of the PPMs under the seven α values, and the performance of the model which uses the entire available training data, is shown in Table 3.2. Here, we report the BCa confidence interval and the bootstrap standard error in addition to the point estimates of the AUPRC, the lack of spread, CITL, calibration slope, and the ICI. We note that under both the lack of spread measure and the ICI, both of which were included in the loss function used to tune the size of subpopulation, a model fit on a similar subpopulation performs better than one fit on the entire training dataset (ie. Proportion = 1.0). Specifically, when $\alpha = 0.1$, the resulting optimal M proportion is 0.29, which is the best performing PPM in terms of lack of spread, with the measure valued at 0.063, compared to the full model, which has a lack of spread value of 0.071. This is expected as low values of α result in a larger emphasis on the discrimination term in the loss function. When $\alpha = 0.9$, thus giving a larger emphasis on calibration in the loss function, the optimal M proportion is 0.2 and the value of the ICI corresponding to that PPM is 0.013 which is the best moderately calibrated model. The difference between the best and worst moderately calibrated model is 0.003, however these small differences are in line with what was found in Figure 3.6 when investigating the relationship between M and the ICI.

The AUPRC, which was not considered when tuning M but is a more commonly used measure of discrimination, is the best when the model is fit to the entire training dataset.

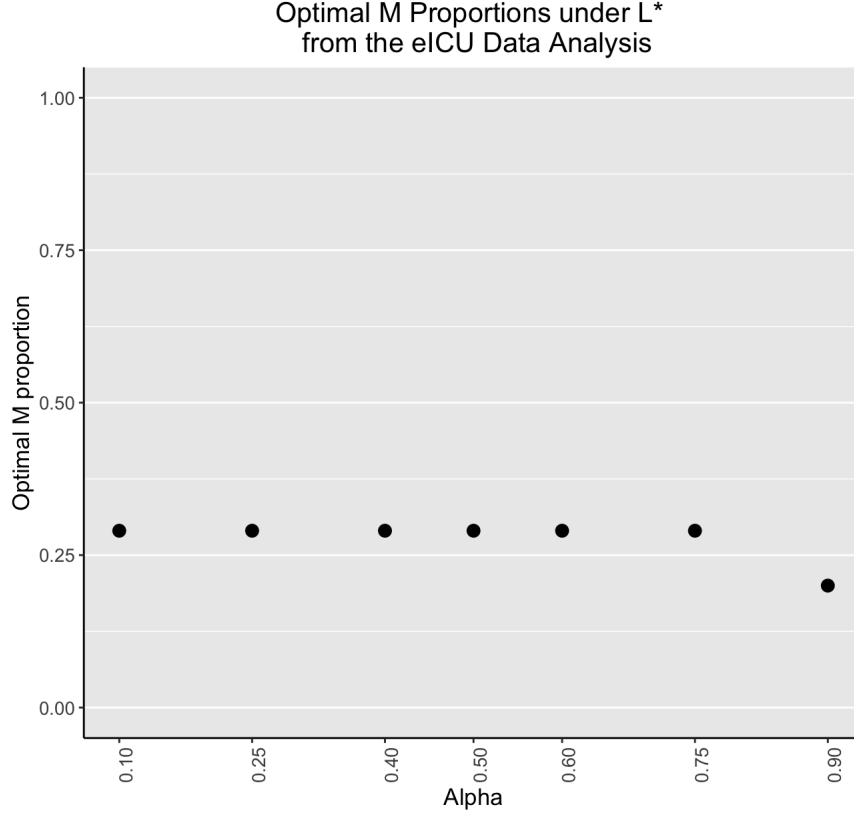


Figure 3.7: Optimal M proportions found using L^* from Equation (3.4) for varying values of α found when tuning M during the training/testing step of applying the proposed algorithm to the eICU dataset.

Under the entire training dataset, the AUPRC is valued at 0.478, while under the optimal M proportion when $\alpha = 0.1$, the resulting PPM has an AUPRC value of 0.475. Based on the investigations shown in Figure 3.6, the optimal M proportion under the AUPRC is expected to be approximately 0.5, with small differences between the AUPRC of models fit on subpopulations with sizes larger than 50% of the training data. We explore the changes in optimal M proportion when changing the measure of discrimination in the loss function from the lack of spread to the AUPRC later in this section.

Under the CITL, the best performing model is the one corresponding to an M proportion of 0.29. Although the point estimates of the CITL for the PPM fit on 29% and 20% of the entire training dataset are the same, the BCa confidence interval of the former shows better calibration than the latter. The model with the worst CITL is the full model.

Table 3.2: Cardiac Validation Under L^*

alpha	Proportion	AUPRC	Lack of Spread	CITL	Slope	ICI
0.1	0.29	0.475 (0.060) (0.363, 0.607)	0.063 (0.004) (0.059, 0.073)	0.021 (0.006) (0.017, 0.023)	1.015 (0.092) (0.914, 1.129)	0.015 (0.005) (0.006, 0.016)
0.25	0.29	0.475 (0.060) (0.363, 0.607)	0.063 (0.004) (0.059, 0.073)	0.021 (0.006) (0.017, 0.023)	1.015 (0.092) (0.914, 1.129)	0.015 (0.005) (0.006, 0.016)
0.4	0.29	0.475 (0.060) (0.363, 0.607)	0.063 (0.004) (0.059, 0.073)	0.021 (0.006) (0.017, 0.023)	1.015 (0.092) (0.914, 1.129)	0.015 (0.005) (0.006, 0.016)
0.5	0.29	0.475 (0.060) (0.363, 0.607)	0.063 (0.004) (0.059, 0.073)	0.021 (0.006) (0.017, 0.023)	1.015 (0.092) (0.914, 1.129)	0.015 (0.005) (0.006, 0.016)
0.6	0.29	0.475 (0.060) (0.363, 0.607)	0.063 (0.004) (0.059, 0.073)	0.021 (0.006) (0.017, 0.023)	1.015 (0.092) (0.914, 1.129)	0.015 (0.005) (0.006, 0.016)
0.75	0.29	0.475 (0.060) (0.363, 0.607)	0.063 (0.004) (0.059, 0.073)	0.021 (0.006) (0.017, 0.023)	1.015 (0.092) (0.914, 1.129)	0.015 (0.005) (0.006, 0.016)
0.9	0.20	0.441 (0.060) (0.358, 0.529)	0.066 (0.004) (0.058, 0.075)	0.021 (0.007) (0.020, 0.025)	0.981 (0.090) (0.905, 1.057)	0.013 (0.007) (0.007, 0.018)
	1.0	0.478 (0.060) (0.375, 0.594)	0.071 (0.004) (0.062, 0.076)	0.022 (0.006) (0.020, 0.026)	1.024 (0.099) (0.878, 1.198)	0.016 (0.005) (0.005, 0.027)

Similarly, the model with the best weak calibration, measured by the calibration slope, is the one corresponding to an M proportion of 0.29, while the worst model is the full model.

In this example, the best performing model would be the one corresponding to any of the α values except $\alpha = 0.9$, since almost all performance measures are better under an M proportion of 0.29. Further, even though this is not the optimal model under the ICI, the difference between the point estimates of this model and the one which is optimal is 0.002, so we sacrifice little in terms of moderate calibration. For simplicity, given the results we observed, we will say that the best performing model corresponds to $\alpha = 0.5$, which had a resulting optimal M proportion of 0.29.

We now discuss the results obtained using the second loss function, L^{**} , described in Equation (3.7), in applying our algorithm to the eICU cardiac dataset. Recall that this loss function consists of the AUPRC and the ICI as measures of discrimination and calibration, respectively. Based on the relationship between M and the AUPRC seen in Figure 3.6, we expected the optimal M proportion under L^{**} to be much higher for low values of α compared to what was optimal using L^* as the loss function, as shown in Figure 3.7. The results of this investigation are shown in Figure 3.8, where our hypothesis is shown to be correct. Unlike the results using the lack of spread as the measure of discrimination, here, for low values of α , the optimal M proportion is 0.49. The optimal M proportion remains

at 0.49 until we set α to be 0.5, which places equal weight on both the AUPRC and the ICI in the loss function, where the proportion then decreases slightly to 0.43. Then, for $\alpha = 0.9$, the optimal M proportion is 0.2, which is exactly what was found under the previous loss function. This is not surprising, as in both loss functions considered, we used the same measure of calibration. These results match the relationships between M and the AUPRC and M and the ICI we observed in Figures 3.6(a) and (e), respectively.

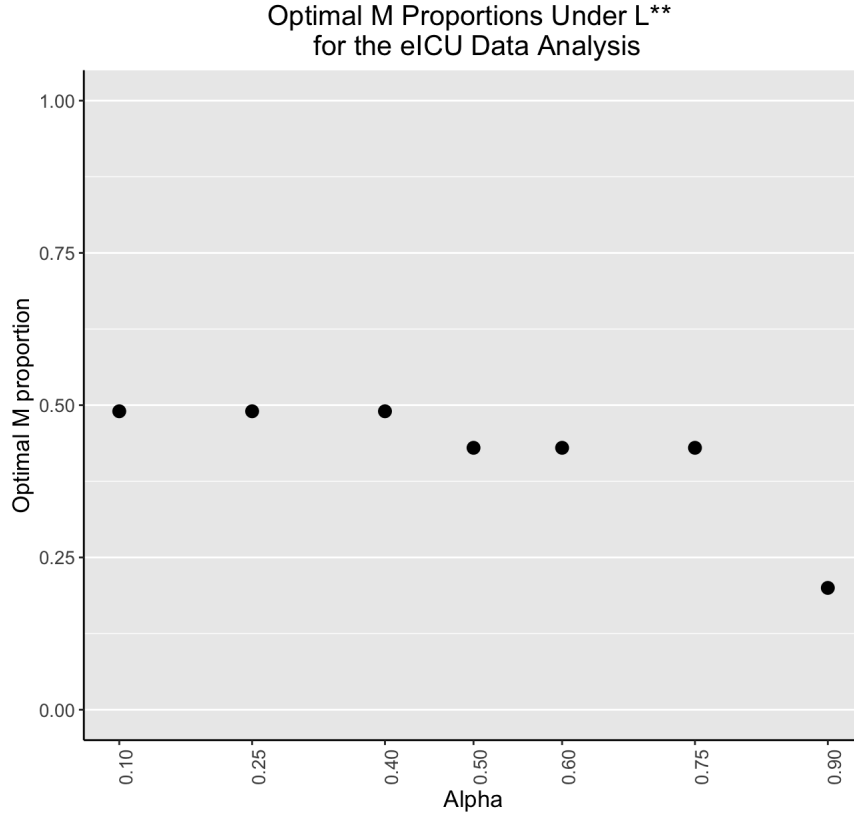


Figure 3.8: Optimal M proportions found using L^{**} from Equation (3.7) for varying values of α found when tuning M during the training/testing step of applying the proposed algorithm to the eICU dataset.

Using the optimal M proportion found under each value of α , we then validated the results and display our findings in Table 3.3. We immediately notice that the AUPRC of the model corresponding to $\alpha = 0.1$ has improved compared to the results using the prior loss function, shown in Table 3.2, and it outperforms the AUPRC of the full model. The resulting AUPRC performance confirms the relationship between M and the AUPRC we

observed in Figure 3.6, where the model with the optimal AUPRC is fit on approximately 50% of the full training dataset, but the difference in AUPRC values between the optimal and full models is very small. Specifically, here, the optimal model, which is fit on a similar subpopulation made up of 49% of the patients from the full training dataset, has an AUPRC value of 0.481, while the full model has an AUPRC value of 0.478. As expected, the lack of spread of the model is not optimal under an M proportion of 0.49, with a value that is 0.006 units higher than what was displayed in Table 3.2 when we set $\alpha = 0.1$, however, the performance is still better than the full model, which, again, confirms the relationships seen in Figure 3.6.

As we did not change the measure of calibration used to tune the size of subpopulation in the loss function, we do not see many changes in the calibration of these models. The model with the best moderate calibration is still the model corresponding to an M proportion of 0.2, which was determined when we put most of the weight in the loss function on the calibration term (ie. $\alpha = 0.9$). The model with the best mean calibration, measured through the CITL, is the one which is fit on 43% on the full training dataset, while the model with the best weak calibration is the one which is fit on 20% of the training dataset.

The overall optimal model using L^{**} as the loss function to tune M is not as clear of a decision as we observed when using L^* as the loss function. Here, the user would have to make a decision as to what they value more and what measure they are willing to sacrifice. As the differences between measures are very small, these sacrifices will in turn be small in size. We define the optimal model of this study as the one which results in the smallest sacrifices in terms of performance, which, in this case, corresponds to $\alpha = 0.1$, $\alpha = 0.25$ or $\alpha = 0.4$, where all three result in an optimal M proportion of 0.49. Without loss of generality, we conclude that the model corresponding to $\alpha = 0.4$ is optimal, where this model has the best discrimination in terms of the AUPRC, and the lack of spread is 0.001 units less than its optimal value, the CITL is 0.001 units smaller than its optimal value, the calibration slope is 0.005 absolute units smaller than its optimal value and the ICI is 0.003 units smaller than its optimal value across all eight models considered.

3.5 Discussion

In this chapter, we introduced a general loss function on which to tune M when fitting a personalized predictive model (PPM) which allows not only the flexibility of weighting one performance measure over another, but also allows flexibility in choosing which measures of model discrimination and calibration to be considered when optimizing the size of subpopulation. As mentioned in Chapter 2, further work is necessary to verify if this loss

Table 3.3: Cardiac Validation Under L^{**}

alpha	Proportion	AUPRC	Lack of Spread	CITL	Slope	ICI
0.1	0.49	0.481 (0.060) (0.375, 0.595)	0.069 (0.004) (0.059, 0.075)	0.022 (0.006) (0.020, 0.026)	1.024 (0.099) (0.878, 1.198)	0.016 (0.006) (0.005, 0.027)
0.25	0.49	0.481 (0.060) (0.375, 0.595)	0.069 (0.004) (0.059, 0.075)	0.022 (0.006) (0.020, 0.026)	1.024 (0.099) (0.878, 1.198)	0.016 (0.006) (0.005, 0.027)
0.4	0.49	0.481 (0.060) (0.375, 0.595)	0.069 (0.004) (0.059, 0.075)	0.022 (0.006) (0.020, 0.026)	1.024 (0.099) (0.878, 1.198)	0.016 (0.006) (0.005, 0.027)
0.5	0.43	0.475 (0.060) (0.369, 0.593)	0.068 (0.004) (0.060, 0.074)	0.021 (0.006) (0.020, 0.030)	0.848 (0.097) (0.660, 0.979)	0.014 (0.005) (0.005, 0.023)
0.6	0.43	0.475 (0.060) (0.369, 0.593)	0.068 (0.004) (0.060, 0.074)	0.021 (0.006) (0.020, 0.030)	0.848 (0.097) (0.660, 0.979)	0.014 (0.005) (0.005, 0.023)
0.75	0.43	0.475 (0.060) (0.369, 0.593)	0.068 (0.004) (0.060, 0.074)	0.021 (0.006) (0.020, 0.030)	0.848 (0.097) (0.660, 0.979)	0.014 (0.005) (0.005, 0.023)
0.9	0.20	0.441 (0.060) (0.358, 0.529)	0.066 (0.004) (0.058, 0.075)	0.021 (0.007) (0.020, 0.025)	0.981 (0.009) (0.905, 1.057)	0.013 (0.007) (0.007, 0.018)
	1.0	0.478 (0.060) (0.375, 0.594)	0.071 (0.004) (0.062, 0.076)	0.022 (0.006) (0.020, 0.026)	1.024 (0.099) (0.878, 1.198)	0.016 (0.005) (0.005, 0.027)

function is a proper scoring rule. Further, we investigated the relationship between M and various measures of model discrimination and calibration, and gave recommendations on bounds for the grid of values on which to tune M with the goal of reducing computation time.

We simulated 12 datasets that differed by their true outcome model, and analyzed the relationships between M and six performance measures: AUROC, AUPRC, lack of spread, CITL, calibration slope and the ICI. In all cases, M had a negatively proportional relationship with measures of discrimination, aside from the lowest values of M when using the AUROC and the AUPRC as performance measures, so a relatively low value of M will result in a PPM with relatively better discriminative ability than a model fit on a larger subpopulation. In many cases, this low value is good enough, if not the best, for model calibration as well. The optimal M was often very similar under the calibration slope and the ICI, which are measures of weak calibration and moderate calibration, respectively. This agreement between the optimal M values did not always extend to the CITL, a measure of mean calibration, which had an optimal M that was sometimes different than that under weak and moderate calibration. Based on the results from this simulation study, we recommend that the grid of values on which to tune M should be bounded below by 20% of the training data, and bounded above by either 50% or 70% of the

training data, depending on whether there is greater interest on model discrimination or calibration. A lower bound of 20% ensures that the sample size in the validation step will not be too small, and corresponds to what we see as optimal under discrimination in many cases we considered. An upper bound of 50% when there is equal interest in discrimination and calibration, or greater interest in discrimination (ie. $\alpha \leq 0.5$) tightens the bound and decreases computation time while still likely capturing the optimal M value under discrimination. An upper bound of 70% when greater interest lies in calibration (ie. $\alpha > 0.5$) increases the chance of capturing the optimal M under calibration if that value is large, while ensuring that we do not have too much of a deterioration in discriminative ability. Since a low value of M is good, if not the best, in many cases in terms of calibration, we can justify not having the entire size of the training dataset as the upper bound on the grid of values on which to tune M . By bounding the grid of values, computation time can be cut down significantly, which is important for when these methods are taken into practice, as PPM algorithms are computationally intensive. The cases we considered here are not part of an exhaustive list, and these recommendations on bounds are based on the 12 cases we did consider. Other datasets may differ in characteristics that may lead to the consideration of a wider grid, however, our recommendations serve, at the very least, as a starting point. Future work includes investigating these relationships under more datasets, as well as more models (ie. random forest), especially the relationships between M and calibration, in order to propose bounds that are more robust.

In Chapter 2, we proposed a loss function that is an extension of the Brier Score, and in this current chapter, we further investigated the calibration component of this loss function to classify it into the calibration hierarchy. We intuitively and empirically showed that the previous loss function is a joint measure of discrimination and mean calibration. Motivated by this and by the differing optimal M values between mean and moderate calibration shown in Section 3.3.2, we proposed a general loss function which is a joint measure of discrimination and calibration, but the user can choose which measures of discrimination and calibration to consider. Through simulation study, we tested a loss function consisting of the lack of spread as a measure of discrimination, and the ICI as a measure of calibration, using multiple α values on simulated data to find the optimal size of subpopulation when fitting a PPM, and found that this new loss function resulted in different values of optimal M when compared to those found using the original loss function. Further, we compared the resulting optimal M values under varying values of α to the relationships we investigated in Section 3.3.2, and as we increase α , heavier weight is given to the calibration term in the loss function, resulting in an optimal M value that matches the relationships we display in Figures 3.1 and 3.2.

We also applied the proposed algorithm to a dataset from the eICU database to pre-

dict the mortality of patients with diseases of the circulatory system. We considered two different loss functions which both had the ICI as the measure of calibration, but differed in the defined measure of discrimination, which was either the lack of spread, as was done in the simulation study, or the AUPRC, a more common measure of discrimination. In the first study, where the discrimination in the loss function was the lack of spread, the optimal M proportion under all values of α was low, specifically, it was 0.29 for all values except for $\alpha = 0.9$, where the optimal M was 0.2. In the second study, there was more variability in the optimal M proportions across α values, and the optimal M proportions when discrimination was emphasized (ie. when α was a value close to 0), was a much higher value at 0.49. These results coincided with the relationships between M and spread, as well as M and the AUPRC which we saw in Figure 3.6: very low values of M correspond to low values of lack of spread, while moderate values of M correspond to high values of the AUPRC. We fit PPMs to the hold-out dataset using the optimal M proportions found in the training/testing step and calculated BCa bootstrap confidence intervals for each of the point estimates of the performance measures in both cases. Under the first loss function, we concluded that the optimal model is the one corresponding to $\alpha = 0.5$, which resulting in an optimal M proportion of 0.29, where all performance measures were optimized except the ICI, which performed optimally under $\alpha = 0.2$ (ie. $M_{proportion} = 0.2$). The model fit to the full training dataset had the worst performance across all performance measures, except under the AUPRC, where it performed slightly better. When using the AUPRC as the discrimination in the loss function, we concluded that the optimal model corresponded to $\alpha = 0.4$ (ie. $M_{proportion} = 0.49$), as choosing this model resulted in the smallest sacrifices made in calibration performance. Further, the full model, under the second loss function, was the worst performing model across all performance measures considered. Through this data analysis, we showed how the optimal model differs when changing the choice of performance measures to use in the loss function. By allowing this flexibility, we give users more autonomy on the choice of optimal M value for their specific study.

This study has a few limitations. Computational intensity continues to be the greatest challenge to overcome before taking these algorithms into practice. We have improved on the computation time by introducing bounds to the grid of values used to tune M , however, this is not sufficient, especially if these methods are to be used in the intensive care unit, where decisions must be made very quickly. As mentioned prior, we focus our efforts on a relatively basic model, logistic regression, to explore these relationships between the size of subpopulation, and the effects of patient similarity methods on prediction, under a simple case. Next steps of this work include investigations under more sophisticated machine learning models, that is, to see what the optimal M values would be using the same data but under different models. However, this exploration needs to await further developments

on computational efficiencies for the proposed algorithm.

Chapter 4

A Personalized Approach to Flexible Modelling of Longitudinal Data

4.1 Introduction

By definition, electronic health records (EHRs) represent longitudinal data that has been collected during routine delivery of health care [Cowie et al., 2017]. These data can be used to make individual predictions for the trajectory of a specific outcome for a given patient, with the purpose, for example, of providing suitable treatment [Bhagwat et al., 2018, De Brouwer et al., 2021, Spence et al., 2024].

One of the simplest models that can be used for longitudinal prediction is the *linear mixed effects model*, where both fixed and random effects are used to model a response, the latter which represents subject-specific effects that are unique to an individual [Fitzmaurice et al., 2012]. Using a model with mixed effects allows for the analysis of both between-subject and within-subject variability in the longitudinal responses.

In developing these prediction models, one needs to be aware of the challenge of dealing with potential nonlinear associations between predictors and the outcome [Eekhout et al., 2023]. One way of mitigating this challenge is to fit a *generalized additive mixed effects model*, which allows the flexibility of considering non-linear association between variables, as well as the ability to account for random effects [Lin and Zhang, 1999]. In these models, part of the linear predictor is specified in terms of smooth functions of predictors, which can then be estimated using methods such as smoothing splines.

In settings with a non-time varying, binary outcome, it has been shown that one-size-

fits-all models do not perform as well as models built for an individual using a similar subpopulation [Celi et al., 2012, Lee et al., 2015]. To determine the similar subpopulation, one must first define a similarity metric to measure similarity between a patient for whom one wishes to make a prediction (ie. the index patient), and any other patient in the training data. When using EHR data, one needs to consider that many predictors may be longitudinal in nature. This challenge has been long investigated in the time-series literature, where they call it the *similarity search*: determining the degree to which similarities exist between two or more time-series data sets [Fang et al., 2015, Yang and Shahabi, 2004]. In personalized predictive modelling, it is common to see data reduction techniques used prior to measuring similarity, so that classic measures such as Mahalanobis distance or cosine similarity can then be used. These data reduction methods include determining summary statistics of the longitudinal data [Lee et al., 2015], or using functional principal components [Yao et al., 2005], the latter having the ability to consider the temporal aspect of the longitudinal predictors [Sharafoddini et al., 2021]. In Chapters 2 and 3, we used functional principal component analysis (fPCA) in the data analysis to extract the functional principal components (FPCs) of longitudinal predictors, and used the FPC scores as predictors when measuring similarity between patients, prior to making a binary response prediction for a given index patient.

As just mentioned, in previous chapters, we considered a binary outcome, and we investigated the nature of the relationships between the size of subpopulation and measures of discrimination and calibration. To our knowledge, a personalized approach has not been used to predict the continuous, longitudinal outcome trajectory of an index patient. Specifically, the relationships between the size of subpopulation and the predictive performance of modeling individual longitudinal trajectories has not been investigated. We want to determine whether a personalized predictive model (PPM), such as the ones fit in Chapters 2 and 3, will result in better predictions in the case of predicting the trajectory of a continuous outcome over time for an index patient compared to a one-size-fits-all model. Further, as it is important to determine the optimal size of subpopulation prior to making these personalized predictions [Lee, 2017], we propose an algorithm to tune the size of subpopulation for fitting a longitudinal prediction model.

The remainder of this chapter is structured as follows. Section 4.2 contains a more detailed description of both the longitudinal model we will be fitting, as well as the proposed algorithm to determining the optimal size of subpopulation for predicting the trajectory of an index patient using a personalized approach. In Section 4.3, we outline the data-generating mechanisms of our simulation study, which involves applying the proposed algorithm to two simulated datasets in order to investigate relationships between the size of subpopulation and a measure of overall performance, the *mean absolute error*, when

predicting an individual's longitudinal trajectory. We then present the results from two follow-up simulation studies to further investigate these relationships. We include a discussion of the results in Section 4.4 and point to future work in this research area.

4.2 Methods

4.2.1 From Linear Mixed Effects Models to Additive Mixed Effects Models

The notation in this chapter differs from the previous two chapters, as we are now dealing with time-varying predictors and, most notably, a time-varying outcome.

Consider the following model:

$$y_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})t_{ij} + \beta_2 x_{2i} + \beta_3 x_{3i} + \epsilon_{ij}. \quad (4.1)$$

Equation (4.1) is a *linear mixed effects model* for continuous, longitudinal data, where y_{ij} is the response for individual i , $i = 1, \dots, N$ at a time point j , $j = 1, \dots, n$, where, in this simplified setup, all individuals have the same time points. Let the number of predictors be represented by p , where $h = 1, \dots, p$. In this example, we have $p = 3$, where this includes the time covariate. The coefficients β_0 , β_1 , β_2 , and β_3 are fixed, population-level effects, and we let x_{hi} be the value of the baseline predictor h corresponding to individual i . The coefficients b_{0i} and b_{1i} are random effects, which represent the between-subject variability around the population-level effects, specifically around the population-level intercept and the time effect, respectively. These random effects are random variables that follow a normal distribution with a single, unknown parameter; specifically, $b_{0i} \sim N(0, \sigma_{b_0}^2)$ and $b_{1i} \sim N(0, \sigma_{b_1}^2)$. If we let q be the number of random effects, where in this case $q = 2$, then $\mathbf{b}_i = \begin{bmatrix} b_{0i} \\ b_{1i} \end{bmatrix} \sim N_q(0, D)$, where D is a $q \times q$ covariance matrix for $\mathbf{b}_i$, and is assumed to be the same for each individual. Thus, the random effects induce a correlation structure for the repeated measures within a subject. The constant ϵ_{ij} represents the within-subject error, and since we are assuming all individuals have the same time points, all individuals will have the same covariance matrices. Specifically, $\boldsymbol{\epsilon}_i \sim N(0, \Sigma)$, where $\boldsymbol{\epsilon}_i$ is an $n \times 1$ vector of within-subject error terms for individual i , and Σ is a $n \times n$ covariance matrix, which may account for an additional correlation structure, such as serial correlation, among an individual's repeated measures. Finally, we assume that each of the random effects and the within-subject errors are independent.

Linear mixed effects models provide the basis for a more flexible type of model for longitudinal data that combines the flexibility of generalized additive models introduced by [Hastie \[2017\]](#) and allows for the ability to account for random effects: the *additive mixed effects model*. Equation (4.2) shows an example of an additive mixed model, which is identical to Equation (4.1) except for the relationship between time and the outcome, where we now represent time as a function, $g(t_j)$. We note that the subscript i has been suppressed from the predictor, t_{ij} , as time is not currently assumed to be subject-specific:

$$y_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})g(t_j) + \beta_2 x_{2i} + \beta_3 x_{3i} + \epsilon_{ij}. \quad (4.2)$$

Using an additive mixed effects model allows for non-linear relationships between predictors and the response variable, and these relationships can be estimated using smooth functions, such as penalized splines, though, note in Equation (4.2) that the only smooth relationship is connected to time (and not other predictors). The penalized spline representation of Equation (4.2) is shown in Equation (4.3), where $\kappa_k, k = 1, \dots, K_T$ is a distinct knot in the range of t_j , and $a_+ = \max(0, a)$.

$$y_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})t_j + \sum_{k=1}^{K_T} b_k^{(T)}(t_j - \kappa_k)_+ + \beta_2 x_{2i} + \beta_3 x_{3i} + \epsilon_{ij} \quad (4.3)$$

Equation (4.3) is subject to the constraint

$$\sum_{k=1}^{K_T} (b_k^{(T)})^2 < B_1$$

for some finite constant, B_1 .

Equation (4.3) is equivalent to a linear mixed effects model where $b_k^T \sim N(0, \sigma_T^2)$ [[Brumback et al., 1999](#)]. In this example, we now have $K_T + 2$ random effects. All standard linear mixed effects models methods and software can be applied when fitting an additive mixed model with penalized splines.

4.2.2 A Personalized Approach to Additive Mixed Models using Penalized Splines

Motivated by previously proposed algorithms in Chapters 2 and 3, we propose a new algorithm for fitting a PPM when there are longitudinal predictors present and the goal is to predict a continuous, longitudinal outcome trajectory for an index patient.

Let each patient i , $i = 1, \dots, n_{train}$, be represented by a Euclidean vector, $\mathbf{X}_i$, in the multi-dimension predictor space defined by p baseline predictors, where some of these predictors are the FPC scores of longitudinal predictors. Further, let $\mathbf{X}_{Index}$, $Index = 1, \dots, n_{test}$, be the Euclidean vector representation of the index patient, for whom we would like to make a prediction. Note that the index patient is not included in the training dataset.

1. Fit the model of interest, such as an additive model using penalized splines, on the entire dataset to determine the functional form.
2. Split the entire population into a $(100q)\%$ hold-out validation set (n_{val}) and $(100[1 - q])\%$ training-testing (TrTe) set (n_{TrTe}), where $q \in [0, 1]$. Here, we suggest $q = 0.3$, as to ensure that the validation sample is large enough to result in accurate predictions of individual trajectories.
3. Determine the FPC scores for each time-varying predictor such that 95% of the variation is accounted for.
4. Tune M by following the steps below:
 - (a) Randomly split the TrTe set into distinct training and testing sets using K -fold cross validation.
 - i. For index patient I in the test set, calculate $d(\mathbf{X}_{Index}, \mathbf{X}_i)$, for $i = 1, \dots, n_{train}$ in the training set, where $d(\mathbf{X}_{Index}, \mathbf{X}_i)$ is the similarity metric between the index patient and the i^{th} training set patient. Use the baseline predictors and the FPC scores as input.
 - ii. Consider a grid of M values and tune M through random grid search.
 - A. Sort all $d(\mathbf{X}_{Index}, \mathbf{X}_i)$ in descending order, and the patients corresponding to the top M scores form the subpopulation.
 - B. Fit the model of interest, such as an additive mixed model using penalized splines, on the subpopulation.
 - C. Average the value of the overall performance measure chosen (eg. mean absolute error, mean squared prediction error, etc.).
 - iii. Repeat the K -fold cross validation v times, where $vK \geq 200$.
 - iv. Optimal M is the one which optimizes the chosen performance measure.
 - v. Optimal M is divided by the number of patients in the training set to obtain the optimal M proportion, ie. $M_{optimal} = M/n_{train}$.

5. Draw B bootstrap samples from the validation sample (each the same size as the validation sample). Do the following steps for each bootstrap sample:
 - (a) Randomly split the bootstrap sample into an 80% training sample and 20% testing sample.
 - (b) Set the integer $M = \text{ceiling}(n_{val.train}P_{optimal})$ where $n_{val.train}$ is the size of the training set.
 - (c) For index patient I in the test set, calculate $d(\mathbf{X}_{Index}, \mathbf{X}_i)$, for $i = 1, \dots, n_{train}$ in the training set, where $d(\mathbf{X}_{Index}, \mathbf{X}_i)$ is the similarity metric between the index patient and the i^{th} patient.
 - (d) Use the M obtained in step 5(b) to form the subpopulation.
 - (e) Fit the model of interest on the subpopulation.
 - (f) Measure the performance of the model using overall measures of interest (eg. mean absolute error, mean squared prediction error, etc.).
6. In order to quantify uncertainty around the performance measures, determine a 95% bootstrap percentile confidence interval for the overall performance measure using the B bootstrap estimates.
7. Repeat Steps 1-6 Z times, ie. repeat the process with Z different validation holdout samples (and, thus, TrTe samples) to view consistency of the tuning of M (step (4) above) as well as the model performance measures in different hold-out validation samples. We suggest Z to be approximately ten.

This algorithm is similar to those proposed in Chapters 2 and 3, although there are some very obvious differences. Since we are fitting a longitudinal model, we need to determine the functional form of the data prior to fitting the PPMs. We assume, naively, that the PPMs will have the same functional form as the one-size-fits-all model, and we determine this functional form in Step 1 by fitting the model of interest on the entire dataset. Another key difference is we include Step 3, which involves determining the FPC scores of the time-varying predictors, which we will use as input into the similarity metric, along with the baseline predictors, in Step 4(a). Although we did this in the data analysis in Chapters 2 and 3, we now explicitly state this as a step in the algorithm. Step 4(a)ii includes fitting a longitudinal model and determining the value of an overall performance measure, such as the mean absolute error, which is different from the algorithms in Chapters 2 and 3 where we defined a loss function that consisted of discrimination and calibration subparts. In this study, we are predicting the continuous, longitudinal outcome trajectory of a model, and

as discrimination is defined as a model’s ability to distinguish between outcome classes, measures of discrimination cannot be used in this context to measure performance.

4.3 Simulation Studies

Below, we describe four simulation studies that are based on different data generating mechanisms and all coded in R [R Core Team, 2020]. We begin with a complex case, and decrease the complexity as we move through the other three simulations. The goal in each case is to determine whether the overall performance of a prediction model can be improved when using a similar subpopulation of patients rather than the full training data. In all four simulation studies, the performance measure used is the mean absolute error, defined as $MAE = \sum_{\text{Index}=1}^{n_{\text{test}}} \sum_{j=1}^n |y_{\text{Index},j} - \hat{y}_{\text{Index},j}|$, where $y_{\text{Index},j}$ is the observed outcome of the index patient at time j , and $\hat{y}_{\text{Index},j}$ is the predicted value for the index patient at time j from the model. Further, in all simulation studies, within each randomly generated dataset, we conduct 20-repeated 10-fold cross validation. We note once again that, for the purposes of this simulation, all individuals are observed at common time points and there are no missing data points.

4.3.1 Simulation 4.1: Predicting a Continuous, Longitudinal Outcome: A Complex Case

We simulated $N = 3000$ individuals, and randomly split the dataset to obtain a TrTe sample of size $n_{\text{TrTe}} = 2100$ and a hold-out validation sample of size $n_{\text{val}} = 900$. Similar to the simulation study done in Chapter 3, we simulated the data such that there were two clusters of patients that generally differed in the distribution of their predictors, each cluster consisting of 1500 individuals. We simulated ten predictors: eight baseline continuous predictors and two time-varying continuous predictors. The distributions that each of the predictors were simulated from for each patient cluster are shown in Table 4.1. The superscript on the predictors indicates the patient cluster for which that predictor was simulated.

To generate the time-varying predictors, which we represent by x_{ij} , we used Equation (4.4), where $\mathbf{b}_i = \begin{bmatrix} b_{0i} \\ b_{1i} \end{bmatrix} \sim N_{q=2}(0, D)$, and the matrix $D = \begin{bmatrix} \tau_0^2 & \tau_0\tau_1\tau_{01} \\ \tau_0\tau_1\tau_{01} & \tau_1^2 \end{bmatrix}$. In the matrix D , τ_0 is the population variance of the random intercept, τ_1 is the population variance of the random slope and τ_{01} is the covariance of the random intercept and random slope.

Table 4.1: Details of the eight simulated baseline predictors for the two clusters of patients in Simulation 4.1. The table on the left corresponds to the first patient cluster, while the table on the right corresponds to second patient cluster.

Predictor	Distribution	Predictor	Distribution
$x_1^{(1)}$	$N(80, 15)$	$x_1^{(2)}$	$N(60, 10)$
$x_2^{(1)}$	$N(40, 5)$	$x_2^{(2)}$	$N(21, 7)$
$x_3^{(1)}$	$N(10, 0.75)$	$x_3^{(2)}$	$N(5, 3)$
$x_4^{(1)}$	$N(0, 1)$	$x_4^{(2)}$	$N(2, 1)$
$x_5^{(1)}$	$N(10, 2)$	$x_5^{(2)}$	$N(5, 1)$
$x_6^{(1)}$	$N(25, 6)$	$x_6^{(2)}$	$N(50, 6)$
$x_7^{(1)}$	$N(20, 2)$	$x_7^{(2)}$	$N(11, 0.75)$
$x_8^{(1)}$	$N(10, 0.5)$	$x_8^{(2)}$	$N(8, 0.5)$

Finally, $\epsilon_{ij} \sim N(0, \sigma^2)$ and t_j is a sequence from 0 to 15 increasing by 1 which represents time.

$$x_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})t_j + \epsilon_{ij} \quad (4.4)$$

The values used in Equation (4.4) to generate the two longitudinal predictors, x_9 and x_{10} , are shown in Table 4.2. Note that the superscript on each of the parameter values corresponds to the cluster of patients for which that parameter value was applied.

Using the time-varying and baseline predictors described above, we simulated a time-varying outcome, y_{ij} , using the model in Equation (4.5), where x_{10ij} is the value of the second time-varying predictor, whose details are shown in Table 4.2, for a patient i at a time point j . Further, x_{3i} and x_{7i} are baseline predictors, which are described in Table 4.1, for a patient i , and $g(t_j)$ is a function of the time predictor. The parameter values of the predictors associated with the outcome were chosen to further emphasize the difference between the outcome trajectories of the two clusters of patients. The remainder of the parameter values were chosen arbitrarily.

$$y_{ij} = (\beta_0 + b_{0i}) + \beta_1 x_{3i} + \beta_2 x_{7i} + \beta_3 x_{10ij} + (\beta_4 + b_{4i})g(t_j) + \epsilon_{ij} \quad (4.5)$$

We let the function $g(t_j)$ be either a linear function of time, where $g(t_j) = \log(3)t_j$, or a curved function of time, where $g(t_j) = \sin(t_j)$. Depending on the definition of the function $g(t_j)$, we defined some of the parameter values differently, the specifications of which are

Table 4.2: Parameter values used to generate the two longitudinal predictors in our simulation study. Each table corresponds to a different longitudinal predictor, the table on the left corresponding to x_9 and the table on the right corresponding to x_{10} . Within each table, the superscript corresponds to the cluster of patients for which that parameter was used.

Parameter	Value	Parameter	Value
$\beta_0^{(1)}$	1.5	$\beta_0^{(1)}$	1.5
$\beta_1^{(1)}$	0.3	$\beta_1^{(1)}$	1.15
$\tau_0^{(1)}$	0.3	$\tau_0^{(1)}$	0.55
$\tau_1^{(1)}$	0.1	$\tau_1^{(1)}$	0.2
$\tau_{01}^{(1)}$	0.4	$\tau_{01}^{(1)}$	0.2
$\sigma^{(1)}$	0.25	$\sigma^{(1)}$	0.25
$\beta_0^{(2)}$	-2.75	$\beta_0^{(2)}$	1.5
$\beta_1^{(2)}$	-0.75	$\beta_1^{(2)}$	-3.75
$\tau_0^{(2)}$	0.6	$\tau_0^{(2)}$	1.2
$\tau_1^{(2)}$	0.25	$\tau_1^{(2)}$	0.45
$\tau_{01}^{(2)}$	0.15	$\tau_{01}^{(2)}$	0.25
$\sigma^{(2)}$	0.25	$\sigma^{(2)}$	0.25

shown in Table 4.3, where the table on the left corresponds to the linear function and the table on the right corresponds to the curved function.

Figure 4.1 shows the trajectories of the longitudinal outcome for each individual based on the generation mechanisms described above. The final result of this data generation is two datasets, each consisting of two distinct patient clusters, where the datasets differed by outcome model, specifically, in the relationship between time and the outcome. Note, again, that we specifically simulated the data to reproduce a situation where there are two clusters of patients with varying trajectories. These trajectories are a function of the levels of the predictors, where these predictors are also used for the purposes of similarity calculations. We generated the data in this way so that, in theory, and based on what we saw in Chapters 2 and 3, the predicted trajectories of an index patient will be more connected to the subpopulation when M is sufficiently small (but not too small).

We fit an additive mixed model to the data with smoothing done via penalized splines. We specified the number of knots to be $K_T = 4$, as per the recommendation by Coull et al. [2001], who state that a reasonable allocation rule is one knot per 4-5 observations. Since

Table 4.3: Parameter values used to generate the longitudinal outcome in our simulation study. The table on the left were the values when the function $g(t_j)$ is linear, and the one on the right corresponds to the curved function $g(t_j)$.

Parameter	Value	Parameter	Value
β_0	1	β_0	-1.75
β_1	0.5	β_1	0.75
β_2	1.25	β_2	-0.5
β_3	1.15	β_3	3.5
β_4	0.5	β_4	3
τ_0	0.25	τ_0	0.25
τ_1	0.25	τ_1	0.25
τ_{01}	0.15	τ_{01}	0.15
σ	0.25	σ	0.25

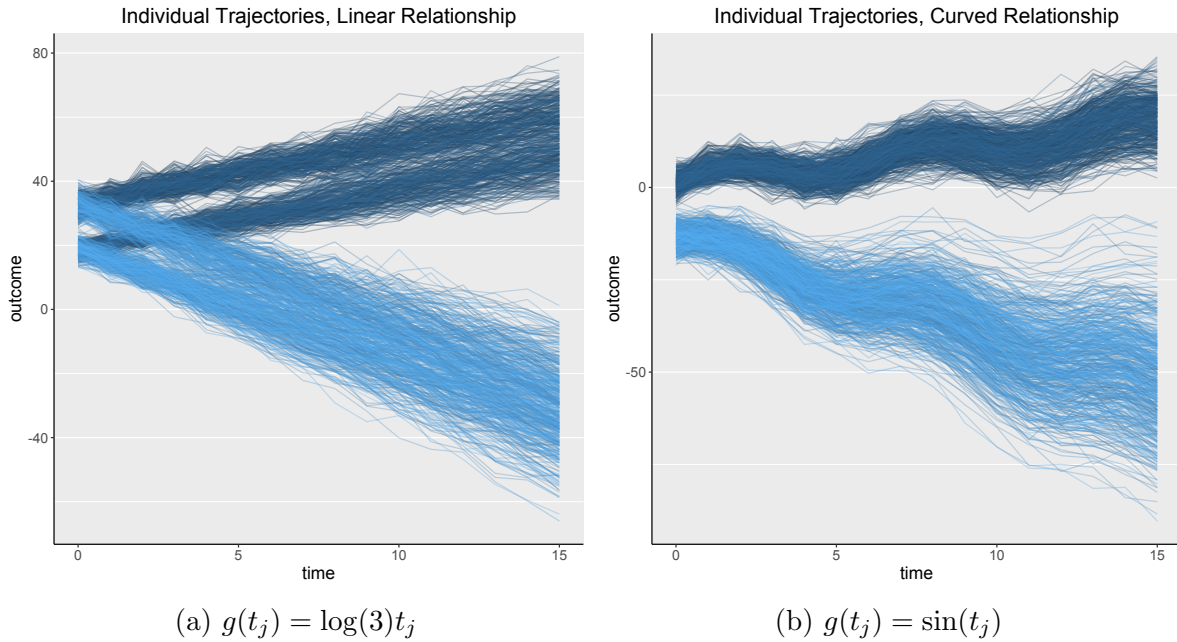


Figure 4.1: Individual trajectories under two different time-outcome relationships in Simulation 4.1.

we have 16 time points, we defined one knot per 4 observations. As discussed in Section 4.2.1, the smoothing parameters of the penalized spline are treated as random effects,

which results in a model equivalent to a linear mixed effects model. We use linear mixed effects methods via the `gamm4` function in the *gamm4* package [Wood et al., 2017] to fit the generalized additive mixed effects model. This package uses the `lmer` function in the *lme4* package [Bates et al., 2014] to fit the linear mixed effects model, and is both faster and more robust than the `gamm` function in Wood [2018]’s *mgcv* package, although does have some limitations compared to the latter package in regards to multi-penalty smoothing, but those limitations do not have an effect on this simulation study.

To determine the similarity between patients, we used the cosine similarity metric, as was done in Chapters 2 and 3.

We investigated the relationship between the mean absolute error and the size of sub-population, M . When tuning M , we considered eight possible values in the grid which was used for tuning, and we list them in terms of the $M_{proportion}$: 10%, 25%, 40%, 50%, 60%, 75%, 90%, 100%. We do not put any bounds on the grid of values as the recommendations we gave in Chapter 3 were based on the prediction of a binary outcome for an index patient and may not be transferable to this case. The optimal M is the one which minimizes the mean absolute error, and the results of this investigation are shown in Figure 4.2.

In both cases, we see the same pattern in relationship between M and the MAE: the performance improves at a high rate up until approximately $M_{proportion} = 0.6$, where the change in MAE between different values of M becomes very small. However, in both cases, the worst model is the one fit on 10% of the training data, and the optimal model is the one fit on the entire training dataset. These results are contradictory to what was seen in Chapters 2 and 3, and counter-intuitive since we deliberately simulated the data such that 50% of patients have a relatively flat trajectory and the other 50% of patients do not. These results lead to questions as to whether this is due to the fact that we are predicting the trajectory of an index patient, or due to the fact that the outcomes for each patient are continuous. We explore these questions in the upcoming simulation studies.

4.3.2 Simulation 4.2: Predicting a Continuous, Longitudinal Outcome: A Simple Case

To check the reasonableness of the somewhat counter-intuitive results in Simulation 4.1, we conduct a simple longitudinal simulation study where the data consists of only one predictor and we are fitting a linear mixed effects model, which is not as complex as the model fit in the previous case.

We simulated $N = 1000$ individuals, and, similar to Simulation 4.1, we simulated the data such that there were two clusters of patients that differed in the distribution of their

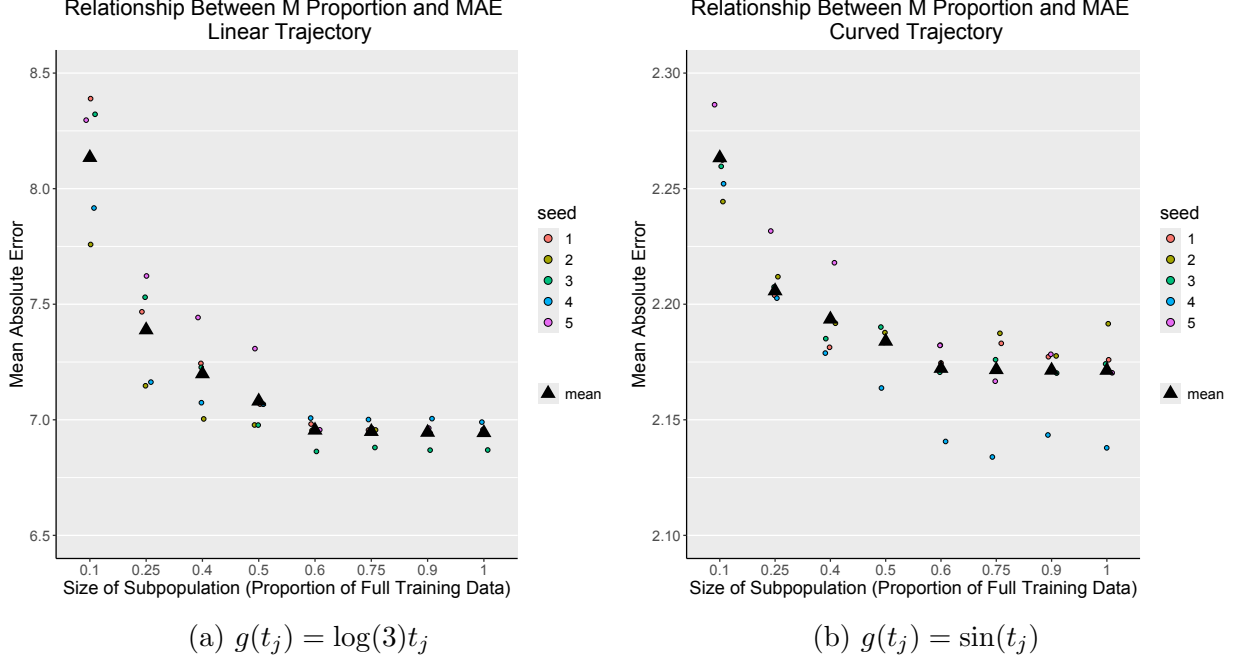


Figure 4.2: The mean absolute error of models fit to subpopulations of varying sizes under five different seeds in Simulation 4.1. The mean across the seeds at each size of subpopulation is represented by the black triangle.

predictors, each cluster consisting of 500 individuals. However, in this case, we generated only one binary predictor, x , so 500 individuals had a corresponding value of $x = 0$, while the rest had a value of $x = 1$.

Using this predictor, we simulated a time-varying outcome using the model in Equation (4.6), where t_j is a sequence from 0 to 4, increasing by 1, which represents time. This set up connects the predictor, which will be used in similarity generation, with the creation of the response. Note that, depending on the value of a patient's predictor, some of the coefficients in the outcome model differ. The corresponding parameter values are shown in Table 4.4, where the superscript indicates the value of the predictor, x .

$$y_{ij} = \begin{cases} (\beta_0^{(0)} + b_{01}) + \beta_1^{(0)}x_i + (\beta_2^{(0)} + b_{2i})t_j + \epsilon_{ij}, & \text{if } x_i = 0 \\ (\beta_0^{(1)} + b_{01}) + \beta_1^{(1)}x_i + (\beta_2^{(1)} + b_{2i})t_j + \epsilon_{ij}, & \text{if } x_i = 1 \end{cases} \quad (4.6)$$

The resulting dataset, which consisted of the predictor, x , a column indicating the time, and the continuous, longitudinal outcome, comprised of two distinct clusters of patients

Table 4.4: Parameter values used to generate the longitudinal outcome in Simulation 4.2.

Parameter	Value
$\beta_0^{(0)}$	1
$\beta_1^{(0)}$	0
$\beta_2^{(0)}$	1
$\beta_0^{(1)}$	1
$\beta_1^{(1)}$	3
$\beta_2^{(1)}$	5
τ_0	0.35
τ_1	0.55
τ_{01}	0.15
σ	0.2

which varied in their trajectories, as shown in Figure 4.3.

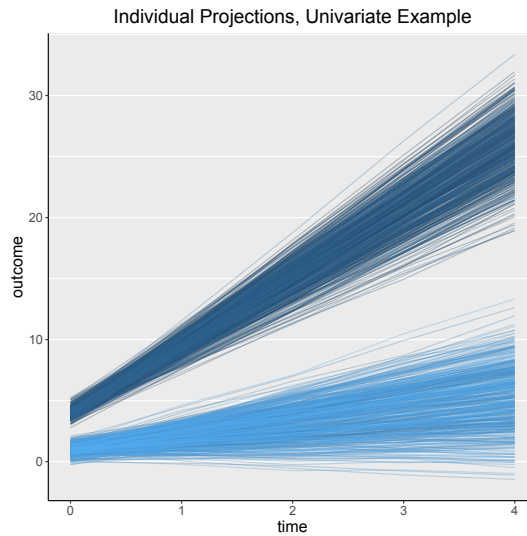


Figure 4.3: Individual trajectories of the outcome for Simulation 4.2.

Since there was only a single predictor, we did not need to use cosine similarity as a metric to determine similarity between patients. We simply employed a matching technique, where if the index patient had a value of $x = 0$, then M randomly selected patients in the training dataset with $x = 0$ were chosen to be included in the similar subpopulation. If M was greater than the number of individuals with $x = 0$, then all individuals

in the training dataset with $x = 0$ were chosen to be members of the subpopulation, and $(M - n_{train})$ randomly selected patients with $x = 1$ from the training dataset were also chosen to be part of the subpopulation. The same process was done in the case where the index patient had a value of $x = 1$.

We considered more values of $M_{proportion}$ as was considered in Simulation 4.1, and defined $M_{proportion} = \{0.1, 0.25, 0.4, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0\}$. We measured performance using the MAE, and the results of fitting a linear mixed effects model to various sizes of subpopulation are shown in Figure 4.4.

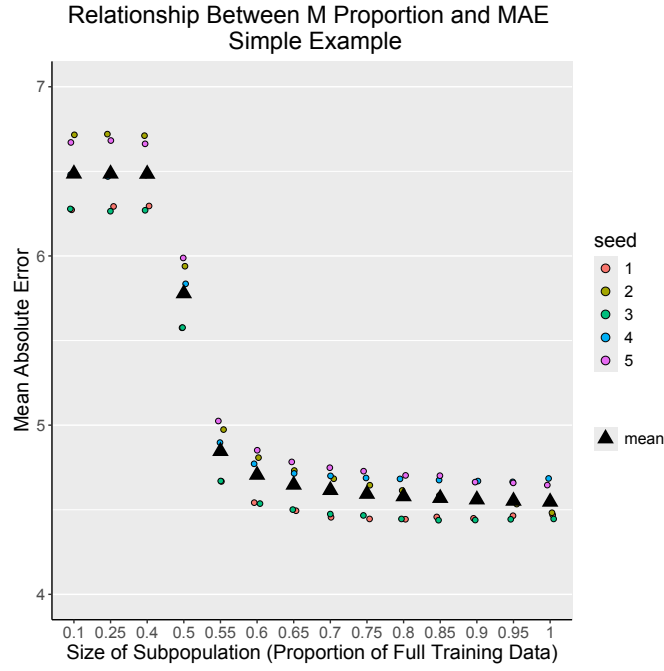


Figure 4.4: The mean absolute error of models fit to subpopulations of varying sizes under five different seeds in Simulation 4.2. The mean across the seeds at each size of subpopulation is represented by the black triangle.

We see a similar pattern as to what was seen in Figure 4.2 under Simulation 4.1: the performance of the longitudinal PPM improves drastically up to a point, here it is approximately $M_{proportion} = 0.7$, where it then begins to stabilize, ultimately resulting in an optimal model that is fit on the entire training dataset. The differences between models fit on subpopulation sizes between $M_{proportion} = 0.7$ and the full dataset are more apparent than they were in the previous, complex study (Simulation 4.1), however they are still small differences, changes being in the tenths or, in many cases, hundredths. Since in both

Simulation 4.1 and 4.2 we obtained similar results, the next question is whether this is due to the time-varying nature of the outcome, or due to the continuous nature of the outcome.

4.3.3 Simulation 4.3: Predicting a Continuous Outcome: A Non-Time-Varying Scenario

We ran a simulation study where we aimed to predict a single continuous (non-longitudinal) outcome to test whether the counter-intuitive results (compared to our speculation and what has been observed in Chapters 2 and 3 of this thesis for predicting binary outcomes) obtained in Simulation 4.1 and 4.2 are due to the fact that the outcome is continuous. We ran an algorithm similar to the one presented in Chapter 3, but fit a linear model and used the MAE as the loss function to tune M and measure the performance of that model.

To simulate the data, we again forced a dataset consisting of two clusters of patients, and in this case, the two clusters had different normal distributional means. We simulated $N = 4000$ individuals, where $n_1 = 2000$ belonged to the first cluster and $n_2 = 2000$ belonged to the second cluster. We generated four predictors: two binary and two continuous. In the first cluster of patients, $x_1^{(1)} \sim \text{Bernoulli}(0.5)$, $x_2^{(1)} \sim \text{Bernoulli}(0.65)$, $x_3^{(1)} \sim N(3, 1)$, and $x_4^{(1)} \sim N(3, 0.5)$, where the superscript on each predictor indicates that the predictor belongs to the first cluster. In the second cluster of patients, $x_1^{(2)} \sim \text{Bernoulli}(0.1)$, $x_2^{(2)} \sim \text{Bernoulli}(0.2)$, $x_3^{(2)} \sim N(0, 1)$, and $x_4^{(2)} \sim N(2, 1)$.

We simulated a linear outcome, y_i , as shown in Equation (4.7). Note here that we do not include the subscript j as was done in the previous two simulation studies, as we are not simulating a longitudinal outcome. Further, ϵ_i is the within-subject error for an individual, i .

$$y_i = 1 + 6.5x_{2i} + 3.75x_{3i} + \epsilon_i \quad (4.7)$$

The resulting distribution of responses is displayed in Figure 4.5, and there is clear separation between the two clusters of patients, with one cluster having a much higher normal distributional mean than the other.

We measured similarity using cosine similarity, and used the same grid of values of M proportions as was used in Simulation 4.2:

$M_{proportion} = \{0.1, 0.25, 0.4, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0\}$. The resulting relationship between M and the performance measure, MAE, is shown in Figure 4.6. As the differences between the models fit using the varying sizes of subpopulation are small,

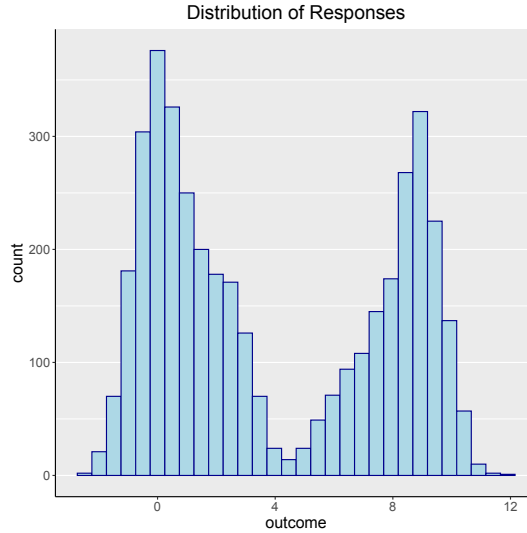


Figure 4.5: Distribution of responses for a single continuous outcome in Simulation 4.3.

it was difficult to see any patterns when we displayed all seeds considered on one plot as was done in previous sections, thus we show the results from four resulting seeds in a grid.

In all seeds considered, though in some cases it is more obvious than others, a model fit on a similar subpopulation performed better than one fit on the entire training dataset. Even though the differences between the models fit on subpopulations of different sizes are very small, there is a clear pattern under each seed which leads us to believe that when predicting a single outcome, as was done in Chapters 2 and 3, a personalized approach is better than a one-size-fits-all approach. This also means that in Simulation 4.1 and 4.2, we obtained the counter-intuitive results because we were predicting the trajectory of an index patient, not because the outcome was continuous. Further research is required to verify that the patterns seen in the longitudinal data simulation studies (Simulation 4.1 and 4.2) from this chapter are truly representative of what will be seen more generally when attempting to use similarity-based modeling for predicting longitudinal trajectories.

4.3.4 Simulation 4.4: Predicting the Difference of Two Time Points

For the final simulation study of this chapter, we simulated a longitudinal trajectory for each patient which consisted of two time points, then aimed to predict the difference

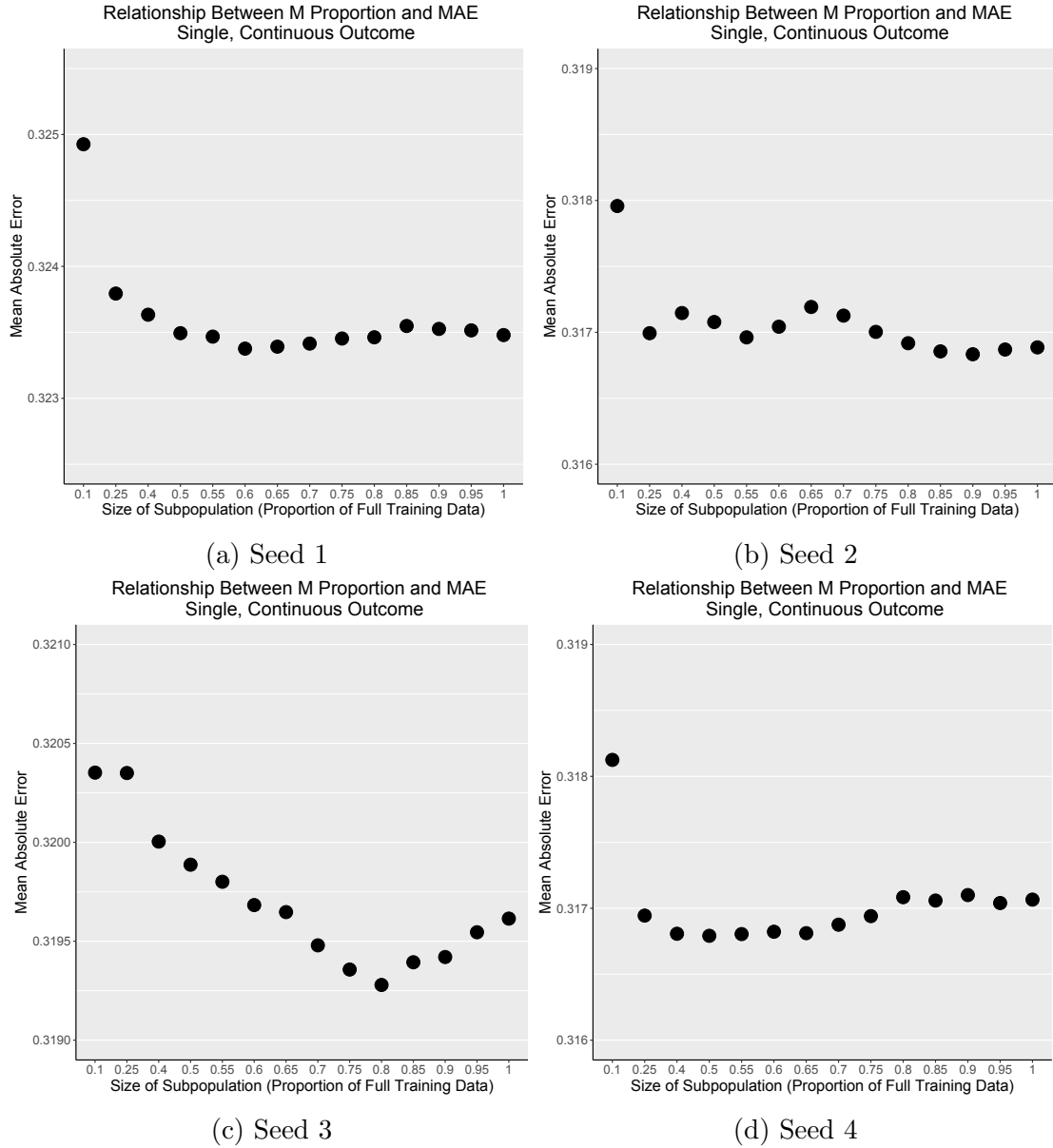


Figure 4.6: Relationship between M and the MAE under four separate seeds when fitting a PPM in Simulation 4.3.

between these two time points. Thus, we still aimed to predict a single, continuous outcome, however, this outcome was now related to the longitudinal trajectory of the index patient.

We ran the same algorithm that was used in Simulation 4.3, noting that we are predicting a single, continuous outcome, thus we fit a linear model and use the MAE to tune M .

We simulated $N = 2000$ individuals, and again, simulated the data such that there were two clusters of patients that differed, generally, in the distribution of their predictors, each cluster consisting of 1000 individuals. We simulated eight continuous predictors, and the distributions of these predictors are the same as was used in Simulation 4.1, as displayed in Table 4.1.

Using these predictors, we simulated a time-varying outcome using the model in Equation (4.8), where $t_j = \{0, 1\}$ and represents the two time points for each patient. The corresponding parameter values are shown in Table 4.4.

$$y_{ij} = (\beta_0 + b_{0i}) + \beta_1 x_{2i} + \beta_2 x_{3i} + \beta_3 x_{7i} + (\beta_4 + b_{4i}) \ln(3)t_j + \epsilon_{ij} \quad (4.8)$$

Table 4.5: Parameter values used to generate the longitudinal outcome in Simulation 4.4.

Parameter	Value
β_0	1
β_1	1.15
β_2	0.5
β_3	1.25
β_4	0.5
τ_0	0.25
τ_1	0.25
τ_{01}	0.15
σ	0.25

We chose $M_{proportion}$: 10%, 25%, 40%, 50%, 60%, 75%, 90%, 100%, similar to what was done in Simulation 4.1, and the resulting MAE performance of the PPMs under these sizes of subpopulation for one seed is shown in Figure 4.7.

Similar to the results in Simulations 4.1 and 4.2, as the size of subpopulation increases, the performance in terms of MAE improves drastically up to a point, in this case that point is $M_{proportion} = 0.75$, where it then stabilizes but ultimately results in the one-size-fits-all model as the optimal model.

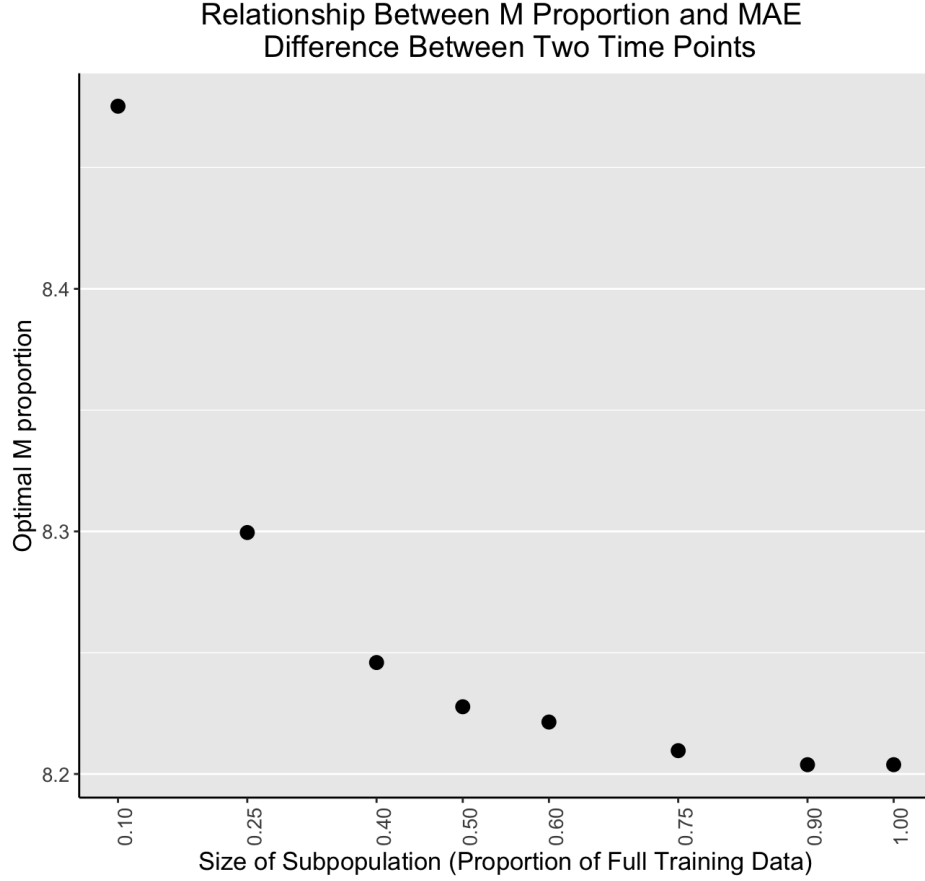


Figure 4.7: Relationship between M and the MAE when fitting a PPM in Simulation 4.4.

4.4 Discussion

In this chapter, we considered a dataset with a time-varying, continuous outcome and proposed an algorithm to fit a personalized predictive model (PPM) to the data with the intention of predicting the trajectory of an individual in the case where there are two distinct trajectories an individual could have. After conducting four separate simulation studies, we conclude that models fit on a similar subpopulation to predict the trajectory of an index patient do not perform better than a one-size-fits-all model.

Our proposed algorithm consisted of a training step, which tuned the size of a subpopulation to determine which proportion of the entire training data will result in the lowest overall performance measure, in our case this measure was the mean absolute error (MAE).

In Simulation 4.1, we considered a case where the relationship between the time predictor and the outcome was linear, and another where it was curved. For each type of dataset, we considered five randomly generated seeds, and showed that in all five cases, the MAE improved up to an M proportion of 0.6, where it then slightly improved leading to the full dataset being the optimal performing model. The difference between the MAE of models fit on subpopulations of sizes $M_{proportion} = 0.6$ to the full dataset are very small, some differences being in the thousandths. We chose the MAE as the overall performance measure as it is less sensitive to outliers compared to other measures, such as the mean squared prediction error (MSPE). However, although not explicitly shown, we also used the MSPE as the performance measure on which to tune M , as it is closer to the Brier Score, which was used in Chapters 2 and 3, compared to the MAE, and using this measure led to the same results as seen in Figure 4.2. It might be of interest to test whether other measures of performance, such as the root mean squared prediction error (RMSPE) or measures of calibration such as those presented in Chapters 2 and 3, have differing relationships with M from what we have seen under the MAE and MSPE when predicting an outcome trajectory. As discrimination is defined as a model’s ability to distinguish between outcome classes, measures of discrimination are not relevant when the outcome is continuous.

We did not know whether these results were obtained because of our data generating mechanisms, so we conducted another simulation study, Simulation 4.2, where, again, we predicted a longitudinal, continuous outcome, but our dataset was much simpler, consisting of a single, binary predictor. We fit a linear mixed effects model, and measured performance using the MAE. Again, we found that the optimal model was the model which was fit to the entire training dataset. The same pattern was seen, where the model performance improved up to a point, then the MAE remained relatively stable.

In the third simulation study, Simulation 4.3, we tested whether or not the previous results were stemming from predicting a trajectory, or from predicting a continuous outcome. Thus, we fit PPMs under various values of M to predict a single, continuous outcome, and we found that a personalized approach does lead to better prediction, unlike what was seen in the prior simulation studies for the longitudinal responses, ie. Simulations 4.1 and 4.2.

In the final simulation study, Simulation 4.4, we aimed to predict a single, continuous response once again, however, this response was now a summary statistic of a longitudinal trajectory. Specifically, we simulated a trajectory for each index patient which consisted of two time points, and aimed to predict the difference in these two time points for an index patient. Again, we simulated the data so that there were clearly two subgroups of patients with different outcome trajectories, and found, similar to the results in Simulations 4.1 and 4.2, that a one-size-fits-all model has better predictive performance than a personalized predictive model fit on a similar subpopulation. However, this performance is only a slight

improvement compared to using an M of 75% or 90% of the entire training data. We have shown that though a personalized approach may perform better than a one-size-fits-all model when predicting a continuous response, if the continuous response is related to a longitudinal trajectory, this may not be the case. This simulation can be further explored without the use of random effects in the simulation data generation process to see if the inclusion of random effects is partly explaining these counter intuitive results.

To our knowledge, this area has not yet been explored, so the information obtained in this chapter is very valuable as it broadens our knowledge of the performance of PPMs. We can only speculate as to why a personalized approach does not result in better performance when predicting a trajectory, but since the performance improves only slightly after $M_{proportion} = 0.6$ in Simulations 4.1 and 4.2, and after $M_{proportion} = 0.9$ in Simulation 4.4, one reason may be due to the fact that although using a similar subpopulation leads to good performance, there is too much variation with the smaller sample size. Thus, we can use a similar subpopulation to fit the model, which would save computation time and result in similar performance to the full model, but it would not improve in performance compared to the full model. These results could change if we considered a dataset much larger than $N = 3000$, such as was done in Chapters 2 and 3, however, because of the computational intensity of these methods, this extension is currently impractical.

There are many extensions of this work that, due to time constraints, we did not get a chance to explore. As mentioned above, computational intensity remains, as in the algorithms proposed in Chapters 2 and 3, a challenge when applying these methods to sizable datasets. We employed the *Rcpp* package [Eddelbuettel and François, 2011], along with parallelization, but there is still room for improvement to increase the usability of this algorithm in practice. From what we have seen, a personalized approach does not lead to better prediction when predicting the trajectory of an index patient, but we only conducted two simulation studies. Applying these methods using different datasets, simulated or real, and under different models, will be important to truly determine if personalized methods should be used in a longitudinal setting. We introduced a very simple example of predicting a single, continuous outcome, but further work can be done in this area as well, especially to determine if differences in the MAE between models fit on various sizes of M are always small, as we saw in our results.

Chapter 5

Discussion

5.1 Summary

In this thesis, we aimed to answer the question, “Do personalized predictive models (PPMs) result in better predictions compared to one-size-fits-all models?”. We approached this problem through three different projects, a comprehensive set of simulation studies, and using multiple performance measures, and concluded that a personalized approach leads to better predictions when predicting a single (non-longitudinal) outcome of an index patient.

It has been criticized that model calibration is not assessed nearly as often as discrimination in prediction models [Van Calster et al., 2019], and this criticism motivated the initial algorithm proposed in Chapter 2, which fit a PPM that jointly optimized discrimination and calibration. We used the fact that the Brier Score can be decomposed into its discrimination and calibration subparts to propose a loss function which considers both discrimination and calibration, along with a mixture loss term that allows the flexibility of emphasizing one performance measure over the other, if desired. Under both discrimination and calibration as performance measures, we empirically show that using a subpopulation of patients to fit a predictive model results in relatively better, if not the best, performance. In both the simulations and data analysis, we confirmed the large data-analysis results of Lee et al. [2015] which state that the discrimination of a predictive model fit on a similar subpopulation for an index patient performs better than a one-size-fits-all model. We investigated the relationship between M and various measures of calibration, as well, and found that, although the best calibrated models are those which use the entire available training data, if one is willing to sacrifice a bit of calibration for better discrimination, then a smaller choice of M is optimal. When conducting the sensitivity analysis on the

mixture term, $\alpha \in [0, 1]$, in both the simulation study and the data analysis, the best overall performing models were those which put similar weights on both the discrimination and calibration terms in the loss function. Specifically, the best performing model in the simulation study corresponded to $\alpha = 0.58$, while in the data analysis the best performing model was seen under $\alpha = 0.51$. Further, the models corresponding to these α values were those fit on a similar subpopulation, thus supporting the use of our proposed algorithm to tune M and α to fit relatively better performing models compared to one-size-fits-all models.

Though the results we obtained in Chapter 2 were promising, there were still many limitations of the algorithm which needed to be addressed. Notably, [Van Calster et al. \[2016\]](#) recommend that prediction models strive to be at least moderately calibrated, however, we left the original Brier Score decomposition to represent the loss function, meaning we were using an inferior calibration measure, ie., mean calibration, in our loss function. Further, we only explored the relationships between M and various performance measures under one simulated dataset, so we did not know whether the results were robust to datasets with different outcome models. Finally, the algorithm was very computationally intensive, and although measures were taken to lessen this burden, through both the *Rcpp* package [\[Eddelbuettel and François, 2011\]](#), and through parallelization, the computation time remained a barrier to the usability of this algorithm in practice. The purpose of Chapter 3 was to address these limitations and improve our proposed algorithm. We proposed a new loss function, which still includes the mixture loss term, α , but now allows more flexibility by giving the user a choice of which measures to include in the function: one discrimination measure and one calibration measure. After investigating the relationships between M and six performance measures, we were also able to recommend bounds on the grid of values on which to tune M , with the intent of lessening the computational burden of the algorithm. Based on our findings, we recommend that the lower bound on this grid of values be 20% of the training data, as this ensures that the sample size in the validation step is not too small. We recommend that the upper bound is either 50% or 70% of the training data, depending on whether there is greater interest on model discrimination or calibration. When there is equal interest in discrimination and calibration, or greater interest in discrimination (ie. $\alpha \leq 0.5$), we suggest implementing an upper bound of 50% which will decrease the computation time while still likely capturing the optimal M value under any of the discrimination measures considered. When there is greater interest in calibration (ie. $\alpha > 0.5$), we suggest an upper bound that is 70% of the training data which will increase the change of capturing the optimal M value under the calibration measures if that value is large, while still ensuring that there is not too much deterioration in discriminative ability. Similar to Chapter 2, through simulation study and a data analysis, we showed that a

PPM performs better than a one-size-fits-all model, overall, and based on the performance measures chosen for the loss function, the optimal size of subpopulation of the PPM may differ.

Across all simulation studies and data analyses conducted in Chapters 2 and 3, we notice that the optimal model corresponds to an α value between 0.4 and 0.6. Values of α outside of these bounds put a much heavier emphasis on one of the two performance measures included in the loss function, and result in one performance measure performing really well at the expense of the other. Thus, we recommend when tuning the size of subpopulation, that α be tuned as well, but only considering α values between 0.4 and 0.6, unless one desires to truly emphasize one performance measure over the other. This said, an α value of 0.5 might be a reasonable default choice for many problems.

Our proposed algorithms and loss functions in Chapters 2 and 3 were based on fitting models to predict a binary outcome. In Chapter 4, we consider a longitudinal, continuous outcome, along with longitudinal predictors, to determine whether a PPM under a different outcome would still result in better predictions. In this case, we used an overall performance measure, the mean absolute error, to tune the size of subpopulation. Motivated by EHR data, which could contain a large number of very diverse patients, we considered a case where there are two very distinct clusters that patients could belong to, each with a specific trajectory of outcome. Although we exaggerated the scenario, the goal was to demonstrate that when these circumstances do exist, a PPM might lead to better performance. In Simulation 4.1, however, where we fit an additive mixed model to similar subpopulations of patients to predict the trajectory of an index patient, the model fit to the entire training dataset performed the best. More specifically, the performance improved drastically up to an M proportion of 60%, where it then stabilized but still resulted in the full model being the optimal model. This counter-intuitive result was further tested in a simpler scenario, Simulation 4.2, where a dataset with only a single, binary predictor was generated, and we fit a linear mixed effects model to predict the outcome trajectory of an index patient. We obtained identical results, which lead to the final question: was this a characteristic of a longitudinal outcome, or a continuous outcome? In the third simulation study, Simulation 4.3, we generated a relatively simple dataset with a single, continuous (non-longitudinal) outcome and found that in this case, a model fit on a similar subpopulation led to better performance. In the final simulation study of this chapter, Simulation 4.4, we aimed to predict the difference in the two-time-point longitudinal trajectory of an index patient, and, similar to Simulations 4.1 and 4.2, found that a one-size-fits-all model performance is better than a PPM. Thus, we concluded, that PPMs, in the longitudinal response cases we considered, do not perform better than one-size-fits-all models when the goal is to predict the trajectory of an index patient. This is even the case when we are predicting a continuous

response that is a summary statistic of a longitudinal trajectory. This could be happening for many reasons, but we currently have two theories. The first is that the inclusion of random effects is partly responsible, and we can explore this theory by generating the data without these random effects and running the simulations again. The second theory is that we are obtaining these counter-intuitive results due to the relatively small datasets we are considering, compared to those we considered in Chapters 2 and 3. Since the performance improves only slightly when subpopulations larger than 60% are used to train the model, we wonder whether using a similar subpopulation does lead to better performance, but in this study, we are losing too much power by considering a subpopulation rather than the full dataset. Due to computational complexity, we are unable to test larger sample sizes for now, but this is discussed further in the next section.

We note that in the simulation studies throughout this thesis, we replicated a scenario where there are clearly separate clusters of patients within a larger dataset. In a case where all patients have the same distributional mean, or the same patient trajectory, then a personalized approach to modelling is not preferred over a one-size-fits-all model, since using a larger dataset with exchangeable patients will result in more power. That being said, we show that in cases where there are clusters of patients, at least when predicting a single outcome, PPMs outperform one-size-fits-all models.

In all three projects, we investigated relationships not previously studied, and proposed methods motivated by Lee [2017] which aim to fit PPMs using an optimal size of subpopulation. We introduce a novel loss function for use when fitting PPMs to predict a binary outcome, and allow flexibility in not only the choice in performance measures to include, but also in the weighting of each performance measure. Throughout this thesis, we provide recommendations on a variety of subjects, including which performance measures to use in the loss function, the bounds on the grid of values on which M is tuned, and which α values may result in the best performing model. We made efforts to make these models more accessible in practice through parallelization and through using the *Rcpp* package. This thesis contributes to the ongoing conversation in precision medicine to fit better performing predictive models.

5.2 Future Work

There are many potential extensions of this work, the primary focus being improving the computational efficiency of these algorithms. One way of helping the computational burden could be considering a binary M search, as opposed to a random grid search, when tuning M . An additional strategy is assessing the similarity between the index patients

and all other patients outside of the main loop, and indexing these similarity values to refer back when needed, specifically, when defining the similar subpopulation. Once the computational efficiency is improved, even slightly, we can explore many different avenues to gain a larger perspective on the performance of PPMs as a whole. For instance, in Chapter 2 and 3, we were only able to fit logistic regression models because more sophisticated machine learning methods would increase the computation time significantly. If this burden is lessened, we can explore the relationships between M and various performance measures of models such as random forest, or gradient boosting. Further, we can add a variable selection component to these models, which is especially important when working with EHR data which can contain a large number of predictors. Further, we can further investigate fitting PPMs to predict the trajectory of patients using larger sample sizes.

Right now, for a given health condition, the optimal M value is the same for each new index patient. An area to explore is whether this optimal M should be different for different patients. Once again, the computational efficiency of the algorithm must be improved before tackling this extension, however, having a unique optimal size of subpopulation for different patients could potentially further improve predictive performance of the personalized predictive models. Further, as these models do take a considerable time to run, it would be worth investigating the cost effectiveness of tuning the size of subpopulation prior to implementation in, say, health settings.

Throughout this thesis, we put a great emphasis on neighbourhood-based algorithms using cosine similarity to define similarity. There are many other ways of defining a similar subpopulation, and a comparison of various methods should be done to determine if certain methods result in better performance under specific scenarios. This could include using a different similarity metric, such as the Generalized Mahalanobis distance, instead of cosine similarity, or changing the type of algorithm completely to, for example, a clustering algorithm to help define the subpopulation.

We explored three types of outcomes: binary, continuous, and longitudinal, continuous. In the latter, we showed that a personalized approach does not result in better performing models in the simulations we ran, but this area does need to be explored more thoroughly. We are not certain why this is the case, though we do speculate that it has to do with the small sample size. In future work, we wish to explore this area more thoroughly, as well as look at other outcomes such as ordinal and time-to-event, to investigate PPM performance under different settings.

To allow for greater usability of these methods, we wish to make recommendations for an optimal M proportion to use in practice under different settings. Since these algorithms are computationally intensive, recommending a rule-of-thumb proportion would make these

methods more accessible in, for instance, health care settings. We also would like to streamline the algorithm so, instead of three separate algorithms which we have presented in Chapters [2](#), [3](#) and [4](#), we can propose one meta-algorithm with three variations. Further, our next step is to tie all methods together into an R package to be used by researchers, an outline of which is provided in Appendix [B](#).

References

- Melissa Assel, Daniel D Sjöberg, and Andrew J Vickers. The brier score does not evaluate the clinical utility of diagnostic tests or prediction models. *Diagnostic and prognostic research*, 1(1):1–7, 2017.
- Peter C Austin and Ewout W Steyerberg. The integrated calibration index (ici) and related metrics for quantifying the calibration of logistic regression models. *Statistics in medicine*, 38(21):4051–4065, 2019.
- Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. *arXiv preprint arXiv:1406.5823*, 2014.
- Nikhil Bhagwat, Joseph D Viviano, Aristotle N Voineskos, M Mallar Chakravarty, Alzheimer’s Disease Neuroimaging Initiative, et al. Modeling and prediction of clinical symptom trajectories in alzheimer’s disease using longitudinal data. *PLoS computational biology*, 14(9):e1006376, 2018.
- Babette A Brumback, David Ruppert, and Matthew P Wand. Variable selection and function estimation in additive nonparametric regression using a data-based prior: Comment. *Journal of the American Statistical Association*, 94(447):794–797, 1999.
- Boris Campillo-Gimenez, Wassim Jouini, Sahar Bayat, and Marc Cuggia. Improving case-based reasoning systems by combining k-nearest neighbour algorithm with logistic regression in the prediction of patients’ registration on the renal transplant waiting list. *PLoS One*, 8(9):e71991, 2013.
- Cody Carroll, Alvaro Gajardo, Yaqing Chen, Xiongtao Dai, Jianing Fan, Pantelis Z Hadjipantelis, K Han, H Ji, HG Mueller, and Jane-Ling Wang. fdapace: Functional data analysis and empirical dynamics. *R package version 0.5*, 4, 2020.

- Leo Anthony Celi, Sean Galvin, Guido Davidzon, Joon Lee, Daniel Scott, and Roger Mark. A database-driven decision support system: customized mortality prediction. *Journal of personalized medicine*, 2(4):138–148, 2012.
- William S Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association*, 74(368):829–836, 1979.
- William S Cleveland and Susan J Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American statistical association*, 83(403):596–610, 1988.
- Brent A Coull, Joel Schwartz, and MP Wand. Respiratory health and air pollution: additive mixed model analyses. *Biostatistics*, 2(3):337–349, 2001.
- Martin R Cowie, Juuso I Blomster, Lesley H Curtis, Sylvie Duclaux, Ian Ford, Fleur Fritz, Samantha Goldman, Salim Janmohamed, Jörg Kreuzer, Mark Leenay, et al. Electronic health records to facilitate clinical research. *Clinical Research in Cardiology*, 106:1–9, 2017.
- David R Cox. Two further applications of a model for binary regression. *Biometrika*, 45(3/4):562–565, 1958.
- Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.
- Edward De Brouwer, Thijs Becker, Yves Moreau, Eva Kubala Havrdova, Maria Trojano, Sara Eichau, Serkan Ozakbas, Marco Onofri, Pierre Grammond, Jens Kuhle, et al. Longitudinal machine learning modeling of ms patient trajectories improves predictions of disability progression. *Computer methods and programs in biomedicine*, 208:106180, 2021.
- Wei Du and Xintao Wu. Enhancing personalized modeling via weighted and adversarial learning. *International Journal of Data Science and Analytics*, 12(1):1–14, 2021.
- Sahibsingh A Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (4):325–327, 1976.
- Dirk Eddelbuettel and Romain François. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011. doi: 10.18637/jss.v040.i08.

- Iris Eekhout, Stef van Buuren, Bram Visser, Marco CAM Bink, and Abe Huisman. Longitudinal individual predictions from irregular repeated measurements data. *Scientific reports*, 13(1):952, 2023.
- LiYing Fang, Min Wan, MingWei Yu, JianZhuo Yan, Zheng Liu, and Pu Wang. Analysis of similarity measure in the longitudinal study using improved longest common subsequence method for lung cancer. *Biomedical Signal Processing and Control*, 15:60–66, 2015.
- Garrett M Fitzmaurice, Nan M Laird, and James H Ware. *Applied longitudinal analysis*. John Wiley & Sons, 2012.
- Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.
- Trevor J Hastie. Generalized additive models. In *Statistical models in S*, pages 249–307. Routledge, 2017.
- Tommy Hielscher, Myra Spiliopoulou, Henry Völzke, and Jens-Peter Kühn. Using participant similarity for the classification of epidemiological data on hepatic steatosis. In *2014 IEEE 27th International Symposium on Computer-Based Medical Systems*, pages 1–7. IEEE, 2014.
- Xiaoqian Jiang, Aditya Menon, Shuang Wang, Jihoon Kim, and Lucila Ohno-Machado. Doubly optimized calibrated support vector machine (doc-svm): an algorithm for joint optimization of discrimination and calibration. *PloS one*, 7(11):e48823, 2012.
- Joon Lee. Patient-specific predictive modeling using random forests: an observational study for the critically ill. *JMIR medical informatics*, 5(1):e6690, 2017.
- Joon Lee, David M Maslove, and Joel A Dubin. Personalized mortality prediction driven by electronic medical data and a patient similarity metric. *PloS one*, 10(5):e0127428, 2015.
- Xihong Lin and Daowen Zhang. Inference in generalized additive mixed models by using smoothing splines. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(2):381–400, 1999.
- David M Maslove, Francois Lamontagne, John C Marshall, and Daren K Heyland. A path to precision in the icu. *Critical Care*, 21(1):1–9, 2017.

- Michael E Miller, Carl D Langefeld, William M Tierney, Siu L Hui, and Clement J McDonald. Validation of probabilistic predictions. *Medical Decision Making*, 13(1):49–57, 1993.
- Kenney Ng, Jimeng Sun, Jianying Hu, and Fei Wang. Personalized predictive modeling and risk factor identification using patient similarity. *AMIA Summits on Translational Science Proceedings*, 2015:132, 2015.
- Brice Ozenne, Fabien Subtil, and Delphine Maucort-Boulch. The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases. *Journal of clinical epidemiology*, 68(8):855–859, 2015.
- Roberto Paredes and Enrique Vidal. Learning weighted metrics to minimize nearest-neighbor classification error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1100–1110, 2006.
- Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):1–13, 2018.
- R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020. URL <https://www.R-project.org/>.
- James O Ramsay. Kernel smoothing approaches to nonparametric item characteristic curve estimation. *Psychometrika*, 56(4):611–630, 1991.
- RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, PBC., Boston, MA, 2020. URL <http://www.rstudio.com/>.
- Kaspar Rufibach. Use of brier score to assess binary predictions. *Journal of clinical epidemiology*, 63(8):938–939, 2010.
- Anis Sharafoddini, Joel A Dubin, and Joon Lee. Patient similarity in prediction models based on health data: a scoping review. *JMIR medical informatics*, 5(1):e7, 2017.
- Anis Sharafoddini, Joel A Dubin, and Joon Lee. Identifying subpopulations of septic patients: A temporal data-driven approach. *Computers in Biology and Medicine*, 130: 104182, 2021.
- Jeffrey S Spence, Monroe P Turner, Bart Rypma, Mark D’Esposito, and Sandra Bond Chapman. Toward precision brain health: accurate prediction of a cognitive index trajectory using neuroimaging metrics. *Cerebral Cortex*, 34(1):bhad435, 2024.

- David J Spiegelhalter. Probabilistic prediction in patient management and clinical trials. *Statistics in medicine*, 5(5):421–433, 1986.
- Ewout W. Steyerberg. *Clinical prediction models*. Springer International Publishing, 2019.
- Ewout W Steyerberg, Andrew J Vickers, Nancy R Cook, Thomas Gerds, Mithat Gonen, Nancy Obuchowski, Michael J Pencina, and Michael W Kattan. Assessing the performance of prediction models: a framework for some traditional and novel measures. *Epidemiology (Cambridge, Mass.)*, 21(1):128, 2010.
- Ben Van Calster, Daan Nieboer, Yvonne Vergouwe, Bavo De Cock, Michael J Pencina, and Ewout W Steyerberg. A calibration hierarchy for risk models was defined: from utopia to empirical data. *Journal of clinical epidemiology*, 74:167–176, 2016.
- Ben Van Calster, David J McLernon, Maarten Van Smeden, Laure Wynants, and Ewout W Steyerberg. Calibration: the achilles heel of predictive analytics. *BMC medicine*, 17(1): 1–7, 2019.
- Kirsten Van Hoorde, Sabine Van Huffel, Dirk Timmerman, Tom Bourne, and Ben Van Calster. A spline-based tool to assess and visualize the calibration of multiclass risk predictions. *Journal of biomedical informatics*, 54:283–293, 2015.
- Ni Wang, Yanqun Huang, Honglei Liu, Zhiqiang Zhang, Lan Wei, Xiaolu Fei, and Hui Chen. Study on the semi-supervised learning-based patient similarity from heterogeneous electronic medical records. *BMC medical informatics and decision making*, 21(2):1–13, 2021.
- Simon Wood. Mixed gam computation vehicle with automatic smoothness estimation. r package version 1.8–12, 2018.
- Simon Wood, Fabian Scheipl, and Maintainer Simon Wood. Package ‘gamm4’. *Am Stat*, 45(339):0–2, 2017.
- Simon N Wood. *Generalized additive models: an introduction with R*. chapman and hall/CRC, 2017.
- Kiyoung Yang and Cyrus Shahabi. A pca-based similarity measure for multivariate time series. In *Proceedings of the 2nd ACM international workshop on Multimedia databases*, pages 65–74, 2004.

Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional data analysis for sparse longitudinal data. *Journal of the American statistical association*, 100(470):577–590, 2005.

Wangmeng Zuo, David Zhang, and Kuanquan Wang. On kernel difference-weighted k-nearest neighbor classification. *Pattern Analysis and Applications*, 11(3):247–257, 2008.

APPENDICES

Appendix A

Additional Plots - Chapter 3

We include additional plots to the simulation study done in Chapter 3. Specifically, in Section 3.3.2, we show the relationships between M and six performance measures under two cases, but we investigated these relationships across 12 varying datasets. We include the results from the other ten cases here.

Recall that the twelve datasets considered differed in the size of their association, which could have been either 10/40 or 20/40, by the prevalence of the outcome, which could have been low (~ 0.05), moderate (~ 0.15) or balanced (~ 0.5), and by the linearity, or lack thereof, of the model. In Section 3.3.2, we displayed the results from the linear model with an association of 20 predictors and a prevalence of outcome that is moderate, as well as the non-linear model with an association of 20 predictors and a prevalence of outcome that is low.

A.1 Linear, 10 Association, Balanced Prevalence

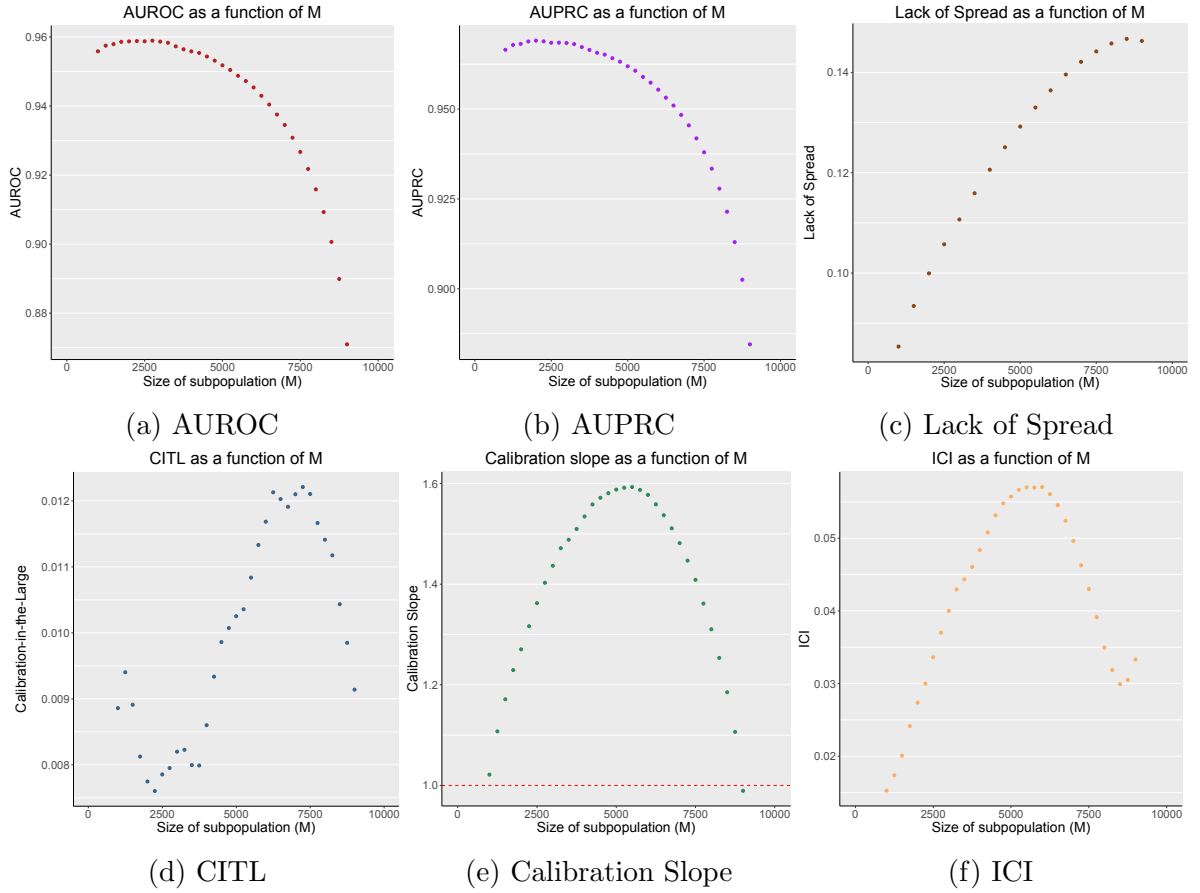


Figure A.1: Performance measures as a function of M when the outcome is simulated by a model which is linear, has 10 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.50.

A.2 Linear, 20 Association, Balanced Prevalence

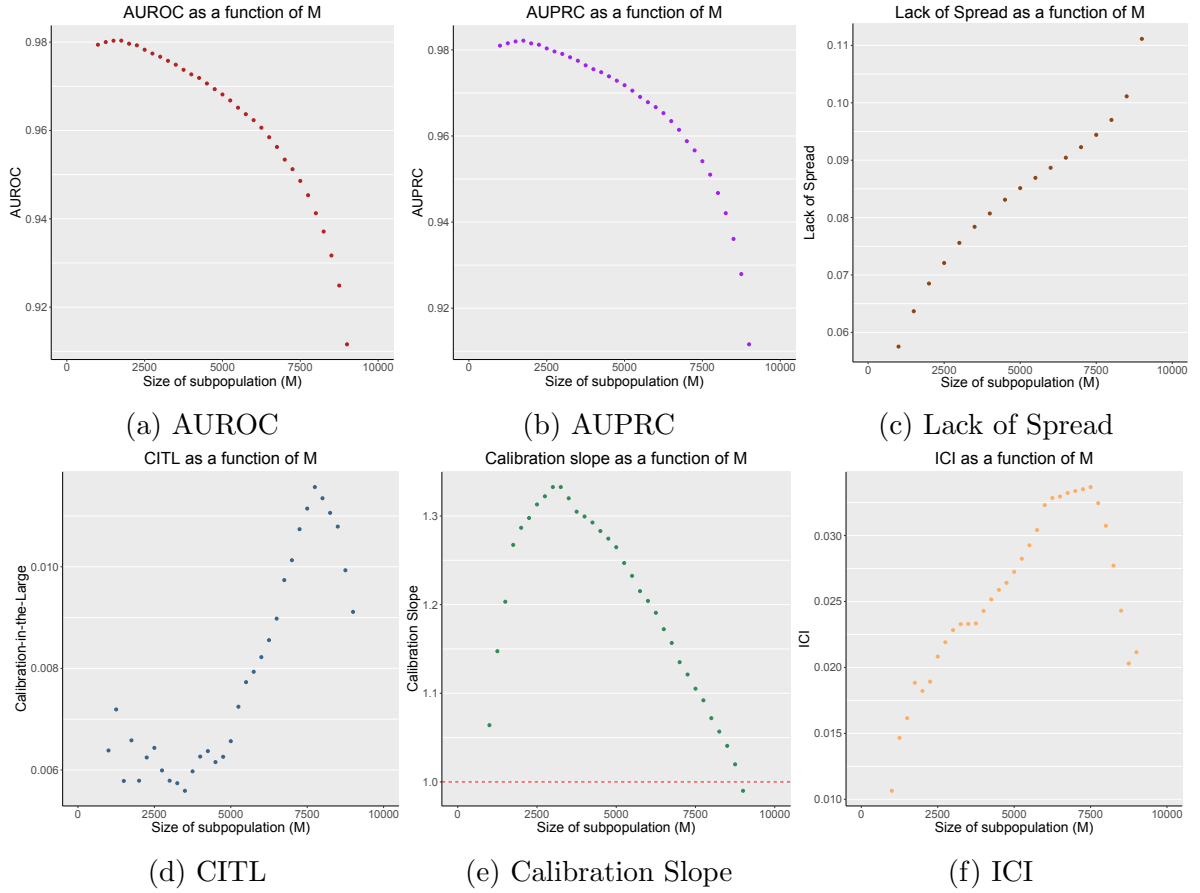


Figure A.2: Performance measures as a function of M when the outcome is simulated by a model which is linear, has 20 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.50.

A.3 Linear, 10 Association, Moderate Prevalence

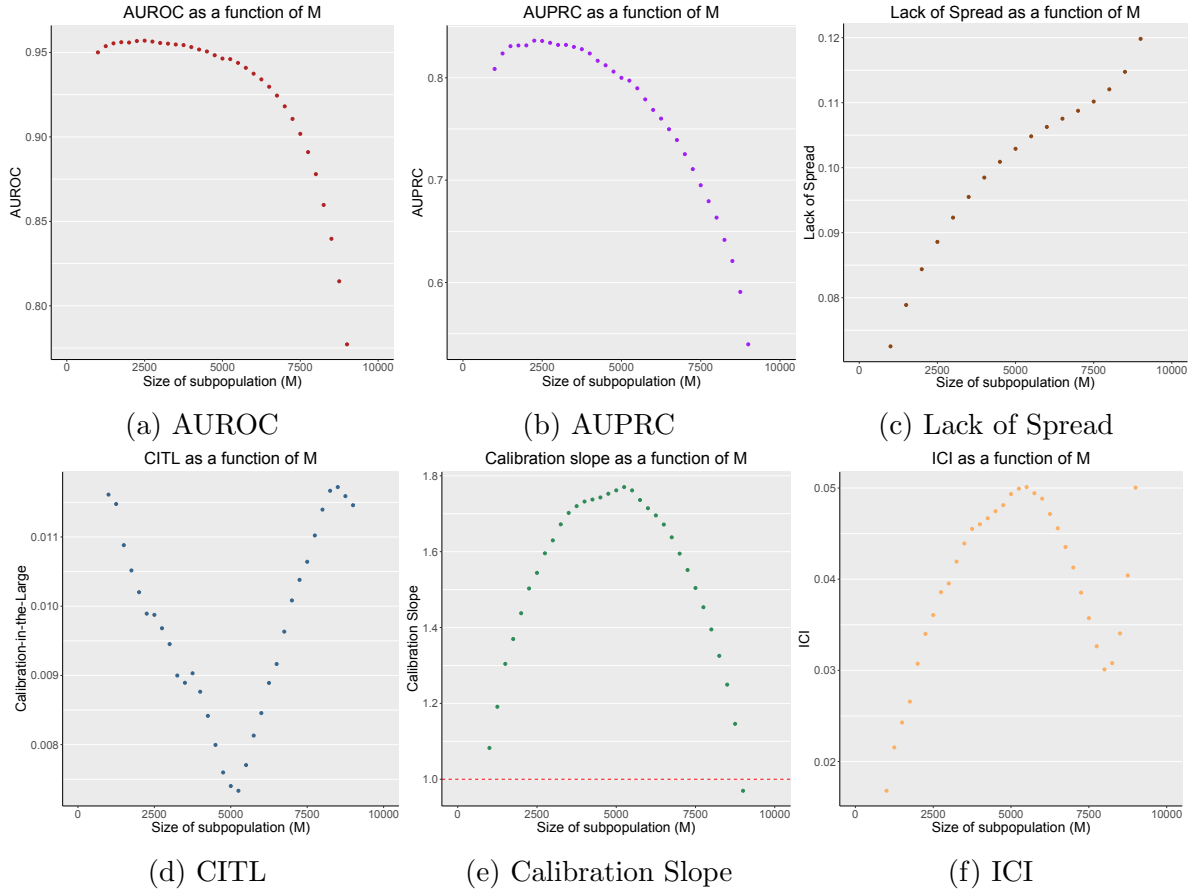


Figure A.3: Performance measures as a function of M when the outcome is simulated by a model which is linear, has 10 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.15.

A.4 Linear, 10 Association, Low Prevalence

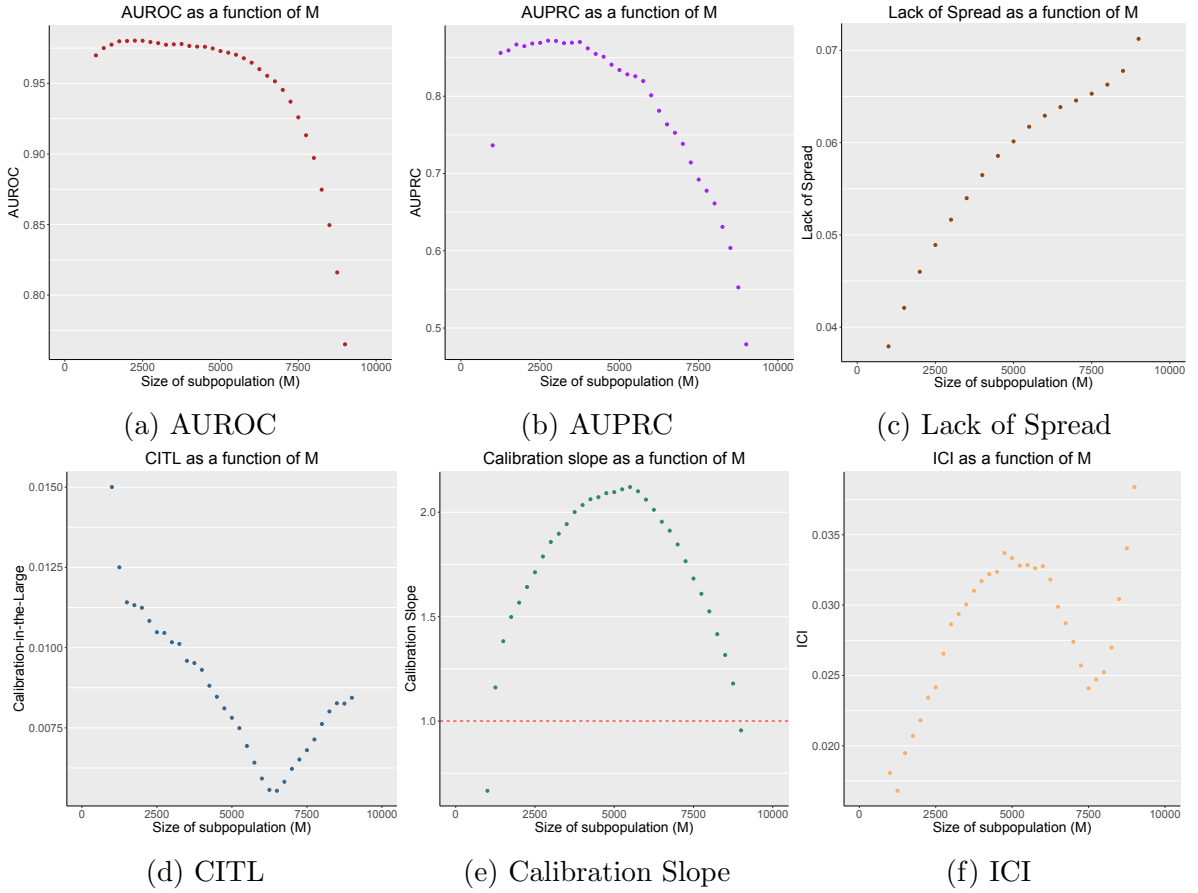


Figure A.4: Performance measures as a function of M when the outcome is simulated by a model which is linear, has 10 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.05.

A.5 Linear, 20 Association, Low Prevalence

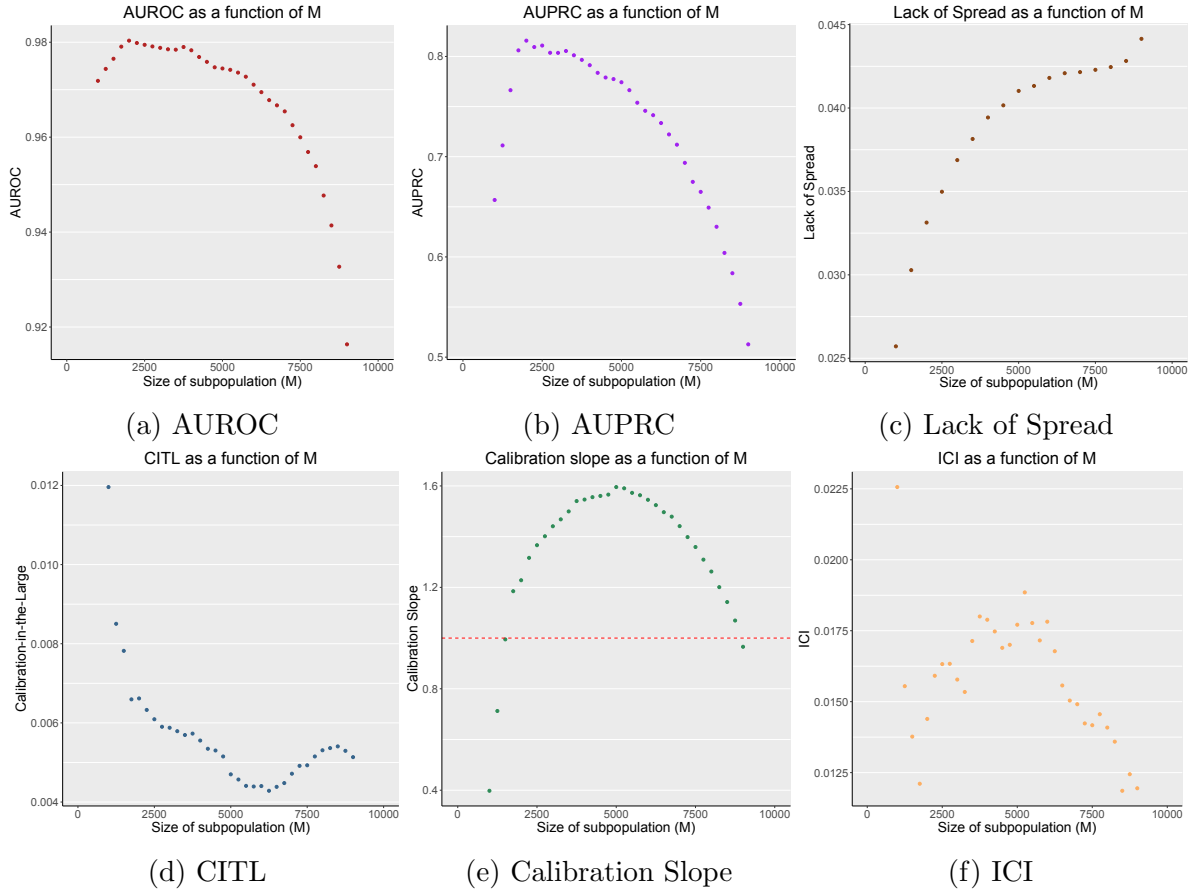


Figure A.5: Performance measures as a function of M when the outcome is simulated by a model which is linear, has 20 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.05.

A.6 Non-Linear, 10 Association, Balanced Prevalence

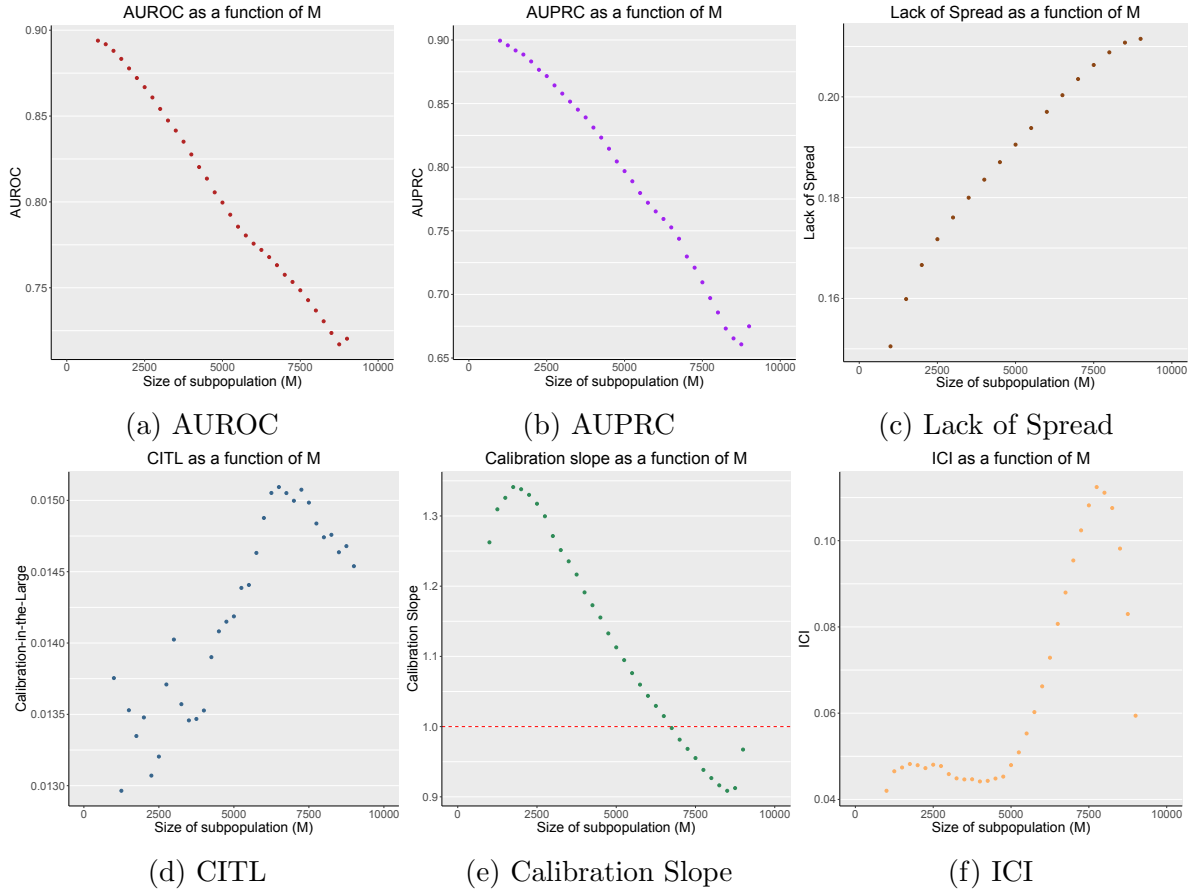


Figure A.6: Performance measures as a function of M when the outcome is simulated by a model which is non-linear, has 10 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.50.

A.7 Non-Linear, 20 Association, Balanced Prevalence

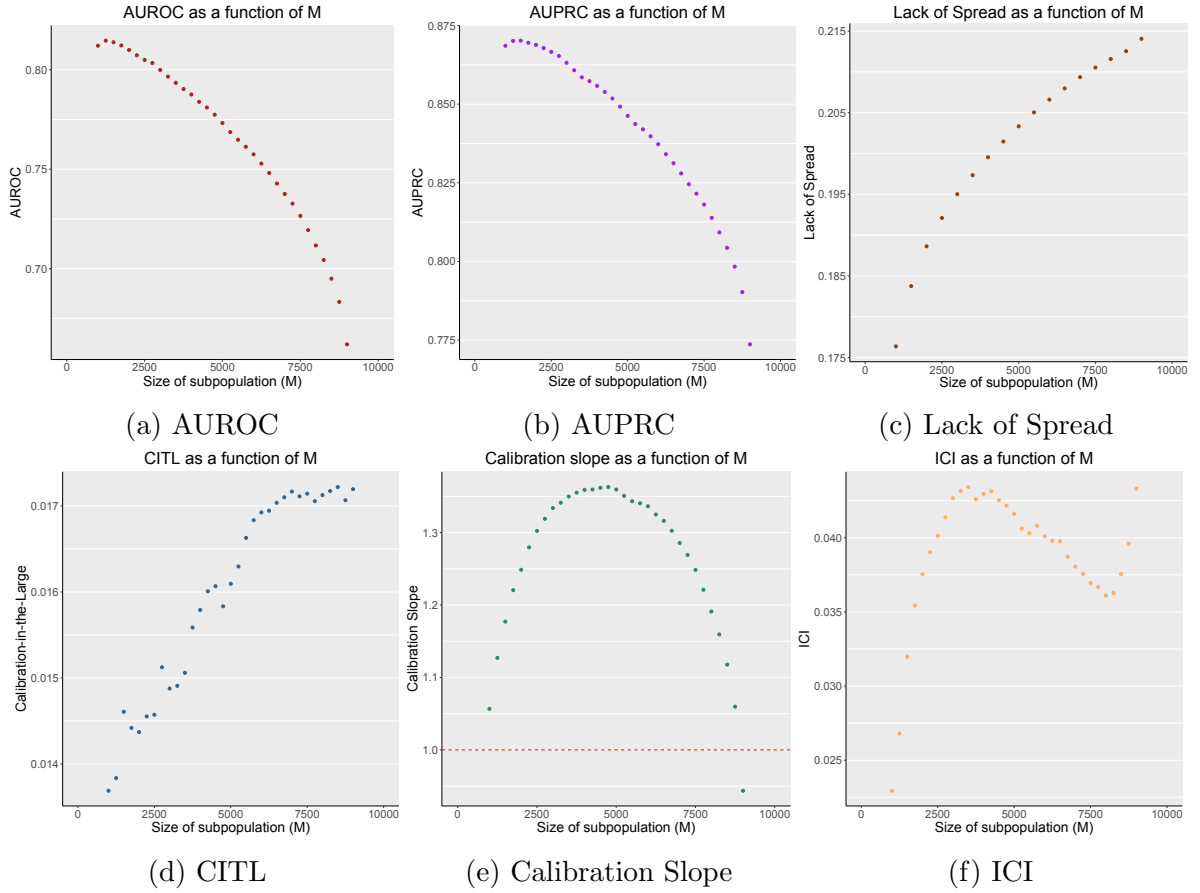


Figure A.7: Performance measures as a function of M when the outcome is simulated by a model which is non-linear, has 20 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.50.

A.8 Non-Linear, 10 Association, Moderate Prevalence

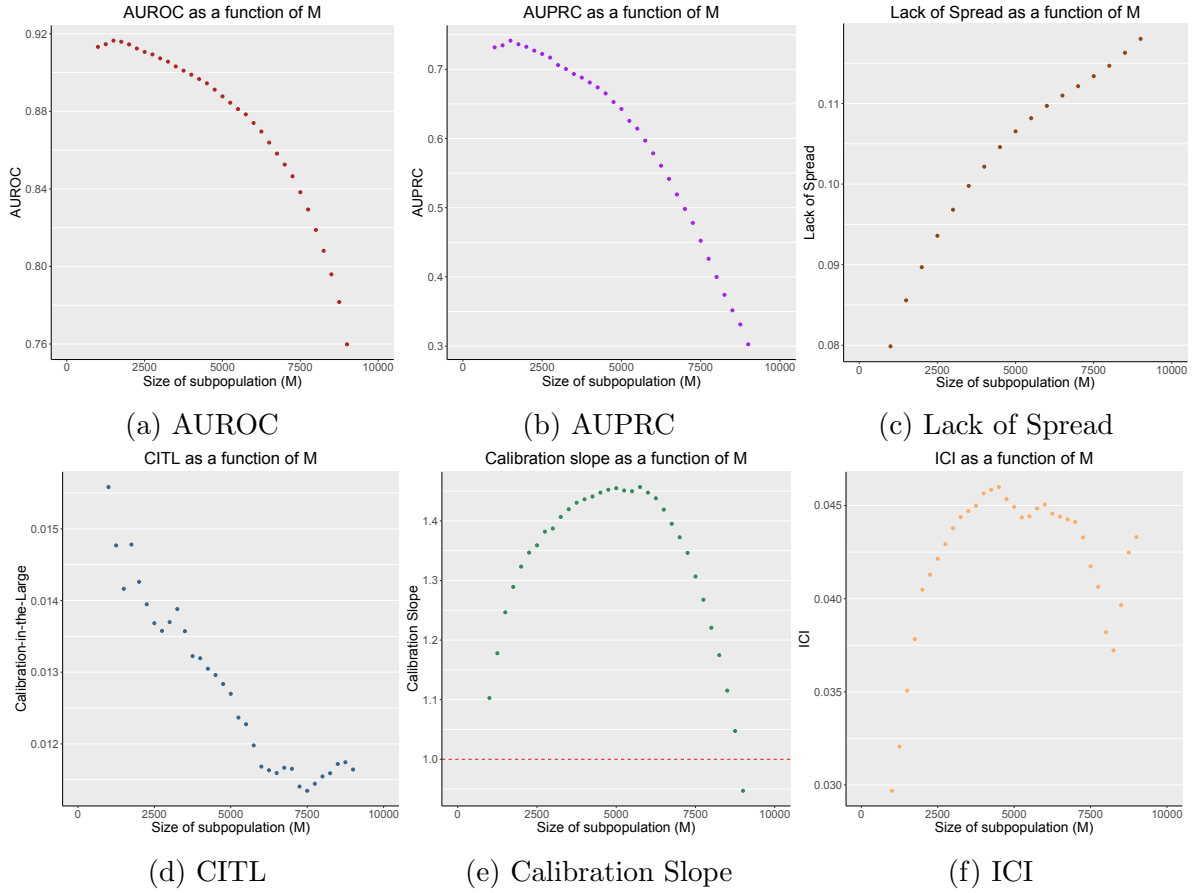


Figure A.8: Performance measures as a function of M when the outcome is simulated by a model which is non-linear, has 10 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.15.

A.9 Non-Linear, 20 Association, Moderate Prevalence

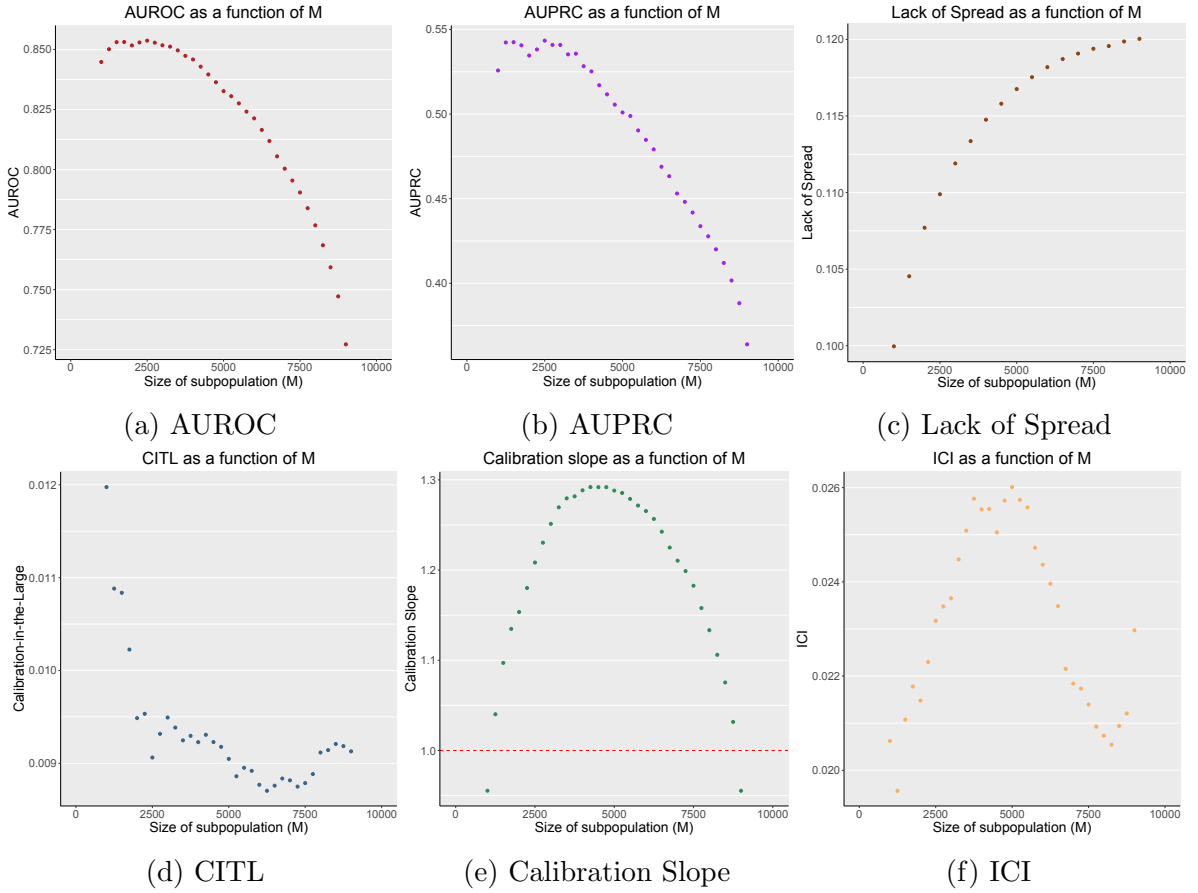


Figure A.9: Performance measures as a function of M when the outcome is simulated by a model which is non-linear, has 20 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.15.

A.10 Non-Linear, 10 Association, Low Prevalence

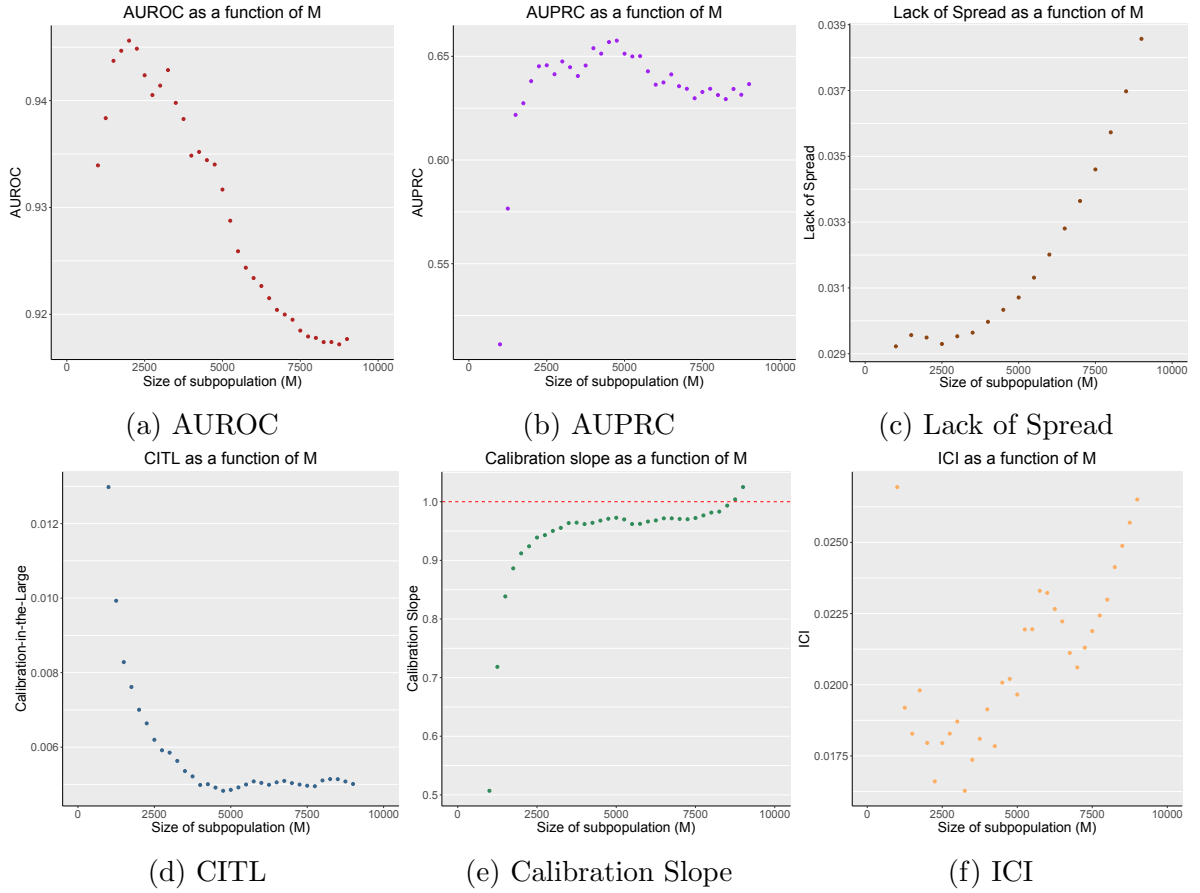


Figure A.10: Performance measures as a function of M when the outcome is simulated by a model which is non-linear, has 10 predictors associated with the outcome and results in a binary outcome with a prevalence of approximately 0.05.

Appendix B

Proposed R Package

To increase the usability of the methods described in Chapters [2](#) and [3](#), we plan on developing an R package which will include a variety of functions to fit personalized predictive models to predict a binary outcome. Below, we outline some key functions that we plan on including in the package.

B.1 Tuning the Size of Subpopulation

One of the functions available will allow the user to tune the size of subpopulation, M , based on a training/testing dataset of their choice. Output will include the optimal choice of M , the optimal M proportion and the value of the loss function across all values of M tested. Some notable arguments to be included in the function are described below.

data	Data to use for the tuning of M .
sim (Similarity Metric)	Choice of measure to determine similarity between an index patient and any other patient. Choices include cosine similarity (“csm”), Mahalanobis distance (“mah”) and generalized Mahalanobis distance (“gmah”). Default is cosine similarity.
lb (Lower Bound)	The desired lower bound (in the form of a proportion, ie. between 0 and 1) on the grid of values on which to tune M . The default will be 20% of the training data, as per our recommendation in Chapter 3.
ub (Upper Bound)	The desired upper bound (in the form of a proportion) on the grid of values on which to tune M . The default will be 50% of the training data, as per our recommendation when there is equal emphasis on discrimination and calibration in Chapter 3.
C (Calibration Measure)	As discussed in Chapter 3, users will have a choice of performance measures to include in the generalized loss function that is used to tune the size of subpopulation. Here, the user defines which measure of calibration they wish to include. Possible measures include calibration-in-the-large (“citl”), calibration slope (“cs”), integrated calibration index (“ici”), with default being the integrated calibration index. We may include other measures of moderate calibration in this list, including the estimated calibration index (“eci”) [Van Hoorde et al., 2015].
D (Discrimination Measure)	Choice of discrimination measure to include in the general loss function that is used to tune the size of subpopulation. Choices include the area under the ROC curve (“auroc”), the area under the precision-recall curve (“auprc”) and the lack of spread (“spread”), with the default being the area under the ROC curve.
K	The desired K for K -fold cross validation. Possible values include 5 or 10, with 10 being the default.
model	Type of model to fit to the similar subpopulation to make a prediction for an index patient. Possible models include logistic regression (“logis”), and will eventually include classification tree (“ct”), random forest (“rf”) and gradient boosting (“gbm”). Logistic regression is the default model.

B.2 Fitting a PPM

Once an optimal M has been determined, one can use this function to fit a personalized predictive model using that M . Output will include the performance of the model, where users can choose which performance measures to calculate, as well as bootstrap confidence intervals of these measures. Selected arguments to be included in the function are described below.

<code>indexData</code>	The dataset that includes the patients for which we want to make a prediction, ie. the index patients.
<code>trainData</code>	The dataset that includes all other patients that make up the training data.
<code>mProp (M Proportion)</code>	The proportion of patients from the entire training dataset to include in the similar subpopulation when fitting the personalized predictive model.
<code>sim</code>	Choice of measure to determine similarity between an index patient and any other patient. Choices include cosine similarity (“csm”), Mahalanobis distance (“mah”) and generalized Mahalanobis distance (“gmah”). Default is cosine similarity.
<code>model</code>	Type of model to fit to the similar subpopulation to make a prediction for an index patient. Possible models include logistic regression (“logis”), and will eventually include classification tree (“ct”), random forest (“rf”) and gradient boosting (“gbm”). Logistic regression is the default model.
<code>perf (Performance Measure)</code>	Measures of performance to calculate. Users can select more than one by putting their choices in a vector. Choices include the area under the roc curve (“auroc”), area under the precision-recall curve (“auprc”), lack of spread (“spread”), calibration-in-the-large (“citl”), calibration slope (“cs”), integrated calibration index (“ici”), Brier Score (“brier”), sensitivity (“sensitivity”) or specificity (“specificity”).
<code>B</code>	Number of bootstrap replicates.
<code>CI (Confidence Interval)</code>	Type of bootstrap confidence interval to calculate. Users can choose more than one option, by including their choices in a vector, including normal (“normal”), basic (“basic”), percentile (“perc”), and bias-corrected and accelerated (“bca”).

B.3 Datasets

We will include a few datasets in this package so that users can test these functions with built-in data. We will simulate this data to be similar to the data generated in Chapters 2 and 3. In addition, we will also consider creating a function where users can simulate their own data by specifying a number of characteristics such as sample size, the number of predictors, the predictor classes, the outcome prevalence, the linearity of the outcome model, and which simulated predictors are associated with the outcome. Doing this will allow users to investigate the relationships between M and the various performance measures using data that may be similar to data they will be working with, with the goal of determining an optimal M to use in practice.