Advances in Similarity-Based Prediction Modeling

by

Minzee Kim

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

Waterloo, Ontario, Canada, 2026

© Minzee Kim 2026

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:	Lisa Lix Professor, College of Community and Global Health, University of Manitoba
Supervisor(s):	Joel A. Dubin Professor, Dept. of Statistics and Actuarial Science, University of Waterloo
Internal Member:	Christian Boudreau Research Associate Professor, Dept. of Statistics and Actuarial Science, University of Waterloo
Internal Member:	Michael Wallace 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

Chapters 2, 3, and 4 were written by Minzee Kim under the supervision of Dr. Joel Dubin. Minzee Kim was responsible for conducting the literature reviews, developing the methods, implementing the simulation studies, performing data analyses, and preparing the chapter manuscripts. Dr. Joel Dubin provided supervision, guidance, intellectual input, and feedback at each stage of the research and manuscript drafts. Chapters 2, 3 and 4 are currently in preparation for submission to peer-reviewed journals. A preprint of the work in Chapter 2 is available on arXiv, the citation of which is included below.

Chapters 1 and 5 were written by Minzee Kim under the supervision of Dr. Joel Dubin, who provided comments, editorial feedback, and guidance on manuscript preparation.

Kim, M., & Dubin, J. A. (2025). Dynamic Prediction of Joint Longitudinal-Survival Models Using a Similarity-Based Approach. arXiv preprint arXiv:2509.05476.

Abstract

Personalized predictive modeling has been growing rapidly in recent years, especially with the availability of Electronic Health Records (EHRs). This approach aims to improve a model's predictive performance by fitting a unique model to each individual. We train the model on a subset of the training data consisting of individuals that are similar to the individual we are predicting for, identified through some patient similarity metric. Studies have shown that using a personalized model trained on a customized subset of the data leads to better prediction than using a global model trained on all the available data in the training data (Krikella and Dubin, 2025; Lee et al., 2015). In this thesis, we discuss advancements in similarity-based prediction modeling through extensive simulation studies and data analyses.

Longitudinal and time-to-event data are often analyzed in biomarker research to study the association between the longitudinal biomarker measurements and the event-time outcome, in which the longitudinal information contributes to the probability of the outcome of interest. An attractive feature of fitting a joint model on this type of data is that we can dynamically predict the survival probability as additional longitudinal information becomes available. In Chapter 2, we propose a new similarity-based method for the dynamic prediction of joint models where we consider training the model on only a targeted subset of the data to obtain an improved outcome prediction. Through a comprehensive simulation study and an application to intensive care unit data on patients diagnosed with sepsis, we demonstrate that the predictive performance of the dynamic prediction of joint models can be improved with our proposed similarity-based approach.

Next, we develop a new patient similarity metric designed to improve the predictive performance of a personalized model for binary response data. Specifically, we introduce a weighted cosine similarity metric in Chapter 3 that extends the standard cosine similarity metric by assigning predictor-specific weights when computing similarity between participants. These weights are estimated using the relaxed adaptive group lasso. Results from our simulation study and an analysis of intensive care unit data involving patients with circulatory system disease show that although the proposed similarity metric leads to a slight deterioration in calibration, it produces substantial gains in discrimination. Overall predictive performance measured by the Brier Score improves because the increase in discrimination outweighs the loss in calibration; therefore, our proposed similarity metric more effectively identifies clinically similar patients, resulting in improved predictive accuracy.

Finally, in Chapter 4, we conduct a comprehensive comparison of several similarity metrics to investigate how the choice of similarity metric influences predictive performance

in personalized modeling, again in the context of binary response data. By fitting models using only a subset of training participants who are most similar to the individual of interest, prediction accuracy for that individual can be improved. Consequently, selecting an appropriate similarity metric that identifies the most relevant subset of data is critical. We compare a range of distance-based and cosine similarity measures alongside clustering-based approaches, an area that is not well explored in the existing literature. In addition, we perform an extensive simulation study to examine how different data-generating mechanisms and underlying dataset characteristics affect the relative effectiveness of each similarity metric.

Finally, we end with a discussion chapter that summarizes the key contributions of the thesis along with highlighting some key areas of future work.

Acknowledgements

First, I would like to thank my supervisor, Joel Dubin, for his guidance, support, and patience throughout my degree. Taking your STAT 932 course during my Master's degree was one of the main reasons I chose to pursue a PhD, and I am so glad that I did. I am deeply grateful for your mentorship and for everything I learned working with you over the years. I will truly miss our weekly meetings. I would also like to thank my committee members, Christian Boudreau, Michael Wallace, Abel Torres Espin, and Lisa Lix, for your time, thoughtful feedback, and valuable suggestions throughout this process. I very much appreciate your guidance and insights.

I would also like to thank Mary Lou Dufton for all your help over the years and for many hallway catch-ups since my Master's. Thank you as well to Chelsea Uggenti for your mentorship and for helping me discover my passion for teaching. I am also grateful to Joslin Goh for mentoring and guiding me as a statistical consultant, and for helping me expand my knowledge of statistics beyond my own research through various opportunities.

To Tatiana and Augustine, I am so grateful that this degree brought us together. Thank you for the coffee breaks, research days, deep conversations, and for celebrating both our successes and challenges along the way. Tatiana, I am so grateful that you are my PhD sister and that we could bounce research ideas off each other anytime and anywhere. Augustine, I am thankful that we got to share an office for a while and experience being coworkers and teaching at the same time.

I am lucky to have incredible friends who have supported me throughout my degree outside of the department as well. Johanna, thank you for believing in me and encouraging me, for our countless calls, and for making the distance feel like nothing through all our trips and visits, even though we haven't lived in the same city since high school. Nicole, thank you for your steady support and for always being someone I could turn to throughout this degree. I am grateful for the many conversations we share, from books and shows to life in general, and I am so excited to be moving closer to you. To Esther, thank you for reminding me that there is life outside of my degree. Over the years, and especially more recently, we have grown closer, and I am so grateful for that. Thank you for our chaotic and funny chats, weekend brunches, Toronto adventures, and for helping me bring out a piece of Korea within me.

The biggest thank you goes to my family. To my sister Hyunzee, thank you for your love, support, and endless laughter. You truly are the funniest person I know. Even though we live in different countries, you never make it feel like we are far apart. Thank you for always being such a great listener, for your advice, and for celebrating my wins as if they

were your own. Dad, thank you for your unwavering support, your push when I needed it, and for sharing your knowledge and advice. Your curiosity inspired me to pursue a PhD. Mom, you are my biggest cheerleader and supporter; thank you for believing in me through everything. I would not be the person I am today without you. Mom and Dad - your love, support, and patience have always been my safety net, giving me the courage and confidence to take risks, grow, and move forward in life. Thank you for everything.

Dedication

To my family.

Table of Contents

Examining Committee Membership ii

Author's Declaration iii

Statement of Contributions iv

Abstract v

Acknowledgements vii

Dedication ix

List of Figures xiii

List of Tables xvi

1 Introduction 1

2 Dynamic Prediction of Joint Longitudinal-Survival Models Using a Similarity-Based Approach 6

2.1 Introduction 6

2.2 Methods 9

2.2.1 Joint Models for Longitudinal and Time-to-event Data 9

2.2.2 Predictive Accuracy of Joint Models 10

2.2.3 Calculating CSM with Time-Varying Covariates . . . 12
2.2.4 Algorithm . . . 13
2.3 Simulations . . . 15
2.3.1 Setup . . . 16
2.3.2 Results . . . 17
2.4 Data Analysis . . . 21
2.4.1 Results . . . 24
2.5 Discussion . . . 26

3 A New Trained Supervised Method for Calculating Patient Similarity 30
3.1 Introduction . . . 30
3.2 Methods . . . 32
3.2.1 Weighted Cosine Similarity Metric . . . 32
3.2.2 Algorithm for PPM . . . 36
3.3 Simulations . . . 37
3.3.1 Setup . . . 38
3.3.2 Simulation Results . . . 40
3.4 Data Analysis . . . 47
3.4.1 Results . . . 47
3.5 Discussion . . . 50

4 Comparison of Similarity Metrics in Personalized Predictive Modeling 54
4.1 Introduction . . . 54
4.2 Methods . . . 56
4.2.1 Algorithm for PPM Using k-NN . . . 59
4.2.2 Algorithm for PPM Using Clustering . . . 60
4.3 Simulations . . . 62
4.3.1 Setup . . . 63
4.3.2 Simulation Results . . . 66
4.4 Discussion . . . 78

5 Discussion **82**

References **86**

APPENDICES **92**

A Chapter 2 Appendix **93**

A.1 A Demonstration of Dynamic Prediction using a Joint Model 93

B Chapter 3 Appendix **99**

B.1 Simulation Results for Case 1 100

B.2 Simulation Results for Case 3 101

B.3 Simulation Results for Case 4 102

B.4 Simulation Results for Case 6 103

C Chapter 4 Appendix **104**

C.1 Simulation Results for Case 2 105

C.2 Simulation Results for Case 4 107

C.3 Simulation Results for Case 6 109

C.4 Simulation Results for Case 8 111

C.5 Simulation Results for Case 10 113

List of Figures

2.1 The estimated Brier score of the five simulated datasets for Scenario 1, with an approximate 95% confidence interval around each estimate. We can see that 0.2 results in the smallest estimated Brier score across all datasets for the five values considered for the proportion of training data. 19

2.2 The estimated Brier score of the five simulated datasets for Scenario 2, with an approximate 95% confidence interval around each estimate. We can see that 0.4 results in the smallest estimated Brier score across all datasets for the five values considered for the proportion of training data. 20

2.3 Flow chart of cohort selection from the eICU database for data analysis. . 22

2.4 The estimated Brier score of the eICU dataset across five different subpopulation proportions. We can see that 0.6 results in the smallest estimated Brier score. 25

3.1 Performance metrics across different subpopulation proportions for the three similarity measures. The dataset has low prevalence with moderate signal (Case 2). 41

3.2 Performance metrics across different subpopulation proportions for the three similarity measures. The dataset has a reasonably high prevalence with moderate signal (Case 5). 44

3.3 Flow chart of cohort selection from the eICU database for data analysis. . 48

3.4 Performance metrics across different subpopulation proportions for the three similarity measures using the ICU dataset. 49

4.1 Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has low prevalence with the first data generation method (Case 1). 68

4.2 Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has low prevalence with the second data generation method (Case 3). 70

4.3 Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has low prevalence with the third data generation method (Case 5). 72

4.4 Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has low prevalence with the fourth data generation method (Case 7). 74

4.5 Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has low prevalence with the fifth data generation method (Case 9). 76

A.1 Traceplots and density plots of the estimated survival parameters in the joint model using **JMbayes2** package in R, showing the convergence in the MCMC process. 94

A.2 Traceplots and density plots of the estimated longitudinal parameters in the joint model using **JMbayes2** package in R, showing the convergence in the MCMC process. 95

A.3 Traceplots and density plots of the estimated elements of matrix $\mathbf{D}$ and τ_1 in the joint model using **JMbayes2** package in R, showing the convergence in the MCMC process. 96

A.4 Plots showing the dynamic prediction of cumulative risk for participant ID 173 as additional longitudinal measurements become available. 97

B.1 Performance metrics across different subpopulation proportions for the three similarity measures. The dataset has low prevalence with weak signal (Case 1). 100

B.2 Performance metrics across different subpopulation proportions for the three similarity measures. The dataset has low prevalence with a strong signal (Case 3). 101

B.3 Performance metrics across different subpopulation proportions for the three similarity measures. The dataset has a high prevalence with a weak signal (Case 4). 102

B.4 Performance metrics across different subpopulation proportions for the three similarity measures. The dataset has a high prevalence with a strong signal (Case 6). 103

C.1 Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has a high prevalence with the first data generation method (Case 2). 105

C.2 Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has a high prevalence with the second data generation method (Case 4). 107

C.3 Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has a high prevalence with the third data generation method (Case 6). 109

C.4 Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has a high prevalence with the fourth data generation method (Case 8). 111

C.5 Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has a high prevalence with the fifth data generation method (Case 10). 113

List of Tables

2.1 Parameter values used to generate the longitudinal data for each of the 2000 individuals in our simulation study. 17

2.2 Parameter values used to generate the event time for each of the 2000 individuals in the two simulation study scenarios. 17

2.3 A table showing the average of the estimated dynamic prediction error of joint models for different subpopulation sizes across five simulation runs for each simulation scenario, displayed in Figures 2.1 and 2.2. The bolded value indicates the smallest estimated Brier score in each simulation case. . . . 21

2.4 The estimated Brier score for five different subpopulation proportions for the eICU data. When the subpopulation proportion is 0.6, the estimated Brier score is minimized. The columns "Lower" and "Upper" represent the 95% confidence interval of the estimates. 24

3.1 Results from the validation study for low prevalence, moderate signal (Case 2). The columns "Lower" and "Upper" represent the 2.5% BCa and 97.5% BCa bootstrap interval estimates, respectively, and the winning M_p from the tuning process used to calculate the performance metrics is 0.1. 42

3.2 Results from the validation study for a reasonably high prevalence, moderate signal (Case 5). The columns "Lower" and "Upper" represent the 2.5% BCa and 97.5% BCa bootstrap interval estimates, respectively, and the winning M_p used to calculate the performance metrics is 0.1 from the tuning process. 45

3.3 Relative difference in the performance measures between using weightbeta CSM and standard CSM, and the tuned M_p value for each simulation case. 45

3.4 Relative difference in the performance measures between using weight01 CSM and standard CSM, and the tuned M_p value for each simulation case. 46

3.5 Results from the validation study for data analysis using the ICU dataset. The columns "Lower" and "Upper" represent the 2.5% BCa and 97.5% BCa bootstrap interval estimates, respectively, and the winning M_p used to calculate the performance metrics is 0.2 from the tuning process. 50

4.1 Results from the validation study for Case 1. $M_p = 0.6$ and $C = 3$ were used from the tuning process. 69

4.2 Results from the validation study for Case 3. $M_p = 0.9$ and $C = 1$ were used from the tuning process. 71

4.3 Results from the validation study for Case 5. $M_p = 0.2$ and $C = 3$ were used from the tuning process. 73

4.4 Results from the validation study for Case 7. $M_p = 0.4$ and $C = 3$ were used from the tuning process. 75

4.5 Results from the validation study for Case 9. $M_p = 1$ and $C = 1$ were used from the tuning process. 77

4.6 A table summarizing important findings from the simulation study. 78

A.1 Estimated parameter values from fitting the joint model in 2.3.1 with data generated using a seed of 1 following the procedure outlined in Section 2.2.1. 98

C.1 Results from the validation study for Case 2. $M_p = 0.8$ and $C = 20$ were used from the tuning process. 106

C.2 Results from the validation study for Case 4. $M_p = 1$ and $C = 1$ were used from the tuning process. 108

C.3 Results from the validation study for Case 6. $M_p = 0.1$ and $C = 3$ were used from the tuning process. 110

C.4 Results from the validation study for Case 8. $M_p = 0.6$ and $C = 3$ were used from the tuning process. 112

C.5 Results from the validation study for Case 10. $M_p = 0.9$ and $C = 1$ were used from the tuning process. 114

Chapter 1

Introduction

A recent development in the area of statistical inference, especially for healthcare and medicine, is patient similarity in prediction models. With a sharp increase in electronic medical records (EMRs) in recent years that contain large but heterogeneous data, researchers have moved towards personalized medicine, which allows for individualized decisions about treatment and care based on a personalized prediction specific to the patient (Wang et al., 2021). Studies have shown that patient similarity can improve the accuracy of predictive models (Krikella and Dubin, 2025; Lee et al., 2015; Xu et al., 2019). This approach, called Personalized Predictive Modeling (PPM), identifies a subset of data that is uniquely tailored to specific individuals to train a predictive model. This type of modeling is not limited to healthcare, and has been implemented successfully in other fields as well, including personalized product recommendations in e-commerce (Adomavicius and Tuzhilin, 2005) and future career trajectory prediction for Major League Baseball players (Bailey et al., 2020).

An important area of PPM is in defining similarity between study participants to improve the predictive accuracy of the model. A useful application of PPM is in the context of precision medicine, where we try to narrow our focus on the effective intervention selection, care planning, and resource allocation for a given subpopulation of individuals with common health-relevant characteristics. In this context, we want to make predictions for an individual, called the index participant, by finding other patients in the index participant's relevant subpopulation in order to learn how to implement treatment decisions. The goal is to identify a group of participants that are similar to an index participant and train the model on this subset of the data to predict the response for the index patient (Sharafoddini et al., 2017). One way to do so is using cluster-based algorithms, where the participants in the training data are grouped based on their profiles and a new participant

is assigned to this pre-determined cluster based on their similarity to the cluster. Other common approaches to measuring similarity are distance metrics, such as Gower's distance, Euclidean distance, and Mahalanobis distance, where the distance between the index participant and the other participants is measured to identify similar participants. The cosine similarity metric is also a common measure that calculates the cosine of an angle between the predictor vectors of two participants to quantify similarity (Sharafoddini et al., 2017).

Then, the next thing to consider in PPM is determining the size of the customized, homogeneous training data. We must determine the optimal size of the subpopulation data since this affects the model's predictive accuracy. When we decrease the size of the subpopulation, this indicates that we are fitting a model on highly homogeneous data, which will improve the prediction performance of the model; however, training a model on a small dataset that contains few patients that are too similar to the index patient may cause other problems, such as optimization error and issues with variability, which may offset at least some of the positive attributes of a personalized predictive model. For example, Krikella and Dubin (2025) empirically show that there is a parabolic relationship between a grid of subpopulation sizes and calibration and, in some scenarios, it may be better to consider larger subpopulations if there is a primary interest in calibration (as opposed to discrimination) of a binary response prediction model. We adopt the methodology proposed by Lee et al. (2015), where we use cross-validation to tune for the "optimal M_p", or the best subpopulation proportion to train a personalized model for each case we are predicting. For the purpose of this thesis, we define the "optimal M_p" as the M_p that will give the best predictive performance measure, which in this case, is the lowest Brier score since it is a summary performance measure that can be decomposed into both discrimination and calibration.

The thesis is organized as follows. My first project demonstrates how PPM can improve the predictive performance of a complex statistical model that is used in healthcare and biostatistics. Longitudinal data, which contains multiple measurements over a certain period of time for given individuals, is extremely effective for evaluating the relationship between the covariates and the longitudinal response over different lengths of time. Longitudinal analysis has many advantages, including establishing a sequence of events, following change over time for particular individuals within a group, and identifying the association between covariates and exposures with regard to presence, timing and chronicity (Caruana et al., 2015). These data are often collected in biomarker research and follow-up studies where biomarkers, e.g., systolic blood pressure or respiratory rate, are collected over time to monitor patient progress. For example, this longitudinal information may be used as a response variable to evaluate the individual change in the response variable over time, or how these changes are related to other covariates. On the other hand, it can be used as

a time-varying covariate in a joint model with, for example, time-to-event (i.e. survival) data to study the association between the longitudinal biomarker measurements and the time-to-event response. In the last 35 years, joint modeling has gained considerable interest and researchers have implemented various extensions, such as handling missing data and fitting a model with multiple biomarker measurements, as well as using different functional forms for associating longitudinal and survival responses (Rizopoulos and Taylor, 2024). We introduce a new joint modeling method that incorporates PPM so that the dynamic prediction of joint models for longitudinal and time-to-event data can be improved by training the model on a personalized subset of the training data with the optimal subpopulation size.

The second project of my thesis focuses on measuring patient similarity more effectively. As mentioned previously, PPM trains the prediction model on a tailored dataset that contains only the most similar observations to the observation of interest; therefore, it is crucial to identify this unique data that consists of similar participants. Often, the cosine similarity metric (CSM) is considered to quantify the similarity between the participants because it is easy to interpret and can handle mixed data types, such as continuous, binary, and categorical variables (Lee et al., 2015). In the setting of healthcare and biostatistics where the data consists of numerous patient features of different types, CSM is a reasonable choice to measure similarity between the patients. Unlike simulation studies, in real life, we are unaware of the true relationship between the predictors and the outcome, therefore we may incorrectly specify the prediction model. Perhaps there are some variables that have a non-linear representation in the linear predictor of the model, or some of the features are not correlated with the outcome but we include them in the prediction models. This inspired us to develop a new patient similarity measure, which is a type of weighted CSM in which we assign specific weights derived from regularization methods to use only the features correlated with the outcome to measure the patient similarity instead of using all available features. Ultimately, the goal is to identify similar participants based on the features that actually matter in the prediction. While a weighted CSM is not a novel method, our proposed method is novel in how it introduces a supervised method to assign weights to the features that can be derived without knowing the type of features in the context of the model - for example, if the features belong to a category of demographic information, lab results, or vital signs - or expert knowledge. We demonstrate that our proposed weighted CSM outperforms the standard CSM, especially when the performance is measured by discrimination.

Finally, my third project investigates commonly used patient similarity metrics to provide guidelines for similarity metrics to use depending on the dataset and the situation at hand, focusing on the context of predicting binary outcomes. Review papers have shown

that there are numerous choices of patient similarity metrics (Sharafoddini et al., 2017). For example, Euclidean distance and Mahalanobis distance quantify the distance between participants, and this distance is used to identify the participants that are the most similar to the participant of interest. CSM is also a popular method that quantifies similarity as an angle between the vectors of predictors. These measures are used in the k-nearest neighbour (k-NN) structure, where the top k number of most similar participants based on the distance or CSM are picked to train the personalized model. There have been recent developments in the patient similarity learning methods as well, such as the large margin nearest neighbour (LMNN), which also applies the k-NN algorithm to identify the personalized training dataset after calculating the distance (Huai et al., 2018). Another approach is clustering, which is not as commonly used in the PPM framework as the k-NN methods. Clustering algorithms, such as k-means and hierarchical clustering, are applied to the training data to form clusters. Then, the participant of interest is assigned to the nearest cluster, and the participants in the cluster are used to train the personalized model (Panahiazar et al., 2015). There is limited work on the comparison of the performance of similarity metrics that are used in the k-NN and clustering framework using the same data. We hope to contribute to the literature by providing a direct comparison of various similarity metrics under multiple data-generating mechanisms to investigate how the data structure can affect the performance of each similarity metric, and ultimately provide a guideline on which patient similarity metric one should use depending on the data.

Our methods proposed in Chapter 2 and Chapter 3 are applied to a publicly available intensive care unit database, called the eICU Collaborative Research Database. This multi-center critical care database was made available by Philips Healthcare with the MIT Laboratory for Computational Physiology and is publicly available on the PhysioNet repository (Pollard et al., 2018). The eICU database holds a large amount of data associated with over 200,000 patient intensive care unit (ICU) stays, containing information about vital sign measurements, nurses' notes, diagnoses, treatment decisions, patient demographic information and more from a combination of critical care units throughout the continental United States in 2014 and 2015. However, certain social variables, such as income, socioeconomic status, and education level, are not available in this dataset. Previous studies in the literature, such as Krikella and Dubin (2025) and Ju and Dubin (2026), use a combination of demographic and clinical variables, including admission characteristics (e.g., age and weight), severity-of-illness score, laboratory measurements, counts of laboratory tests performed, vital signs, and diagnostic information in their respective data analyses.

The remainder of this thesis is organized as follows. In Chapter 2, we present our first project on improving the predictive performance of joint modeling with PPM. This chapter includes background on joint modeling, our proposed framework, a series of simulation

studies, and a real-world data analysis using the eICU database for patients with sepsis. In Chapter 3, we discuss our second project, which is a newly proposed similarity metric to improve the predictive performance of PPM. We present our methodology, simulation studies, and data analysis using the eICU database for patients with circulatory system disease to demonstrate the effectiveness of our proposed method. In Chapter 4, we provide a comprehensive overview and comparison of various patient similarity metrics, and how the relative performance of these metrics can change depending on the data-generating mechanism. We present this result using an extensive simulation study. We conclude this thesis in Chapter 5, summarizing our contributions to the literature along with ideas and directions for future work.

Chapter 2

Dynamic Prediction of Joint Longitudinal-Survival Models Using a Similarity-Based Approach

2.1 Introduction

A biomarker is an indication and a measure of a medical state that can be measured objectively and reproducibly - it is "any substance, structure, or process that can be measured in the body or its products and influence or predict the incidence of response or disease" (World Health Organization, 2001). In particular, biomarkers are used in research for disease diagnosis and personalized medication and treatment (Gosho et al., 2012). Longitudinal and time-to-event data are often analyzed in biomarker research to study the association between the longitudinal biomarker measurements and an event-time response, in which the longitudinal information contributes to the probability of the outcome of interest (Rizopoulos, 2012). More specifically, in follow-up studies, different measurements are collected for each individual over time, including longitudinal responses and the time until an event occurs for that individual. It may be of interest to study the association between the longitudinal biomarker information and time-to-event response, instead of conducting separate analyses for each, which is often the goal of biomarker research where studies aim to identify biomarkers with strong prognostic capabilities for an event-time response (Rizopoulos, 2012). For example, T4 and TSH (thyroid-stimulating hormone) levels are measured from blood samples to monitor patients at risk of thyroid cancer and to track the disease progression by investigating the association between these

biomarkers and the time of cancer development. Another unique characteristic of follow-up studies is that it has a dynamic nature - the rate of progression is different between each individual and it also dynamically changes over time uniquely for individuals (Rizopoulos, 2012).

Joint longitudinal-survival (JLS) models were developed to accommodate the unique characteristics of longitudinal and time-to-event data. They are used to efficiently study the effects of endogenous time-varying covariates on the survival response; we can study the strength of association between the hazard of the event and the time-varying covariate by implementing joint models. Furthermore, joint models can dynamically predict the survival probability as additional longitudinal measurements become available, which reflects the unique nature of follow-up studies (Rizopoulos, 2016). That is, as time progresses and additional information becomes available for each participant, we can update the survival prediction for each participant using the new information. There has been much development in JLS modeling in the past 35 years; numerous extensions have been implemented, including flexible modeling of longitudinal trajectories, incorporating multiple longitudinal markers, and modeling multiple failure times (Rizopoulos, 2012). In this work, we propose a new method as an extension of JLS modeling by using a similarity-based approach to improve the dynamic predictive performance of JLS models.

Patient similarity has been considered a groundbreaking new paradigm and has been implemented to create customized models that lead to better prediction performance, particularly with model discrimination (Lee et al., 2015). In recent years, there has been much interest in applying patient similarity to biomedicine, where a sharp increase in electronic health records (EHR) has resulted in a sizeable amount of available patient information (Sharafoddini et al., 2017). Often, we use these data to fit a model for patient response prediction as it is directly related to intervention and treatment selection, care planning, and resource allocation. Traditionally, large and heterogeneous data such as EHR were analyzed for predictions. However, using the entire training data shows the "average best choice"; thus, physicians and practitioners cannot rely on just this information when they want to make a prediction for a new patient with characteristics that diverge from the average. This is the motivation behind Personalized Predictive Modeling (PPM) based on patient similarity - the goal is to provide personalized predictions by identifying patients in training data similar to an index patient for whom we are trying to make a prediction (Sharafoddini et al., 2017). Past research has shown that implementing patient similarity can improve the predictive accuracy of different models, such as logistic regression, decision trees, and random survival forests (Lee et al., 2015; Xu et al., 2019).

As mentioned in Chapter 1, quantifying similarity between the patients is an important step in PPM to improve the predictive accuracy of the model. Numerous methods were

developed to measure similarity, including but not limited to distance-based similarity metrics such as the Euclidean distance and Mahalanobis distance; cosine similarity metrics; correlation-based similarity metrics where the correlation coefficient was used to identify similar patients; and cluster-based algorithms that group the patients in the training data based on their profiles (Sharafoddini et al., 2017). In this work, we implement the cosine similarity metric (CSM) which measures the cosine of the angle between the predictor vectors of two participants. More specifically, it is expressed as

$$CSM(S_i, S_j) = \frac{\mathbf{S_i} \cdot \mathbf{S_j}}{\|\mathbf{S_i}\|\|\mathbf{S_j}\|},$$

where $\mathbf{S_i}$ and $\mathbf{S_j}$ are the predictor vectors of Participant i and Participant j, respectively, and $||\cdot||$ represents the Euclidean vector magnitude. Since the patient similarity is quantified as an angle between two vectors, it is normalized to be between -1, representing minimum similarity where the two vectors would be in exactly opposite directions, and 1, representing the maximum similarity where the two vectors are identical (Lee et al., 2015). One convenient feature of using cosine similarity over Euclidean or Mahalanobis distance is that it can naturally accommodate categorical variables. By calculating the CSM of an index participant with each of the participants in a training data set, we can rank the participants in the training data set in descending order to identify the participants that are the most similar to the index participant.

By training the model on a more personalized subset of data consisting of similar individuals, we can increase the predictive accuracy; however, we must strike a balance in choosing the right sample size. We may offset the positive results from the patient similarity and develop a model with worse prediction accuracy than the original model when we increase personalization of the data and limit the sample size to be too small (Lee et al., 2015). Krikella and Dubin (2025) empirically showed that though model discrimination tends to improve as the size of the subpopulation decreases, model calibration may be more complicated and settling on a size of the subpopulation to optimize calibration is less obvious. This said, sometimes we consider the Brier score to evaluate predictive model performance, which is a summary measure that comprises both discrimination and calibration. In this work, we aim to show that the prediction of joint models for longitudinal and survival data measured by the Brier score (i.e., specifically, time-varying Brier score (Henderson et al., 2002)), can be improved using a similarity-based approach. We use cross-validation to determine the optimal subpopulation size that produces the best prediction accuracy in the joint model. We also demonstrate that the Brier score of the joint model can be decreased when we train the model on the subpopulation of the most similar individuals in the training data to each of the individuals in the testing data. Thus, we make a novel

contribution to the literature by improving the predictive performance of joint models using PPM and evaluating model performance across varying training data sizes consisting of similar observations through cross-validation.

This chapter is organized as follows. Section 2.2 begins with a brief discussion on fitting a joint model, evaluating its predictive accuracy, and handling time-varying predictors when calculating the CSM. These components are used to implement our proposed algorithm for improving the predictive accuracy of the joint model using a similarity-based approach, presented in Section 2.2.4. We then describe the details of a comprehensive simulation study in Section 2.3, including data generation procedure and our simulation results. In Section 2.4, we apply our proposed algorithm in Section 2.2.4 to a dataset obtained from the publicly-available eICU Collaborative Research Database (Pollard et al., 2018) to attempt to confirm the results of our simulation study. Finally, we conclude with a discussion of our findings and future work in Section 2.5.

2.2 Methods

2.2.1 Joint Models for Longitudinal and Time-to-event Data

Joint models for longitudinal and time-to-event data can accommodate endogenous covariates in survival analysis. We can use longitudinal information to better aid with survival prediction, over just using baseline data, and can dynamically update the survival prediction as more longitudinal data becomes available. A standard relative risk model to jointly fit longitudinal and time-to-event data is defined as the following (Rizopoulos, 2023):

$$h_i(t|\mathbf{M}_i(t)) = h_0(t)\exp\left\{\boldsymbol{\gamma}^T\mathbf{w}_i + \alpha\mathbf{m}_i(t)\right\}, \tag{2.1}$$

where $\mathbf{M}_i(t)$ is the longitudinal history of individual i such that $\mathbf{M}_i(t) = \{\mathbf{m}_i(s), 0 \leq s < t\}$, α quantifies the association between the endogenous covariate and the risk of an event, $h_0(t)$ is the baseline hazard function, and $\mathbf{w}_i$ is a vector of baseline covariates with corresponding regression parameters $\boldsymbol{\gamma}$ in the Cox proportional hazards model (Cox, 1972). $\mathbf{m}_i(t)$ is defined from a linear mixed effects model, where

$$\begin{aligned}\mathbf{Y}_i(t) &= \mathbf{m}_i(t) + \boldsymbol{\epsilon}_i(t)\\ &= \mathbf{X}_i^T(t)\boldsymbol{\beta} + \mathbf{Z}_i^T(t)\mathbf{b}_i + \boldsymbol{\epsilon}_i(t).\end{aligned}$$

Here, $\mathbf{X}_i^T(t)$ is a $(n_i \times p)$ matrix of covariates; $\boldsymbol{\beta}$ is a $(p \times 1)$ vector of fixed effects; $\mathbf{Z}_i^T(t)$ is a $(n_i \times q)$ matrix of covariates, typically a subset of $\mathbf{X}_i^T(t)$, with $q \leq p$; $\mathbf{b}_i$ is a $(q \times 1)$

vector of random effects with $\mathbf{b}_i \sim MVN(\mathbf{0}, \mathbf{D})$; and $\boldsymbol{\epsilon}_i$ is the random error associated with individual i such that $\boldsymbol{\epsilon}_i \sim MVN(\mathbf{0}, \sigma^2\mathbf{I}_{n_i})$ (Fitzmaurice et al., 2012). This last assumption on the model error terms, referred to as the "conditional independence" assumption, can be relaxed, if the model so warrants, by imposing a (serial) correlation structure on these error terms. The **JMbayes2** package in R by Rizopoulos et al. (2023) is often used to fit joint models. An example of fitting a joint model to perform dynamic prediction of cumulative risk for an individual is demonstrated in Appendix A, using the first simulated dataset described in Section 2.3.1.

2.2.2 Predictive Accuracy of Joint Models

Often, the motivation behind fitting a statistical model to a dataset is to predict the response. With longitudinal and time-to-event data, it offers a unique advantage that we can dynamically predict the probability of an event; thus, we can use all the available information at hand, including both the baseline information and accumulated biomarker measurements, to update our predictions of the probability of the event as more biomarker measurements become available. Furthermore, we can make a personalized prediction that is specific to the individual of interest. With growing interest in prognostic models and personalized medicine in medical care, dynamic prediction of joint models has been especially helpful in informing physicians about future prospects of a patient to adjust medical care (Barrett and Su, 2017).

Formally put, we have longitudinal measurements available up to time point t for a new participant j,

$$\mathbf{Y}_j(t) = \{y_j(s), 0 \leq s \leq t\},$$

and we are interested in the survival probability of patient j at some future time u using both baseline covariates and longitudinal information available until time t, which can be written as

$$\pi_j(u|t) = P\{T_j^* \geq u | T_j^* > t, \mathbf{Y}_j(t), D_n\},$$

where T_j^* is the true underlying event time of individual j, $u > t$, and D_n is the sample in which the joint model was fitted.

In general, calibration and discrimination are used to evaluate the performance of the model, where the former measures how well the model predicts the observed data and how closely the predicted probability agrees numerically with the true response (D'Agostino and Nam, 2003) and the latter focuses on how well the model can discriminate between participants who experienced the event and those who did not (D'Agostino and Nam, 2003).

Brier scores are also often used to evaluate model performance, as they can be decomposed into separate components that respectively capture both discrimination and calibration (Krikella and Dubin, 2025; Murphy, 1973). Two things must be considered when measuring the predictive accuracy of joint models for longitudinal and time-to-event data. The first is censoring - censoring may depend on the observed longitudinal measurements or on the baseline covariates, so the performance measures must be corrected for censoring using model-based estimators of the censoring distribution (Rizopoulos et al., 2017). The second is the fact that due to its dynamic nature, the model predictions are updated as more longitudinal information becomes available. Then, we redefine discrimination so that we are interested in the event occurrence in a time frame $(t, t + \Delta t]$ when we have longitudinal measurements for participant j until time t. For calibration and Brier score, we are interested in predicting the probability of an event at time $t + \Delta t$ given longitudinal information until t.

In this work, we have decided to use the time-dependent Brier score to measure the predictive performance since, as just conveyed above, it captures both discrimination and calibration, though we will discuss utilizing these performance measures separately in the Discussion section (Krikella and Dubin, 2025; Murphy, 1973). The expected quadratic error of prediction, or the Brier score, conditional on time $u > t$, is

$$PE(u|t) = E\left[\{N_j(u) - \pi_j(u|t)\}^2\right],$$

where $N_j(t) = I(T_j^* > t)$, or the event status at time t, and $u = t + \Delta t$. Then the estimate of the Brier score that accounts for censoring is

$$\widehat{PE}(u|t) = (R(t))^{-1} \sum_{j:T_j \geq t} (\widehat{L}(u|t)), \tag{2.2}$$

where

$$\begin{aligned}\widehat{L}(u|t) = {} & I(T_j \geq u)\left(1 - \hat{\pi}_j(u|t)\right)^2 \\ & + \delta_j I(T_j < u)\left(0 - \hat{\pi}_j(u|t)\right)^2 \\ & + (1 - \delta_j) I(T_j < u)\left\{\hat{\pi}_j(u|T_j)\left(1 - \hat{\pi}_j(u|t)\right)^2 + \left(1 - \hat{\pi}_j(u|T_j)\right)\left(0 - \hat{\pi}_j(u|t)\right)^2\right\},\end{aligned} \tag{2.3}$$

and $R(t)$ is the number of participants at risk at time t (Henderson et al., 2002). In Equation (2.3) combined with Equation (2.2), the first and second terms of $\widehat{L}(u|t)$ correspond to patients who were alive after time u and dead in the interval $[t, u)$, respectively, and the third term corresponds to those who were censored in the interval $[t, u)$ (Rizopoulos et al., 2017).

2.2.3 Calculating CSM with Time-Varying Covariates

Recall that CSM is expressed as the following, which defines the similarity between the index participant and the participant in the training data:

$$CSM(S_i, S_j) = \frac{\mathbf{S_i} \cdot \mathbf{S_j}}{\|\mathbf{S_i}\|\|\mathbf{S_j}\|}.$$

We need to identify $\mathbf{S_i}$ and $\mathbf{S_j}$, the predictor vectors for Participant i and Participant j. While it is straightforward to include the baseline covariates as part of the vectors, it can be advantageous to take into account longitudinal measurements as well. To represent the multiple longitudinal measurements into a finite number of vectors, we use functional principal component analysis (FPCA), a multivariate analysis technique that extracts information from functional data, such as longitudinal data (Wang et al., 2016). It reduces the dimension of data with a large number of interrelated variables and transforms the data to a new set of variables, called functional principal components, while retaining as much of the total variation as possible (Ullah and Finch, 2013). The principal components are uncorrelated and can be ordered so that the first few will retain most of the variation present in the original variable. This is implemented using the Principal Analysis by Conditional Estimation (PACE) algorithm in the R package **fdapace** (Zhou et al., 2022) where the expansion for the longitudinal measurement $Y_i(t)$ with only the first r eigenfunctions $(\hat{\phi}_k(t), k = 1, 2, ..., r)$ is

$$\hat{Y}_i^r(t) = \hat{\mu}(t) + \sum_{k=1}^{r} \hat{\xi}_{ik}\hat{\phi}_k(t),$$

where $\hat{\mu}(t)$ is the estimated mean function and $\hat{\xi}_{ik}$ are the FPCs. Therefore, in our simulation to be discussed in Section 2.3, we used FPCA to reduce the dimensionality of the longitudinal data and use FPCs to explain the variation in the trajectories, which are used in the calculation of the cosine similarity metric along with the baseline covariates to quantify the similarity between the participants in the study. r can be chosen using a criterion such as the number of FPCs that exceed a certain threshold of explained variation (e.g., 95%) in a given longitudinal variable. Note that with the PACE algorithm, we can apply FPCA to irregular and even sparse longitudinal measurements. Even when measurements are missing at certain time points for a participant, PACE borrows strength from the entire data to estimate the global mean and covariance, and uses these estimates to calculate the FPCs by conditioning on the observed data.

2.2.4 Algorithm

In this section, we describe our new proposed algorithm to improve the dynamic prediction of joint models for longitudinal and time-to-event data at a specific time point t for a specific future time u that has some subject-matter interest, called *similarity-based dynamic prediction using joint models*.

1. Let D_n be the training data that contains longitudinal and survival information for n individuals. Then, for each individual j in the testing data, we use the cosine similarity metric and the features of the individuals in D_n to calculate the similarity between the participant in D_n with individual j.

2. Let $M_p \in (0, 1]$ be the proportion of the training data for fitting a personalized model. Let $m = M_p \times n$ be the subpopulation size for fitting the personalized model. We define $D^*_{m,j}$, which is the subset of the training data containing m individuals that are the most similar to patient j in the testing data, determined by the cosine similarity metric.

3. Fit a joint model with $D^*_{m,j}$ using the following equation:

$$h^*_i(t|\mathbf{M}^*_i(t)) = h^*_0(t) \exp\left\{\boldsymbol{\gamma}^T \mathbf{w}^*_i + \alpha \mathbf{m}^*_i(t)\right\}, \tag{2.4}$$

   where $\mathbf{M}^*_i(t)$ is the longitudinal history of individual i in $D^*_{m,j}$, $\mathbf{w}^*_i$ is the baseline covariate vector of individual i in $D^*_{m,j}$, and $\mathbf{m}_{i*}(t) = \mathbf{X}^T_{i*}(t)\boldsymbol{\beta}^* + \mathbf{Z}^{*T}_i(t)\mathbf{b}_{i*}$.

4. Using the fitted joint model in Step 3, calculate the probability of survival at time u given biomarker measurements until time t with $D^*_{m,j}$:

$$\pi^*_j(u|t) = P\{T^*_j \geq u | T^*_j \geq t, \mathbf{Y}^*_j(t), D^*_{m,j}\}. \tag{2.5}$$

In other words, by training the joint model on $D^*_{m,j}$ instead of D_n, we are using personalized data that contains $M_p \times 100\%$ of the original data D_n that is the most similar to the testing participant j to dynamically predict its response.

By decreasing M_p, we can create personalized training data for predicting the response of participant j as it will contain smaller and smaller subsets of participants from D_n that are more similar to participant j. However, a smaller dataset may cause other problems, such as overfitting the data (e.g., if too many covariates are considered relative to sample size), having optimization problems for fitting the underlying LME model, or fitting an

inadequate model due to a small sample size. Therefore, it is of interest to identify the optimal M_p that will maximize the accuracy in predicting the survival probability at future time u given baseline covariates and longitudinal information available until time t. Below, we present the algorithm to tune for M_p using R-repeated K-fold cross-validation:

1. Split the data into K different folds for K-fold cross-validation.
2. Let k be the hold-out fold for testing data, and the other $K-1$ folds be the training data.
3. For training and testing data separately, apply FPCA using the PACE algorithm to extract the first few FPCs for the longitudinal measurement that cumulatively explain at least $S\%$ of the total variation in the trajectory of the longitudinal marker. We use $S = 95\%$ as the total variation of the longitudinal biomarker to be used by the FPCs in our simulation study below. Note that we use the longitudinal measurements available up to time t to calculate the first few FPCs to avoid data leakage.
4. Consider a grid of M_p values to tune M_p.
   (a) For an index participant j in the testing data, calculate $CSM(S_j, S_i)$ for all participants i in the training data. For the features to calculate the similarity between two participants using CSM, we consider the baseline covariates and FPCs to explain the variation in the trajectory of the time-varying covariates calculated in Step 2.
   (b) Sort the participants in the training data in descending order by the similarity metric.
   (c) Fit the joint model in Equation (2.4) with $D^*_{m,j}$, or the top $M_p \times 100\%$ of the sorted training data. For a pre-specified t and $u > t$, calculate the loss function for participant j defined as the following:

$$\begin{aligned}\widehat{L}^*(u|t) = & I(T_j \geq u)\left(1 - \hat{\pi}^*_j(u|t)\right)^2 \\ & + \delta_j I(T_j < u)\left(0 - \hat{\pi}^*_j(u|t)\right)^2 \\ & + (1-\delta_j) I(T_j < u)\left\{\hat{\pi}^*_j(u|T_j)\left(1 - \hat{\pi}^*_j(u|t)\right)^2 + \left(1 - \hat{\pi}^*_j(u|T_j)\right)\left(0 - \hat{\pi}^*_j(u|t)\right)^2\right\},\end{aligned}$$

   where $\hat{\pi}^*_j(u|t)$ is defined in Equation (2.5).

(d) Repeat Step 4(a) to Step 4(c) for every participant j in the testing data and average over the values calculated in Step 4(c) to find the Brier score given by the following equation:

$$\widehat{PE}^{*}(u|t) = (R(t))^{-1} \sum_{j:T_j \geq t} (\widehat{L}^{*}(u|t)).$$

5. Repeat Steps 3-4 for each k in the K folds. Average the value of the Brier scores over the K folds.

6. Repeat Step 2 to Step 5 R times and average over the repeated K-fold cross validation runs. The optimal M_p is the value with the smallest estimated Brier score for calculating the dynamic prediction at a specific time point t for a specific future time u that has some subject-matter interest.

7. In the case where there is a desire to evaluate prediction performance on a hold-out validation sample, use the M_p that was tuned using the training-testing data outlined in the steps above. We consider this step in the data analysis section in Section 2.4.

2.3 Simulations

Below, we describe the simulation study performed to show how we can improve the dynamic predictive performance of joint models using a similarity-based approach. Tuning for the optimal subpopulation size using 5-fold cross-validation repeated ten times for each dataset under dynamic prediction for joint modeling is very computationally extensive. Therefore, we generate five datasets for two different simulation scenarios in this simulation study. Still, we plan to increase the number of repeats of the cross-validation and the simulation scenarios with more efficient coding in future research. We consider five possible values for the proportion of the training data to determine the size of the targeted subpopulation data - $20\%, 40\%, 60\%, 80\%$, and 100% - and calculated the Brier score of the model for predicting the response at time $u > t$ with biomarker measurements available until time $t = 1$. While it is useful to use longitudinal information to update the survival prediction, we cannot always continue to collect biomarker data for an extended period of time before making a decision based on the survival prediction. For example, in the ICU, healthcare professionals measure biomarkers such as blood pressure, oxygen level, and temperature to monitor the patients and predict their risks, but they must make a decision before it is too late to take action and operate on the patient. Section 2.3.1 defines the parameter values and the methods to generate longitudinal and time-to-event data for this simulation study, and the results are presented in Section 2.3.2.

2.3.1 Setup

We consider two cases for the simulation study, coded in R version 4.4.0 (R Core Team, 2023). We first outline the steps to generate longitudinal and time-to-event data for fitting the joint model expressed in Equation (2.1). For both scenarios, we use the following structure of the joint model:

$$h_i(t|\mathbf{M}_i(t)) = h_0(t) \exp\left\{\gamma_1 w_{1i} + \gamma_2 w_{2i} + \alpha \mathbf{m}_i(t)\right\},$$

where

$$\mathbf{m}_i(t) = \beta_0 + b_{0i} + (\beta_1 + b_{1i}) \times time$$

and w_{1i} and w_{2i} are each non-time-varying covariates associated with the hazard.

To generate longitudinal data for $n = 2000$ in the simulation study, we use the following equation:

$$Y_{ij} = \beta_0 + b_{0i} + (\beta_1 + b_{1i}) \times time + \boldsymbol{\epsilon}_i,$$

where $\mathbf{b}_i \sim MVN(\mathbf{0}, \mathbf{D})$ such that $\mathbf{D} = \begin{pmatrix} \tau_0^2 & \tau_0\tau_1\tau_{01} \\ \tau_0\tau_1\tau_{01} & \tau_1^2 \end{pmatrix}$; $\boldsymbol{\epsilon}_i$ is the random error associated with individual i such that $\boldsymbol{\epsilon}_i \sim MVN(\mathbf{0}, \sigma^2\mathbf{I}_{n_i})$; and $time$ is a sequence of variables from 0 to 10 increasing by 0.5. The values used to generate the longitudinal data are identified in Table 2.1.

Using these longitudinal measurements, we generate the event time for each individual. We use the equation created by Austin (2012) for generating survival times to simulate Cox proportional hazard models with continuous time-varying covariates, assuming that the survival times follow a Weibull distribution. Survival times for the participants in our simulation study are generated using Equation 2.6, where $w_1 \sim Unif(-1.73, 1.73)$, $w_2 \sim N(0, 0.7)$.

$$T = \left[\log\left\{\frac{-\log(u)(1+v)}{\exp(\gamma_1 w_1 + \gamma_2 w_2 + \alpha(\beta_0 + b_0))\lambda v \alpha} + 1\right\}\frac{1}{\beta_1 + b_1}\right]^{1/(1+v)} \tag{2.6}$$

As mentioned previously, we consider two simulation scenarios. The parameter values used to generate the two cases are identified in Table 2.2. Notice that in Scenario 1, α is much greater than γ_1 and γ_2 compared to that in Scenario 2; the association between the biomarker and the event time is much greater than the association between the baseline covariates and the event time in Scenario 1. In Scenario 2, the magnitude of association between the baseline covariates and the biomarker with the event time is more similar. Finally, we consider censoring. In this simulation study, we choose right random censoring,

where the censoring time of individual i is generated as $C_i \sim Unif(1, C)$. C is 7 in Scenario 1, and 4 in Scenario 2.

Table 2.1: Parameter values used to generate the longitudinal data for each of the 2000 individuals in our simulation study.

Parameter	**Value**
β_0	-1.35
β_1	0.3
τ_0	0.27
τ_1	0.08
τ_{01}	0.2
σ	0.25

Table 2.2: Parameter values used to generate the event time for each of the 2000 individuals in the two simulation study scenarios.

Scenario 1[a]		**Scenario 2[b]**	
Parameter	**Value**	**Parameter**	**Value**
λ	0.5	λ	0.5
v	1.03	v	1.03
α	4.5	α	4.5
γ_1	0.5	γ_1	3.5
γ_2	1.5	γ_2	3.5

[a] The coefficient of the biomarker to the event-time response variable is much greater compared to the coefficients of the fixed covariates.

[b] The coefficient of the biomarker to the event-time response variable is similar in magnitude to the coefficients of the fixed covariates.

2.3.2 Results

In this simulation study, we generate five datasets for the two scenarios. Table 2.3 shows the average estimated Brier score for different subpopulation proportions across the five

datasets. As explained in Section 2.2.4, we use repeated cross-validation to determine the optimal M_p, i.e. the proportion of the training data for fitting a personalized joint model. We implement a 5-fold CV repeated ten times, meaning that we can also find the standard error and obtain an approximate $100 \times (1-\alpha)\%$ confidence interval around the estimate of the Brier score for each subpopulation proportion using the following equation (Bates et al., 2023; Hastie et al., 2009):

$$\left(\bar{e} - z_{1-\alpha/2}\widehat{SE}, \bar{e} + z_{1-\alpha/2}\widehat{SE}\right),$$

where

$$\widehat{SE} = \frac{1}{\sqrt{n}} \times \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(e_i - \bar{e})^2}$$

such that n is the number of folds (50 in this scenario), e_i is the estimate from the ith fold, $\bar{e}$ is the average of e_i across all folds, and $z_{1-\alpha/2}$ is the $1-\alpha/2$ quantile of the standard normal distribution. This approximation should be reasonable assuming the Brier score is not too close to 0.

Recall that a smaller Brier score indicates a better predictive performance. Figure 2.1 and Figure 2.2 show the results from the five simulated datasets for each scenario with confidence intervals around the estimate for each subpopulation proportion. For Scenario 1, we can see that the estimated Brier score increases as the subpopulation proportion increases; for Scenario 2, the estimated Brier score is the lowest for $M_p = 0.4$ across the five values considered for M_p. For Scenario 1, where the association between the biomarker and the response is greater in comparison to the association between the non-time-varying covariates and the response, the average estimated Brier score is minimized when M_p is 0.2 - that is, fitting the joint model on 20% of the training data that is the most similar increases the predictive performance of the model than using the entire training data. For Scenario 2, where the association between the biomarker and the response is similar in size to that of non-time-varying covariates and the response, the average estimated Brier score is minimized when M_p is 0.4. When we fit a model on targeted training data, specifically 40% of the training data, we can improve the predictive performance of the model, especially when compared to using a much larger percentage of similar individuals in the training data.

In conclusion, this simulation study has shown that by training the model on only the individuals in the training data that are the most similar (on a set of a priori chosen covariates) to the individual that we are predicting, we can improve the predictive performance of the model, as shown by the estimated Brier score. The joint model has better

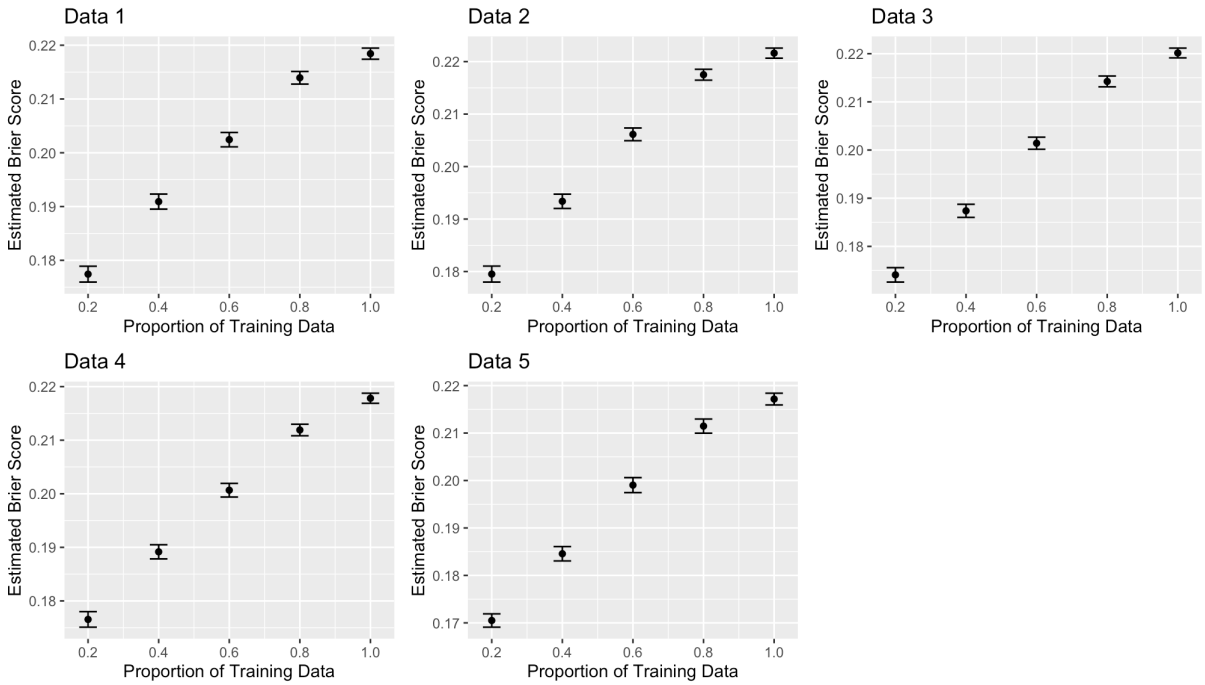


Figure 2.1: The estimated Brier score of the five simulated datasets for Scenario 1, with an approximate 95% confidence interval around each estimate. We can see that 0.2 results in the smallest estimated Brier score across all datasets for the five values considered for the proportion of training data.

prediction as the dataset becomes more targeted and there is less noise in the data for making a targeted prediction. However, the subpopulation proportion that minimizes the estimated Brier score varies depending on the dataset. For Scenario 1, it is shown that 20% of the original training data is the best subpopulation size among the proportion of training set size considered; for Scenario 2, it is 40% of the original training data. Note that the estimated Brier score of Scenario 2 is lower than that of Scenario 1 across all five values considered for the subpopulation proportion. This is because in Scenario 2, we use longitudinal information available up to time $t = 1$ and make a prediction for time $u = 3$, whereas in Scenario 1, we use longitudinal information available up to time $t = 1$ and make a prediction for a time period further into the future, which is $u = 4$. Therefore, as expected, the accuracy of the model is better for $u = 3$, or Scenario 2.

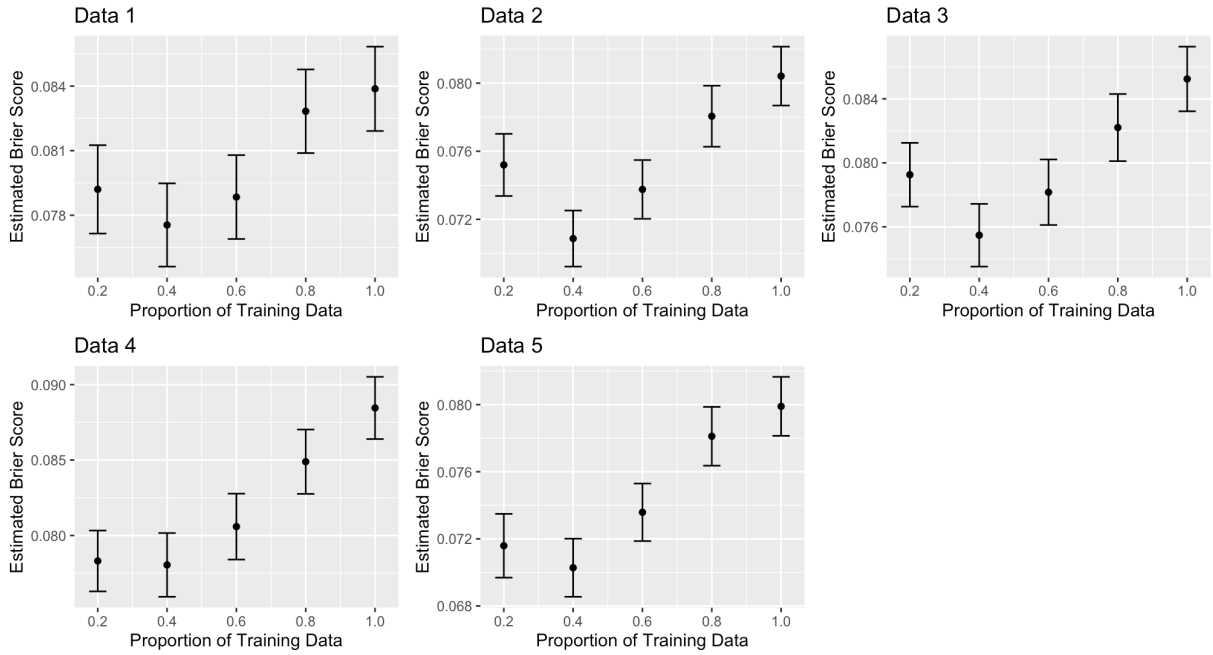


Figure 2.2: The estimated Brier score of the five simulated datasets for Scenario 2, with an approximate 95% confidence interval around each estimate. We can see that 0.4 results in the smallest estimated Brier score across all datasets for the five values considered for the proportion of training data.

Table 2.3: A table showing the average of the estimated dynamic prediction error of joint models for different subpopulation sizes across five simulation runs for each simulation scenario, displayed in Figures 2.1 and 2.2. The bolded value indicates the smallest estimated Brier score in each simulation case.

Scenario 1		**Scenario 2**	
M_p	**Estimated Brier score**	M_p	**Estimated Brier score**
0.2	**0.1756**	0.2	0.0767
0.4	0.1891	**0.4**	**0.0744**
0.6	0.2019	0.6	0.0770
0.8	0.2138	0.8	0.0812
1.0	0.2190	1.0	0.0836

2.4 Data Analysis

We apply our proposed algorithm in Section 2.2.4 to the dataset available in the eICU Collaborative Research Database, which covers patients admitted to critical care units in 2014 and 2015 throughout the continental United States, containing 200 859 unit encounters for 139 367 unique patients. Noting some patients have more than one hospital stay in the database, or even multiple ICU stays within a single hospital stay, we focus only on the first hospital stay of a given patient, and the first unit (ICU) stay within that first hospital stay, in order to get an independent dataset with each patient having only one record. This narrows down the number of patients to 113 782. From this dataset, we consider a subset of 77 330 patients who were in the ICU for at least 48 hours so that we have enough longitudinal information to make a prediction of the survival probability given biomarker measurements for the next 24 hours following their first day in the ICU. We require the first 24 hours in the ICU in order to allow the severity of illness score, APACHE IVa (Zimmerman et al., 2006) to be calculated, which is a commonly used prediction tool in critical care medicine. In our data analysis, we consider the patients with complete APACHE IVa score diagnosed with sepsis - a systemic inflammatory reaction to infection that may lead to inflammation in organs remote from the initial wound and eventually lead to end-of-organ dysfunction and failure (Bone et al., 1997) - which contains 8996 patients. This process is illustrated in Figure 2.3.

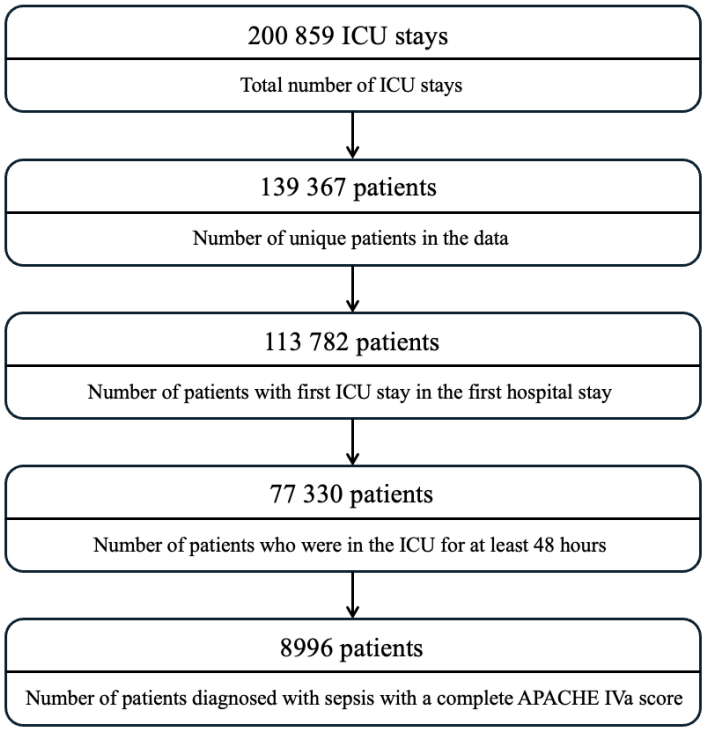


Figure 2.3: Flow chart of cohort selection from the eICU database for data analysis.

We randomly select 2000 patients from this dataset for computational efficiency reasons to demonstrate that we can improve the dynamic prediction of joint longitudinal-survival models using a similarity-based approach. We use the following features to fit a personalized joint model:

- Demographic data of the patient: age and gender
- Information on the hospital: hospital location
- Patient status: unit discharge status, signifying whether the patient left the unit alive or expired; unit discharge offset, or the number of minutes from unit admit time that the patient was discharged from the unit; APACHE IVa score, a predictor for mortality that incorporates patient information collected within the first 24 hours of the ICU stay, including both demographic information about the patient and physiologic data (Zimmerman et al., 2006)

- Vital signs measurements: SpO2 and respiration; note that these values are available every five minutes, representing five-minute median values in the eICU database, but we take the average of each of the vital sign measurements every three hours to obtain a smoother collection of these vital signs for each patient.

Then, the following joint model is fit using the dataset:

$$\begin{aligned} h_i(t|\mathbf{M}_i(t)) = h_0(t)\exp\Big\{&\gamma_1 w_{1i} + \gamma_2 w_{2i} + \gamma_3 w_{3i} + \gamma_4 w_{4i} + \gamma_5 w_{5i} + \gamma_6 w_{6i} \\ &+ \gamma_7 w_{7i} + \gamma_8 w_{8i} + \gamma_9 w_{9i} + \gamma_{10} w_{10i} + \gamma_{11} w_{11i} + \gamma_{12} w_{12i} + \gamma_{13} w_{13i} \\ &+ \gamma_{14} w_{14i} + \gamma_{15} w_{15i} + \gamma_{16} w_{16i} + \alpha_1 \mathbf{m}_{1i}(t) + \alpha_2 \mathbf{m}_{2i}(t)\Big\}, \end{aligned}$$

where $\mathbf{m}_{1i}(t)$ is the standardized measurement of SpO2 at time t such that

$$\mathbf{m}_{1i}(t) = \beta_{0(1)} + b_{0i(1)} + (\beta_{1(1)} + b_{1i(1)}) \times time,$$

and $\mathbf{m}_{2i}(t)$ is the standardized measurement of respiration at time t such that

$$\mathbf{m}_{2i}(t) = \beta_{0(2)} + b_{0i(2)} + (\beta_{1(2)} + b_{1i(2)}) \times time,$$

and w_{1i}, w_{2i}, w_{3i}, and w_{4i} signify gender, age (standardized), APACHE IVa score (standardized), and hospital location of patient i respectively, with the response variable being the number of minutes from the first 24 hours in the ICU that the patient was discharged from the unit. w_{5i}, w_{6i}, and w_{7i} represent the three functional principal components that explain at least 95% of the variation in the data for SpO2 and w_{8i}, w_{9i}, and w_{10i} represent the three functional principal components that explain at least 95% of the variation in the data for respiration, for the first 24 hours since entering the ICU. w_{11i}, w_{12i}, and w_{13i} represent the maximum, minimum, and median of the SpO2 measurements of the first 24 hours, and w_{14i}, w_{15i}, and w_{16i} represent the maximum, minimum, and median of the respiration measurements of the first 24 hours. Note that in the three FPCs to explain the variation in the data for SpO2 and respiration for the first 24 hours of ICU stay are baseline covariates and are also being used to calculate similarity. FPCs to explain the variation in the biomarkers for the next 24 hours after the first 24 hours are being used in similarity calculation only. More details on the covariates being used for the cosine similarity metric are provided below.

To measure the similarity between the patients for fitting a personalized predictive model, we use all of the following predictors, which can be categorized into two groups.

The first set of predictors is the baseline covariates and the information we gather from the first 24 hours of being admitted into the ICU. These include each patient's standardized age, gender, hospital location, teaching hospital status, standardized APACHE IVa score, the first three functional principal components and the maximum, minimum, and median of SpO2 and respiration in the first 24 hours. The second set of predictors are the information we gather from the biomarker measurements in the next 24 hours since being admitted to the ICU for one day (i.e., hours 24+ to 48): the three functional principal components for each of the vital sign measurements that explain at least 95% of the variation in the data for the next 24 hours after the first day since admitted to the ICU. We use the biomarker measurements for 24 hours after the first day of being admitted to the ICU only for similarity calculation because we want to make a prediction for the survival rate at some future time u given biomarker information up to t, which is 24 hours in this data analysis. We consider $20\%, 40\%, 60\%, 80\%$, and 100% of the population of the training data as the subpopulation size to compare the estimated Brier score of the joint model for predicting the probability of survival at 5000 minutes (approximately 83 hours or 3.5 days) after the first 24 hours in the ICU, given longitudinal measurements from hour 24+ to 48 in the ICU.

2.4.1 Results

Table 2.4: The estimated Brier score for five different subpopulation proportions for the eICU data. When the subpopulation proportion is 0.6, the estimated Brier score is minimized. The columns "Lower" and "Upper" represent the 95% confidence interval of the estimates.

M_p	**Estimated BrS**	**Lower**	**Upper**
0.2	0.0978	0.0936	0.1020
0.4	0.0659	0.0629	0.0688
0.6	**0.0653**	0.0623	0.0682
0.8	0.0668	0.0631	0.0704
1.0	0.0661	0.0626	0.0700

From Figure 2.4, we can see that the model performs relatively poorly in estimating the survival probability at time 5000 minutes with longitudinal measurements until 1440 minutes (i.e., 24 hours) after the first day in the ICU when the proportion of the data is 20% of the training data - the estimated Brier score is approximately 0.0978. However, as

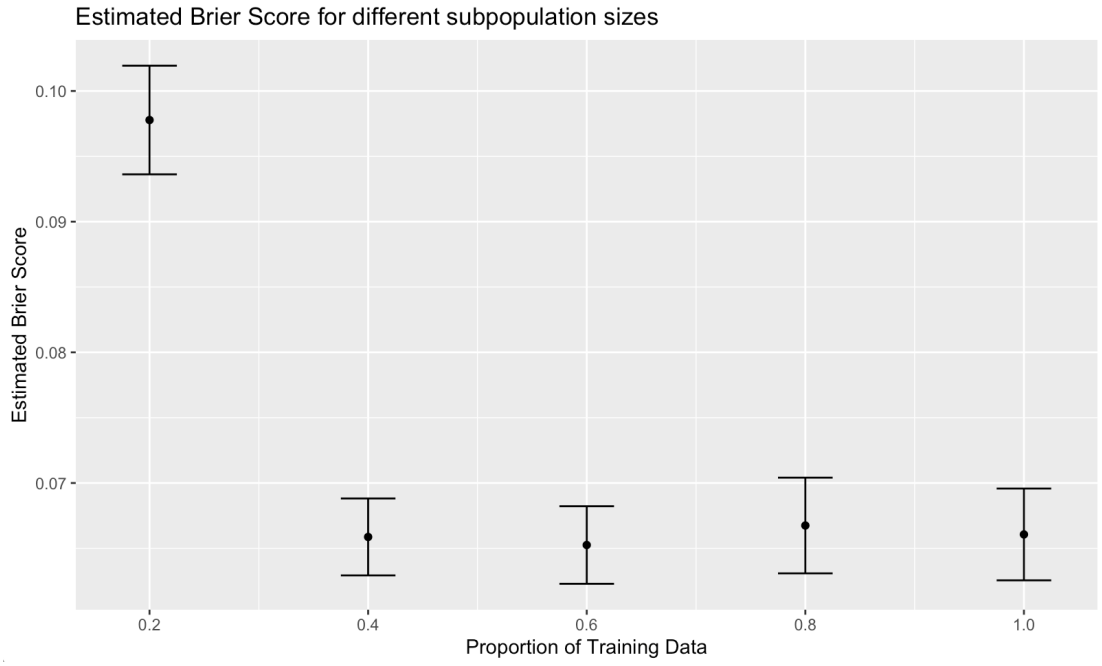


Figure 2.4: The estimated Brier score of the eICU dataset across five different subpopulation proportions. We can see that 0.6 results in the smallest estimated Brier score.

shown in Table 2.4, the estimated Brier scores for 40%, 60%, 80%, and 100% of the data are 0.0659, 0.0653, 0.0668, and 0.0661, respectively, showing that the model's predictive performance is maximized when we are fitting the joint model when the subpopulation proportion is 0.6. Note that these estimated values are not comparable to the results from the simulation study - not only is the model from the data analysis more complex, but the prevalence in the dataset also differs. In the simulation study, around 48% of the patients in the data experienced the event, whereas only about 11% of the patients experienced the event in the data analysis. While the pattern of the estimated Brier score plotted against M_p is different between the simulation study and the data analysis, the conclusion is the same in that the predictive performance of joint models with longitudinal and time-to-event data can be improved by fitting the model on a more personalized data - 20% and 40% being the optimal subpopulation proportions from the two cases in the simulation study, and 60% for the data analysis, though certainly the improvement with the subpopulation approach (compared to using the entire training data) is not as dramatic as seen in the two simulation results.

In the validation step, we evaluate the predictive performance of the joint model with

personalized predictive modeling for the proportion of training data equal to 60%, which was shown to have the most optimal outcome from the data analysis. We randomly sample n patients from the hold-out validation sample and apply the algorithm outlined in Section 2.2.4, with the proportion M_p set to 60%. The sample sizes considered are $750, 900, 1500,$ and 2000. For $n = 750$ and $n = 900$, we perform 3-fold CV repeated 10 times since the sample size is small; for $n = 1500$ and $n = 2000$, we perform 5-fold CV repeated 10 times. When sampling the data from the hold-out validation set, we used stratified sampling so that the event rate is similar to that of the training-testing dataset - for a sample size n, $0.11 \times n$ experience the event and $0.89 \times n$ do not experience the event. The predictive performance across four different sample sizes for $M_p = 0.6$ is equal to $0.0785, 0.0747, 0.0658,$ and 0.0654, respectively, for $n = 750, 900, 1500,$ and 2000. As the sample size increases, the estimated Brier score decreases and thus improves the predictive performance of the model, as expected.

Note that the estimated Brier score in the validation sample, at least for the larger sample sizes, is quite close to that of the testing and training sample results reported earlier in this section. While this is not very common in other statistical analyses involving hold-out validation set and training-testing set - the model trained on the training-testing set is used to fit the validation data, so the accuracy in the validation step is usually lower - it makes sense in the context of personalized predictive modeling that the performance measure is similar when the validation sample size is equal to the size of the training-testing data. This is because by the nature of personalized predictive modeling, we fit a new personalized model using $K - 1$ fold data for every patient in the remaining fold for cross-validation. Thus, we are not using a universal model trained on training data from the cross-validation step and applying this model to the validation data, meaning that the predictive performance should be similar in the training-testing stage and validation stage. This "closeness" phenomenon would not be seen if we had a true external validation sample, such as with ICU patients in a different country.

2.5 Discussion

In this work, we have shown how we can improve the dynamic predictive performance of joint models for longitudinal and time-to-event data using our proposed similarity-based modeling approach. Joint models are commonly used in follow-up studies and have three notable features that make them quite attractive for model predictions: first, they use the entire longitudinal information available up to a time point instead of using only the most recent measurement; second, they allow for dynamic predictions by updating the survival

probability as more longitudinal measurements become available; third, they allow for individualized predictions by incorporating random effects into the model. Our goal is to improve the predictive accuracy of joint models by personalizing the training data to the individual we are making the predictions for, using our proposed method outlined in Section 2.2.4. Note that the cross-validation process to identify the optimal M_p was a complex extension of the process outlined in Lee et al. (2015), but applying the similarity-based approach to train the joint model on personalized data to improve the dynamic prediction of joint models is the novelty introduced here.

More specifically, we use patient similarity to identify the individuals in the training data that are the most similar to the individual that we want to predict for, and train the model on $M_p \times 100\%$ of the most similar training data. The optimal size of this personalized subset of the data varied depending on the simulation scenario, but ultimately, we can decrease the estimated Brier score and thus improve the predictive performance of the joint models by training the model on a subpopulation of the dataset. Our simulation study with repeated 5-fold cross-validation showed that 20% was the best subpopulation proportion among the considered values for Scenario 1, and 40% was the best subpopulation proportion among the considered values for Scenario 2. In other words, our research can indicate that some advancements and improvements can be seen in the dynamic predictive performance of joint models by taking a more targeted sample from the training data - this agrees with past research in patient similarity literature.

Note that Scenario 1 in the simulation study shows that as we decrease the proportion of the training data that we are fitting the joint model - that is, as our dataset becomes more targeted with less noise - the predictive performance continues to improve. We do not show the results for proportions smaller than $M_p = 0.2$ because it leads to optimization errors when fitting a linear mixed-effects model. In our simulation study, we train the model on a personalized dataset of size $m = 360$ to estimate the Brier score when $M_p = 0.2$. However, when this value is smaller, it leads to optimization errors for some datasets. In other words, the optimal value of 20% is specific to the dataset simulated in our study - for a different dataset with more samples, for example, the optimal M_p might be less than 20% of the original training data since a smaller proportion of the data may not lead to optimization errors if the original training data size is sufficiently large.

In the data analysis section, we apply our algorithm to a real dataset, specifically the eICU database, to demonstrate how our method can be useful in real-life scenarios by effectively increasing the predictive performance of joint models on patients in intensive care units. The eICU database offers rich information in both the patients' demographic information and lab and vital sign results during the ICU stay - we increase the complexity of the joint model from the joint models in the simulation study by incorporating more co-

variates. The point estimate of the Brier score suggests that it is minimized when $M_p = 0.6$ compared to $M_p = 0.1$, and thus we can improve the predictive performance of joint models by training the model on a targeted subpopulation of the training data. However, the magnitude of this improvement is considerably smaller than what was observed in our two simulation scenarios. We suspect that this may be due to the correlation structure among the predictors, which are not present in the simulated datasets, or a different functional form of the longitudinal predictors must be specified in the joint model (such as incorporating the slope of the time-dependent features, instead of just using the current value of the underlying longitudinal trajectory).

There are several areas of future work in this study. First, as mentioned above, our results show more favourability in terms of model predictive performance in the simulation results than in the data analysis, in that there is a greater improvement when the joint model is trained on personalized data for the simulation study compared to the data analysis; thus, further investigation is necessary. Perhaps a different subset of patients in the eICU dataset, such as those with circulatory system disease, may show results that more closely align with the simulation study. It would also be worthwhile to investigate the correlation structure among predictors and the functional form of the longitudinal predictors in the joint model for the data analysis and evaluate how these factors influence the effectiveness of the similarity-based approach. Second, we only consider one measure of predictive accuracy, namely the Brier score. We hope to study the log loss and calibration, such as integrated calibration index (Austin and Steyerberg, 2019), to evaluate the predictive performance of the models when using other predictive performance measures. In addition, we will consider the scaled Brier score to account for outcome prevalence, which allows for more meaningful comparisons across different datasets. Third, we have considered very specific values of t and u in our simulation study. The choices for t and u are context-dependent. For example, relatively small values of t and u are used with the eICU database, as clinical decisions must be made fast before patient deterioration. In contrast, other applications, such as monitoring cancer biomarkers to predict recurrence, may involve much longer time horizons, ranging from months to years. Therefore, future work will extend the simulation study to incorporate a wider range of t and u values in order to assess whether these choices influence tuning for the optimal M_p. Furthermore, we only work with the cosine similarity metric to quantify the similarity between the individuals in the training data and the testing data - it may be worthwhile to study how the predictive accuracy of joint models changes under the similarity setting with different approaches to measure similarity, such as Mahalanobis distance or clustering methods. Finally, we hope to study the performance of joint models with the introduction of missing data in both biomarkers and time-invariant covariates. Currently, in our simulation study, we assume

that all measurements are available - we would like to study how our results change with missing data, both for the type of missing and when we use different approaches to deal with missing data, such as different imputation methods.

Finally, as mentioned earlier, a big challenge of our work was the computational burden. We have taken measures to increase the computational efficiency, such as fitting the Bayesian model coded in C++ and parallelizing our algorithm. However, since we fit a personalized model for each individual in the testing data for every fold in the repeated 5-fold cross-validation, we fit thousands of models to tune the optimal subpopulation size - this increases the computational time dramatically. Fortunately, the computation will not be as extensive in real-life scenarios when we want to predict the survival probability for a given new patient. We only have to implement our procedure on one available training dataset to tune for M_p that will give the best predictive performance specific to that dataset, under the assumption that the new individual is drawn from the same underlying population as the available training data. Afterwards, to predict the survival probability of a new patient, we fit a joint model on a personalized dataset containing the top $M_p \times 100\%$ of the most similar patients from the available training dataset. This will be quicker than fitting a joint model on all of the available training data, since fitting a joint model when $M_p < 1$, thus resulting in a smaller size of the training dataset, will decrease the computation time.

Chapter 3

A New Trained Supervised Method for Calculating Patient Similarity

3.1 Introduction

With the rapid development of technology, electronic health record (EHR) data has been readily available for predictive modeling. EHR data offer a huge amount of information - still, previous studies have shown that personalized predictive modeling (PPM) that uses a carefully chosen subset of the data that is more homogeneous leads to better predictive results, particularly regarding model discrimination performance, compared to using the entire data that contains more heterogeneous patients (Xu et al., 2019; Krikella and Dubin, 2025). This was also shown to be the case with joint modeling for longitudinal and time-to-event data in Chapter 2.

A key challenge in this area of research is quantifying the similarity between patients using their features in a given dataset to identify those in the training data that are considered to be similar to each patient in the testing data. Calculating similarity using distance-based metrics can be classified into two broad areas, where one calculates similarity using a single similarity measure, such as Euclidean distance or Mahalanobis distance, and the other uses a different feature similarity calculation for each data type and then subsequently combines the measurements (Wang et al., 2021).

To calculate the similarity between patients, recent studies have incorporated different weights for each feature, or predictor in the EHR data, to improve the predictive performance of the model. For example, Wang et al. (2019) calculate the feature similarity for

four categories - age, sex, laboratory test, and disease diagnoses - and combine them into a single metric using predetermined weights of 0.1, 0.1, 0.4, and 0.4, respectively. The weights are determined experimentally from their previous study, where they proposed three weighting schemes for the four categories - 0.25, 0.25, 0.25, and 0.25 for the first scheme, 0.1, 0.1, 0.4, and 0.4 for the second scheme, and 0.1, 0.1, 0.6, and 0.2 for the third scheme - and the one with the best predictive performance is chosen (Huang et al., 2019). Gu et al. (2017) use an adopted weighted Euclidean distance to calculate the similarity for both continuous and discrete variables. The weights are derived using the genetic algorithm, which adopts the idea of natural selection and natural genetics to learn the weights based on sample cases. Gottlieb et al. (2011) categorize the features into different groups, calculate the distance between each feature with different metrics depending on the group, and combine them into a single similarity measure using a weighted geometric mean. Campillo-Gimenez et al. (2013) consider an exclusive OR-based patient similarity with the logistic regression model, where they use the Wald statistic of each predictor variable in the logistic regression to determine the weights of each feature. Li et al. (2025a) group the features by type, such as demographics, comorbidities, lab tests, and procedures and use predefined empirical weights, expert-assigned weights, and different grid-search weight combinations before combining them into a single scalar similarity metric. Lambert et al. (2025) use a weighted cosine similarity metric to build a patient network using the Markov Cluster Algorithm, where they assign different weights to the features. The cosine similarity is weighted by Inverse Document Frequency (IDF) (Sparck Jones, 1972), Wu & Palmer measure (Wu and Palmer, 1994), or by the Lin measure (Lin et al., 1998). However, these weights are not learned or optimized but are calculated mathematically from the hierarchical structure of the feature and their frequency. Some papers approach calculating patient similarity using feature selections, where some selected features have a weight of 1 and other unselected features have a weight of 0. For example, Suo et al. (2018) calculate the similarity using the patients' predicted future feature values and apply group lasso to determine these features. Pai and Bader (2018) use machine learning algorithms that use a network of features to tune for the optimal selection of the features that have the highest discriminative power.

We hope to contribute to this area of statistics by developing a new method to calculate patient similarity. In particular, we introduce a new supervised measure of patient similarity that uses a regularization technique to develop a new weighted cosine similarity metric. Cosine similarity metric (CSM) is an effective unsupervised measure of similarity, expressed as

$$CSM(S_i, S_j) = \frac{\sum_{p=1}^{P} S_{ip} \times S_{jp}}{\sqrt{\sum_{p=1}^{P} S_{ip}^2}\sqrt{\sum_{p=1}^{P} S_{jp}^2}},$$

where P is the total number of features in the vector for each participant $\mathbf{S}_i$ and $\mathbf{S}_j$ (Li and Han, 2013). However, when the data contain predictors that are highly correlated with each other, CSM can be overly biased and does not account for the number of shared features between the two vectors (Li and Han, 2013). For example, consider the following three vectors, $A = (1, 2, 0)^T, B = (0, 1, 0)^T$, and $C = (1, 2, 2)^T$. When we consider the similarity between A and B and A and C, we might say that C is more similar to A than B since the first two features of A and C are equal. However, $CSM(A, B) = 0.894$ and $CSM(A, C) = 0.745$, and so A and B are more similar in terms of the cosine similarity metric (Li and Han, 2013).

Our method in this chapter adapts the soft similarity measure developed by Sidorov et al. (2014) and updates the existing cosine similarity metric with weights derived from regularization, further described in Section 3.2.1. Unlike existing weighted cosine similarity methods, our proposed weighted CSM assigns supervised weights and does not involve categorizing different types of predictors into groups to assign weights depending on the group characteristic. Thus, we do not require the subject-matter expert's opinion to create clusters of different features. Instead, the weights are derived from the predictor's relationship with the outcome of interest by fitting a model with all potential predictors and then identifying those that are correlated with the outcome. Furthermore, our method effectively assigns weights of 0 to less significant features and assigns flexible weights to the predictors that are selected for the model. In Section 3.2, we introduce the algorithm to calculate the weighted CSM with details on the regularization, along with the algorithm to implement PPM with the new similarity measurement. In Sections 3.3 and 3.4, respectively, we illustrate the improved predictive performance of a statistical model with our proposed method through a simulation study and data analysis using the publicly-available eICU Collaborative Research Database (Pollard et al., 2018). We conclude with a discussion and some suggestions on future work in Section 3.5.

3.2 Methods

3.2.1 Weighted Cosine Similarity Metric

We propose a new weighted cosine similarity metric to quantify the similarity between two models in a vector space model, which can be implemented in personalized predictive modeling (PPM) as well to measure the similarity between two individuals (Sidorov et al., 2014):

$$CSM_w(S_i, S_j) = \frac{\sum_{p=1}^{P} w_p \times S_{ip} \times S_{jp}}{\sqrt{\sum_{p=1}^{P} w_p \times S_{ip}^2}\sqrt{\sum_{p=1}^{P} w_p \times S_{jp}^2}}.$$

We introduce two new approaches to determine w_p that allow for curvilinear or non-linear representations of some variables in the linear predictor. The regularization technique that we use to compute w_p is the relaxed adaptive group lasso, which enables determining w_p supervised without requiring further knowledge about the types of predictors or expert knowledge, as previously published (Wang et al., 2019). Note that while relaxed adaptive group lasso is not a new contribution to the literature, to our knowledge, it has not been used before to compute the weights of the weighted CSM to be used in the context of PPM.

Since we have a large number of predictors to calculate the similarity between the participants in the data, it is common to consider a regularization method to retain only a subset of the predictors. We consider the relaxed adaptive group lasso for several reasons. First, lasso (least absolute shrinkage and selection operator) is a generally effective regularization approach to reduce the dimension of the parameter space in modeling (often high-dimensional) data, particularly using generalized linear models (Hastie et al., 2009). However, the model selection result could be inconsistent, mostly because lasso applies the same amount of shrinkage to each coefficient. As a solution, adaptive lasso is introduced to allow for different amounts of shrinkage on different regression coefficients, thereby reducing bias (Zou, 2006). Adaptive group lasso is a natural extension of adaptive lasso, best suited for situations where it is desirable to select the predictors in a grouped manner instead of selecting them individually. For example, a collection of indicators representing a categorical predictor with three levels should be included or excluded from the model as a whole, rather than having only some of its indicator variables selected. It allows for different tuning parameters to be used for different groups of predictors to shrink a subset of coefficients to zero (Wang and Leng, 2008). Mathematically, let $y \in \mathbb{R}^n$ denote the outcome, or response, vector and $X \in \mathbb{R}^{n \times p}$ the design matrix. The p predictors are partitioned into m non-overlapping groups $\{G_1, \ldots, G_m\}$, where $G_g \subset \{1, \ldots, p\}$ and $|G_g| = p_g$. Let $\beta = (\beta_{G_1}^\top, \ldots, \beta_{G_m}^\top)^\top \in \mathbb{R}^p$ denote the corresponding coefficient vector. β is estimated using the adaptive group lasso as

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \sum_{g=1}^{m} w_g \|\beta_{G_g}\|_2 \right\},$$

where the adaptive weights are defined as

$$w_g = \left(\|\tilde{\beta}_{G_g}\|_2 \right)^{-\gamma}, \quad \gamma > 0,$$

where $\tilde{\beta}$ is a group lasso estimator (Wang and Leng, 2008).

Finally, we extend this to relaxed adaptive group lasso, which can reduce estimation bias while maintaining sparsity by refitting a model on the predictors selected by the adaptive group lasso (Meinshausen, 2007).

Let the set of selected groups from adaptive group lasso be defined as

$$\mathcal{A} = \left\{ g : \|\hat{\beta}_{G_g}\|_2 > 0 \right\}.$$

Conditioning on the selected group set $\mathcal{A}$, we obtain the relaxed adaptive group lasso estimator by solving

$$\hat{\beta}^{\text{RAGL}} = \arg\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda_1 \sum_{g \in \mathcal{A}} w_g \|\beta_{G_g}\|_2 \right\} \quad \text{subject to } \beta_{G_g} = 0 \ \ \forall g \notin \mathcal{A},$$

where $0 \leq \lambda_1 \leq \lambda$. A common special case is $\lambda_1 = 0$, corresponding to refitting the model using only the predictors in $\mathcal{A}$. Applying the relaxed adaptive group lasso strategy, we introduce two new weighted CSM methods as follows:

1. Created an augmented design matrix $\mathbf{X}^*$ that contains the original values for the predictors in the data and non-linear transformations of the predictors. For example, this could be an augmented design matrix of the original predictors, squared values of the predictors, and/or the exponential of the values of the predictors; we do not limit the number of transformations, although we consider two transformations of the original predictors in the simulation study in Section 3.3. This is to account for real-life scenarios where we do not know the true outcome model, and some variables have curvilinear or non-linear representations in the linear predictor. Of course, with greater knowledge of the underlying relationships (say via historical knowledge) between the predictors and the response, we could be more precise in how we construct $\mathbf{X}^*$.

2. Standardize each column of $\mathbf{X}^*$.

3. Define the grouping structure for the augmented design matrix $\mathbf{X}^*$. Let P denote the number of original predictors. For each original predictor $p \in \{1, \ldots, P\}$, define a group $G_p \subset \{1, \ldots, p^*\}$ that contains the column index corresponding to predictor p and the column indices of all its associated nonlinear transformations in $\mathbf{X}^*$, where p^* denotes the total number of columns in $\mathbf{X}^*$. Thus, the augmented predictors are partitioned into $G = P$ non-overlapping groups $\{G_1, \ldots, G_P\}$, with each group corresponding to one original predictor and its transformations.

4. Perform relaxed adaptive group lasso as described in Section 3.2.1.

5. Using the estimated coefficients from the relaxed adaptive group lasso, we define predictor-specific weights w_p for $p = 1, \ldots, P$ to construct the weighted cosine similarity metric. Let the set of selected predictor groups at tuning parameter value λ be defined as
$$\mathcal{G}_\lambda = \left\{ p \in \{1, \ldots, P\} : \|\hat{\beta}_{G_p,\lambda}\|_2 > 0 \right\}.$$

   (a) **Method 1 (weight01 CSM).** Define
$$w_p = \begin{cases} 1, & p \in \mathcal{G}_\lambda, \\ 0, & p \notin \mathcal{G}_\lambda. \end{cases}$$
   That is, all predictors whose corresponding groups are selected by the adaptive group lasso are weighted equally, while unselected predictors receive zero weight.

   (b) **Method 2 (weightbeta CSM).** For each selected predictor group $p \in \mathcal{G}_\lambda$, define a group-level importance score
$$s_p = \|\hat{\beta}_{G_p,\lambda}\|_2 = \left(\sum_{q \in G_p} \hat{\beta}_{q,\lambda}^2 \right)^{1/2}.$$
   The predictor-specific weights are then given by
$$w_p = \begin{cases} 0, & p \notin \mathcal{G}_\lambda, \\ \dfrac{s_p}{\sum_{p' \in \mathcal{G}_\lambda} s_{p'}}, & p \in \mathcal{G}_\lambda. \end{cases}$$

For convenience, especially in the simulation section, we refer to Method 1 as "weight01 CSM", since it is a weighted CSM method with a weight of 1 for the selected predictors

and a weight of 0 for the non-selected predictors, and Method 2 as "weightbeta CSM" since it is a weighted CSM method where the nonzero weights are calculated using the beta estimates. To compare the two new proposed methods, we refer to "standard CSM" as the regular CSM introduced in Section 3.1 that uses all available predictors with an equal weight of 1 across all predictors.

3.2.2 Algorithm for PPM

In this section, we describe our new proposed algorithm to improve the predictive performance of generalized linear models with the weighted CSM calculation described in Section 3.2.1.

1. Split the data into training and testing set (TrTe data) and validation set (Validation data), where $100\times q\%$ of the data is used as TrTe data and the remaining $100\times(1-q)\%$ of the data is used as the Validation data.
2. Split the TrTe data into K folds for K-fold cross-validation (CV).
3. Let k be the hold-out fold for testing data, and the other $K-1$ folds be the training data.
4. Consider a grid of M_p values to tune M_p.
   (a) For an index participant j in the testing data, apply the weighted CSM algorithm described in Section 3.2.1 to the training dataset.
   (b) Sort the participants in the training dataset in descending order by the similarity metric.
   (c) Fit a generalized linear model on the top $M_p \times 100\%$ of the sorted training data and estimate the outcome for participant j.
   (d) Repeat Steps 4(a) - 4(c) for each participant j in the testing data and calculate the model performance measures of interest, such as discrimination, calibration, or Brier score.
5. Repeat Steps 3-4 for each k in K folds. Average over the values of performance metrics over the K folds.

6. Repeat Steps 2-5 for all K folds and repeat the entire K-fold cross-validation procedure R times. This leads to a total of $R \times K$ evaluations, each using a different training-testing split. The performance metric is averaged over all $R \times K$ evaluations, and the optimal M_p is selected as the value achieving the best averaged performance according to a pre-specified evaluation metric. In our simulation study and data analysis, we use Brier score as the evaluation metric since it evaluates overall predictive accuracy, incorporating both discrimination and calibration (Murphy, 1973).

7. We validate the model using the Validation data obtained in Step 1.

   (a) From the Validation data, create a bootstrap sample of the same length by sampling the data rows with replacement.

   (b) Randomly split the bootstrap sample into 80% training sample and 20% testing sample.

   (c) For each participant j in the testing sample, measure the similarity between participant j and all the participants in the training sample.

   (d) Sort the participants in the training sample by descending order of similarity. Using the winning M_p value obtained from Steps 2-6, obtain a subpopulation of the training set that is the most similar to participant j from the testing set.

   (e) Evaluate the predictive performance by fitting a generalized linear model on the subpopulation obtained in Step 7(c), for each j in the testing sample.

   (f) Repeat Steps 7(a)-7(e) B number of times to obtain a bias-corrected and accelerated (BCa) bootstrap confidence interval around the performance measure of the validation sample.

3.3 Simulations

In this section, we describe a simulation study conducted on various datasets to investigate if using our proposed weighted CSM can improve the predictive performance of models in PPM. We follow the proposed algorithm outlined in Section 3.2.2 to implement a 5-fold cross-validation repeated twenty times to tune for the best subpopulation proportion for the dataset generated in this section. We consider the following subpopulation proportion sizes M_p - $0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$, and 1 - and use both discrimination and calibration to evaluate the predictive performance of the model. Note that the subpopulation proportion of 1 is equivalent to using all available data in the training data to fit

the same global model for all participants in the testing data. In the validation step, we use $B = 500$ bootstrap samples to generate the bootstrap confidence interval around the performance measures of the validation sample. Our algorithm is tested on five different datasets generated.

Area Under the Receiver Operating Characteristic Curve (AUROC) and Area under the Precision-Recall Curve (AUPRC) are considered to assess discrimination. Receiver Operating Characteristic Curve displays the trade-off between sensitivity and specificity across all possible threshold values, and AUROC is the area under this curve that ranges between 0.5 and 1 (Ozenne et al., 2015). AUPRC is an alternative approach to measuring discrimination that uses the precision-recall curve that shows the trade-off between precision and sensitivity (also called recall), where precision is the number of true positives divided by the total number of predicted positives. Unlike AUROC, possible values of AUPRC depend on the prevalence of the data and decrease towards 0 as the prevalence gets lower (Ozenne et al., 2015). To evaluate discrimination, we calculate AUPRC for datasets with lower prevalence and AUROC for datasets with higher prevalence because AUPRC is a more informative measure of the performance of the model with highly imbalanced data (Hancock et al., 2023). For both measures, high values indicate better predictive performance. We use the Integrated Calibration Index (ICI) to assess calibration, which is a measure of moderate calibration that represents the weighted average absolute difference between the observed and predicted probabilities (Austin and Steyerberg, 2019). That is, it measures the weighted difference between observed and predicted probabilities, where the observations are weighted by the empirical density function of the predicted probabilities. Finally, to consider a combination of both discrimination and calibration, we use the Brier score (BrS) which is an overall predictive performance measure that can be decomposed into both discrimination and calibration (Murphy, 1973).

We begin with Section 3.3.1 where we describe in detail how the different datasets are generated. In Section 3.3.2, we discuss the simulation results comparing our proposed methods, weight01 CSM and weightbeta CSM, with standard CSM.

3.3.1 Setup

We consider six cases for this simulation study, coded in R version 4.4.0 (R Core Team, 2023) using RStudio (Posit team, 2023). In all six cases, we generate the data using a sample size of $n = 10000$ with 20 predictors. 15 of these predictors are continuous variables generated using an independent normal distribution of $N(0, 2^2)$ and five are binary predictors with an independent Bernoulli distribution of $Bernoulli(0.5)$. Among these 20

predictors, seven continuous predictors and three binary predictors are used to generate a binary outcome Y for each sample in the data with $\varepsilon \sim N(0, 2.5^2)$. We consider 0.1 and 0.5 as options for the outcome prevalence in the dataset. The value of 0.1 is motivated by the observed mortality prevalence in the ICU dataset, while 0.5 represents a higher-prevalence scenario for comparison. In addition, we examine three settings (weak, moderate, and strong signal) to reflect varying strengths of association between the predictors and the outcome. Specifically, the linear predictors for the six cases are as follows:

Case 1: Prevalence = 0.1, weak signal

$$z = -2 + 1x_1 + 0.5x_2 - 0.5x_3 - 1x_4 + 0.5x_5 - 0.5\exp(x_6) - 1x_7^2 - 1x_{16} - 1x_{17} + 0.5x_{18} + \varepsilon$$

Case 2: Prevalence = 0.1, moderate signal

$$z = -3 + 3x_1 + 1.5x_2 - 1.5x_3 - 3x_4 + 1.5x_5 - 1.5\exp(x_6) - 3x_7^2 - 3x_{16} - 3x_{17} + 1.5x_{18} + \varepsilon$$

Case 3: Prevalence = 0.1, strong signal

$$z = -4.5 + 5x_1 + 3x_2 - 3x_3 - 5x_4 + 3x_5 - 3\exp(x_6) - 5x_7^2 - 5x_{16} - 5x_{17} + 3x_{18} + \varepsilon$$

Case 4: Prevalence = 0.5, weak signal

$$z = -0.75 + 1x_1 + 0.5x_2 + 0.5x_3 + 1x_4 + 0.5x_5 - 0.5\exp(x_6) + 1x_7^2 - 1x_{16} - 1x_{17} + 0.5x_{18} + \varepsilon$$

Case 5: Prevalence = 0.5, moderate signal

$$z = -1.75 + 3x_1 + 1.5x_2 + 1.5x_3 + 3x_4 + 1.5x_5 - 1.5\exp(x_6) + 3x_7^2 - 3x_{16} - 3x_{17} + 1.5x_{18} + \varepsilon$$

Case 6: Prevalence = 0.5, strong signal

$$z = -2.5 + 5x_1 + 3x_2 + 3x_3 + 5x_4 + 3x_5 - 3\exp(x_6) + 5x_7^2 - 5x_{16} - 5x_{17} + 3x_{18} + \varepsilon$$

To generate the binary outcome using the z above from the six cases, we use

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

$$Y \sim Bernoulli(\pi).$$

We fit a logistic regression model using all 20 predictors, no interactions, and no non-linear transformations of the predictors to predict the outcome of the index patient, thus introducing multiple forms of misspecification at the model fitting stage (i.e. too many noise predictors and ignoring nonlinearities for two of the signal predictors).

3.3.2 Simulation Results

In this section, we present the results of the simulation study. Although the algorithm was evaluated on five independently generated datasets for each of the six cases, we focus on one dataset per case, since the results were consistent across all datasets. First, for Case 2 (low prevalence with moderate signal) and Case 5 (a reasonably high prevalence with moderate signal), we display plots showing AUROC or AUPRC, ICI, and BrS across ten different subpopulation proportions to tune for the optimal subpopulation proportion using the TrTe data as described in Section 3.2.2. Note that for AUROC and AUPRC, a greater value represents better model discrimination; for ICI and BrS, a small value represents better predictive performance. We implement a 5-fold CV repeated twenty times, meaning that we can also find the standard error and obtain an approximate $100 \times (1 - \alpha)\%$ confidence interval around the estimate of the performance measures for each subpopulation proportion using the following equation (Bates et al., 2023; Hastie et al., 2009):

$$\left(\bar{e} - z_{1-\alpha/2}\widehat{SE}, \bar{e} + z_{1-\alpha/2}\widehat{SE}\right),$$

where

$$\widehat{SE} = \frac{1}{\sqrt{n}} \times \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(e_i - \bar{e})^2}$$

such that n is the number of folds (100 in this scenario), e_i is the estimate from the ith fold, $\bar{e}$ is the average of e_i across all folds, and $z_{1-\alpha/2}$ is the $1-\alpha/2$ quantile of the standard normal distribution. We will mention an improvement we can make to this approximate interval in Section 3.5. Next, we display the results from the validation study using the optimal subpopulation proportion found in the tuning process for the four different metrics for Case 2 and Case 5. The interval around each estimate is found using the bias-corrected and accelerated bootstrap intervals. Note that while we discuss the results of all six cases, the general shape of the plots for Cases 1, 2, and 3 is similar to each other, and the plots for Cases 4, 5, and 6 are similar to each other. Therefore, the detailed results, including

the plots for Cases 1, 3, 4, and 6, can be found in the Appendix. Finally, we present two tables showing the relative difference in the performance metrics for weightbeta CSM and weight01 CSM relative to the standard CSM for all six cases.

Case 2

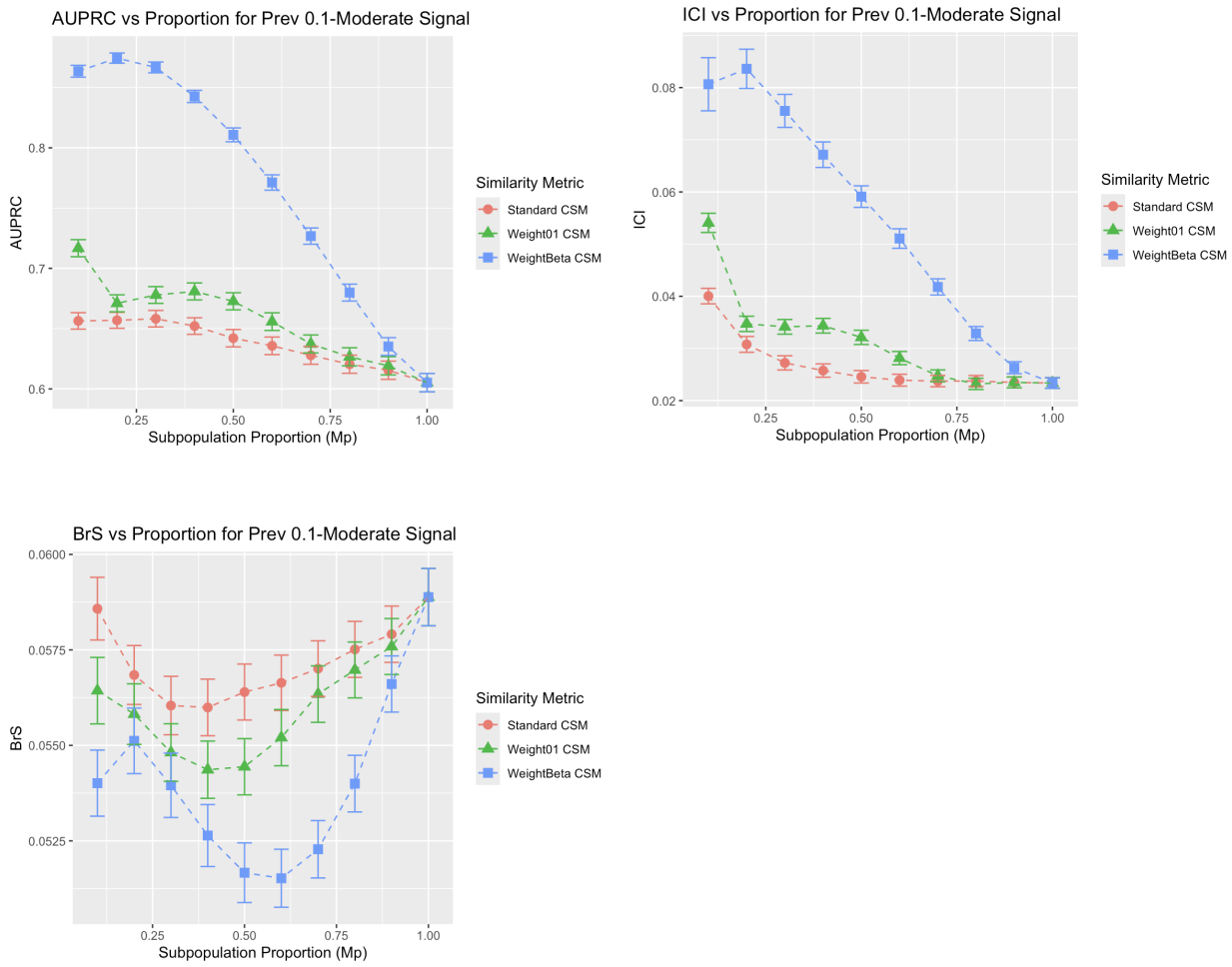


Figure 3.1: Performance metrics across different subpopulation proportions for the three similarity measures. The dataset has low prevalence with moderate signal (Case 2).

The simulation results of Case 2, shown in Figure 3.1, agree with the existing literature that we can improve discrimination while training the model on targeted and more homogeneous data, specific to each person in the testing data. The first plot comparing AUPRC

Table 3.1: Results from the validation study for low prevalence, moderate signal (Case 2). The columns "Lower" and "Upper" represent the 2.5% BCa and 97.5% BCa bootstrap interval estimates, respectively, and the winning M_p from the tuning process used to calculate the performance metrics is 0.1.

Metric	**CSM**	**Est**	**Lower**	**Upper**
AUPRC	standard	0.633	0.505	0.757
	weight01	0.646	0.514	0.775
	weightbeta	**0.776**	0.642	0.876
ICI	**standard**	**0.029**	0.012	0.051
	weight01	0.030	0.013	0.056
	weightbeta	0.057	0.026	0.092
BrS	standard	0.064	0.049	0.079
	weight01	0.063	0.049	0.078
	weightbeta	**0.059**	0.040	0.077

across different subpopulation proportions shows that as the subpopulation proportion increases, AUPRC decreases. This effect is amplified with our proposed methods - when we assign a weight of 0 or 1 on the features to calculate similarity (weight01 CSM), the model consistently has better discrimination than standard CSM. When we assign weights derived from the beta estimates from relaxed adaptive group lasso to calculate the similarity (weightbeta CSM), we observe a huge improvement in model discrimination for all subpopulation proportions compared to standard CSM. In particular, we can achieve maximum discrimination when we use the top 20% of the most similar data to train the model to predict the outcome. However, with calibration measured by ICI, different patterns are observed. First, we see that generally, training the data on a targeted subpopulation does not have a huge improvement compared to fitting a global model. However, there is still a very slight improvement, especially with using weight01 CSM to calculate the similarity - training the model on a subpopulation proportion of 80% has a negligibly better calibration at 0.0232 than using the global dataset with a calibration of 0.0234. Another pattern to note is that standard CSM generally performs better than weight01 CSM and weightbeta CSM across all subpopulation proportions except at 80%, where weight01 CSM outperforms standard CSM. In other words, the best calibration is achieved when using weight01 CSM, with a subpopulation proportion of 80%. For BrS, which incorporates both discrimination and calibration, we see that the best result is achieved with weightbeta CSM when the subpopulation proportion is 60%. Weightbeta CSM consistently outperforms

weight01 CSM, and weight01 CSM consistently outperforms standard CSM. This result is as expected - BrS can be decomposed into discrimination, where weightbeta CSM greatly improves discrimination compared to standard CSM for smaller subpopulation proportions, and calibration, where the model had better predictive performance with greater subpopulation proportion.

After tuning for the optimal subpopulation proportion, which is 0.6 determined by BrS, we validate the model performance using the holdout validation set and compare AUPRC, ICI, and BrS using the three similarity metrics. This result is shown in Table 3.1, which includes the estimate of the performance metrics along with the bias-corrected and accelerated bootstrap intervals. We can see that when we use a subpopulation proportion of 0.6, weightbeta CSM outperforms standard CSM in AURPC and BrS.

Case 5

Figure 3.2 shows that Case 5, which has a reasonably high prevalence of 0.5 and a moderate signal, has a similar shape of AUROC against different subpopulation proportions compared to Case 2. Weight01 CSM and weightbeta CSM outperform standard CSM, and in particular, the discriminative ability of weightbeta CSM has a dramatic improvement compared to standard CSM. In Case 5, best discrimination is achieved by training the model on the top 0.1, or 10%, of the most similar participants in the available data. Calibration has an interesting outcome compared to Case 2. Recall that in Case 2, the best calibration is achieved when using weight01 CSM with the subpopulation proportion of 0.8. In Case 5 also, the best calibration is found with a subpopulation proportion of 0.9 using weight01 CSM. However, when the dataset has a reasonably high prevalence, we see a unique shape in the plot of ICI against different subpopulation proportions for weightbeta CSM. It has an inverse parabolic shape, where the calibration at a subpopulation proportion of 0.1 is much better than that of standard CSM and weight01 CSM. With BrS, which considers both discrimination and calibration, we also observe a different pattern compared to Case 2. For all three similarity measures, lower subpopulation proportion results in better BrS, and the minimum value is achieved with weightbeta CSM at a subpopulation proportion of 0.1.

Tuning for the optimal subpopulation proportion that will result in the best BrS demonstrates that with Case 5, using the top 10% most similar patients in the training data leads to the best predictive performance of the predictive model. Using this value, we can validate the performance of the model using the holdout validation set, where the results can be found in Table 3.2. From the tuning process using the TrTe data, we note that while the best calibration is achieved with weight01 CSM when the subpopulation proportion is

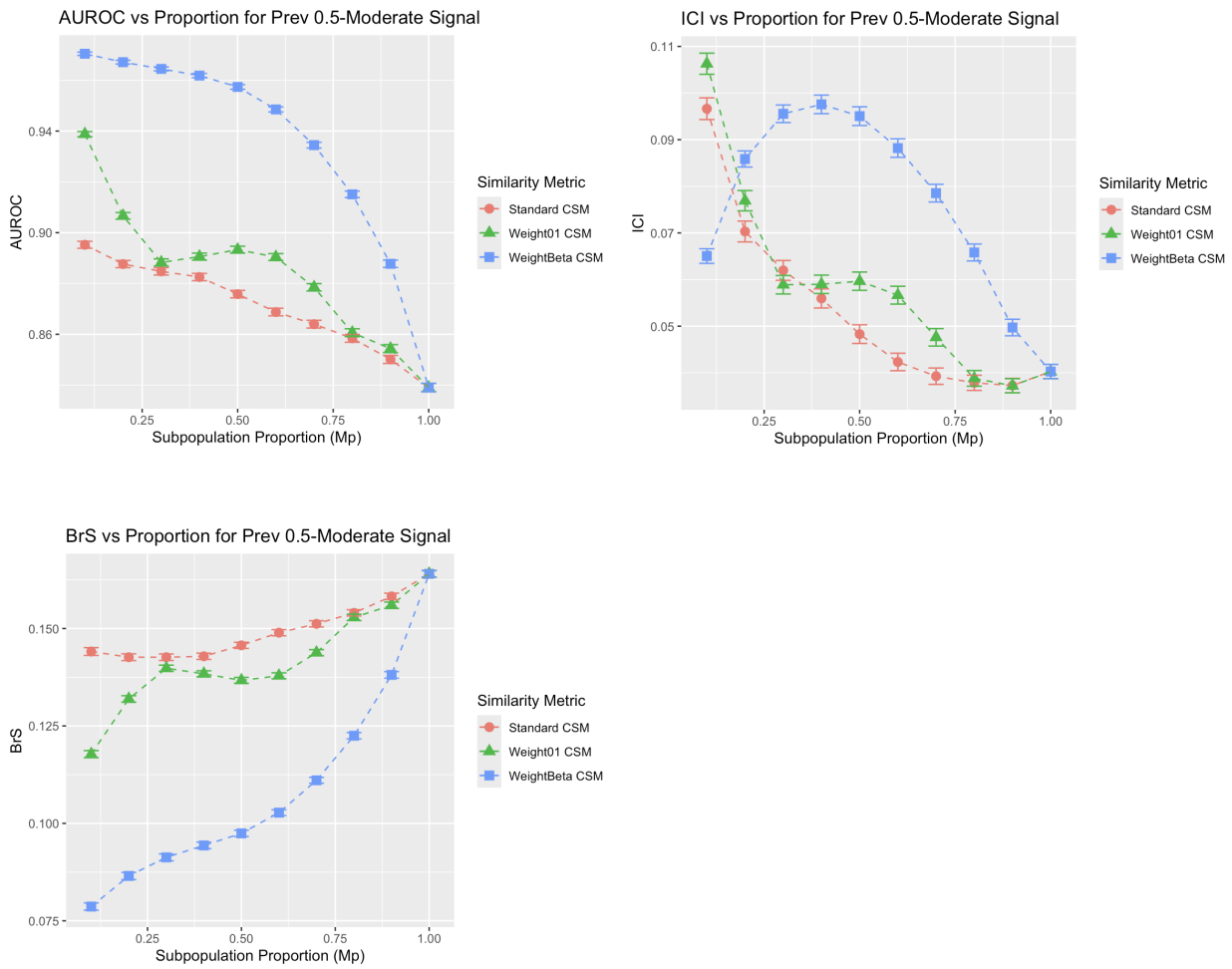


Figure 3.2: Performance metrics across different subpopulation proportions for the three similarity measures. The dataset has a reasonably high prevalence with moderate signal (Case 5).

Table 3.2: Results from the validation study for a reasonably high prevalence, moderate signal (Case 5). The columns "Lower" and "Upper" represent the 2.5% BCa and 97.5% BCa bootstrap interval estimates, respectively, and the winning M_p used to calculate the performance metrics is 0.1 from the tuning process.

Metric	**CSM**	**Est**	**Lower**	**Upper**
AUROC	standard	0.879	0.843	0.914
	weight01	0.932	0.888	0.956
	weightbeta	**0.942**	0.913	0.966
ICI	standard	0.100	0.055	0.151
	weight01	0.104	0.065	0.150
	weightbeta	**0.084**	0.058	0.121
BrS	standard	0.150	0.121	0.180
	weight01	0.116	0.087	0.149
	weightbeta	**0.087**	0.063	0.116

0.9, gains in discrimination are much bigger with weightbeta CSM when the subpopulation proportion is 0.1; then using 0.1 as our chosen subpopulation proportion from the tuning process, we note that even with calibration being considered, weightbeta CSM outperforms standard CSM across all three performance metrics.

Table 3.3: Relative difference in the performance measures between using weightbeta CSM and standard CSM, and the tuned M_p value for each simulation case.

	AUROC$_{\Delta rel}$	**AUPRC**$_{\Delta rel}$	**ICI**$_{\Delta rel}$	**BrS**$_{\Delta rel}$	**Tuned** M_p
Case 1: Prev 0.1, weak		0.2050	0.9989	0.0249	0.6
Case 2: Prev 0.1, mod		0.2254	1.0137	0.0806	0.6
Case 3: Prev 0.1, strong		0.2429	1.0221	0.1027	0.6
Case 4: Prev 0.5, weak	0.0738		0.0141	0.2244	0.1
Case 5: Prev 0.5, mod	0.0712		0.1409	0.4223	0.1
Case 6: Prev 0.5, strong	0.0660		0.1815	0.4332	0.1

Table 3.3 displays the relative difference in calculating similarity using weightbeta CSM versus standard CSM for each of the four metrics across different scenario cases, obtained by applying the tuned M_p values to the holdout validation set. Relative difference is

calculated using the following equation:

$$RelDiff_{wbeta;stand} = \frac{|metric_{weightbetaCSM} - metric_{standardCSM}|}{metric_{standardCSM}}.$$

The tuned M_p values applied to the validation set to obtain the performance metrics are shown in the final column. We can observe that generally, as the signal grows stronger - i.e. the beta values used to generate the outcome get larger - the relative difference in the measures increases. This is as expected, since the outcome is generated using only ten predictors and our proposed weightbeta CSM aims to identify and use these predictors to measure the similarity between the participants instead of using all predictors available. Thus, if the signal is stronger, then we expect the calculated similarity to be more different between assigning weights on these selected features versus using all 20 predictors equally to evaluate the similarity between participants.

Table 3.4: Relative difference in the performance measures between using weight01 CSM and standard CSM, and the tuned M_p value for each simulation case.

	AUROC$_{\Delta rel}$	**AUPRC**$_{\Delta rel}$	**ICI**$_{\Delta rel}$	**BrS**$_{\Delta rel}$	**Tuned** M_p
Case 1: Prev 0.1, weak		0.0218	0.0946	0.0070	0.6
Case 2: Prev 0.1, mod		0.0211	0.0542	0.0155	0.6
Case 3: Prev 0.1, strong		0.0416	0.1733	0.0322	0.6
Case 4: Prev 0.5, weak	0.0393		0.0891	0.1100	0.1
Case 5: Prev 0.5, mod	0.0601		0.0631	0.2288	0.1
Case 6: Prev 0.5, strong	0.0514		0.0120	0.2233	0.1

Table 3.4 displays the relative difference in calculating similarity using weight01 CSM versus standard CSM for each of the four metrics across different scenario cases, obtained by applying the tuned M_p values to the holdout validation set. Relative difference is calculated using the following equation:

$$RelDiff_{w01;stsand} = \frac{|metric_{weight01CSM} - metric_{standardCSM}|}{metric_{standardCSM}}.$$

There are several subtle differences between Table 3.4 and Table 3.3. First, the relative differences in the performance metrics are generally larger in Table 3.3. Second, unlike Table 3.3, which reports the relative differences between weightbeta CSM and standard CSM, Table 3.4 does not show a clear monotonic increase in relative differences for weight01

CSM versus standard CSM as the signal strength increases. Although stronger signals tend to produce larger relative differences than weaker signals, the pattern is not strictly monotonic. We speculate that this lack of monotonicity may be due to the way M_p is tuned. M_p is determined as the value that will give the smallest BrS across all three CSM methods. Since weightbeta CSM outperforms weight01 CSM and standard CSM measured by BrS from our simulation study, M_p is selected based on the predictive performance of the model using weightbeta CSM.

3.4 Data Analysis

We apply our proposed algorithms in Section 3.2.1 and 3.2.2 to the dataset available in the eICU Collaborative Research Database, which covers patients admitted to critical care units in 2014 and 2015 throughout the continental United States, containing 200 859 unit encounters for 139 367 unique patients. Note that this is the same dataset as discussed in Chapter 2, but instead of patients diagnosed with sepsis, we consider the patients diagnosed with circulatory system disease. To get an independent dataset with each patient having only one record, we focus on only the first ICU stay of the first hospital stay, where identifiable, of the patients who were in the ICU for at least 24 hours, have no missing APACHE (i.e. APACHE IVa) score, and have no missing vital sign data in the first 24 hours of their ICU stay. This workflow is outlined in Figure 3.3.

For the predictor variables, we consider age, gender, densely collected longitudinal vital signs - including oxygen saturation (Sao2), respiration rate, and heart rate - for the first 24 hours after ICU admission, and APACHE IVa score, which is a severity-of-illness score predicted using demographic data such as age and gender, acute physiology score, and chronic health status collected within the first 24 hours of the ICU stay. We use the hospital discharge status of dead or alive as the binary outcome variable. To represent most of the variation in the longitudinal vital signs measurements for the first 24 hours of the ICU stay, we apply FPCA and extract the first three FPCs for each vital sign, which end up explaining at least 95% of the variation in the data, as described in Chapter 2.

3.4.1 Results

We apply PPM to the eICU data, with $0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$, and 1 considered for the M_p values, and calculate the patient similarity using standard CSM, weight01 CSM, and weightbeta CSM. 80% of the available dataset is used for training and testing

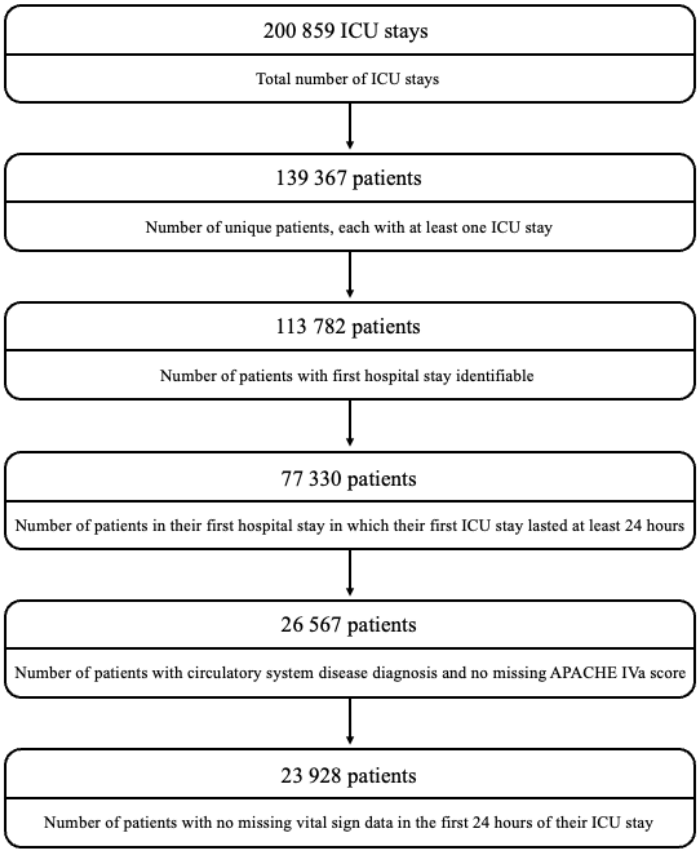


Figure 3.3: Flow chart of cohort selection from the eICU database for data analysis.

the model, using a 10-fold CV repeated ten times across different subpopulation proportions (Figure 3.4). We measure discrimination using AUPRC, since this dataset has a low prevalence of 0.11. The general pattern is similar to the simulation study, where as the subpopulation proportion increases, AUPRC decreases for weight01 CSM and weightbeta CSM. For a standard CSM that uses all the available predictors, AUPRC peaks when the subpopulation proportion is 0.6 and decreases thereafter. Best AUPRC is achieved with weightbeta CSM at a subpopulation proportion of 0.2. For calibration measured with ICI, using weight01 CSM almost always outperforms using standard CSM or weightbeta CSM. Best calibration is achieved with weight01 CSM when the subpopulation proportion is 0.2. For BrS, we also see the same result that calculating patient similarity using weight01 CSM and weightbeta CSM outperform standard CSM, with the lowest, and thus best BrS value, achieved with weightbeta CSM when the subpopulation proportion is 0.2.

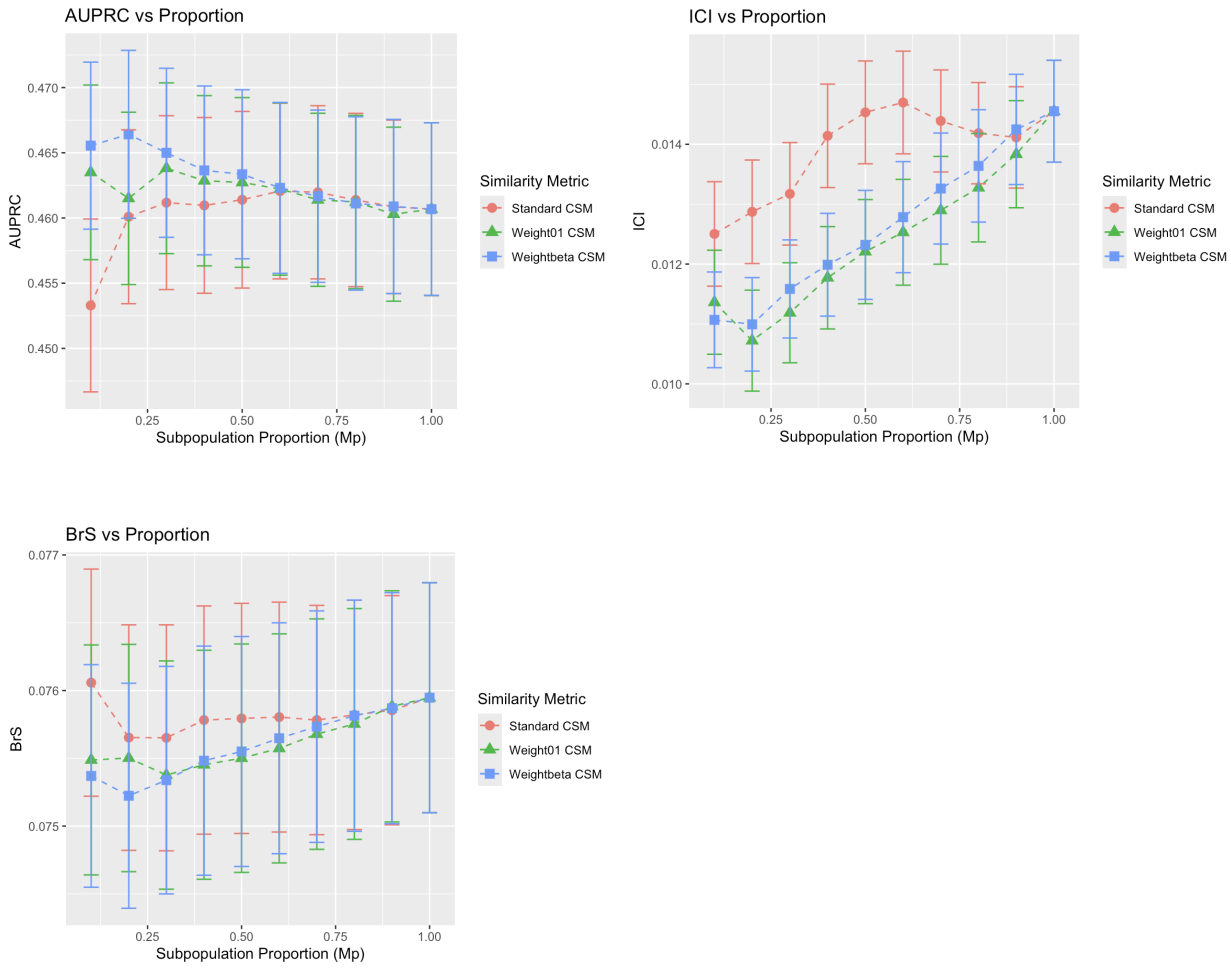


Figure 3.4: Performance metrics across different subpopulation proportions for the three similarity measures using the ICU dataset.

From AUPRC, ICI, and BrS, the best predictive performance of the model can be achieved for a subpopulation proportion of 0.2. Using the holdout validation set, we evaluate the performance of the model when we train the predictive model on the top 20% most similar participants for the participant of interest and calculate the performance metrics. The results are found in Table 3.5. Compared to the simulation study, the difference in the predictive performance of the model when using standard CSM and weightbeta CSM is much smaller. However, the estimates and the bias-corrected and accelerated bootstrap confidence intervals show that generally, AUPRC is higher and ICI and BrS are lower (i.e. all in the preferred direction) when we use weightbeta CSM instead of weight01 CSM or standard CSM to implement PPM and predict the outcome.

Table 3.5: Results from the validation study for data analysis using the ICU dataset. The columns "Lower" and "Upper" represent the 2.5% BCa and 97.5% BCa bootstrap interval estimates, respectively, and the winning M_p used to calculate the performance metrics is 0.2 from the tuning process.

Metric	**CSM**	**Est**	**Lower**	**Upper**
AUPRC	standard	0.4562	0.3603	0.5596
	weight01	0.4612	0.3606	0.5704
	weightbeta	**0.4648**	0.3633	0.5633
ICI	standard	0.0195	0.0095	0.0340
	weight01	**0.0192**	0.0094	0.0350
	weightbeta	0.0196	0.0095	0.0350
BrS	standard	0.0724	0.0606	0.0852
	weight01	0.0719	0.0607	0.0849
	weightbeta	**0.0716**	0.0600	0.0844

3.5 Discussion

In this chapter, we have shown that we can improve the predictive performance of the model using PPM when we implement our new proposed methods to calculate the patient similarity. In PPM, we aim to identify a subset of the data that is the most similar to the participant of interest so that we can train a model for each person in the testing data on a more tailored and homogeneous dataset and improve the prediction. Therefore, measuring similarity between participants to identify this subset of the dataset is crucial.

CSM is a popular method to do so, but it can be biased and lead to suboptimal results (Li and Han, 2013); thus, researchers have introduced a weighted version of CSM where features are assigned different levels of importance in similarity calculation. However, in current literature, the weights are derived using the properties of the data, such as feature frequency, or from expert knowledge that reflects the importance of features in a given context. In this chapter, we introduce a new weighted similarity metric that determines the weights without any prior knowledge about the types of predictors and remains effective even under model misspecification. We introduce two similarity metrics using relaxed adaptive group lasso: weight01 CSM, which is a weighted cosine similarity metric that assigns a weight of 0 or 1 to each available predictor, and weightbeta CSM, which assigns weights to each predictor based on the beta estimates after implementing relaxed adaptive group lasso.

Using the simulation study with generated data and data analysis using the ICU dataset, we demonstrate that we can greatly improve discrimination, and calibration in some cases, of a prediction model when implementing our proposed method for similarity calculation. Although a decrease in calibration performance is observed in certain settings, the improvement in discrimination is big enough to offset this reduction, resulting in an overall improvement in predictive performance as measured by the Brier score. The simulation study considered six cases: two levels of prevalence, and three levels of the strength of association between the predictors and the outcome. We consider AUROC, AUPRC, ICI and BrS to evaluate the model performance. When the prevalence is low at 0.1, training the model on a subset of the available training data improves discrimination when using weightbeta CSM or weight01 CSM compared to standard CSM. However, when considering calibration, standard CSM outperforms weight01 CSM or weightbeta CSM. When we consider both discrimination and calibration using BrS, weightbeta CSM and weight01 CSM outperform standard CSM at all considered subpopulation proportions, with BrS minimized using weightbeta CSM and a subpopulation proportion of 0.6. When the prevalence of the data is high at 0.5, discrimination shows the same pattern as the dataset with low prevalence, where using weightbeta CSM and weight01 CSM results in better prediction measures with AUROC maximized at subpopulation proportion of 0.1. However, there is a difference in calibration results, where it is better to use weightbeta CSM compared to standard CSM when the subpopulation proportion is 0.1. Therefore, BrS can be minimized when the subpopulation proportion is 0.1 using weightbeta CSM. When considering the three levels of association between the predictors and the outcome, we see that generally, as the strength increases, the relative difference in the performance metric when implementing PPM with weightbeta CSM versus standard CSM increases as well. In other words, when the signal is stronger, we expect PPM using weightbeta CSM

to have better discrimination.

Data analysis using ICU shows similar results. Considering AUPRC, ICI, and BrS to evaluate the discrimination and calibration of the prediction model, using weightbeta CSM to implement PPM results in better predictive performance compared to using standard CSM. Highest AUPRC, lowest ICI, and lowest BrS values are achieved when training the prediction model on the top 20% of the most similar data using weightbeta CSM. The biggest difference in the results of the simulation study and data analysis is in calibration. In the simulation study, calibration is usually worse when using weight01 CSM or weightbeta CSM instead of standard CSM for different subpopulation proportions when the prevalence is 0.1. Calibration is better when using weightbeta CSM only at the subpopulation proportion of 0.1 when the prevalence is 0.5. Furthermore, ICI is the smallest across all similarity metrics when the subpopulation proportion is 0.7 or 0.8. However, with the ICU data, ICI is the smallest when the subpopulation proportion is 0.2, and using weightbeta CSM consistently performs better than using standard CSM across all subpopulation proportions. We suspect that this may be due to different characteristics of the datasets, and further investigation is required to define how certain characteristics of the data, such as the number of predictors, the association between these predictors and the outcome, and the types of model fit to the data, can affect these results. Although n was relatively large in both the simulation study and the data analysis - $n = 10000$ and $n = 23938$, respectively - a large sample size is not strictly required for the proposed method to perform effectively. For smaller datasets, the tuned value of M_p may exceed 0.1 or 0.2, as sufficient data are needed to properly train the model and ensure that the parameter estimates converge. Nevertheless, the tuning process will account for this trade-off by selecting a subset that is large enough to train the prediction model while remaining small enough to contain only the most homogeneous subset of the original training data.

There are several areas of work for future study. First, more investigation with simulations should be conducted to be able to explain the different patterns in the calibration results. Calibration is very sensitive to how the dataset is generated, as we have seen from the different conclusions in the simulation results and the data analysis. We also plan to evaluate different calibration adjustment methods to improve the overall predictive performance of the model, such as intercept and slope recalibration (Van Calster et al., 2019). Furthermore, we want to explore implementing a nested CV algorithm to create a nested CV interval estimate introduced by Bates et al. (2023) around the performance measures to create intervals that do not underestimate the variance. Among other limitations is extending the assumptions of the predictor relationships, such as allowing for dependence among some predictors as well as allowing for interactions, in the data generation to investigate how weight01 CSM and weightbeta CSM perform compared to standard CSM. In

addition, it would be important to investigate the impact of model misspecification on the performance of these similarity metrics when the prediction model is misspecified and does not match the true data-generating mechanism. Another limitation is not confirming our results using a sophisticated machine learning model, in large part due to the computational burden of the methodology. Implementing PPM is extremely heavy computationally, because we have to fit a new model for each participant in the test data and repeat this process for every fold when using repeated CV to tune for the optimal subpopulation proportion. In our simulation study and data analysis, we use a generalized linear model, so the process is manageable; using a machine learning model, such as random forest, for example, would certainly increase the computation time. However, when we implement PPM with a simple generalized linear model using weightbeta CSM to calculate the similarity between participants, we see an improvement in the model's predictive performance at a subpopulation proportion less than 1 compared to training a random forest model on the entire training data. The next step would be to implement PPM with a random forest model to investigate how the performance metrics change with varying subpopulation proportions and similarity metrics.

Chapter 4

Comparison of Similarity Metrics in Personalized Predictive Modeling

4.1 Introduction

In Personalized Predictive Modeling (PPM), we aim to improve the predictive performance of a model by training the prediction model on customized data for an individual participant. Instead of using a one-size-fits-all model for all of the participants, we take a subset consisting of only the most similar participants in the training data to form a tailored and homogeneous dataset. Therefore, it is extremely important to select the correct subset of patients, because the advantage of fitting a model on a more similar but smaller dataset is lost if these data are not chosen carefully (Ng et al., 2015). Ng et al. have shown that when k randomly selected participants - where k is less than the total number of available participants in the training data - are used to train a personalized logistic regression model, then the predictive performance is lower than using a global logistic regression model that is trained on all available participants. The performance of a model trained on k randomly selected participants steadily increases towards the performance of a global model as k increases. This is as expected because for parametric models such as logistic regression, there must be sufficient data to train the model. However, when the k selected participants are chosen by their similarity with the participant of interest, the predictive performance of the model is greater than that of a global model trained on all available training data (Ng et al., 2015). Thus, correctly identifying the participants to be included in the customized training model in PPM is crucial and there has been substantial work on investigating and developing effective similarity metrics. A systematic review of patient

similarity in prediction models by Sharafoddini et al. (2017) and precision medicine by Parimbelli et al. (2018) show that there are many kinds of similarity metrics that can be grouped into distance-based metrics, cosine similarity-based metrics, and clustering.

Recent papers have compared various similarity metrics on the same dataset to evaluate how their new similarity metric can outperform current similarity measures to improve the predictive performance of a personalized prediction model. Li et al. (2025a) compare different patient similarity measures in a large-scale retrospective analysis of electronic health records (EHR). They use the Euclidean distance with different weighting schemes for the features to apply a k-nearest neighbour (k-NN) method and compare how the similarity metrics can affect the model's performance metrics. The four similarity metrics they compare are: vanilla Euclidean distance, where all features have equal weights; a combination of the Euclidean distance and the Jaccard index with features weighted by kidney-function-associated factors with predetermined weights; a combination of the Euclidean distance and the Jaccard index with features weighted by their types with predetermined weights; and a combination of the Euclidean distance and the Jaccard index with features weighted by their types with grid-searched weights.

Lambert et al. (2025) focus on variations of the cosine similarity metric (CSM) to build a patient network using the Markov Chain Algorithm to identify similar patients. They explore the standard CSM, cosine similarity weighted by the Inverse Document Frequency (Sparck Jones, 1972), cosine similarity weighted by the Wu and Palmer measure (Wu and Palmer, 1994), and cosine similarity weighted by the Lin measure (Lin et al., 1998) and use health data from the Échantillon Généraliste des Bénéficiaires (EGB), a French medico-administrative health insurance database.

Li et al. (2025b) use autoencoders (AEs) to address challenges of high dimensionality, mixed feature types, noise, bias, and sparsity in modeling the EHR data and compare how different AE designs and distance measures can affect the performance metrics applied on two real-world datasets. They compare the performance of using the Euclidean distance with various AE designs, and the Mahalanobis distance used with large margin nearest neighbour (LMNN), neighbourhood components analysis (NCA), and metric learning for kernel regression (MLKR) with various AE designs.

Oh et al. (2021) compare various existing similarity measures and their proposed convolution neural network-based model that jointly combines feature learning and similarity learning. They compare Euclidean distance, CSM, Mahalanobis distance, and localized supervised metric learning (LSML) with their proposed joint learning method using data from the National Health Insurance Sharing Service (NHIS) of South Korea.

Huai et al. (2018) compare CSM, Euclidean distance, LowRank (Zhan et al., 2016),

Reduced-rank local distance metric learning (R2ML) (Huang et al., 2013), LMNN (Weinberger and Saul, 2009), Information-theoretic metric learning (ITML) (Davis et al., 2007), Geometric mean metric learning (GMML) (Zadeh et al., 2016), and their proposed Uncorrelated Patient Similarity Learning approach (UnPSL) that selects relevant features with low correlations with each other. They use three real-world datasets and simulated data to compare the performance of these metrics, although they do not demonstrate how the size of the personalized data is chosen.

In this project, we hope to contribute to the ongoing research in comparing various patient similarity measures through a comprehensive simulation study. While prior work has evaluated selected similarity metrics, existing studies often rely primarily on real-world datasets and provide limited investigation into how data characteristics influence comparative performance. Our study differs in several important ways. First, we provide a systematic comparison of commonly used patient similarity approaches under multiple controlled data-generating mechanisms, allowing us to directly assess how the underlying data structure affects the performance of various similarity metrics. Second, unlike much of the existing literature that compares similarity metrics, we incorporate a tuning procedure that selects the optimal size of customized training data through cross-validation, reflecting how these methods would be implemented in practice. Third, in addition to evaluating widely used k-NN approaches, we compare them with clustering-based strategies, such as k-means and hierarchical clustering, which are not very common in the patient similarity literature in the context of predictive modeling. Thus, our work provides practical guidance for selecting patient similarity measures based on performance and the underlying data structure. In Section 4.2, we describe the similarity measures considered in this work and their implementation in the PPM framework. Section 4.3 presents an extensive simulation study evaluating performance across two outcome prevalence levels and five data-generating mechanisms, resulting in ten distinct scenarios. We conclude with a discussion of directions for future research in Section 4.4.

4.2 Methods

In this section, we provide the details of the similarity metrics and the algorithm to implement PPM for various types of similarity metrics. The first group of similarity metrics we compare are utilized with the k-NN method. These include the Euclidean distance, Mahalanobis distance, Gower's distance, CSM, weightedbeta CSM (proposed in Chapter 3), and LMNN. The second group of similarity metrics is clustering methods, including k-means clustering and hierarchical clustering, since they are most frequently used

in patient similarity (Parimbelli et al., 2018). Let $\mathbf{S}_i = (S_{i1}, S_{i2}, \ldots, S_{iP})^T \in \mathbb{R}^P$ and $\mathbf{S}_j = (S_{j1}, S_{j2}, \ldots, S_{jP})^T \in \mathbb{R}^P$ denote the P-dimensional predictor vectors for sample i and sample j, respectively. For $p = 1, \ldots, P$, S_{ip} and S_{jp} represent the values of predictor p for sample i and sample j, respectively.

Euclidean distance represents the most basic measure of dissimilarity, which adopts the l^2-norm distance to measure the similarity between two samples (Fang et al., 2021):

$$d_{\text{Euc}}(\mathbf{S}_i, \mathbf{S}_j) = \sqrt{\sum_{p=1}^{P} (S_{ip} - S_{jp})^2}.$$

Because the measure is sensitive to variable scale, predictors are typically standardized prior to analysis. Therefore, we consider two methods of standardizing the continuous predictors: (1) z-score normalization for the continuous variables, and (2) min-max scaling between 0 and 1, so that the continuous variables are on the same scale as the one-hot-encoded binary variables.

Mahalanobis distance extends Euclidean distance by incorporating the covariance structure of the predictors:

$$d_{\text{Mah}}(\mathbf{S}_i, \mathbf{S}_j) = \sqrt{(\mathbf{S}_i - \mathbf{S}_j)^T \boldsymbol{\Sigma}^{-1} (\mathbf{S}_i - \mathbf{S}_j)},$$

where $\boldsymbol{\Sigma}$ denotes the covariance matrix of the predictors (Mahalanobis, 1936). Since Mahalanobis distance measures distance in standard deviation units relative to the data distribution, standardizing the data is unnecessary.

Gower's distance is especially useful for datasets containing mixed data, such as continuous and binary predictors (Gower, 1971). It is the complement of the weighted average of similarities across all variables:

$$d_{\text{Gower}}(\mathbf{S}_i, \mathbf{S}_j) = 1 - \frac{\sum_{p=1}^{P} w_p \, \delta_{ijp} \, s_{ijp}}{\sum_{p=1}^{P} w_p \, \delta_{ijp}},$$

where w_p and δ_{ijp} are typically set to 1, and s_{ijp} denotes the variable-specific similarity, which depends on the data type. For binary variables,

$$s_{ijp} = \begin{cases} 1, & \text{if } S_{ip} = S_{jp}, \\ 0, & \text{if } S_{ip} \neq S_{jp}, \end{cases}$$

while for continuous variables,

$$s_{ijp} = 1 - \frac{|S_{ip} - S_{jp}|}{R_p},$$

where $R_p = \max(S_{\cdot p}) - \min(S_{\cdot p})$ represents the observed range of predictor p.

CSM and weightbeta CSM quantify similarity as an angle between two feature vectors (Sidorov et al., 2014). They are calculated using the following:

$$CSM_w(\mathbf{S}_i, \mathbf{S}_j) = \frac{\sum_{p=1}^{P} w_p \times S_{ip} \times S_{jp}}{\sqrt{\sum_{p=1}^{P} w_p \times S_{ip}^2}\sqrt{\sum_{p=1}^{P} w_p \times S_{jp}^2}},$$

where w_p is set to 1 for all predictors in the standard CSM, and w_p is derived using the estimates from the relaxed adaptive group lasso for weightbeta CSM, described in detail in Section 3.2.1. Thus, the corresponding cosine distance is defined as

$$d_{\cos}(\mathbf{S}_i, \mathbf{S}_j) = 1 - CSM_w(\mathbf{S}_i, \mathbf{S}_j).$$

Large margin nearest neighbour (LMNN) is a supervised metric learning approach that learns a linear transformation of the predictor space to improve the k-NN method (Weinberger and Saul, 2009). Used with Mahalanobis distance, its goal is to pull observations with similar outcomes closer together:

$$d_{\text{LMNN}}(\mathbf{S}_i, \mathbf{S}_j) = \sqrt{(\mathbf{S}_i - \mathbf{S}_j)^T \mathbf{M} (\mathbf{S}_i - \mathbf{S}_j)},$$

where $\mathbf{M}$ is a positive semidefinite matrix learned from labelled data.

For the clustering methods used with PPM, we implement k-means clustering and hierarchical clustering (Xu and Wunsch, 2005). k-means clustering uses the Euclidean distance by definition to group the observations into a pre-specified number of clusters by minimizing the within-cluster sum of squared distances. Thus, similar to the Euclidean distance in k-NN methods, we compare k-means clustering with continuous variables standardized between 0 and 1, and also k-means clustering with the continuous variables standardized using z-score normalization.

Hierarchical clustering constructs a nested sequence of clusters based on pairwise distances between observations. Each observation is treated as a single cluster and the closest clusters are merged until a single cluster remains. The desired number of clusters is obtained by cutting the dendrogram at a chosen level, where a dendrogram is a tree-like

diagram that visualizes the hierarchical arrangement of clusters. With hierarchical clustering, we compare measuring the distance using Euclidean distance, Mahalanobis distance, and Gower's distance.

4.2.1 Algorithm for PPM Using k-NN

We describe the algorithm for implementing PPM with various similarity metrics for the k-NN approach. Using a repeated cross-validation algorithm, we can identify the optimal M_p to determine the best size of customized data to improve the predictive performance of a prediction model, and compare the performance of various similarity metrics at different M_p values.

1. Split the data into training and testing set (TrTe data) and validation set (Validation data), where $100\times q\%$ of the data is used as TrTe data and the remaining $100\times(1-q)\%$ of the data is used as the Validation data.
2. Split the TrTe data into K folds for K-fold cross-validation (CV).
3. Let k be the hold-out fold for testing data, and the other $K-1$ folds be the training data.
4. Consider a grid of M_p values to tune M_p.
   (a) For an index participant j in the testing data, calculate the distance between participant j and each participant in the training data using the distance metrics introduced in Section 4.2.
   (b) Sort the participants in the training dataset in increasing order of the distance.
   (c) Fit a prediction model on the top $M_p \times 100\%$ of the sorted training data and estimate the outcome for participant j.
   (d) Repeat Steps 4(a) - 4(c) for each participant j in the testing data and calculate the model performance evaluation metric, such as discrimination, calibration, or Brier score.
5. Repeat Steps 3-4 for each k in K folds. Average over the values of performance metrics over the K folds.

6. Repeat Steps 2-5 for all K folds and repeat the entire K-fold cross-validation procedure R times. This leads to a total of $R \times K$ evaluations, each using a different training-testing split. The performance metric is averaged over all $R \times K$ evaluations, and the optimal M_p is selected as the value achieving the best averaged performance according to a pre-specified evaluation metric. In our simulation study and data analysis, we use Brier score as the evaluation metric since it evaluates overall predictive accuracy, incorporating both discrimination and calibration.

7. We validate the model using the Validation data obtained in Step 1.

   (a) From the Validation data, create a bootstrap sample of the same length by sampling with replacement.

   (b) Randomly split the bootstrap sample into 80% training sample and 20% testing sample.

   (c) For each participant j in the testing sample, measure the distance between participant j and all the participants in the training sample using the distance metrics introduced in Section 4.2.

   (d) Sort the participants in the training sample by increasing order of distance. Using the optimal M_p value obtained from Steps 2-6, obtain a subpopulation of the training set that is the most similar to participant j from the testing set.

   (e) Evaluate the predictive performance by fitting a prediction model on the subpopulation obtained in Step 7(d), for each j in the testing sample.

   (f) Repeat Steps 7(a)-7(e) B number of times to obtain a bias-corrected and accelerated (BCa) bootstrap confidence interval around the performance measure(s) of interest from the validation sample.

4.2.2 Algorithm for PPM Using Clustering

Clustering methods require a slightly different approach than the k-NN method. Instead of tuning for the optimal M_p that lets us specify the size of the training data to fit a personalized model for the participant of interest, we tune for the optimal number of clusters instead. Thus, we do not control the size of the training data for the personalized model, since we cannot guarantee the same number of observations in each cluster.

1. Split the data into training and testing set (TrTe data) and validation set (Validation data), where $100 \times q\%$ of the data is used as TrTe data and the remaining $100 \times (1-q)\%$ of the data is used as the Validation data.

2. Split the TrTe data into K folds for K-fold cross-validation (CV).

3. Let k be the hold-out fold for testing data, and the other $K-1$ folds be the training data.

4. Consider a grid of possible numbers of clusters, C, to tune C.

   (a) Using a clustering method (hierarchical clustering or k-means clustering), create C clusters using the training data.

   (b) Fit a prediction model on each of the clusters so that there are C unique models.

   (c) For an index participant j in the testing data, determine which cluster from the training model is closest by using a distance metric (Euclidean distance, Mahalanobis distance or Gower's distance) to calculate the distance between participant j and each of the cluster centroids.

   (d) Using the model trained on that cluster, predict the outcome of participant j.

   (e) Repeat Steps 4(c) - 4(d) for each participant j in the testing data and calculate the model performance evaluation metric, such as discrimination, calibration, or Brier score.

5. Repeat Steps 3-4 for each k in K folds. Average over the values of performance metrics over the K folds.

6. Repeat Steps 2-5 for all K folds and repeat the entire K-fold cross-validation procedure R times. This leads to a total of $R \times K$ evaluations, each using a different training-testing split. The performance metric is averaged over all $R \times K$ evaluations, and the optimal C is selected as the value achieving the best averaged performance according to a pre-specified evaluation metric. In our simulation study and data analysis, we again use Brier score as the evaluation metric since it evaluates overall predictive accuracy, incorporating both discrimination and calibration.

7. We validate the model using the Validation data obtained in Step 1.

   (a) From the Validation data, create a bootstrap sample of the same length by sampling with replacement.

   (b) Randomly split the bootstrap sample into 80% training sample and 20% testing sample.

   (c) Create clusters using the training data. The number of clusters to form is the tuned C using the TrTe data in Steps 2-6.

(d) Fit a prediction model on each cluster.

(e) For each participant j in the testing sample, identify the closest cluster and use the model trained on this cluster to estimate the outcome of participant j. Determine the closest cluster by calculating the distance between participant j and the cluster centroids using a distance metric such as Euclidean distance, Mahalanobis distance, or Gower's distance.

(f) After repeating for every j in the testing data, calculate the performance metrics.

(g) Repeat Steps 7(a)-7(f) B number of times to obtain a bootstrap confidence interval around the performance measure of the validation sample.

4.3 Simulations

In this section, we describe a simulation study conducted on various datasets to compare different similarity metrics in Section 4.2, applying the algorithms introduced in Section 4.2.1 and Section 4.2.2, depending on the type of similarity metric. We focus on generalized linear models in this chapter, specifically logistic regression with a log link, but we mention in Chapter 5 that this work can be extended to more complicated and flexible models, such as generalized additive models, machine learning models, etc. We implement a 5-fold cross-validation repeated 20 times to tune the subpopulation proportion M_p by considering the following values for M_p: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1. We also implement a 5-fold cross-validation repeated 20 times to tune the cluster size C by considering the following values for C: 1, 2, 3, 4, 5, 7, 10, and 20. Note that M_p of 1 or C of 1 are equivalent to fitting a global model using all participants in the training data.

In the validation step, we use $B = 500$ bootstrap samples to generate the bias-corrected and accelerated (BCa) bootstrap confidence interval around the performance measures using the holdout validation set. The winning values of M_p and C are determined using the mean cross-validated Brier score in the tuning step, and are applied to evaluate the performance of the personalized models on the holdout validation data. Similar to Chapter 3, we use AUPRC, ICI, and BrS to evaluate the discrimination and calibration of the prediction model when the outcome prevalence is low, and AUROC, ICI, and BrS to evaluate the discrimination and calibration of the prediction model when the outcome prevalence is relatively high.

In Section 4.3.1, we describe the various data generation methods that we use to conduct an extensive simulation study to evaluate the effects of similarity metrics on PPM. In

Section 4.3.2, we discuss the results comparing various outputs from the different datasets and similarity metrics.

4.3.1 Setup

In this section, we describe ten cases that we investigate through the simulation study, coded in R version 4.4.0 (R Core Team, 2023) using RStudio (Posit team, 2023). There are two prevalence levels of the outcome Y, 0.1 and 0.5, for each of the five data generation methods described below. We first discuss the process for generating the predictors and the linear predictor z for each of the five data-generating methods. Then, we describe generating the binary outcome Y based on the linear predictor z using Equations 4.1 and 4.2, which is shared across all five methods.

Case 1 (lower prevalence) and Case 2 (higher prevalence)

We generate the data using a sample size of $n = 10000$ with 20 independent predictors. 15 of these predictors are continuous variables generated independently from $N(0, 2^2)$ and five are binary predictors independently from $Bernoulli(0.5)$. Among these 20 predictors, seven continuous predictors and three binary predictors are used to generate a linear predictor z for each sample, with $\varepsilon \sim N(0, 2.5^2)$. Here, $\beta_0 = -2.5$ for prevalence of 0.1 and $\beta_0 = 10.5$ for prevalence of 0.5:

$$z = \beta_0 + 2x_1 + 1x_2 - 1x_3 - 2x_4 + 1x_5 - 1\exp(x_6) - 2x_7^2 - 2x_{16} - 2x_{17} + 1x_{18} + \varepsilon.$$

Case 3 (lower prevalence) and Case 4 (higher prevalence)

The predictors are generated using the same method as **Case 1** and **Case 2**; however, there are no transformations on the predictors x_6 and x_7. Then, with $\beta_0 = -10$ or $\beta_0 = 1.5$ to account for a prevalence of 0.1 and 0.5, respectively, the linear predictor is generated as follows:

$$z = \beta_0 + 2x_1 + 1x_2 - 1x_3 - 2x_4 + 1x_5 - 1x_6 - 2x_7 - 2x_{16} - 2x_{17} + 1x_{18} + \varepsilon.$$

Case 5 (lower prevalence) and Case 6 (higher prevalence)

We generate a synthetic dataset with $n = 10000$ observations arising from three latent clusters not of all equal size $(3000, 3000, 4000)$. Each observation contains 20 predictors, consisting of 15 continuous variables and five binary variables. Let $C_i \in \{1, 2, 3\}$ denote the true cluster membership of individual i. Cluster labels are assigned deterministically based on the specified cluster sizes.

Conditional on cluster membership $C_i = c$, the continuous predictors $\mathbf{X}_i^{(cts)} \in R^{15}$ are generated independently from a multivariate normal distribution

$$\mathbf{X}_i^{(cts)} = (x_{i,1}, \ldots, x_{i,15}) \mid C_i = c \sim \mathcal{N}(\boldsymbol{\mu}_c, \mathbf{I}_{15}),$$

where the cluster-specific mean vectors are $\boldsymbol{\mu}_1 = \mathbf{0}$, $\boldsymbol{\mu}_2 = \mathbf{1}$, and $\boldsymbol{\mu}_3 = -\mathbf{1}$, and $\mathbf{I}_{15}$ denotes the 15×15 identity matrix. The binary predictors $\mathbf{X}_i^{(bin)} = (x_{i,16}, \ldots, x_{i,20})$ are generated independently, conditional on cluster membership. For each binary variable $j = 1, \ldots, 5$ and cluster k, we sample

$$x_{i,15+j} \mid Z_i = c \sim \text{Bernoulli}(p_{jc}),$$

with cluster-specific success probabilities

$$\begin{aligned}
(p_{1c}) &= (0.4, 0.5, 0.6), \\
(p_{2c}) &= (0.3, 0.5, 0.7), \\
(p_{3c}) &= (0.5, 0.6, 0.4), \\
(p_{4c}) &= (0.4, 0.5, 0.6), \\
(p_{5c}) &= (0.5, 0.4, 0.6),
\end{aligned}$$

for clusters $c = 1, 2, 3$, respectively.

A binary outcome is generated using a cluster-specific logistic regression model. Let $\mathbf{X}_i = (\mathbf{X}_i^{(cts)}, \mathbf{X}_i^{(bin)})$ denote the full 20-dimensional predictor vector. $\beta_0 = -8$ or $\beta_0 = -1.5$ is a common intercept across clusters, depending on whether prevalence is 0.1 or 0.5, and $\varepsilon \sim N(0, 2.5^2)$. Then, the linear predictor is

$$z = \begin{cases} \beta_0 + 1x_1 + 2x_2 - 1x_3 + 1x_4 - 2x_5 + 1x_6 + 2x_7 + 1x_{16} + 2_{17} - 1x_{18} + \varepsilon, & c = 1, \\ \beta_0 + 1x_1 + 1x_2 - 2x_3 - 1x_4 + 1x_5 + 2x_6 - 1x_7 + 2x_{16} - 1_{17} - 1x_{18} + \varepsilon, & c = 2, \\ \beta_0 - 2x_1 - 1x_2 + 1x_3 - 1x_4 + 2x_5 - 1x_6 + 1x_7 + 1x_{16} + 2_{17} - 2x_{18} + \varepsilon, & c = 3. \end{cases}$$

Case 7 (lower prevalence) and Case 8 (higher prevalence)

The predictors are generated using the same method as **Case 5** and **Case 6**; however, instead of using cluster-specific equations, we use the same equation across all clusters, implementing transformations on x_6 and x_7 as seen in **Case 1** and **Case 2**. Then, with $\beta_0 = -16$ or $\beta_0 = -5.5$ to account for a prevalence of 0.1 and 0.5, the linear predictor is generated as follows:

$$z = \beta_0 + 1x_1 + 2x_2 - 1x_3 + 1x_4 - 2x_5 + 1\exp(x_6) + 2x_7^2 + 1x_{16} + 2x_{17} - 1x_{18} + \varepsilon.$$

Case 9 (lower prevalence) and Case 10 (higher prevalence)

The predictors are generated using the same method as **Case 5** and **Case 6**; however, instead of having cluster-specific equations for the linear predictor, we use the same equation across all clusters without any transformations on x_6 and x_7. Then, with $\beta_0 = -9.5$ or $\beta_0 = -0.5$ to account for a prevalence of 0.1 and 0.5, the linear predictor is generated as follows:

$$z = \beta_0 + 1x_1 + 2x_2 - 1x_3 + 2x_4 - 2x_5 + 1x_6 + 2x_7 + 1x_{16} + 2x_{17} - 1x_{18} + \varepsilon.$$

Using the linear predictor z for all ten cases, we generate the binary outcome using

$$\pi = 1/(1 + \exp(-z)) \tag{4.1}$$

$$Y \sim Bernoulli(\pi). \tag{4.2}$$

In summary, the simulation study includes two prevalence levels and five data-generating mechanisms. The first data-generating method is similar to the simulation study in Chapter 2, where ten out of 20 predictors are used to generate the outcome, and two predictors are transformed before calculating the linear predictor. The second data-generating method also uses ten out of 20 predictors to generate the outcome, without any transformations of the predictors. The third data-generating method introduces latent clusters in the predictors and also a cluster-specific outcome-generating equation using ten out of 20 predictors. The fourth data-generating method includes latent clusters in the predictors, but uses the same outcome-generating equation across all clusters. However, two of the ten predictors used to generate the outcome are transformed before calculating the linear predictor. The final and fifth data-generating mechanism includes latent clusters in the predictors, uses the same outcome-generating equation across all clusters, and has no transformations of the predictors. When we are fitting a logistic regression model to predict the outcome of the index patient, we do so using all 20 predictors without any interactions or transformations on the predictors.

4.3.2 Simulation Results

In this section, we present the results of the ten simulation cases and report the findings. Although the algorithm was evaluated on five independently generated datasets for each of the ten cases, we focus on one dataset per case, since the results were consistent across all datasets. A similarity metric performs better if the prediction model has higher AUROC or AUPRC, and lower ICI and BrS. For each case, a figure consisting of six plots displays the outcomes from the tuning process using the TrTe data. The three plots in the first column are the results of tuning for the optimal M_p using similarity metrics in the k-NN method. These include CSM (csm), Euclidean distance with the continuous predictors scaled between 0 and 1 (euc_scaled), Euclidean distance with the continuous predictors normalized (euc_z), Gower's distance (gowers), LMNN (lmnn), Mahalanobis distance with the continuous predictors normalized (mahal_scaled), and weightbeta CSM method (wcsm) introduced in Chapter 3. After identifying the most similar participants using the similarity metrics presented above, we take the top $M_p \times 100\%$ most similar participants from the training data to train the prediction model. The three plots in the right column are the results of tuning for the optimal C using the clustering method. These include hierarchical clustering with Euclidean distance, where the continuous predictors are scaled between 0 and 1 (hierarch_euc_scaled), hierarchical clustering with Euclidean distance, where the continuous predictors are normalized (hierarch_euc_z), hierarchical clustering with Gower's distance (hierarch_gower), hierarchical clustering with Mahalanobis distance (hierarch_mah), k-means clustering with the continuous predictors scaled between 0 and 1 (kmeans_scaled), and k-means clustering with the continuous predictors normalized (kmeans_z).

We also include a table of the results from the validation study for each case. From the tuning process, we identify the winning M_p and C, defined as the values that produce the smallest BrS value regardless of the similarity method. We chose BrS as the performance metric in the tuning process because it can be decomposed into both discrimination and calibration. These winning M_p and C are then used to evaluate the performance of PPM on the holdout validation data. Note that while we discuss all ten cases, we show the figures and the tables for Cases 1, 3, 5, 7, and 9 in this section and the rest of the results can be found in Appendix C.

First, we note that for Case 3 (Figure 4.2 and Table 4.2), Case 4 (Figure C.2 and Table C.2), Case 9 (Figure 4.5 and Table 4.5) and Case 10 (Figure C.5 and Table C.5), PPM does not have a noticeable impact on the predictive performance of the model. Recall that in Cases 3 and 4, the data generation process has no latent clusters in the predictors, and has no transformations on the predictors. In Cases 9 and 10, latent clusters exist in the predictors but there was a single outcome-generating equation across all clusters

without any transformations on the predictors. In these cases, the optimal M_p is either 0.9 or 1 depending on the prevalence, but the improvement in predictive performance in the model, if any, is negligible. There is no advantage to fitting a personalized model for each participant in the testing data - in fact, the best predictive performance of the model, measured by discrimination and calibration, is achieved when fitting a global model by using all available training data.

However, we observe an increase in predictive performance by identifying a homogeneous dataset and training a personalized model for the other six cases. For Case 1 (Figure 4.1 and Table 4.1), Case 2 (Figure C.1 and Table C.1), Case 7 (Figure 4.4 and Table 4.4), and Case 8 (Figure C.4 and Table C.4), weightbeta CSM outperforms all other methods in terms of discrimination and BrS when using the k-NN approach. The data-generating mechanism for Cases 1 and 2 includes two transformed predictors before generating the outcome, and Cases 7 and 8 include latent clusters in the predictors and also two transformed predictors when generating the outcome. For Case 7 and Case 8 specifically, we can see that using k-means clustering and hierarchical clustering with the Euclidean distance and normalized continuous predictors perform similarly to the weightbeta CSM in terms of discrimination, but perform even better in terms of calibration.

For Case 5 (Figure 4.3 and Table 4.3) and Case 6 (Figure C.3 and Table C.3), where the data contain latent clusters in the predictors and a cluster-specific outcome-generating equation, the clustering methods outperform k-NN methods in terms of prediction accuracy when fitting a personalized model. The best discrimination is achieved using k-means clustering and hierarchical clustering with Euclidean distance and normalized continuous predictors, followed by LMNN and Euclidean distance with normalized continuous predictors in the k-NN method.

These findings above are highlighted in Table 4.6, where we emphasize the important results of the simulation investigation. A few other things to note are that first, fitting a global model has a better predictive performance compared to fitting a personalized model using a subset of the available training data when the similarity is determined by Gower's distance. Furthermore, standardizing continuous predictors using normalization instead of min-max scaling when using Euclidean distance results in much better performance in both the k-NN method and in k-means or hierarchical clustering methods. When min-max scaling is applied to continuous predictors and Euclidean distance is used to identify similar participants in the k-NN method, the model's predictive performance is very similar to that of using CSM. Euclidean distance with min-max scaling and CSM do not perform as well as other similarity metrics in the k-NN method.

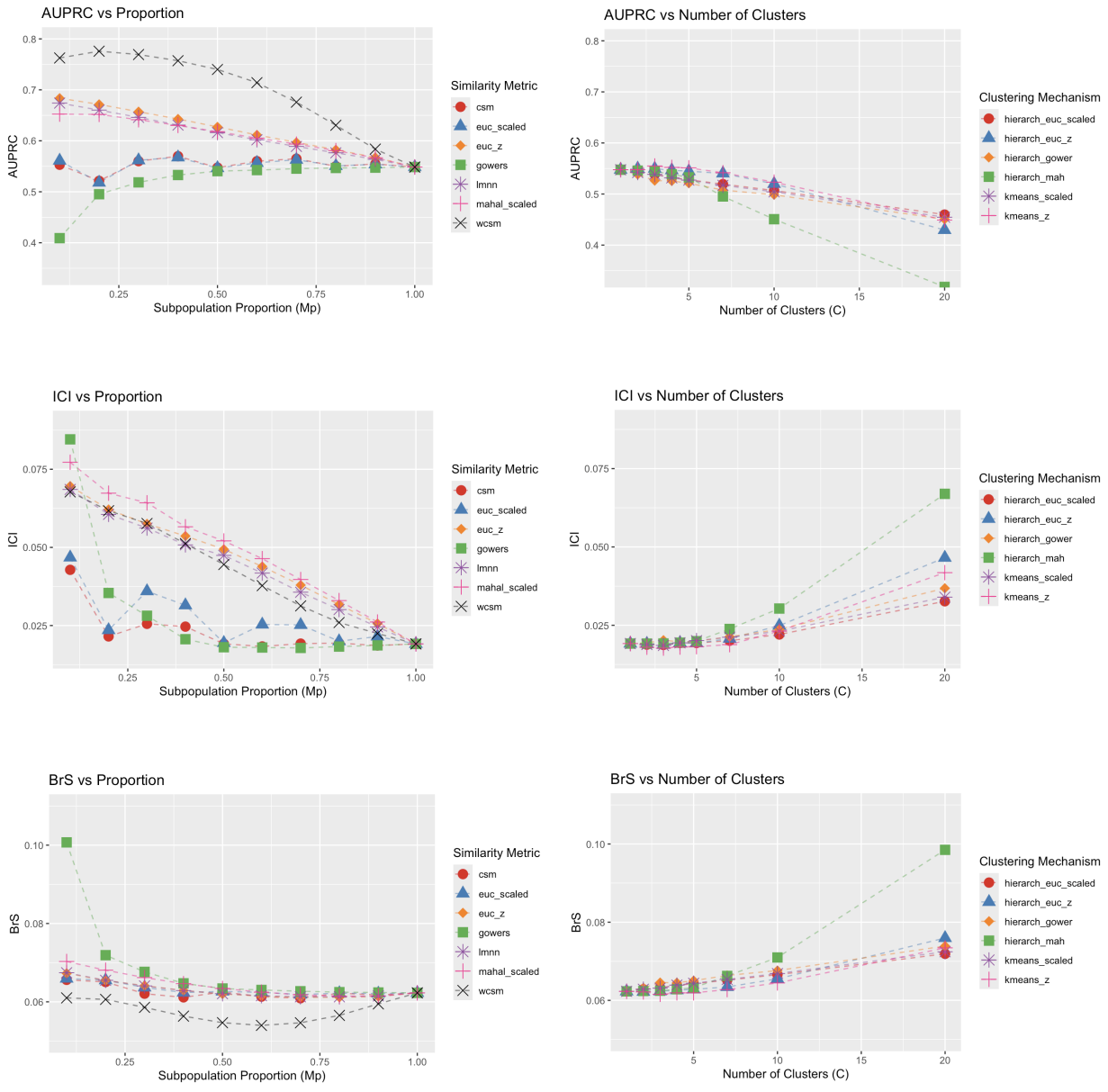


Figure 4.1: Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has low prevalence with the first data generation method (Case 1).

Table 4.1: Results from the validation study for Case 1. $M_p = 0.6$ and $C = 3$ were used from the tuning process.

Metric	Similarity measure	Est	2.5% BCa	97.5% BCa
	csm	0.609	0.439	0.757
	euc_scaled	0.614	0.456	0.761
	euc_z	0.648	0.481	0.777
	gowers	0.549	0.392	0.695
	lmnn	0.644	0.480	0.780
	mahal_scaled	0.646	0.474	0.786
AUPRC	**wcsm**	**0.701**	0.538	0.828
	hierarch_euc_scaled	0.590	0.403	0.736
	hierarch_euc_z	0.527	0.324	0.694
	hierarch_gowers	0.548	0.316	0.706
	hierarch_mah	0.581	0.382	0.736
	kmeans_scaled	0.594	0.418	0.746
	kmeans_z	0.595	0.419	0.738
	csm	**0.0251**	0.0132	0.0479
	euc_scaled	0.0274	0.0139	0.0558
	euc_z	0.0401	0.0142	0.0727
	gowers	0.0313	0.0142	0.0543
	lmnn	0.0393	0.0151	0.0717
	mahal_scaled	0.0406	0.0147	0.0774
ICI	wcsm	0.0405	0.0143	0.0867
	hierarch_euc_scaled	0.0300	0.0155	0.0575
	hierarch_euc_z	0.0403	0.0208	0.0712
	hierarch_gowers	0.0360	0.0171	0.0883
	hierarch_mah	**0.0251**	0.0122	0.0463
	kmeans_scaled	0.0294	0.0141	0.0494
	kmeans_z	0.0310	0.0142	0.0550
	csm	0.0571	0.0403	0.0745
	euc_scaled	0.0569	0.0409	0.0754
	euc_z	0.0572	0.0394	0.0747
	gowers	0.0640	0.0477	0.0827
	lmnn	0.0575	0.0396	0.0753
	mahal_scaled	0.0575	0.0385	0.0755
BrS	**wcsm**	**0.0530**	0.0359	0.0685
	hierarch_euc_scaled	0.0591	0.0424	0.0764
	hierarch_euc_z	0.0660	0.0456	0.0885
	hierarch_gowers	0.0640	0.0460	0.0887
	hierarch_mah	0.0591	0.0423	0.0750
	kmeans_scaled	0.0586	0.0409	0.0745
	kmeans_z	0.0588	0.0413	0.0766

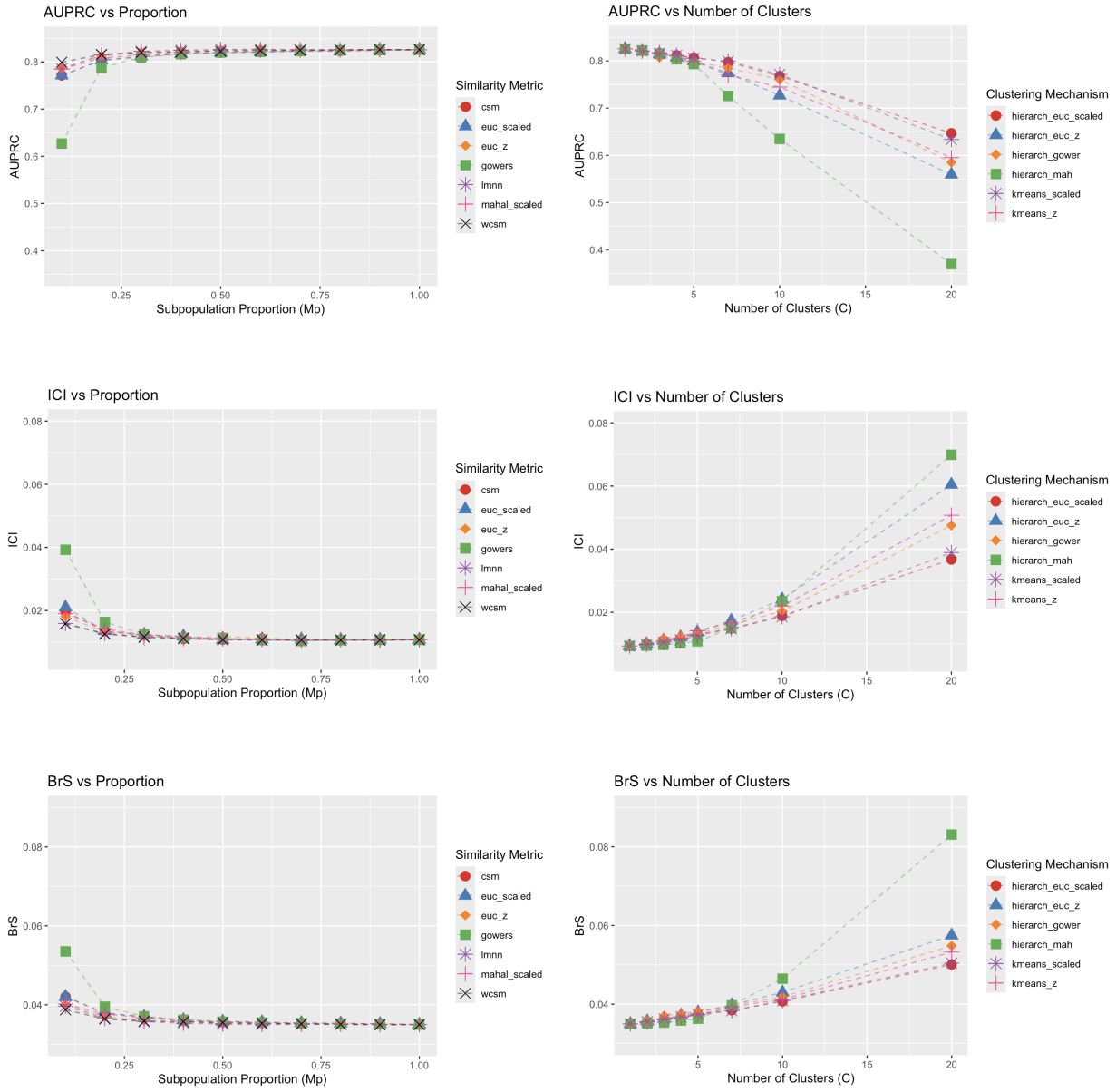


Figure 4.2: Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has low prevalence with the second data generation method (Case 3).

Table 4.2: Results from the validation study for Case 3. $M_p = 0.9$ and $C = 1$ were used from the tuning process.

Metric	Similarity measure	Est	2.5% BCa	97.5% BCa
AUPRC	csm	0.841	0.711	0.922
	euc_scaled	0.839	0.718	0.923
	euc_z	**0.847**	0.725	0.927
	gowers	0.840	0.705	0.920
	lmnn	0.846	0.721	0.924
	mahal_scaled	0.846	0.719	0.927
	wcsm	0.845	0.709	0.926
	hierarch_euc_scaled	0.837	0.718	0.923
	hierarch_euc_z	0.837	0.718	0.923
	hierarch_gowers	0.837	0.718	0.923
	hierarch_mah	0.837	0.718	0.923
	kmeans_scaled	0.837	0.718	0.923
	kmeans_z	0.837	0.718	0.923
ICI	csm	0.0239	0.0105	0.0693
	euc_scaled	0.0239	0.0116	0.0791
	euc_z	0.0236	0.00958	0.0617
	gowers	0.0236	0.0105	0.0591
	lmnn	**0.0232**	0.0109	0.0678
	mahal_scaled	0.0234	0.00947	0.0616
	wcsm	0.0237	0.0107	0.0617
	hierarch_euc_scaled	0.0245	0.0111	0.0765
	hierarch_euc_z	0.0245	0.0111	0.0765
	hierarch_gowers	0.0245	0.0111	0.0765
	hierarch_mah	0.0245	0.0111	0.0765
	kmeans_scaled	0.0245	0.0111	0.0765
	kmeans_z	0.0245	0.0111	0.0765
BrS	csm	0.0324	0.0206	0.0457
	euc_scaled	0.0327	0.0218	0.0460
	euc_z	**0.0318**	0.0214	0.0457
	gowers	0.0325	0.0209	0.0465
	lmnn	0.0319	0.0215	0.0459
	mahal_scaled	0.0319	0.0213	0.0457
	wcsm	0.0320	0.0213	0.0458
	hierarch_euc_scaled	0.0326	0.0201	0.0464
	hierarch_euc_z	0.0326	0.0201	0.0464
	hierarch_gowers	0.0326	0.0201	0.0464
	hierarch_mah	0.0326	0.0201	0.0464
	kmeans_scaled	0.0326	0.0201	0.0464
	kmeans_z	0.0326	0.0201	0.0464

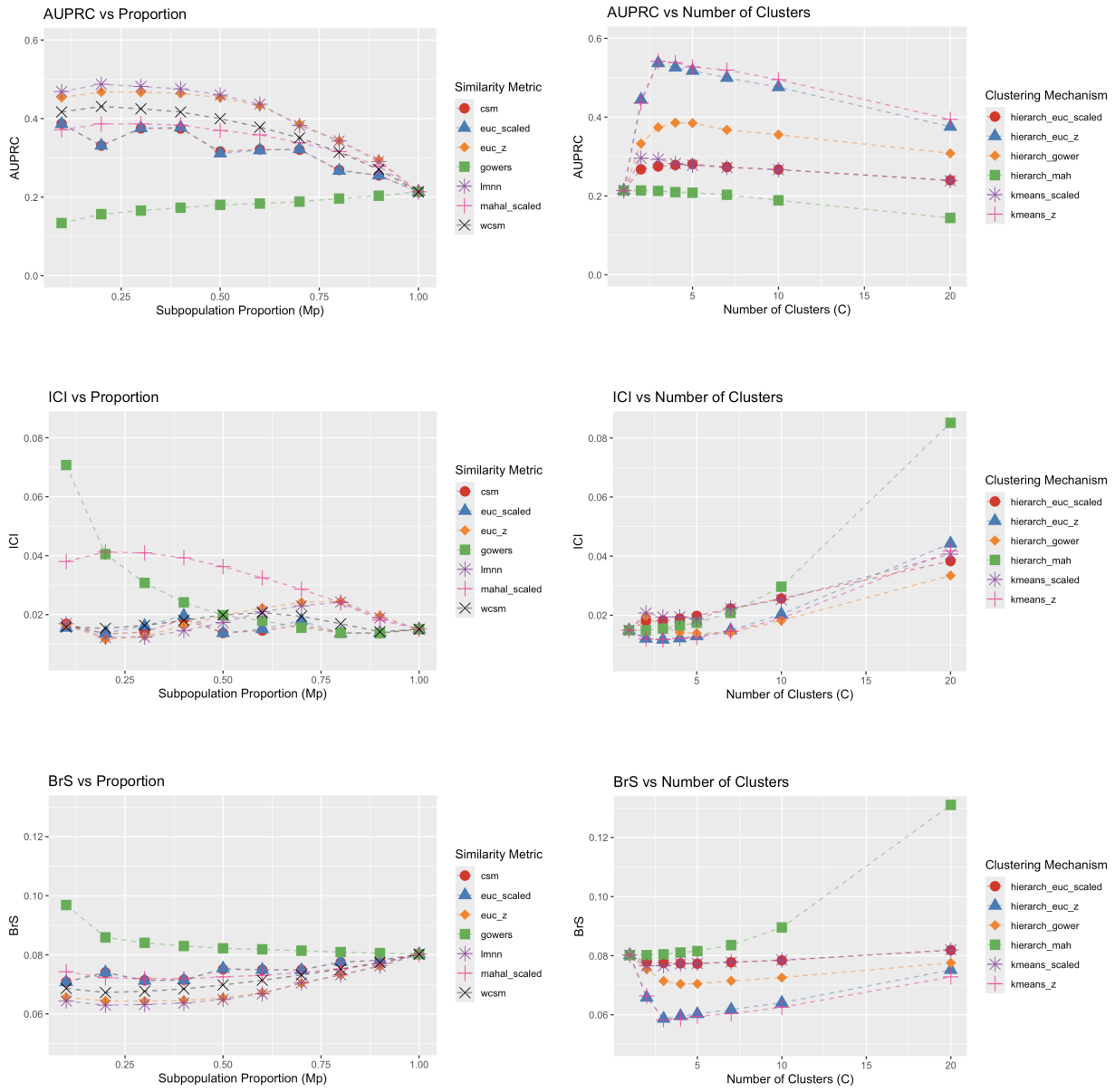


Figure 4.3: Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has low prevalence with the third data generation method (Case 5).

Table 4.3: Results from the validation study for Case 5. $M_p = 0.2$ and $C = 3$ were used from the tuning process.

Metric	Similarity measure	Est	2.5% BCa	97.5% BCa
AUPRC	csm	0.318	0.174	0.494
	euc_scaled	0.311	0.167	0.484
	euc_z	0.413	0.222	0.585
	gowers	0.131	0.0714	0.239
	lmnn	0.407	0.216	0.576
	mahal_scaled	0.321	0.181	0.506
	wcsm	0.374	0.216	0.580
	hierarch_euc_scaled	0.268	0.143	0.458
	hierarch_euc_z	0.447	0.276	0.634
	hierarch_gowers	0.337	0.160	0.517
	hierarch_mah	0.231	0.124	0.409
	kmeans_scaled	0.287	0.151	0.488
	kmeans_z	**0.462**	0.290	0.647
ICI	csm	0.0458	0.0237	0.0716
	euc_scaled	0.0462	0.0226	0.0727
	euc_z	0.0440	0.0215	0.0777
	gowers	0.103	0.0708	0.134
	lmnn	0.0431	0.0202	0.0759
	mahal_scaled	0.0704	0.0380	0.119
	wcsm	0.0450	0.0237	0.0754
	hierarch_euc_scaled	0.0360	0.0180	0.0610
	hierarch_euc_z	**0.0302**	0.0148	0.0573
	hierarch_gowers	0.0392	0.0192	0.0965
	hierarch_mah	0.0324	0.0155	0.0563
	kmeans_scaled	0.0345	0.0181	0.0615
	kmeans_z	0.0317	0.0150	0.0569
BrS	csm	0.0734	0.0550	0.0965
	euc_scaled	0.0739	0.0549	0.0946
	euc_z	0.0673	0.0467	0.0899
	gowers	0.107	0.0788	0.134
	lmnn	0.0673	0.0483	0.0906
	mahal_scaled	0.0816	0.0588	0.105
	wcsm	0.0704	0.0513	0.0925
	hierarch_euc_scaled	0.0728	0.0524	0.0978
	hierarch_euc_z	0.0619	0.0431	0.0802
	hierarch_gowers	0.0714	0.0504	0.120
	hierarch_mah	0.0747	0.0517	0.0971
	kmeans_scaled	0.0714	0.0495	0.0941
	kmeans_z	**0.0612**	0.0431	0.0829

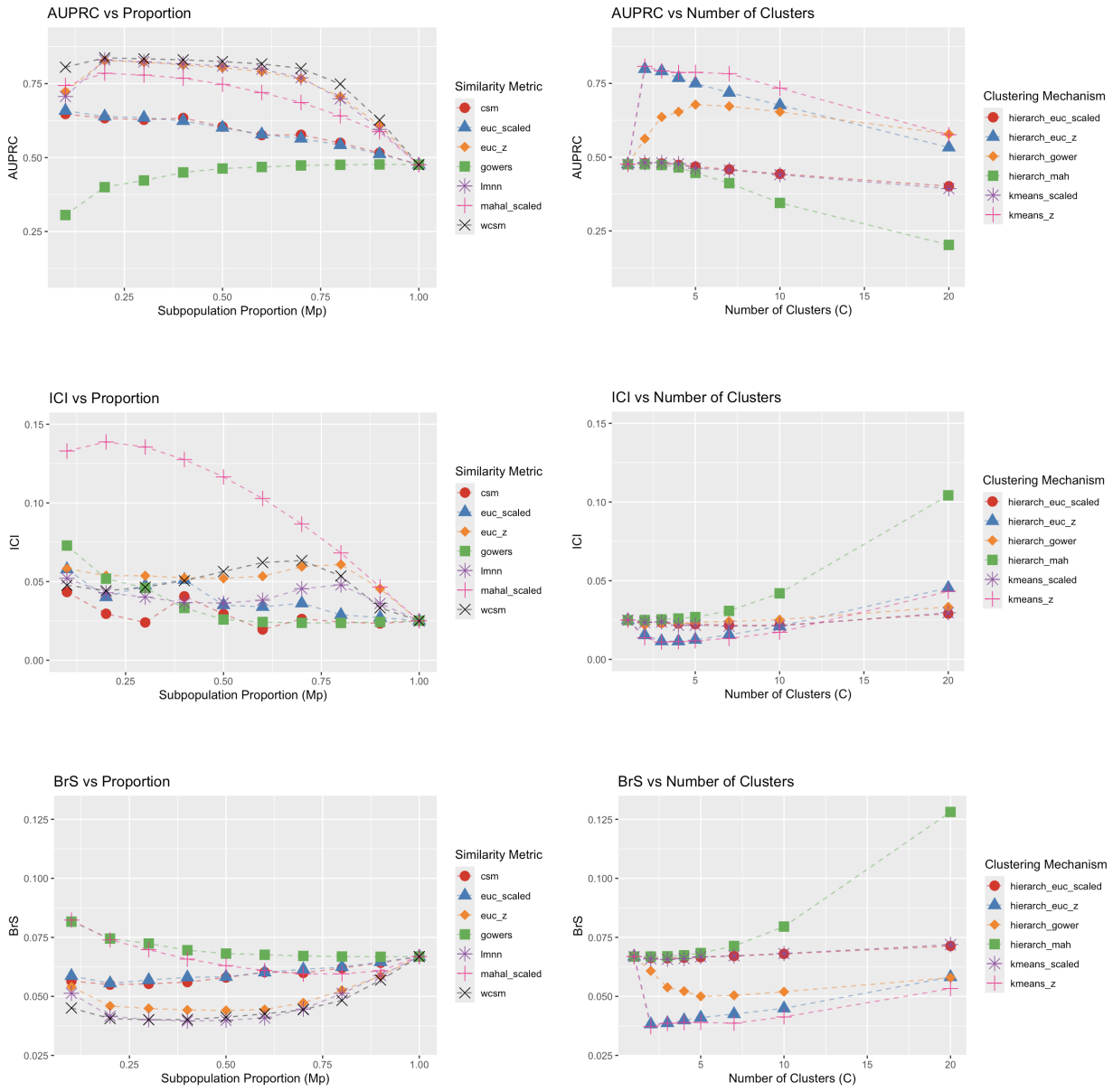


Figure 4.4: Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has low prevalence with the fourth data generation method (Case 7).

Table 4.4: Results from the validation study for Case 7. $M_p = 0.4$ and $C = 3$ were used from the tuning process.

Metric	Similarity measure	Est	2.5% BCa	97.5% BCa
AUPRC	csm	0.590	0.431	0.739
	euc_scaled	0.594	0.427	0.745
	euc_z	0.734	0.512	0.878
	gowers	0.371	0.212	0.524
	lmnn	0.740	0.531	0.868
	mahal_scaled	0.697	0.548	0.828
	wcsm	**0.794**	0.625	0.908
	hierarch_euc_scaled	0.433	0.287	0.615
	hierarch_euc_z	0.752	0.586	0.871
	hierarch_gowers	0.655	0.246	0.807
	hierarch_mah	0.440	0.267	0.607
	kmeans_scaled	0.443	0.284	0.599
	kmeans_z	0.762	0.606	0.879
ICI	csm	0.0394	0.0208	0.0684
	euc_scaled	0.0493	0.0273	0.0816
	euc_z	0.0417	0.0178	0.0703
	gowers	0.0540	0.0326	0.0792
	lmnn	0.0441	0.0188	0.0700
	mahal_scaled	0.0842	0.0390	0.142
	wcsm	0.0403	0.0201	0.0652
	hierarch_euc_scaled	0.0325	0.0161	0.0558
	hierarch_euc_z	0.0253	0.0131	0.0460
	hierarch_gowers	0.0318	0.0170	0.184
	hierarch_mah	0.0357	0.0176	0.0606
	kmeans_scaled	0.0306	0.0137	0.0541
	kmeans_z	**0.0251**	0.0134	0.0442
BrS	csm	0.0606	0.0449	0.0752
	euc_scaled	0.0618	0.0464	0.0776
	euc_z	0.0460	0.0313	0.0651
	gowers	0.0764	0.0531	0.0970
	lmnn	0.0460	0.0317	0.0640
	mahal_scaled	0.0620	0.0432	0.0836
	wcsm	**0.0413**	0.0267	0.0555
	hierarch_euc_scaled	0.0698	0.0501	0.0905
	hierarch_euc_z	0.0426	0.0282	0.0596
	hierarch_gowers	0.0535	0.0367	0.192
	hierarch_mah	0.0700	0.0517	0.0930
	kmeans_scaled	0.0691	0.0507	0.0910
	kmeans_z	0.0416	0.0271	0.0588

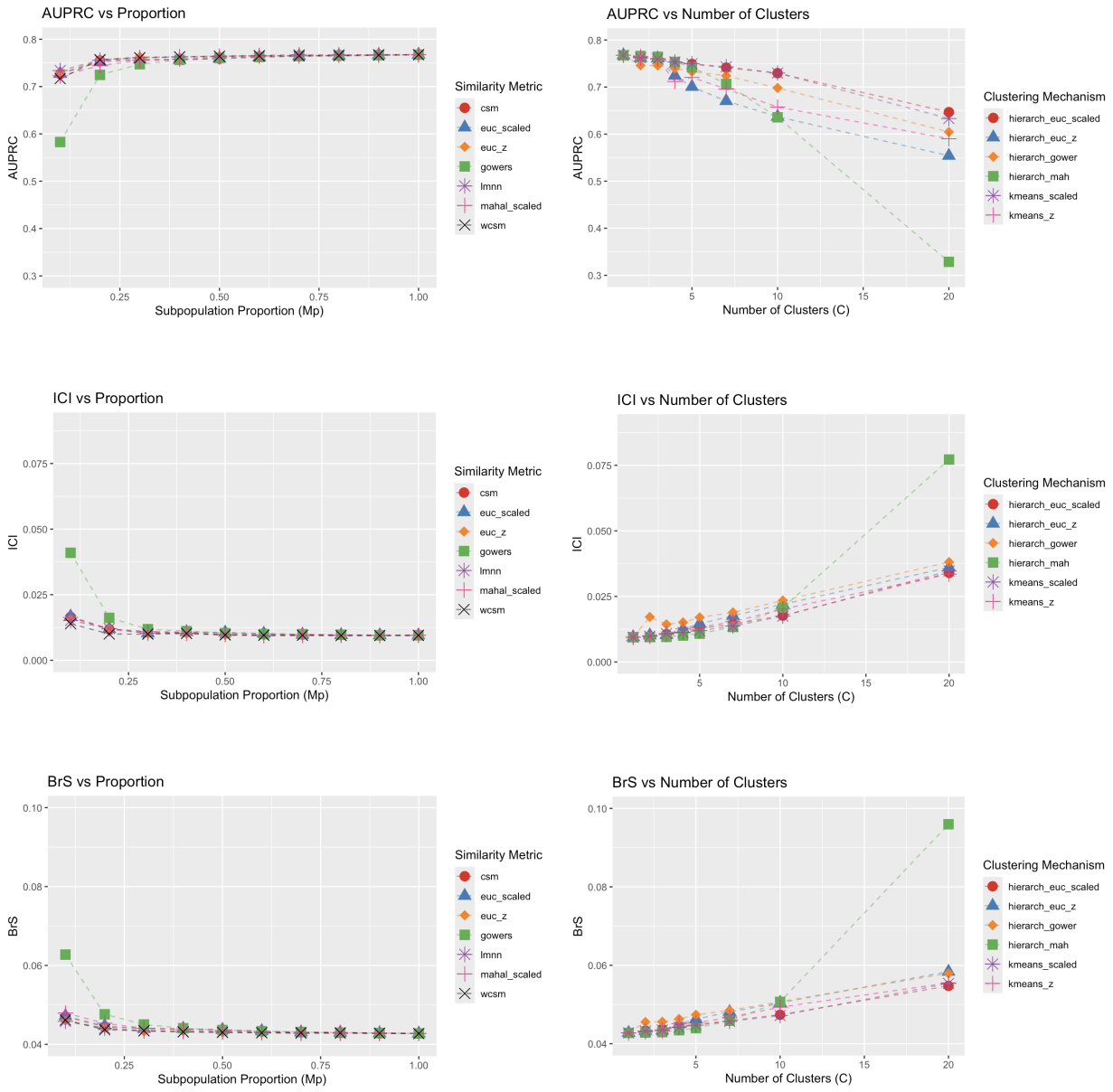


Figure 4.5: Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has low prevalence with the fifth data generation method (Case 9).

Table 4.5: Results from the validation study for Case 9. $M_p = 1$ and $C = 1$ were used from the tuning process.

Metric	Similarity measure	Est	2.5% BCa	97.5% BCa
AUPRC	csm	0.787	0.639	0.883
	euc_scaled	0.787	0.639	0.883
	euc_z	0.787	0.639	0.883
	gowers	0.787	0.639	0.883
	lmnn	0.787	0.639	0.883
	mahal_scaled	0.787	0.639	0.883
	wcsm	0.787	0.639	0.883
	hierarch_euc_scaled	0.787	0.639	0.883
	hierarch_euc_z	0.787	0.639	0.883
	hierarch_gowers	0.787	0.639	0.883
	hierarch_mah	0.787	0.639	0.883
	kmeans_scaled	0.787	0.639	0.883
	kmeans_z	0.787	0.639	0.883
ICI	csm	0.0219	0.00967	0.0553
	euc_scaled	0.0219	0.00967	0.0553
	euc_z	0.0219	0.00967	0.0553
	gowers	0.0219	0.00967	0.0553
	lmnn	0.0219	0.00967	0.0553
	mahal_scaled	0.0219	0.00967	0.0553
	wcsm	0.0219	0.00967	0.0553
	hierarch_euc_scaled	0.0219	0.00967	0.0553
	hierarch_euc_z	0.0219	0.00967	0.0553
	hierarch_gowers	0.0219	0.00967	0.0553
	hierarch_mah	0.0219	0.00967	0.0553
	kmeans_scaled	0.0219	0.00967	0.0553
	kmeans_z	0.0219	0.00967	0.0553
BrS	csm	0.0367	0.0249	0.0511
	euc_scaled	0.0367	0.0249	0.0511
	euc_z	0.0367	0.0249	0.0511
	gowers	0.0367	0.0249	0.0511
	lmnn	0.0367	0.0249	0.0511
	mahal_scaled	0.0367	0.0249	0.0511
	wcsm	0.0367	0.0249	0.0511
	hierarch_euc_scaled	0.0367	0.0249	0.0511
	hierarch_euc_z	0.0367	0.0249	0.0511
	hierarch_gowers	0.0367	0.0249	0.0511
	hierarch_mah	0.0367	0.0249	0.0511
	kmeans_scaled	0.0367	0.0249	0.0511
	kmeans_z	0.0367	0.0249	0.0511

Table 4.6: A table summarizing important findings from the simulation study.

Dataset	**Findings**
Cases 1 and 2	Using weightbeta CSM leads to the best discrimination and BrS in PPM. Using standard CSM leads to the best calibration in PPM.
Cases 5 and 6	Clustering methods lead to better discrimination and calibration in PPM than using similarity metrics with k-NN methods.
Cases 7 and 8	Clustering methods lead to better calibration in PPM than using similarity metrics with k-NN methods.
Cases 3, 4, 9, and 10	Training a model on personalized data has minimal improvement in the predictive performance of a model compared to training the model on all available data. In other words, we do not see an advantage in using PPM for these datasets.
Overall	Generally, wCSM, LMNN, and Euclidean distance with normalized continuous predictors perform well among the k-NN methods. k-means clustering and hierarchical clustering with Euclidean distance and normalized continuous predictors perform well among the clustering methods. Results are data-dependent in terms of whether to use k-NN or clustering methods.

4.4 Discussion

With PPM, we can improve the prediction accuracy for a participant of interest by training a predictive model on a targeted, customized dataset that contains only the most similar participants to the participant of interest. Therefore, identifying the correct subset of the training data to fit the personalized model is crucial. This is especially true since the personalized model will be smaller in size compared to a global model, but for parametric models, such as generalized linear models, sufficient data are needed to train the model. Similarity metrics are a popular research topic in the PPM literature. In this chapter, we provide a guideline for which patient similarity metric should be used by systematically comparing similarity metrics across various datasets that were produced with different characteristics. These metrics can be broadly grouped into two categories: k-NN and clustering.

For the k-NN approach, we identify the top k most similar participants to the partic-

ipant of interest to train the model. The value of k is determined by tuning the optimal subpopulation proportion M_p using repeated cross-validation on the training-testing data. Ultimately, $k = n \times M_p$, where n is the total number of observations in the training data. The similarity metrics compared are CSM, weightbeta CSM from Chapter 3, Mahalanobis distance, Gower's distance, LMNN, and Euclidean distance, with continuous predictors that are standardized using z-score normalization or min-max scaling between 0 and 1.

For the clustering approach, we create clusters of the training data and fit a predictive model to each cluster. Then we calculate the distance from the cluster centres to the participant of interest to identify the closest cluster, and use the model trained on that cluster to predict the outcome for that participant. Unlike the k-NN method, we do not tune for the size of the subpopulation proportion; instead, we tune for the number of clusters C. We investigate the performance of k-means clustering and hierarchical clustering using Gower's distance, Mahalanobis distance, and Euclidean distance. For Euclidean distance used in both k-means and hierarchical clustering, we standardize the continuous predictors using z-score normalization and min-max scaling between 0 and 1. We use k-means clustering and hierarchical clustering because these are the most commonly used forms of clustering in patient similarity (Parimbelli et al., 2018).

We conduct an extensive simulation study to investigate how these patient similarity metrics behave and if they can improve the predictive performance of the model. In total, ten simulation cases were considered: all combinations of two levels of the outcome prevalence at 0.1 and 0.5, and five data-generating mechanisms. With our simulation and data generation setup, including but not limited to the parameter values, number of predictors, number of predictors used for generating the outcome, and the number of observations, our findings are as follows. When the same outcome-generating equation is applied on all predictors without any transformations on the predictors, the performance of the model does not improve with PPM (Case 3 and Case 4). Even if there is an improvement, this is minimal. When latent clusters exist only in the predictors and not in outcome-generation as well, there is no improvement in the predictive performance of the model with PPM if the model does not have any transformations on the predictors (Case 9 and Case 10). Again, if there is any improvement, it is minimal.

However, there is an improvement in the predictive performance of the model after implementing PPM when at least one of the following holds true: (1) some predictors are transformed before generating the outcome, and (2) the outcome-generating equation is cluster-specific when there are latent clusters in the predictors. In Case 1 and Case 2, there are no latent clusters in the predictors; however, two of the predictors used to generate the model are transformed. In Case 5 and Case 6, latent clusters exist in the predictors, and these clusters have their own specified outcome-generating equation. Finally, in Cases

7 and 8, latent clusters exist, and a single outcome-generating equation is used across all clusters, but this equation includes transformations on two predictors used to generate the outcome. Our simulation results show that in Cases 1, 2, 5, 6, 7, and 8, training a model on a personalized subset of the training data leads to a better predictive performance of the generalized linear model than using a global model.

Depending on the data generation mechanism discussed in our work, certain similarity metrics perform better than others. In Case 1 and Case 2, where predictors have no clusters but include transformations, weightbeta CSM has the best discrimination. Overall, PPM with k-NN similarity metrics generally performs better than clustering-based metrics. In Case 5 and Case 6 where the predictors and the outcome are cluster-specific, k-means clustering and hierarchical clustering with Euclidean distance and normalized continuous data perform the best. It is followed by LMNN and the Euclidean distance with normalized continuous data from the k-NN approach. For Case 7 and Case 8 where the predictors have latent clusters but there is only one outcome-generating equation across all clusters with a few transformed predictors in the outcome equation, weightbeta CSM performs the best in terms of discrimination. However, clustering methods generally provide better calibration than the distance-based methods used with the k-NN approach.

There are several directions for future work. First, we want to extend the comparison of various similarity metrics to a real-world dataset. We plan to continue this study and use the eICU dataset for patients with circulatory system disease to investigate how the different similarity measures perform on this dataset. Other extensions include going beyond generalized linear models by exploring more complex machine learning approaches, such as random forests and stochastic gradient boosting, and focusing on other types of responses, including continuous, count, and survival outcomes. We also want to explore other patient similarity metrics to expand the comparison, such as the random forest proximity measure. Furthermore, we aim to consider additional scenarios for data generation to better understand how different similarity metrics perform in PPM. Since our results indicate that the similarity metric performance varies depending on the data-generating mechanism, investigating other generation cases will help us better understand how they perform. Finally, we want to incorporate patient-weighting into the clustering methods. After assigning a participant to the nearest cluster, all participants in that cluster are used with equal weight to train the predictive model. However, we plan to investigate how the predictive performance of the model changes when participants closer to the participant of interest are weighted more heavily, and those further away are weighted less. This approach would allow for an additional layer of personalization beyond tuning for C, the number of clusters. Currently, C is tuned using the dataset, and thus is specific to the training data but not to the individual participants. By weighting patients within a clus-

ter according to their similarity to the index participant, we can introduce a more refined, participant-specific level of personalization.

Despite the amount of future work to be investigated, this chapter generates novel findings in a comprehensive overview of the performance of different similarity metrics in PPM. Not only do we compare various distance metrics and cosine similarity metrics with different clustering algorithms, but we also conduct this comparative analysis on various datasets with different data-generating mechanisms to understand how the similarity metrics behave with different data characteristics. Based on the simulation study in this chapter, our suggestions for the choice of similarity metric when implementing PPM are as follows. Determining between using clustering methods or k-NN methods depends on the data characteristics; when latent clusters exist in the predictors, the clustering methods tend to perform better than the k-NN methods. Among the clustering methods, k-means clustering and hierarchical clustering with Euclidean distance with normalized continuous predictors outperform other clustering methods. When using the k-NN methods, weightbeta CSM, LMNN, and Euclidean distance with normalized continuous predictors outperform other metrics.

Chapter 5

Discussion

In this thesis, we explore different areas of advancement in similarity-based prediction modeling. Past research has shown that personalized predictive modeling (PPM) can improve the predictive performance of a statistical model, with the majority of research focusing on the prediction of binary outcomes. This can be measured using various performance measures such as discrimination, calibration, or a combination of both via the Brier score, and it shows that we can improve the prediction accuracy of the model by training the model on a personalized subset of the original training data (Krikella and Dubin, 2025; Lee et al., 2015; Xu et al., 2019). Naturally, this raises a few follow-up questions. First, how can we quantify the similarity between the participants so that we can correctly identify the observations to be included in this personalized subset of the original training data? Next, how do we determine the size of this personalized subset of the training data? We must strike a balance between containing enough data to train the predictive model, but being small enough to have homogeneous and customized data. Finally, can PPM be implemented with complex models and data structures to improve the predictive performance? Chapter 3 and Chapter 4 focus on the first question, where we investigate new similarity metrics to improve the predictive performance of personalized predictive models, and compare existing metrics to evaluate which has superior performance in certain settings. The second question is targeted in Chapters 2, 3, and 4 through the tuning process with repeated cross-validation; the optimal personalized data size is defined as the value that yields the smallest Brier score when the predictive model is trained on a training subset of that size. Finally, the third question is tested in Chapter 2, where we demonstrate that improving the predictive performance of joint models for longitudinal and time-to-event data can be improved with PPM.

Joint models are used to fit longitudinal and time-to-event data, which are often used in

biomarker research and follow-up studies where the time-varying covariates are associated with the event-time response. Furthermore, they have an attractive characteristic in that we can dynamically predict the survival response as more longitudinal measurements become available. In Chapter 2, we introduce two simulation cases: one where the coefficient of the longitudinal biomarker is much greater compared to the coefficients of the fixed covariates, and the second where the coefficient of the biomarker is similar in magnitude to the coefficients of the fixed covariates. While the M_p value that results in the smallest Brier score in repeated cross-validation is different in both cases, our simulation studies have shown that taking the top $M_p \times 100\%$ of the most similar participants in the training data to fit a joint model can improve the predictive performance of the joint model. Our data analysis using the eICU data for patients diagnosed with sepsis also shows that fitting a joint model on a subpopulation proportion of $M_p = 0.6$ can lead to a lower Brier score, indicating a better predictive performance of joint models, though we note the results of the data analysis were not as convincing as those from the comprehensive simulation study.

In Chapter 3, we introduce a new patient similarity metric to quantify the similarity between participants and identify a homogeneous subset of the training data. We extend the existing CSM and propose a new algorithm for quantifying similarity between individuals for selecting personalized subpopulations. We introduce a weighted CSM, where the weights of each predictor are determined by a relaxed adaptive group lasso. The two proposed methods are weight01 CSM, which assigns a weight of 0 or 1 to the available predictors, and weightbeta CSM, which assigns a weight proportionate to the size of the predictor's beta estimate. These metrics are used to calculate the similarity between the participants in the training data and the participant of interest to determine the subset of the training data used to fit a personalized model. In a simulation study containing six cases - two levels of prevalence and three levels of association between the predictors and the outcome - we compare the discrimination and calibration of a personalized predictive model after implementing the standard CSM, weight01 CSM, and weightbeta CSM. While weightbeta CSM leads to reduced calibration performance, the improvement in discrimination is substantial enough to offset this loss, leading to the best overall performance (as per the Brier Score). We apply our method to the eICU database for patients diagnosed with circulatory system disease and demonstrate that using our new similarity metric can lead to a better predictive performance in the model, especially with smaller M_p values.

In Chapter 4, we perform a systematic comparison of various similarity metrics to investigate how they behave compared to each other and based on the dataset. We use two main algorithms: the k-NN method, where we identify the k most similar individuals to train the personalized model, and the clustering method, where we create clusters using the training data and train the predictive model for a participant of interest on the cluster to

which they are closest. To find k, we tune for M_p; to find the number of clusters, we tune for C using the training and testing data. We investigate this through a simulation study that has ten cases, with two levels of prevalence and five data-generating mechanisms. To the best of our knowledge, a direct comparison of the predictive performance of personalized models between clustering and k-NN methods on controlled data-generating mechanisms using a simulation study is a novel contribution to the literature. We have shown that when the predictors have latent clusters, naturally, the clustering methods perform well; in particular, k-means clustering and hierarchical clustering with Euclidean distance and normalized continuous predictors have the best performance among the clustering methods consistently. Among the distance measures that use the k-NN methods, weightbeta CSM from Chapter 3, LMNN, and Euclidean distance with the normalized continuous predictors perform the best.

Thus, we contribute to the advancements in similarity-based prediction modeling as follows: in Chapter 2, we demonstrate that PPM can be extended to complex models, specifically joint longitudinal-survival models, with various mixed data types to improve their predictive performance; in Chapter 3, we introduce a new similarity metric that is an extension of the existing CSM, which leads to a better predictive performance than the standard CSM for a binary response; finally, in Chapter 4, we investigate the effect of different similarity measures on the success of a similarity-based prediction modeling for a binary response.

We conclude this thesis by outlining limitations and several directions for future research. First, the procedures outlined in this research would require the users to have some background in statistics and machine learning to be able to implement our proposed methods in practice. We first need to tune the subpopulation proportion or the number of clusters using the available data to determine the optimal subpopulation size. Then, for a new participant, we construct a personalized dataset by selecting the most similar patients based on this tuning parameter and fit a personalized model on this participant to make predictions. Note that this means that we are fitting a different model for each participant. As a result, the model parameters are not as straightforward to interpret, since the coefficients are specific to the selected subpopulation. While some interpretation is still possible, it is participant-specific and depends on the subset of data used to fit the model. Second, while the prediction models in Chapters 3 and 4 were based on generalized linear models because we had binary outcomes (not to mention computational challenges; see comment on this point below), future work should investigate whether the observed findings extend to other outcome types. In particular, it would be valuable to examine continuous, count, survival, and longitudinal responses, which would require fitting different models and may influence the performance of the proposed similarity metrics. Third,

we want to incorporate more flexible and complex machine learning models in Chapters 3 and 4, such as random forests and stochastic gradient boosting, to evaluate whether the main conclusions remain consistent under other model assumptions. Also, it would be worthwhile to develop R packages that implement the methods proposed in this thesis, while also tuning for the subpopulation proportion or the number of clusters to make a personalized prediction. Furthermore, we want to include additional validation using diverse real-world datasets. In Chapter 2, our analysis focused on patients diagnosed with sepsis, and future studies should assess whether similar conclusions hold for other clinical datasets, such as datasets of individuals with circulatory system diseases. Likewise, for Chapter 4, applying the proposed similarity metrics to real datasets would help determine whether the performance patterns observed in the simulation study are seen in practice. Although the dataset examined in this thesis is derived from healthcare data involving ICU patients, the methods proposed are not limited to this area. These approaches can be extended to other contexts that require precision in predictions, such as utilizing sales or commercial data to generate personalized predictions for consumers. Therefore, it would be valuable to evaluate the effectiveness of the proposed methods in completely different application contexts using datasets beyond the eICU, and investigate how the heterogeneity in the dataset affects the effectiveness of PPM.

Finally, it is computationally intensive to tune for the optimal subpopulation proportion M_p and the number of clusters C for PPM, especially when we use more complex prediction models. Exploring strategies to improve the computational efficiency of our algorithm will surely be worthwhile. However, the computational burden is lower for the clustering methods since only C models need to be fit - one per cluster - rather than fitting a personalized model for every individual in the testing data. This computational advantage will be even more pronounced relative to k-NN methods when parallelizing across folds while tuning for C, or when using more complex and computationally demanding prediction models. Furthermore, this computational burden introduces a trade-off between potential gains in predictive performance from PPM and the time and resources required to tune the subpopulation size. Reflecting on the results presented in Chapter 4, which indicate that the effectiveness of PPM depends on the underlying data structure, it is important to first examine the characteristics of the dataset before implementation. These characteristics include homogeneity in the data, such as whether there are any latent structures in the predictor space, to assess whether the expected improvement in predictive performance is substantial enough to justify the additional computational cost associated with a similarity-based approach.

References

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749.

Austin, P. C. (2012). Generating survival times to simulate cox proportional hazards models with time-varying covariates. *Statistics in medicine*, 31(29):3946–3958.

Austin, P. C. and Steyerberg, E. W. (2019). The integrated calibration index (ici) and related metrics for quantifying the calibration of logistic regression models. *Statistics in medicine*, 38(21):4051–4065.

Bailey, S. R., Loeppky, J., and Swartz, T. B. (2020). The prediction of batting averages in major league baseball. *Stats*, 3(2):84–93.

Barrett, J. and Su, L. (2017). Dynamic predictions using flexible joint models of longitudinal and time-to-event data. *Statistics in medicine*, 36(9):1447–1460.

Bates, S., Hastie, T., and Tibshirani, R. (2023). Cross-validation: what does it estimate and how well does it do it? *Journal of the American Statistical Association*, pages 1–12.

Bone, R. C., Grodzin, C. J., and Balk, R. A. (1997). Sepsis: a new hypothesis for pathogenesis of the disease process. *Chest*, 112(1):235–243.

Campillo-Gimenez, B., Jouini, W., Bayat, S., and Cuggia, M. (2013). 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.

Caruana, E. J., Roman, M., Hernández-Sánchez, J., and Solli, P. (2015). Longitudinal studies. *Journal of thoracic disease*, 7(11):E537.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202.

D'Agostino, R. B. and Nam, B.-H. (2003). Evaluation of the performance of survival analysis models: discrimination and calibration measures. *Handbook of statistics*, 23:1–25.

Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. (2007). Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216.

Fang, H. S. A., Tan, N. C., Tan, W. Y., Oei, R. W., Lee, M. L., and Hsu, W. (2021). Patient similarity analytics for explainable clinical risk prediction. *BMC medical informatics and decision making*, 21(1):207.

Fitzmaurice, G. M., Laird, N. M., and Ware, J. H. (2012). *Applied longitudinal analysis*. John Wiley & Sons.

Gosho, M., Nagashima, K., and Sato, Y. (2012). Study designs and statistical analyses for biomarker research. *Sensors*, 12(7):8966–8986.

Gottlieb, A., Stein, G. Y., Ruppin, E., and Sharan, R. (2011). Predict: a method for inferring novel drug indications with application to personalized medicine. *Molecular systems biology*, 7(1):496.

Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics*, 27(4):857–871.

Gu, D., Liang, C., and Zhao, H. (2017). A case-based reasoning system based on weighted heterogeneous value distance metric for breast cancer diagnosis. *Artificial intelligence in medicine*, 77:31–47.

Hancock, J. T., Khoshgoftaar, T. M., and Johnson, J. M. (2023). Evaluating classifier performance with highly imbalanced big data. *Journal of Big Data*, 10(1):42.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Henderson, R., Diggle, P., and Dobson, A. (2002). Identification and efficacy of longitudinal markers for survival. *Biostatistics*, 3(1):33–50.

Huai, M., Miao, C., Suo, Q., Li, Y., Gao, J., and Zhang, A. (2018). Uncorrelated patient similarity learning. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 270–278. SIAM.

Huang, Y., Li, C., Georgiopoulos, M., and Anagnostopoulos, G. C. (2013). Reduced-rank local distance metric learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 224–239. Springer.

Huang, Y., Wang, N., Liu, H., Zhang, H., Fei, X., Wei, L., and Chen, H. (2019). Study on patient similarity measurement based on electronic medical records. In *MEDINFO 2019: Health and Wellbeing e-Networks for All*, pages 1484–1485. IOS Press.

Ju, H. and Dubin, J. A. (2026). Unsupervised identification of sepsis subpopulations in the eicu database: A multi-method clustering approach with validation. *Computers in Biology and Medicine*, 204:111546.

Krikella, T. and Dubin, J. A. (2025). A personalized predictive model that jointly optimizes discrimination and calibration. *Statistics in Medicine*, 44(10-12):e70077.

Lambert, J., Leutenegger, A.-L., Baudot, A., and Jannot, A.-S. (2025). Improving patient clustering by incorporating structured variable label relationships in similarity measures. *BMC Medical Research Methodology*, 25(1):72.

Lee, J., Maslove, D. M., and Dubin, J. A. (2015). Personalized mortality prediction driven by electronic medical data and a patient similarity metric. *PloS one*, 10(5):e0127428.

Li, B. and Han, L. (2013). Distance weighted cosine similarity measure for text classification. In *Intelligent Data Engineering and Automated Learning–IDEAL 2013: 14th International Conference, IDEAL 2013, Hefei, China, October 20-23, 2013. Proceedings 14*, pages 611–618. Springer.

Li, D., Alan, S., and Liu, M. (2025a). A comparative analysis of patient similarity measures for outcome prediction. *AMIA Summits on Translational Science Proceedings*, 2025:270.

Li, D., Shukla, A., Chandaka, S., Taylor, B., Xu, J., and Liu, M. (2025b). Autoencoder-based representation learning for similar patients retrieval from electronic health records: Comparative study. *JMIR Medical Informatics*, 13:e68830.

Lin, D. et al. (1998). An information-theoretic definition of similarity. In *Icml*, volume 98, pages 296–304.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2:49–55.

Meinshausen, N. (2007). Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1):374–393.

Murphy, A. H. (1973). A new vector partition of the probability score. *Journal of Applied Meteorology and Climatology*, 12(4):595–600.

Ng, K., Sun, J., Hu, J., and Wang, F. (2015). Personalized predictive modeling and risk factor identification using patient similarity. *AMIA Summits on Translational Science Proceedings*, 2015:132.

Oh, S. H., Back, S., and Park, J. (2021). Measuring patient similarity on multiple diseases by joint learning via a convolutional neural network. *Sensors*, 22(1):131.

Ozenne, B., Subtil, F., and Maucort-Boulch, D. (2015). The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases. *Journal of clinical epidemiology*, 68(8):855–859.

Pai, S. and Bader, G. D. (2018). Patient similarity networks for precision medicine. *Journal of Molecular Biology*, 430(18, Part A):2924–2938. Theory and Application of Network Biology Toward Precision Medicine.

Panahiazar, M., Taslimitehrani, V., Pereira, N. L., and Pathak, J. (2015). Using ehrs for heart failure therapy recommendation using multidimensional patient similarity analytics. *Studies in health technology and informatics*, 210:369.

Parimbelli, E., Marini, S., Sacchi, L., and Bellazzi, R. (2018). Patient similarity for precision medicine: A systematic review. *Journal of biomedical informatics*, 83:87–96.

Pollard, T. J., Johnson, A. E., Raffa, J. D., Celi, L. A., Mark, R. G., and Badawi, O. (2018). The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):1–13.

Posit team (2023). *RStudio: Integrated Development Environment for R*. Posit Software, PBC, Boston, MA.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rizopoulos, D. (2012). *Joint models for longitudinal and time-to-event data: With applications in R.* CRC press.

Rizopoulos, D. (2016). The r package jmbayes for fitting joint models for longitudinal and time-to-event data using mcmc. *Journal of Statistical Software*, 72(7):1–46.

Rizopoulos, D. (2023). Joint modeling of longitudinal and time-to-event data with applications in r.

Rizopoulos, D., Molenberghs, G., and Lesaffre, E. M. (2017). Dynamic predictions with time-dependent covariates in survival analysis using joint modeling and landmarking. *Biometrical Journal*, 59(6):1261–1276.

Rizopoulos, D., Papageorgiou, G., and Miranda Afonso, P. (2023). *JMbayes2: Extended Joint Models for Longitudinal and Time-to-Event Data*. R package version 0.4-5.

Rizopoulos, D. and Taylor, J. M. (2024). Optimizing dynamic predictions from joint models using super learning. *Statistics in Medicine*.

Sharafoddini, A., Dubin, J. A., Lee, J., et al. (2017). Patient similarity in prediction models based on health data: a scoping review. *JMIR medical informatics*, 5(1):e6730.

Sidorov, G., Gelbukh, A., Gómez-Adorno, H., and Pinto, D. (2014). Soft similarity and soft cosine measure: Similarity of features in vector space model. *Computación y Sistemas*, 18(3):491–504.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Suo, Q., Zhong, W., Ma, F., Ye, Y., Huai, M., and Zhang, A. (2018). Multi-task sparse metric learning for monitoring patient similarity progression. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 477–486.

Ullah, S. and Finch, C. F. (2013). Applications of functional data analysis: A systematic review. *BMC medical research methodology*, 13:1–12.

Van Calster, B., McLernon, D. J., Van Smeden, M., Wynants, L., and Steyerberg, E. W. (2019). Calibration: the achilles heel of predictive analytics. *BMC medicine*, 17(1):230.

Wang, H. and Leng, C. (2008). A note on adaptive group lasso. *Computational statistics & data analysis*, 52(12):5277–5286.

Wang, J.-L., Chiou, J.-M., and Müller, H.-G. (2016). Functional data analysis. *Annual Review of Statistics and its application*, 3:257–295.

Wang, N., Huang, Y., Liu, H., Fei, X., Wei, L., Zhao, X., and Chen, H. (2019). Measurement and application of patient similarity in personalized predictive modeling based on electronic medical records. *Biomedical engineering online*, 18:1–15.

Wang, N., Huang, Y., Liu, H., Zhang, Z., Wei, L., Fei, X., and Chen, H. (2021). Study on the semi-supervised learning-based patient similarity from heterogeneous electronic medical records. *BMC medical informatics and decision making*, 21:1–13.

Weinberger, K. Q. and Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244.

World Health Organization (2001). Who international programme on chemical safety biomarkers in risk assessment: validity and validations.

Wu, Z. and Palmer, M. (1994). Verb semantics and lexical selection. *arXiv preprint cmp-lg/9406033*.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678.

Xu, Y., Lee, J., and Dubin, J. A. (2019). Similarity-based random survival forest. *arXiv preprint arXiv:1903.01029*.

Zadeh, P., Hosseini, R., and Sra, S. (2016). Geometric mean metric learning. In *International conference on machine learning*, pages 2464–2471. PMLR.

Zhan, M., Cao, S., Qian, B., Chang, S., and Wei, J. (2016). Low-rank sparse feature selection for patient similarity learning. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1335–1340. IEEE.

Zhou, Y., Bhattacharjee, S., Carroll, C., Chen, Y., Dai, X., Fan, J., Gajardo, A., Hadjipantelis, P. Z., Han, K., Ji, H., Zhu, C., Müller, H.-G., and Wang, J.-L. (2022). *fdapace: Functional Data Analysis and Empirical Dynamics*. R package version 0.5.9.

Zimmerman, J. E., Kramer, A. A., McNair, D. S., and Malila, F. M. (2006). Acute physiology and chronic health evaluation (apache) iv: hospital mortality assessment for today's critically ill patients. *Critical care medicine*, 34(5):1297–1310.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429.

APPENDICES

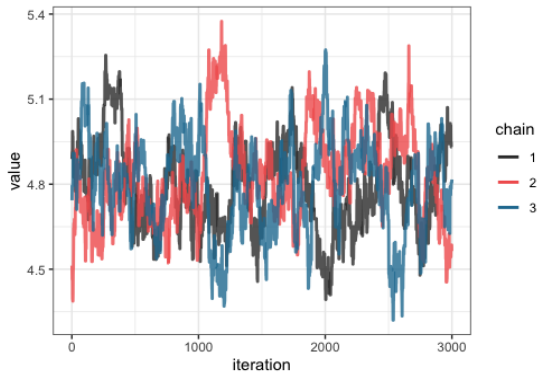
Appendix A

Chapter 2 Appendix

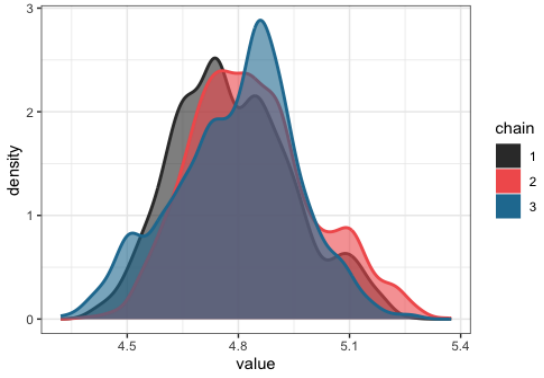
A.1 A Demonstration of Dynamic Prediction using a Joint Model

Below, we provide more detail on dynamically predicting the survival probability of a new participant using the **JMbayes2** package in R (Rizopoulos et al., 2023) and demonstrate how to fit the joint model in Equation 2.3.1. To do so, we first generate longitudinal and time-to-event data as described in Section 2.3.1 and train a joint model on 1800 participants that are randomly selected using noninformative priors. The parameter estimates from fitting the model are shown in Table A.1, with Figures A.1, A.2, and A.3 showing the traceplots and the density plots for each parameter estimate. The default number of iterations per chain with **JMbayes2** is 4000, with 500 burn-in iterations per chain. The traceplot of α showed evidence of greater divergence than other parameters, and the convergence seemed to improve with an increase in the number of chains. However, this came with a substantial computational cost and the potential scale reduction factor was 1.05 with 4000 iterations per chain, so we decided to run the simulation study with the default number of iterations provided in the package.

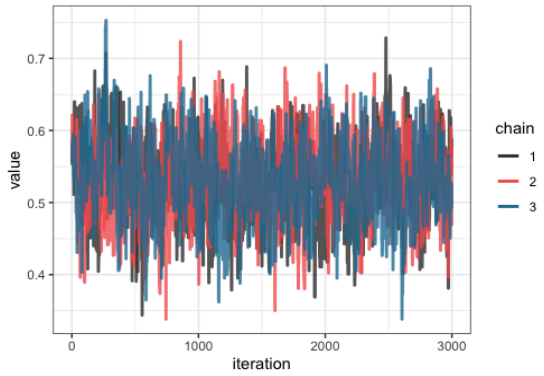
(a) Traceplot of α.



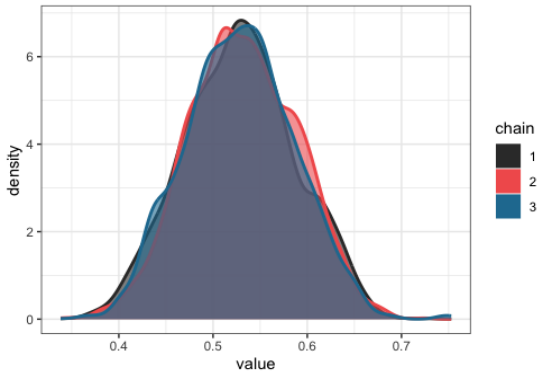
(b) Density plot of α.



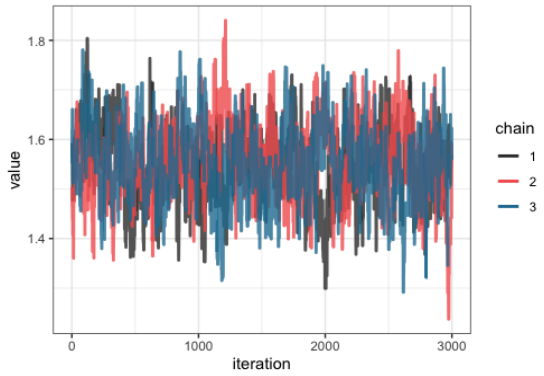
(c) Traceplot of γ_1.



(d) Density plot of γ_1.



(e) Traceplot of γ_2.



(f) Density plot of γ_2.

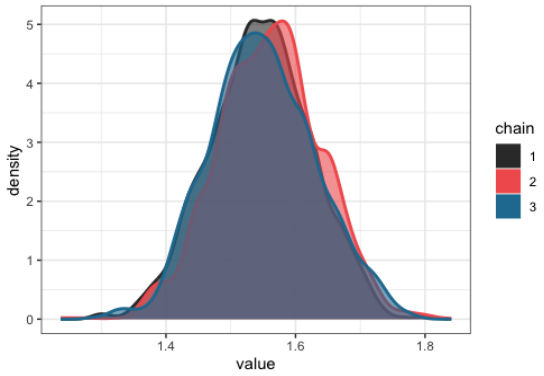
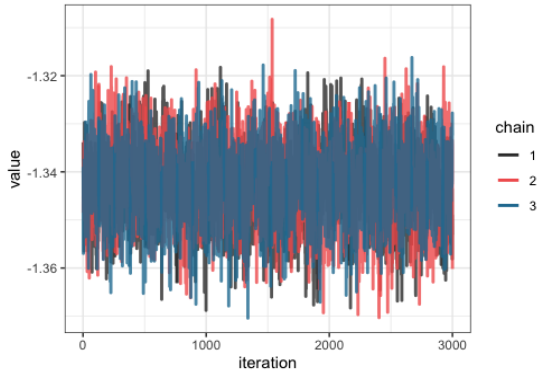
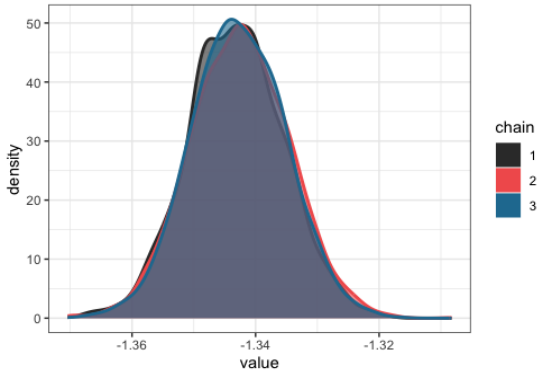


Figure A.1: Traceplots and density plots of the estimated survival parameters in the joint model using **JMbayes2** package in R, showing the convergence in the MCMC process.

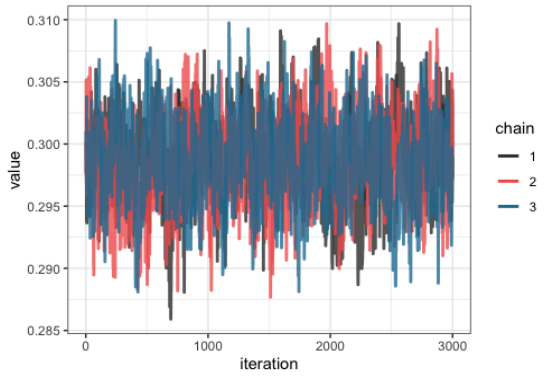
(a) Traceplot of β_0.



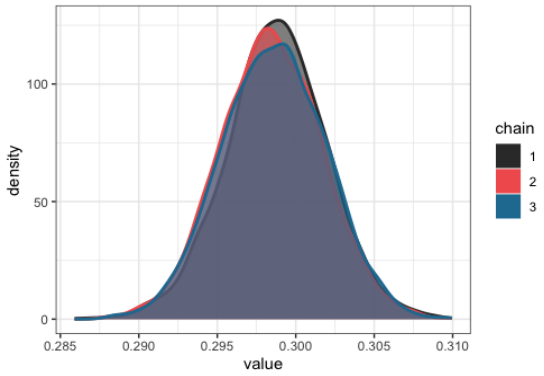
(b) Density plot of β_0.



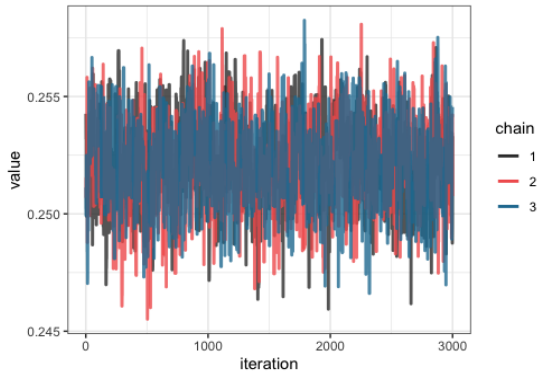
(c) Traceplot of β_0.



(d) Density plot of β_1.



(e) Traceplot of σ.



(f) Density plot of σ.

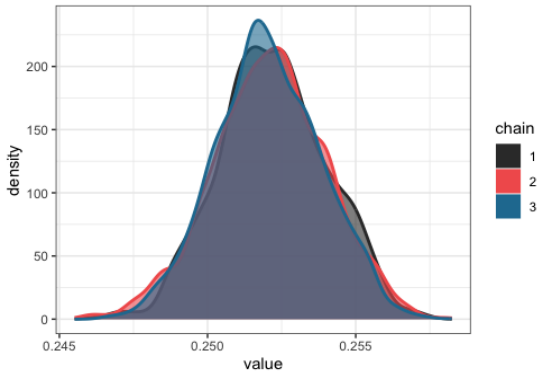
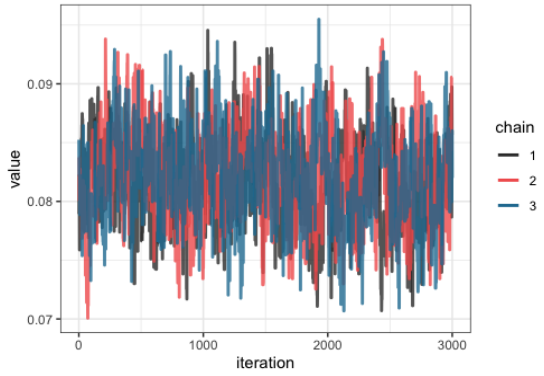
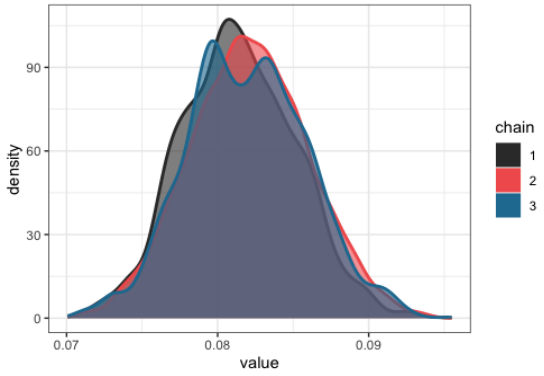


Figure A.2: Traceplots and density plots of the estimated longitudinal parameters in the joint model using **JMbayes2** package in R, showing the convergence in the MCMC process.

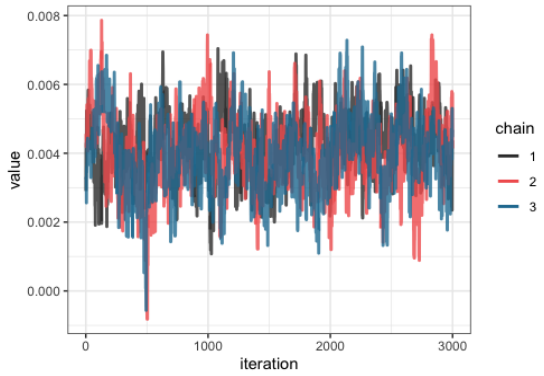
(a) Traceplot of τ_0^2.



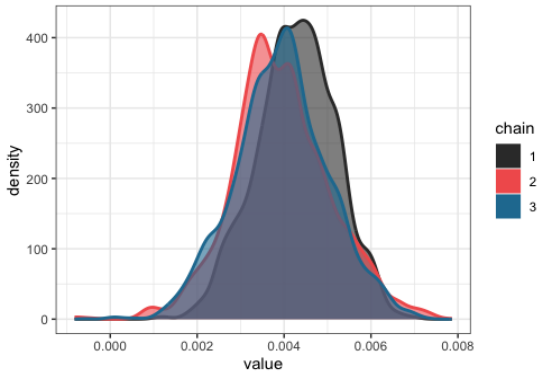
(b) Density plot of τ_0^2.



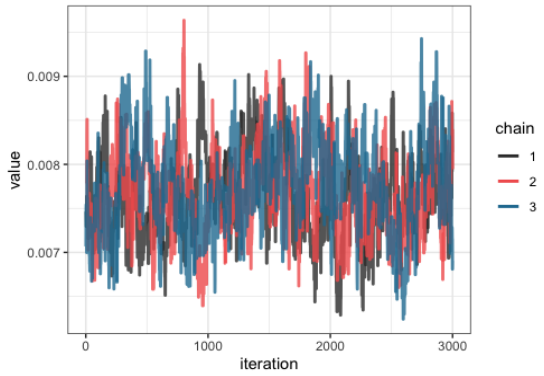
(c) Traceplot of $\tau_0\tau_{01}\tau_1$.



(d) Density plot of $\tau_0\tau_{01}\tau_1$.



(e) Traceplot of τ_1^2.



(f) Density plot of τ_1^2.

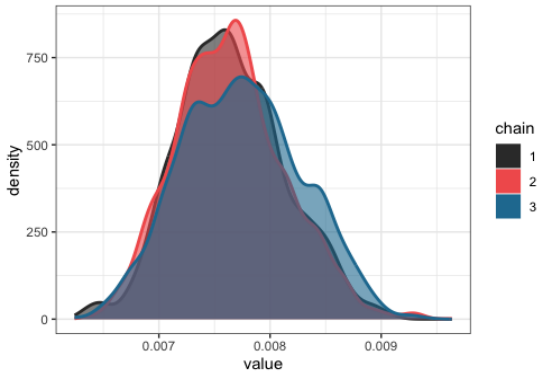


Figure A.3: Traceplots and density plots of the estimated elements of matrix **D** and τ_1 in the joint model using **JMbayes2** package in R, showing the convergence in the MCMC process.

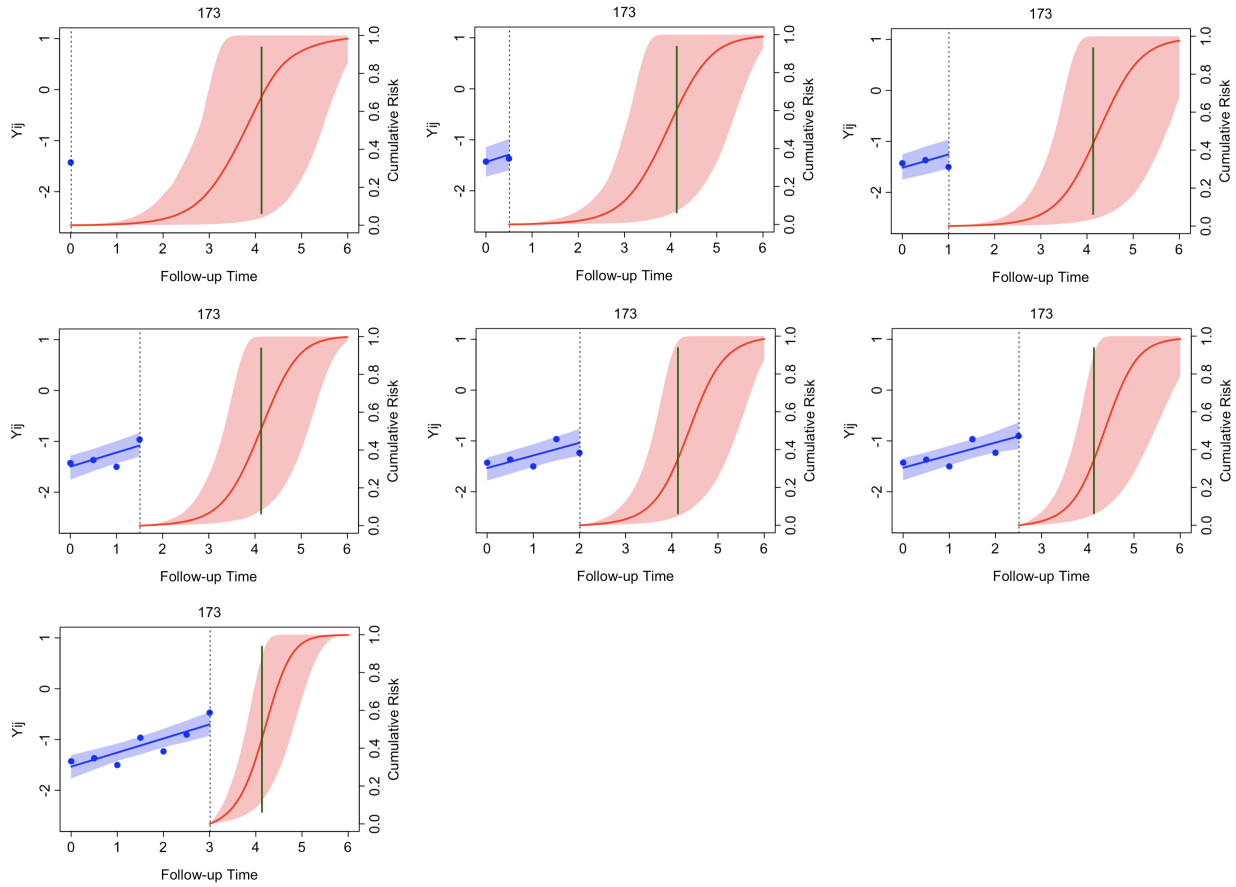


Figure A.4: Plots showing the dynamic prediction of cumulative risk for participant ID 173 as additional longitudinal measurements become available.

As previously mentioned, a major advantage of using a joint model with longitudinal and time-to-event data is that we can dynamically update the prediction of the survival probability as more longitudinal information becomes available, and these predictions are personalized for each participant in the study. For example, consider an individual with ID 173, who was not included in the sample of 1800 participants used to train the joint model. Using the fitted model, we can estimate the cumulative risk, or 1 minus the survival probability, of this participant as more longitudinal measurements become available. Figure A.4 visualizes this - the blue dots represent each longitudinal measurement as time increases in increments of 0.5 and the red line depicts the predicted cumulative risk with the 95% percentile interval around the prediction. The green vertical line represents the true, unobserved time of the event for this participant. Since the plot ends before longitudinal

Table A.1: Estimated parameter values from fitting the joint model in 2.3.1 with data generated using a seed of 1 following the procedure outlined in Section 2.2.1.

Parameter	**True value**	**Estimate**	**2.5%**	**97.5%**
γ_1	0.5	0.5297	0.4203	0.6399
γ_2	1.5	1.5517	1.3967	1.7091
α	4.5	4.8062	4.4944	5.1426
β_0	-1.35	-1.3428	-1.3579	-1.3274
β_1	0.3	0.2986	0.2922	0.3050
σ	0.25	0.2521	0.2485	0.2556
$D[1,1]$	0.0729	0.0818	0.07436	0.08964
$D[1,2]$	0.00432	0.0041	0.00195	0.00610
$D[2,2]$	0.0064	0.00768	0.00676	0.00870

measurements are available closer to the green line, we can see that this participant was censored sometime between time 3 and 3.5. The sequential plots show that as we gain more biomarker information, the red curve showing the predicted risk is updated. Furthermore, note that the cumulative risk always starts from 0 after each biomarker measurement because calculating the probability is conditional on the individual surviving until the biomarker measurement is obtained.

Appendix B

Chapter 3 Appendix

We provide the simulation results for Case 1, Case 3, Case 4, and Case 6 from the simulation study in Chapter 3. While these results are mentioned in Section 3.3.2, the detailed output from the tuning process on TrTe data and holdout Validation data for each case is shown below. When the prevalence is low at 0.1, we show AUPRC, ICI, and BrS plotted against various M_p values, with the performance measures validated using holdout validation data and the winning M_p of 0.6. When the prevalence is reasonably high at 0.5, we show AUROC, ICI, and BrS plotted against various M_p values, with the performance measures validated using holdout validation data and the winning M_p of 0.1.

B.1 Simulation Results for Case 1

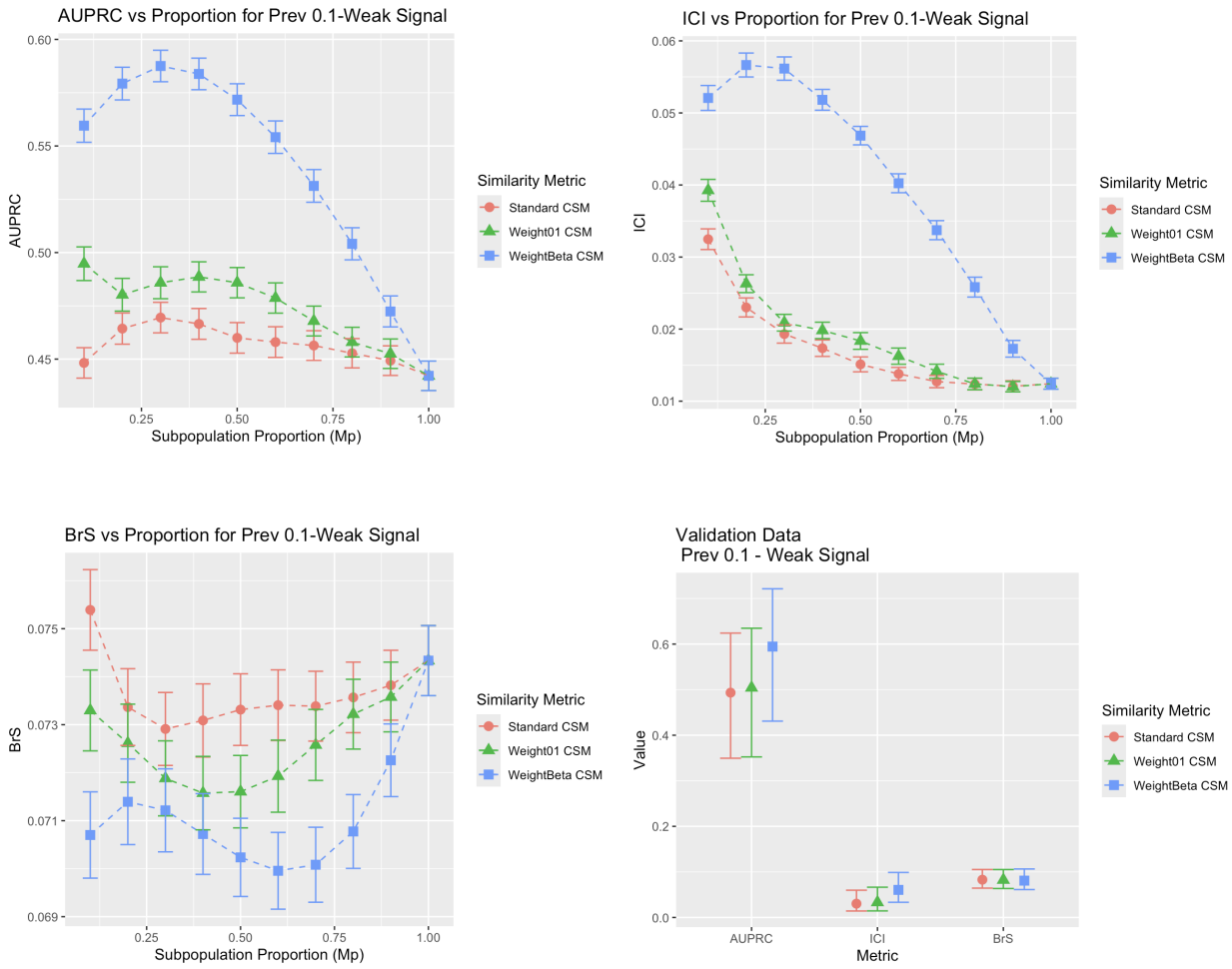


Figure B.1: Performance metrics across different subpopulation proportions for the three similarity measures. The dataset has low prevalence with weak signal (Case 1).

B.2 Simulation Results for Case 3

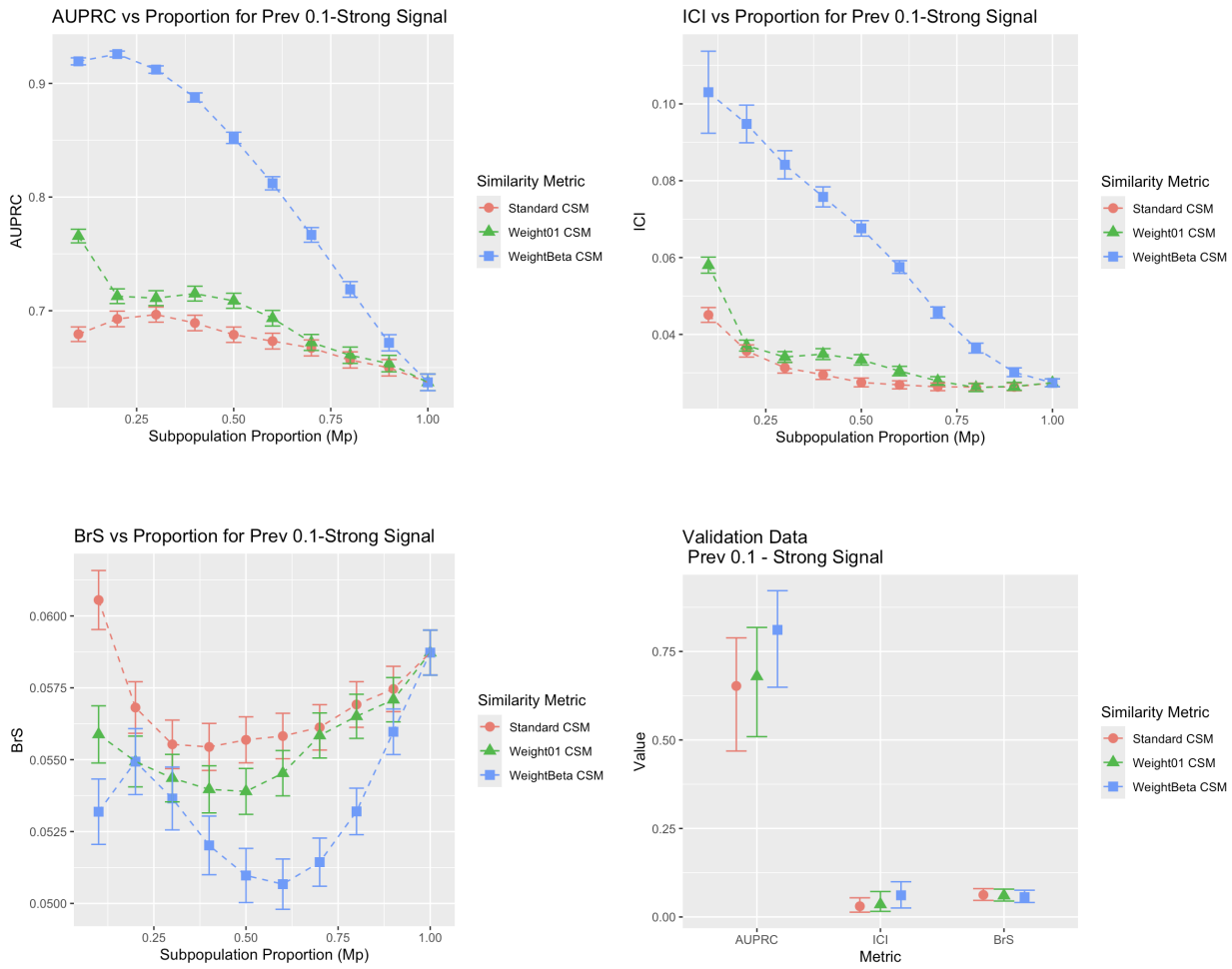


Figure B.2: Performance metrics across different subpopulation proportions for the three similarity measures. The dataset has low prevalence with a strong signal (Case 3).

B.3 Simulation Results for Case 4

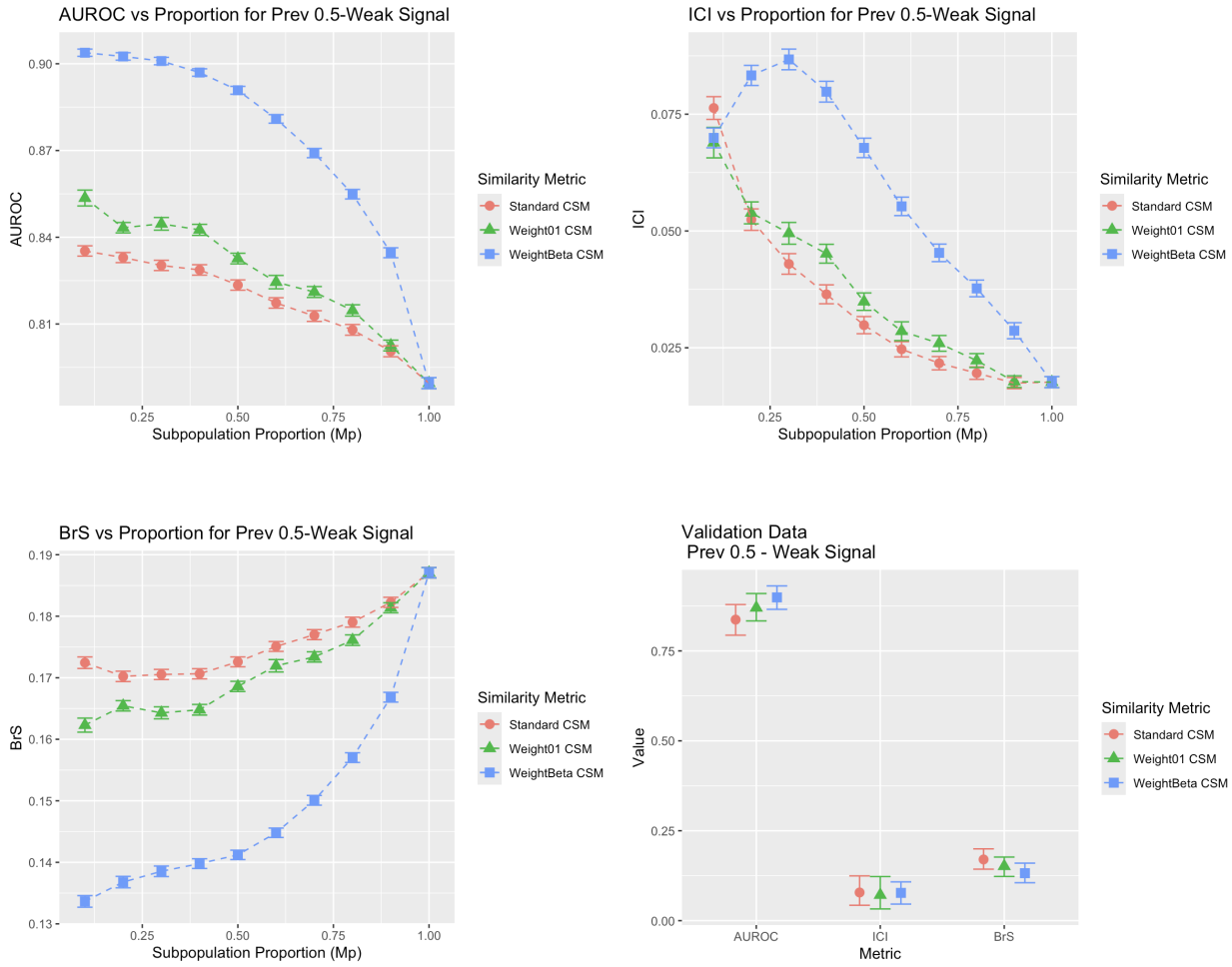


Figure B.3: Performance metrics across different subpopulation proportions for the three similarity measures. The dataset has a high prevalence with a weak signal (Case 4).

B.4 Simulation Results for Case 6

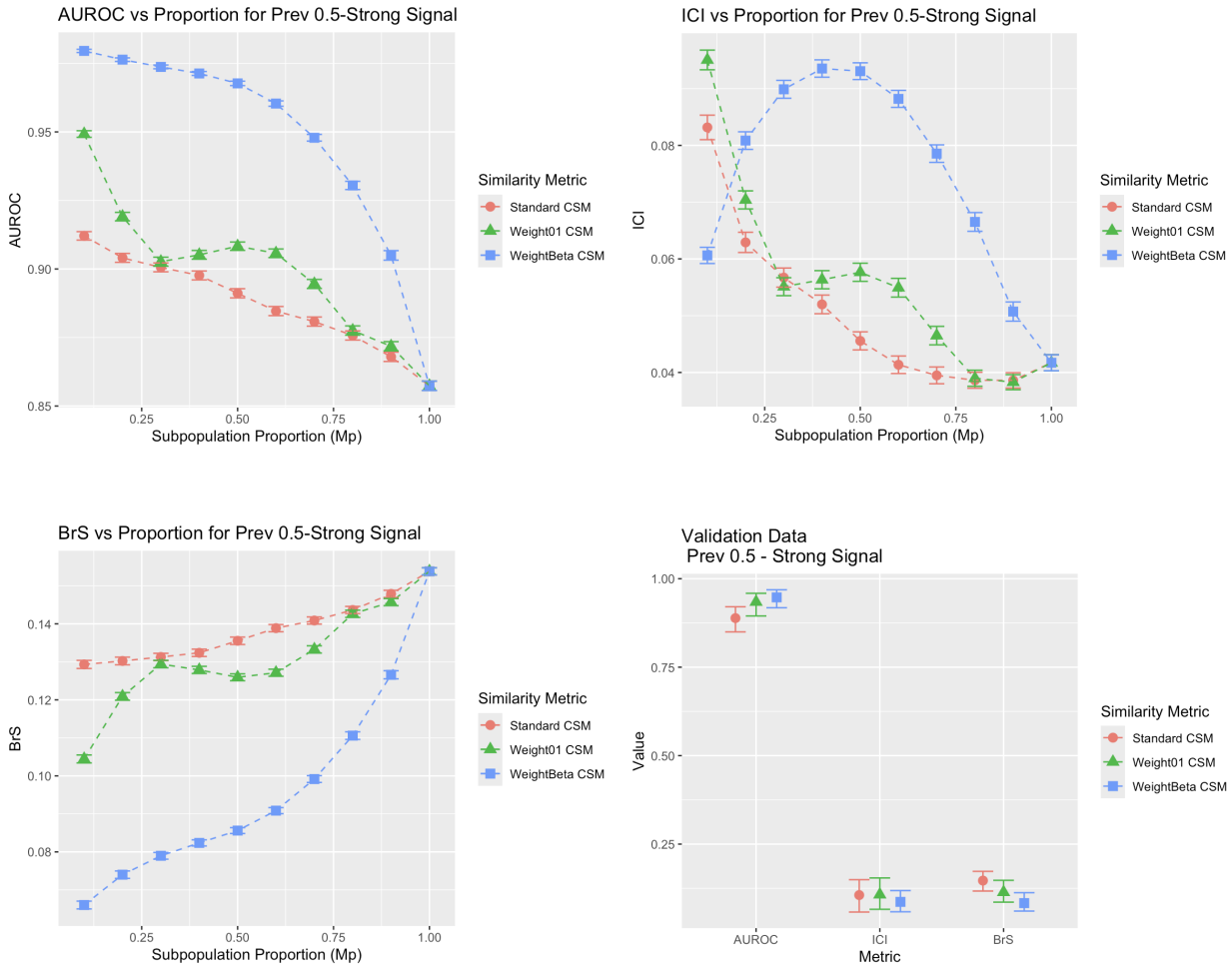


Figure B.4: Performance metrics across different subpopulation proportions for the three similarity measures. The dataset has a high prevalence with a strong signal (Case 6).

Appendix C

Chapter 4 Appendix

We provide the simulation results for Case 2, Case 4, Case 6, Case 8, and Case 10 from the simulation study in Chapter 4. While these results are mentioned in Section 4.3.2, the detailed output from the tuning process using TrTe data and the results from applying the winning M_p and C to the holdout validation data for each case are shown below. When the prevalence is low at 0.1, we show AUPRC, ICI, and BrS plotted against various M_p and C values. When the prevalence is reasonably high at 0.5, we show AUROC, ICI, and BrS plotted against various M_p and C values. The tables show the results from the validation data, which helps us directly compare how the predictive performance changes based on the chosen similarity metric.

C.1 Simulation Results for Case 2

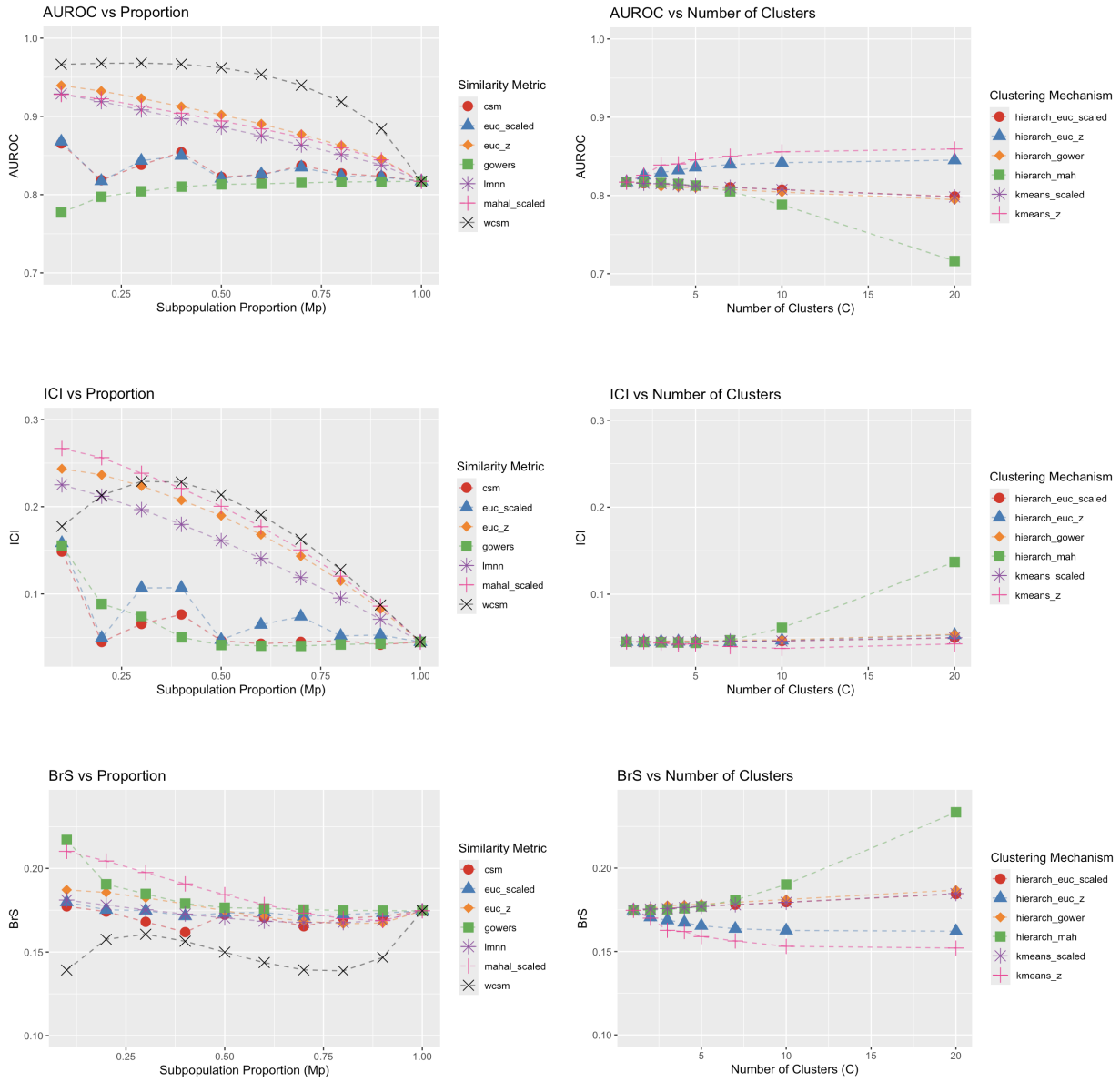


Figure C.1: Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has a high prevalence with the first data generation method (Case 2).

Table C.1: Results from the validation study for Case 2. $M_p = 0.8$ and $C = 20$ were used from the tuning process.

Metric	Similarity measure	Est	2.5% BCa	97.5% BCa
	csm	0.816	0.777	0.855
	euc scaled	0.813	0.771	0.851
	euc_z	0.860	0.828	0.893
	gowers	0.800	0.755	0.837
	lmnn	0.856	0.823	0.890
	mahal_scaled	0.855	0.819	0.889
AUROC	**wcsm**	**0.906**	0.878	0.936
	hierarch_euc_scaled	0.820	0.775	0.864
	hierarch_euc_z	0.775	0.722	0.829
	hierarch_gowers	0.781	0.713	0.833
	hierarch_mah	0.725	0.649	0.782
	kmeans_scaled	0.817	0.774	0.859
	kmeans_z	0.853	0.795	0.890
	csm	**0.0427**	0.0189	0.0776
	euc_scaled	0.0472	0.0182	0.0811
	euc_z	0.123	0.0744	0.168
	gowers	0.0474	0.0206	0.0803
	lmnn	0.118	0.0663	0.161
	mahal_scaled	0.128	0.0792	0.178
ICI	wcsm	0.107	0.0615	0.142
	hierarch_euc_scaled	0.119	0.0809	0.163
	hierarch_euc_z	0.196	0.155	0.245
	hierarch_gowers	0.145	0.0980	0.212
	hierarch_mah	0.136	0.0837	0.203
	kmeans_scaled	0.120	0.0835	0.158
	kmeans_z	0.136	0.101	0.179
	csm	0.175	0.157	0.194
	euc_scaled	0.177	0.159	0.196
	euc_z	0.171	0.151	0.193
	gowers	0.183	0.164	0.202
	lmnn	0.172	0.151	0.195
	mahal_scaled	0.176	0.155	0.198
BrS	**wcsm**	**0.140**	0.123	0.158
	hierarch_euc_scaled	0.183	0.150	0.215
	hierarch_euc_z	0.217	0.176	0.260
	hierarch_gowers	0.206	0.170	0.262
	hierarch_mah	0.230	0.196	0.282
	kmeans_scaled	0.184	0.155	0.215
	kmeans_z	0.152	0.119	0.194

C.2 Simulation Results for Case 4

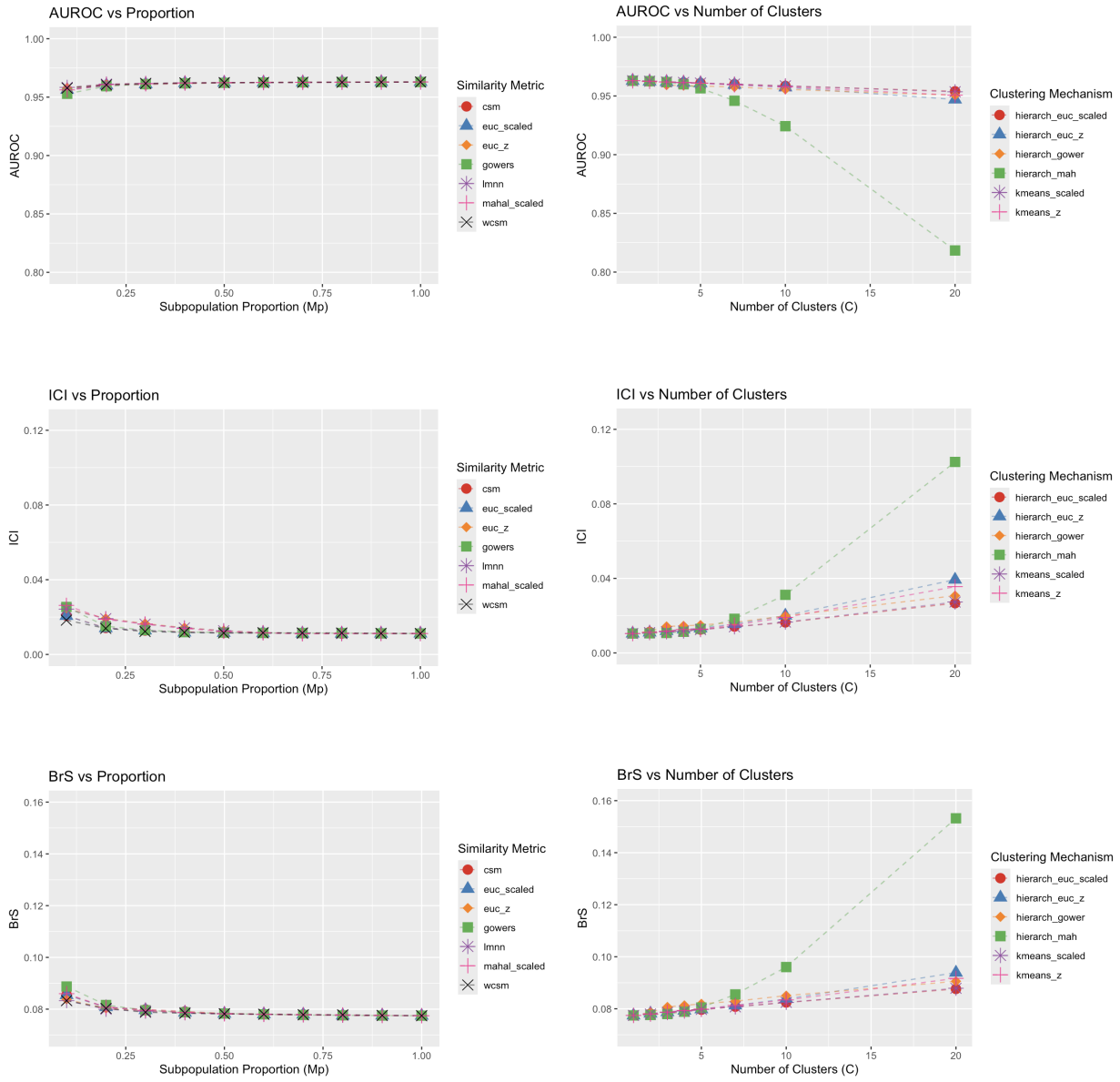


Figure C.2: Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has a high prevalence with the second data generation method (Case 4).

Table C.2: Results from the validation study for Case 4. $M_p = 1$ and $C = 1$ were used from the tuning process.

Metric	Similarity measure	Est	2.5% BCa	97.5% BCa
AUROC	csm	0.960	0.939	0.974
	euc_scaled	0.960	0.939	0.974
	euc_z	0.960	0.939	0.974
	gowers	0.960	0.939	0.974
	lmnn	0.960	0.939	0.974
	mahal_scaled	0.960	0.939	0.974
	wcsm	0.960	0.939	0.974
	hierarch_euc_scaled	0.960	0.939	0.974
	hierarch_euc_z	0.960	0.939	0.974
	hierarch_gowers	0.960	0.939	0.974
	hierarch_mah	0.960	0.939	0.974
	kmeans_scaled	0.960	0.939	0.974
	kmeans_z	0.960	0.939	0.974
ICI	csm	0.0230	0.00811	0.0440
	euc_scaled	0.0230	0.00811	0.0440
	euc_z	0.0230	0.00811	0.0440
	gowers	0.0230	0.00811	0.0440
	lmnn	0.0230	0.00811	0.0440
	mahal_scaled	0.0230	0.00811	0.0440
	wcsm	0.0230	0.00811	0.0440
	hierarch_euc_scaled	0.0230	0.00811	0.0440
	hierarch_euc_z	0.0230	0.00811	0.0440
	hierarch_gowers	0.0230	0.00811	0.0440
	hierarch_mah	0.0230	0.00811	0.0440
	kmeans_scaled	0.0230	0.00811	0.0440
	kmeans_z	0.0230	0.00811	0.0440
BrS	csm	0.0802	0.0629	0.100
	euc_scaled	0.0802	0.0629	0.100
	euc_z	0.0802	0.0629	0.100
	gowers	0.0802	0.0629	0.100
	lmnn	0.0802	0.0629	0.100
	mahal_scaled	0.0802	0.0629	0.100
	wcsm	0.0802	0.0629	0.100
	hierarch_euc_scaled	0.0802	0.0629	0.100
	hierarch_euc_z	0.0802	0.0629	0.100
	hierarch_gowers	0.0802	0.0629	0.100
	hierarch_mah	0.0802	0.0629	0.100
	kmeans_scaled	0.0802	0.0629	0.100
	kmeans_z	0.0802	0.0629	0.100

C.3 Simulation Results for Case 6

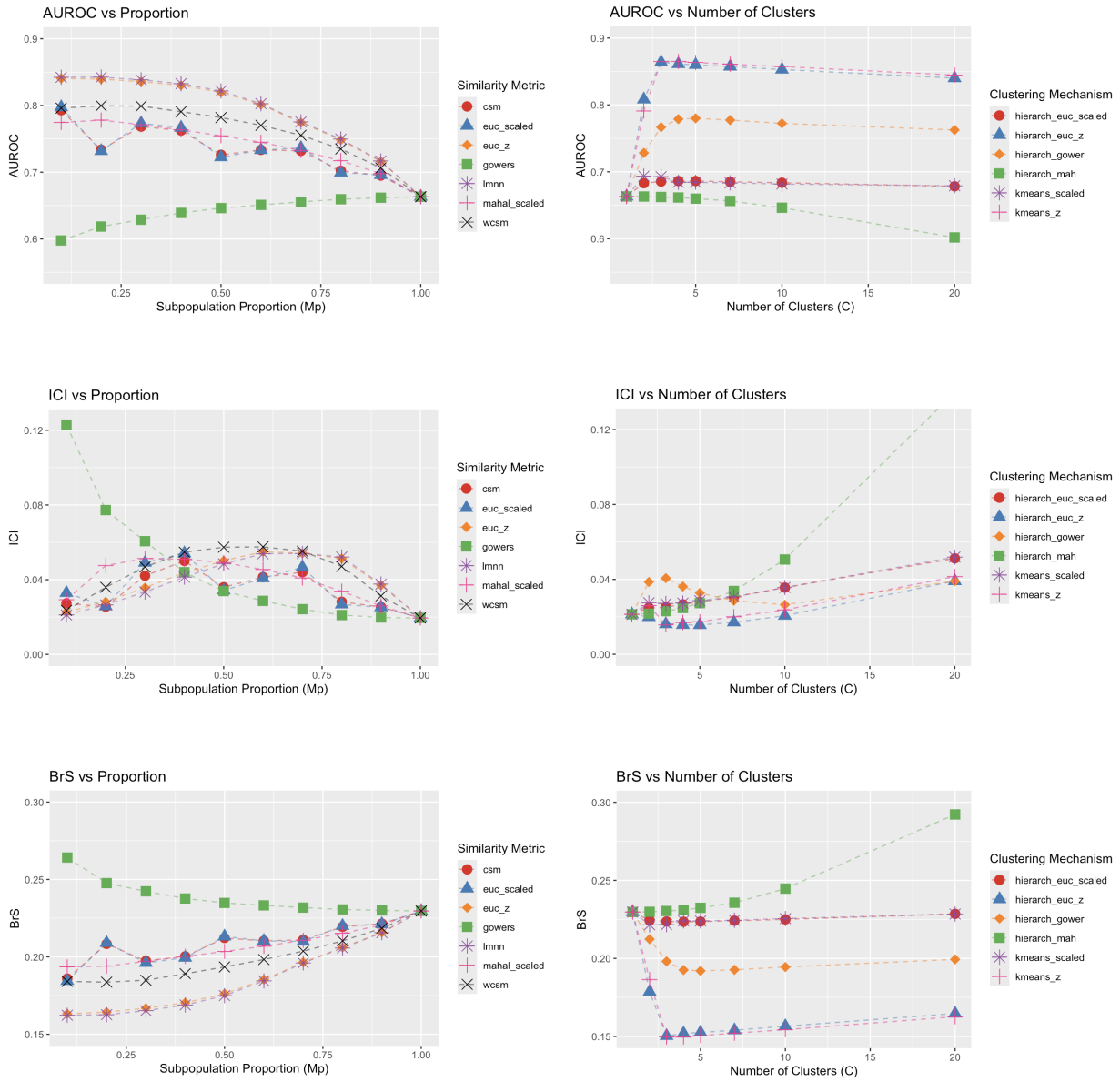


Figure C.3: Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has a high prevalence with the third data generation method (Case 6).

Table C.3: Results from the validation study for Case 6. $M_p = 0.1$ and $C = 3$ were used from the tuning process.

Metric	Similarity measure	Est	2.5% BCa	97.5% BCa
AUROC	csm	0.792	0.747	0.837
	euc_scaled	0.792	0.750	0.838
	euc_z	0.825	0.779	0.863
	gowers	0.569	0.504	0.626
	lmnn	0.823	0.775	0.863
	mahal_scaled	0.777	0.729	0.822
	wcsm	0.802	0.756	0.839
	hierarch_euc_scaled	0.693	0.632	0.743
	hierarch_euc_z	0.840	0.789	0.877
	hierarch_gowers	0.758	0.679	0.810
	hierarch_mah	0.665	0.614	0.717
	kmeans_scaled	0.697	0.636	0.746
	kmeans_z	**0.852**	0.803	0.884
ICI	csm	0.0733	0.0374	0.114
	euc_scaled	0.0737	0.0376	0.115
	euc_z	0.0769	0.0434	0.129
	gowers	0.276	0.226	0.327
	lmnn	0.0769	0.0464	0.131
	mahal_scaled	0.0803	0.0454	0.134
	wcsm	0.0795	0.0454	0.123
	hierarch_euc_scaled	0.0590	0.0253	0.106
	hierarch_euc_z	**0.0422**	0.0155	0.0803
	hierarch_gowers	0.0614	0.0281	0.141
	hierarch_mah	0.0488	0.0229	0.0937
	kmeans_scaled	0.0580	0.0280	0.107
	kmeans_z	**0.0422**	0.0172	0.0781
BrS	csm	0.191	0.165	0.215
	euc_scaled	0.191	0.165	0.215
	euc_z	0.176	0.152	0.206
	gowers	0.336	0.300	0.376
	lmnn	0.177	0.154	0.210
	mahal_scaled	0.199	0.173	0.225
	wcsm	0.187	0.165	0.214
	hierarch_euc_scaled	0.224	0.205	0.248
	hierarch_euc_z	0.164	0.142	0.191
	hierarch_gowers	0.203	0.179	0.249
	hierarch_mah	0.230	0.216	0.246
	kmeans_scaled	0.222	0.203	0.244
	kmeans_z	**0.157**	0.138	0.185

C.4 Simulation Results for Case 8

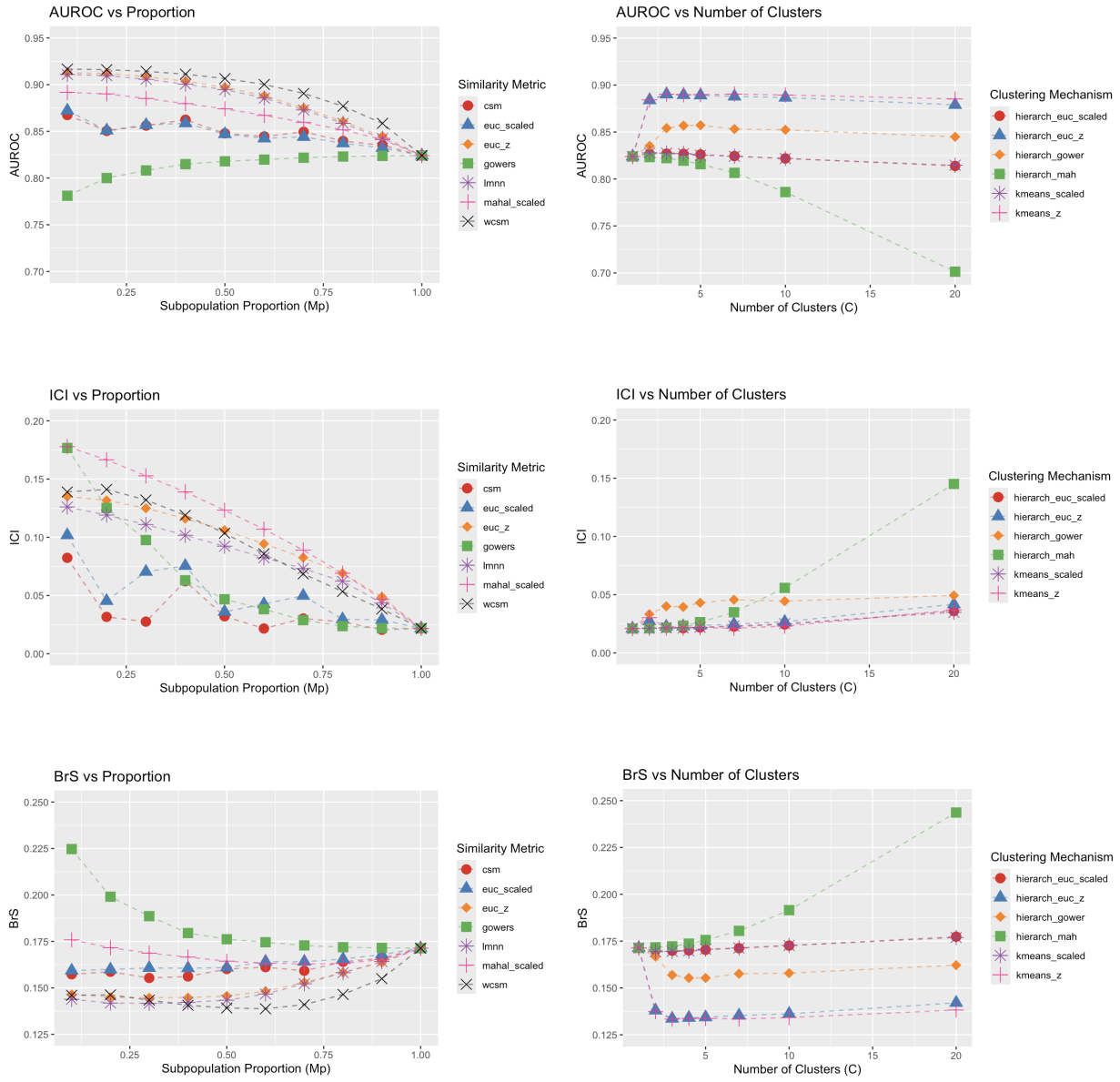


Figure C.4: Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has a high prevalence with the fourth data generation method (Case 8).

Table C.4: Results from the validation study for Case 8. $M_p = 0.6$ and $C = 3$ were used from the tuning process.

Metric	Similarity measure	Est	2.5% BCa	97.5% BCa
AUROC	csm	0.842	0.798	0.875
	euc_scaled	0.843	0.801	0.879
	euc_z	0.884	0.844	0.913
	gowers	0.794	0.750	0.836
	lmnn	0.883	0.844	0.912
	mahal_scaled	0.857	0.817	0.891
	wcsm	**0.896**	0.860	0.924
	hierarch_euc_scaled	0.819	0.780	0.855
	hierarch_euc_z	0.893	0.862	0.918
	hierarch_gowers	0.853	0.764	0.893
	hierarch_mah	0.812	0.772	0.852
	kmeans_scaled	0.824	0.782	0.864
	kmeans_z	**0.896**	0.864	0.922
ICI	csm	0.0347	0.0156	0.0622
	euc_scaled	0.0576	0.0228	0.100
	euc_z	0.0926	0.0468	0.135
	gowers	0.0656	0.0326	0.105
	lmnn	0.0985	0.0508	0.143
	mahal_scaled	0.102	0.0465	0.159
	wcsm	0.0946	0.0502	0.137
	hierarch_euc_scaled	0.0438	0.0175	0.0821
	hierarch_euc_z	0.0328	0.0137	0.0634
	hierarch_gowers	0.0530	0.0211	0.125
	hierarch_mah	0.0399	0.0168	0.0710
	kmeans_scaled	0.0428	0.0155	0.0779
	kmeans_z	**0.0320**	0.0135	0.0596
BrS	CSM	0.163	0.145	0.184
	euc_scaled	0.165	0.146	0.186
	euc_z	0.150	0.129	0.171
	gowers	0.189	0.166	0.213
	LMNN	0.152	0.131	0.175
	mahal_scaled	0.167	0.144	0.189
	wCSM	0.143	0.125	0.164
	hierarch_euc_scaled	0.175	0.155	0.196
	hierarch_euc_z	0.132	0.114	0.153
	hierarch_gowers	0.157	0.132	0.204
	hierarch_mah	0.177	0.158	0.197
	kmeans_scaled	0.172	0.149	0.193
	kmeans_z	**0.130**	0.111	0.151

C.5 Simulation Results for Case 10

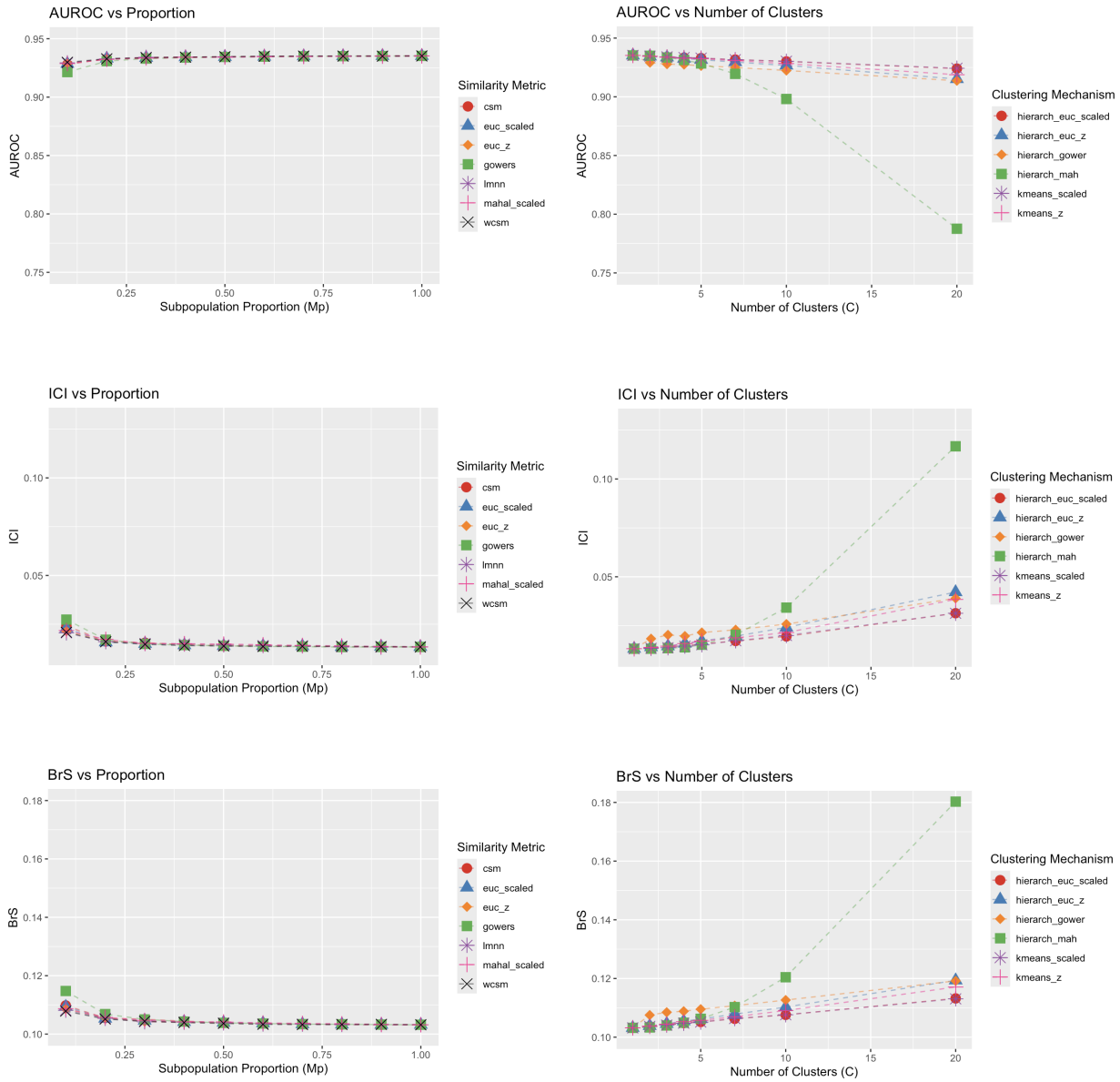


Figure C.5: Performance metrics across different subpopulation proportions and cluster sizes for different similarity metrics. The dataset has a high prevalence with the fifth data generation method (Case 10).

Table C.5: Results from the validation study for Case 10. $M_p = 0.9$ and $C = 1$ were used from the tuning process.

Metric	Similarity measure	Est	2.5% BCa	97.5% BCa
AUROC	csm	0.936	0.913	0.955
	euc_scaled	0.936	0.913	0.955
	euc_z	0.936	0.913	0.955
	gowers	0.936	0.912	0.955
	lmnn	0.936	0.913	0.956
	mahal_scaled	**0.937**	0.913	0.957
	wcsm	**0.937**	0.913	0.956
	hierarch_euc_scaled	**0.937**	0.914	0.956
	hierarch_euc_z	**0.937**	0.914	0.956
	hierarch_gowers	**0.937**	0.914	0.956
	hierarch_mah	**0.937**	0.914	0.956
	kmeans_scaled	**0.937**	0.914	0.956
	kmeans_z	**0.937**	0.914	0.956
ICI	csm	0.0264	0.0108	0.0521
	euc scaled	0.0263	0.0107	0.0510
	euc_z	0.0261	0.00993	0.0499
	gowers	0.0263	0.0107	0.0510
	lmnn	0.0261	0.00984	0.0497
	mahal_scaled	0.0265	0.0105	0.0500
	wcsm	0.0260	0.00927	0.0507
	hierarch_euc scaled	**0.0251**	0.00995	0.0500
	hierarch_euc_z	**0.0251**	0.00995	0.0500
	hierarch_gowers	**0.0251**	0.00995	0.0500
	hierarch_mah	**0.0251**	0.00995	0.0500
	kmeans_scaled	**0.0251**	0.00995	0.0500
	kmeans_z	**0.0251**	0.00995	0.0500
BrS	csm	0.102	0.0834	0.122
	euc scaled	0.102	0.0832	0.122
	euc_z	0.102	0.0835	0.121
	gowers	0.102	0.0840	0.122
	lmnn	0.102	0.0832	0.121
	mahal_scaled	0.102	0.0834	0.122
	wcsm	0.102	0.0831	0.121
	hierarch_euc scaled	**0.101**	0.0836	0.121
	hierarch_euc_z	**0.101**	0.0836	0.121
	hierarch_gowers	**0.101**	0.0836	0.121
	hierarch_mah	**0.101**	0.0836	0.121
	kmeans_scaled	**0.101**	0.0836	0.121
	kmeans_z	**0.101**	0.0836	0.121