New Methods for Improving Accuracy in Three Distinct Predictive Modeling Problems

by

Yingying Xu

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, 2018

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: Paul Gustafson
Professor, Dept. of Statistics, University of British Columbia

Supervisor(s): Joel A. Dubin
Associate Professor
Dept. of Statistics and Actuarial Science, University of Waterloo
Joon Lee
Associate Professor
School of Public Health and Health Systems
Dept. of Statistics and Actuarial Science
University of Waterloo

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

Internal-External Member: Yaoliang Yu

Assistant Professor

Cheriton School of Computer Science, University of Waterloo

I hereby declare that I am the sole author of this 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.

Abstract

People are often interested in predicting a new or future observation. In clinical prediction, the uptake of Electronic Health Records (EHRs) has generated massive health datasets that are big in volume and diverse in variety. The outcomes can be of different types, e.g., continuous, binary, time-to-event, etc., and covariates can be either time-fixed or longitudinal. These datasets can provide rich and diverse information for modeling and prediction but also pose challenges to fast and accurate prediction of outcomes of interest.

One challenge of predicting is that when the data are heterogeneous in the relationship between the covariates and the outcome. In this case, it is quite possible that localizing a subset of data in an informative manner to aid in making predictions will lead to better performance than including all information. Chapter 3 deals with a continuous outcome, and I have developed methodology that gives an interpretable and meaningful definition of similarity, and an algorithm to uncover the similarity structure to improve the prediction accuracy by making similarity-based predictions. In Chapter 4, the similarity-based prediction is extended to a survival outcome, with possible independent or dependent censoring. The algorithm is developed under the random forest framework, and I showed through both simulations and a real data example that incorporating the similarity structure indeed improves prediction accuracy in these cases.

Another challenge in prediction arises when longitudinal covariates are present, and that there are scenarios when one needs to make an early prediction as soon as practical and thus cannot monitor the full trajectory of longitudinal covariates (before the prediction is required). In Chapter 5, I address this concern by quantifying the relationship between the earliness of prediction and the prediction accuracy. A penalization approach with a graphical method is introduced to select a monitoring window length given specific predic-

tion accuracy. Comprehensive simulations are conducted to investigate the performance of the algorithm in selecting the length of the monitoring window in different scenarios.

Acknowledgments

I would like to acknowledge my supervisors Prof. Joel Dubin and Prof. Joon Lee for their patience, support, encouragement, and their insightful guidance on developing the ideas for the thesis and throughout my pursuit of the degree. I am very grateful to have them as my advisers.

I would like to extend my thanks to Prof. Yeying Zhu, Prof. Pengfei Li, Prof. Yaoliang Yu and Prof. Paul Gustafson for their valuable time and effort to serve on my committee, for their helpful comments and valuable suggestions on improving the thesis.

I wish to thank Prof. Heather Keller (Department of Kinesiology, University of Waterloo) for the opportunity to work on several interesting projects. I learned a lot from her and gained valuable experience.

I also would like to thank my fellow graduate students from the department for their help and friendship that made my studies here more enjoyable.

Finally, and most importantly, I would like to thank my parents. Without their constant love and support, I would not be where I am today.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Statistical Prediction

People are often interested in predicting a new or future observation. In some of the predictions, statistical inference is made from a sample of a population and is generalized to the whole population whose outcome is uncertain or unknown. Predictions can also be made for future outcomes based on history - that is, generalization of a given predictive model into the future. In e-commerce for example, people are interested in predicting what a customer might be interested in buying based on the purchasing histories of other customers (one example is in Wang 2016 [56]). Image recognition is also a prediction problem. An example is the identification of hand-written digits (for one example, Niu 2012 [36]). Typically, a training set of images with known digits is used to build a prediction model. Then the digit of a new hand-written digit picture will be identified according to that model. In clinical decision making, physicians need to know what is the survival likelihood of a patient after getting a heart transplant. It is important to identify the

differences between association and prediction. Association is usually model dependent, as all models are only an approximation of the truth. Predictions can usually be validated, using a training-test data split for example. A stronger association does not necessarily imply higher prediction power, as association does not necessarily imply causation.

1.1.1 Prediction Models

I will begin with a brief overview of common types of statistical prediction models. The simplest model is a linear regression model where a linear relationship is assumed between outcome Y and input matrix X. If Y is categorical, a multivariate logistic regression model can be used where the logit of the probabilities $P(Y = y)$ is linear in X.

In many cases, Y is not linear in X, and a more general way of representing the relationship is needed. This can be accomplished using more flexible model specifications, for which one example utilizes regression splines. Smoothing splines, a particular form of penalized spline (e.g., Ruppert , Wand, and Carroll text, 2003 [44]) is another option. This is done by dividing X into different regions and model Y with polynomials of X within each region. The order of the spline, the location and number of the knots need to be specified. Kernel methods are often used as well. Kernel methods estimate the conditional mean using local information. Larger neighborhoods lead to larger bias and smaller variance, and smaller neighborhoods lead to smaller bias and larger variance. Common choices of the kernel includes the Gaussian kernel, the Epanechnikov kernel, and the tri-cube kernel [14].

K-Nearest-Neighbor methods use the k closest neighbors of x to predict Y, i.e.,

$$\hat{Y}(x) = 1/k \sum_{x_i \in N_k(x)} y_i,$$

where $N_k(x)$ is the k nearest neighborhood of x and is defined using the Euclidean distance.

Tree-based methods partition the feature space into rectangles by splitting a feature variable at each node to maximize some similarity measure within each daughter node. To correct for the over-fitting tendency of a tree, a random forest can be used. A random forest consists of trees with the splitting feature variables selected from a bootstrap sample of all features. The average prediction from all trees are combined to make a final prediction. [30]. Boosting methods are commonly used as well. It combines the outputs of "weak" classifiers to form a strong one [14].

1.1.2 Prediction Performance

The performance of prediction models can be assessed by a variety of measures. In general, we want to find predictors with low prediction error, high discrimination ability, and accurate calibration [49]. Often they cannot all be achieved at the same time.

There are two types of statistical predictors, point estimator and probabilistic estimator. A point estimator $\hat{Y} = \hat{f}(X)$ is a function of the input covariates and the function is estimated based on the training data. Probabilistic estimator gives the probabilities that Y is smaller than a value. If Y is categorical, it gives the probability that Y is equal to some number, $P(Y = a)$. The accuracy of point estimators is usually measured by loss functions. Two common choices of the loss functions are the absolute error loss

$$L(Y, \hat{f}(x)) = |Y - \hat{f}(x)|,$$

and the squared error loss

$$L(Y, \hat{f}(x)) = (Y - \hat{f}(X))^2.$$

If Y is categorical, the 0-1 loss function,

$$L(Y, \hat{Y}) = I(Y = \hat{Y}),$$

where I is the indicator function, is commonly used. The prediction error is defined as the expected value over the loss function. For probabilistic estimators, one way of measuring the prediction error is to dichotomize it into different categories based on a subjectively selected threshold and apply the loss function for point estimators.

Sensitivity and specificity can be used to quantify the discriminative power of a probabilistic predictor. That is, how many cases with the outcome have a higher predicted probability of getting that outcome. "Higher probability" can be defined as a probability that is greater than any chosen threshold between 0 and 1. The area under the ROC curve eliminates the need to select a threshold subjectively. It summarizes the sensitivity and specificity of a predictor at all thresholds between 0 and 1. In medical diagnostic testing, each point on the ROC curve can be interpreted as "being a conditional probability of a test result from a random diseased subject exceeding that from a random non-diseased subject" [39]. It can be applied to measure the performance of a survival model as well [20]. Some newer methods for measuring discriminative ability include variants of the c statistic for survival models, reclassification tables, net reclassification improvement (NRI), and integrated discrimination improvement (IDI). NRI was later found to have a high false-positive rate, and a statistically significant NRI statistic should not be relied on as sufficient evidence for improved prediction performance in the evaluation of biomarkers [38].

Calibration can be measured using a goodness of fit test. One example that the three properties cannot be achieved at the same time is, if the data is highly imbalanced between two classes, classification accuracy might be very high for a model that tends to predict the major class for all predictions. Such a model is not useful in practice since it does

not have any discriminative power. A confusion matrix that display the false-positive and false-negative rates will provide more informative, and an alternative model with higher prediction error but better discriminative ability might have better prediction performance.

1.2 Improving Prediction with Similarity Measure

Traditionally, a global model is fitted to the entire dataset to optimize the average model performance. It is quite possible that the prediction of some subsets of data under the global model does not perform as well. Moreover, if the size of the subset is small, the parameter estimates are dominated by the majority cases. It is possible that this group can be identified using some of the available covariates. If we can use information from the available features to identify this subgroup, and build separate models for them, the resulting model will be more useful for that group, and improves prediction performance of the overall model as well. This motivates my proposal to define **similarity** based on the relationship between the covariates of the outcomes instead of just the closeness of the feature vectors or the outcomes. The idea of using similarity to improve prediction performance can be seen in a number of papers, some discussed in Subsections 1.2.1, 1.2.2, and Sections 2.1 and 2.2. Based on how similarity is defined, these methods can be generally divided into two categories, unsupervised and supervised.

1.2.1 Unsupervised Methods

Liu et al. (2007) [31] applied the similarity-based approach to neural networks. They used unsupervised clustering to divide sample data into K groups and then trained separately an RBF network in each cluster. A new case is then assigned to one of the clusters

for prediction. They showed that when applied to financial time series prediction the method reduced the computation time and the complexity of the model and improved trend accuracy in prediction. Lowskey et al. (2013) [32] applied similarity to model the effect of covariates on a survival outcome. Instead of generating one survival curve for the entire training dataset, for a new patient j in the test dataset, their algorithm generates a Kaplan-Meier survival curve based on K-nearest neighbors of that patient in the training set. The Mahalanobis distance between the covariates was used as the similarity measure to find the K-nearest neighbors.

Trivedi, Pardos & Heffernan (2015) [53] investigated the utility of clustering in improving prediction accuracy, and also provided some explanations on why this approach may increase prediction accuracy. They applied k-means clustering first and then obtained k sets of predictors for each cluster. The k sets of predictions were then combined using a naive ensemble for prediction. When similarity measure is defined in an unsupervised way, it will only depend on the covariates. The relative explanatory ability or the relationship between the covariates and the outcome is ignored.

1.2.2 Supervised and Semi-Supervised Methods

Ishwaran et al. (2008) [22] extended a random forest model to survival outcomes. By splitting nodes to maximize survival difference, dissimilar cases will end up in different nodes while cases in the same node will be more homogeneous. A cumulative hazard function (CHF) is calculated for each tree and then averaged to obtain the ensemble CHF. As a result, the CHF obtained from similar cases are used to predict the CHF of a new case. Xu, Nettleton, & Nordman (2014) [57] developed case-specific random forests (CSRFs). While the standard random forest (RF) model uses uniform weights in the re-sampling

scheme and generates a global random forest for all cases, Xu, Nettleton, & Nordman (2014) [57] applied weights to assign higher probabilities to cases that are more similar to the target case. The similarity is defined using bagging of trees, that is, firstly, a standard RF model is fit to the data. The greater the degree trees group two cases into the same node, the more similar these two cases are. However, the drawback of applying random forest models, e.g., either Ishwaran et al. (2008) or Xu, Nettleton & Nordman (2014) [57], to clinical data is computational burden and a lack of interpretability.

1.3 Thesis Outline

Chapter 2 will discuss general prediction problems in clinical settings and especially the challenges of predictions in the ICU. The MIMIC-III (Medical Information Mart for Intensive Care III) dataset and the motivating study from that dataset will be introduced. In Chapter 3, I will give an interpretable and meaningful definition of neighborhoods and use simulations to show the idea of building localized models to improve prediction. I will propose a new method to estimate similarity in a supervised way. In Chapter 4, I will extend our method to survival analysis where the outcome is the time to occurrence of some event. In Chapter 5, I will quantify the relationship between prediction accuracy and the length of the monitoring window of longitudinal covariates.

Chapter 2

Clinical Prediction

Clinical prediction models provide physicians with evidence-based decision-making by estimating individual probabilities of risks and benefits [11]. The dataset that we mainly focus on comes from an observational study. Observational studies are used primarily to identify risk and prognostic factors and in cases where randomized controlled trials would be impossible or unethical. They often have lower costs (than clinical trials), greater follow-up time for patients, and include a broader range of patients. It has been shown in recent studies that well-designed observational studies provide results similar to randomized controlled trials[2, 8].

2.1 Overview of Similarity-based Prediction in Medicine

The uptake of Electronic Health Records (EHRs) has generated massive health data sets that are big in volume and diverse in variety. Traditional methods assume that all patients in a given dataset have the same relationship between the outcome and the feature vector,

and focus on a global model to fit the entire training dataset. However this approach may not be optimal for big health data, and using all patient data in the prediction might only be adding computational burden and decreases prediction accuracy.

It has been shown in previous papers that patient similarities from EHR can be utilized to improve prediction of health risks, and to target medical treatments,. For example, Gottlieb et al. (2013) used basic patient-specific information gathered at admission, to identify similar patients and then predicting the eventual discharge diagnoses[17]. Panahiazar et al. (2015) used EHR information such as medical co-morbidities, lab measurements, ejection fraction, vital status and demographics to identify similar patients, and subsequently assign medication plans based on the similarity index [37].

2.2 MIMIC III Database

MIMIC-III is a freely accessible critical care database for 53,423 distinct hospital admissions for adult patients. It is an update to the MIMIC-II database [27, 46, 45] with added patients from 2008-2012. Data includes vital signs, medications, diagnostic code, survival data and high resolution data including lab results and bedside monitoring data [23]. They can provide rich and diverse information for modeling and prediction but also pose challenges to fast and accurate prediction of outcome of interest. Sun et al. (2015) used localized supervised learning to incorporates expert feedback in defining patient similarity, and demonstrated the efficacy of their approach with MIMIC-II data, a precursor to MIMIC-III[50]. Lee, Maslove & Dubin (2015) [26] showed empirically that using a patient similarity metric (PSM) to only include a subset of similar patients improves mortality prediction in the MIMIC-II database.

These work showed that a personalized decision support system is promising. With a

rich database like MIMIC, and with its large sample size, it is possible to identify similar patients and then effectively reduce the sample size while improving the patient outcome performance.

2.3 An Example of a MIMIC Study

One of the studies based on MIMIC database is an Acute Kidney Injury (AKI) study in the ICU (Intensive Care Unit). Acute kidney injury (AKI) is a sudden episode of kidney failure or kidney damage that happens within a few hours or a few days [1].There are various ways to diagnose the onset of AKI, and the most widely adopted definition is an increase in creatinine measurement of 0.3mg/dl or a 50% increase compared to the baseline measurement within 48 hours [34]. The definitions of the key concepts and variables are listed as follows:

Basic definitions:

AKI: An increase in creatinine measurement of 0.3mg/dl or 50% compared to ICU admission value within 48 hours after ICU admission.

AKI onset time: The time when creatinine exceeds the threshold.

Peak creatinine: The highest creatinine value within 24 hours or 48 hours after AKI onset for AKI patients

Recovery: Return to below 10% above admission value.

Censoring: Neither recovered from AKI nor died before hospital discharge

Creatinine Percentage Increase Definition1: The percentage increase of peak creatinine within 48 hours compared to admission value. (44.51% of patients have peak creatinine is at AKI onset, 23.39% between 0-24 hours, 32.10% between 24-48 hours)

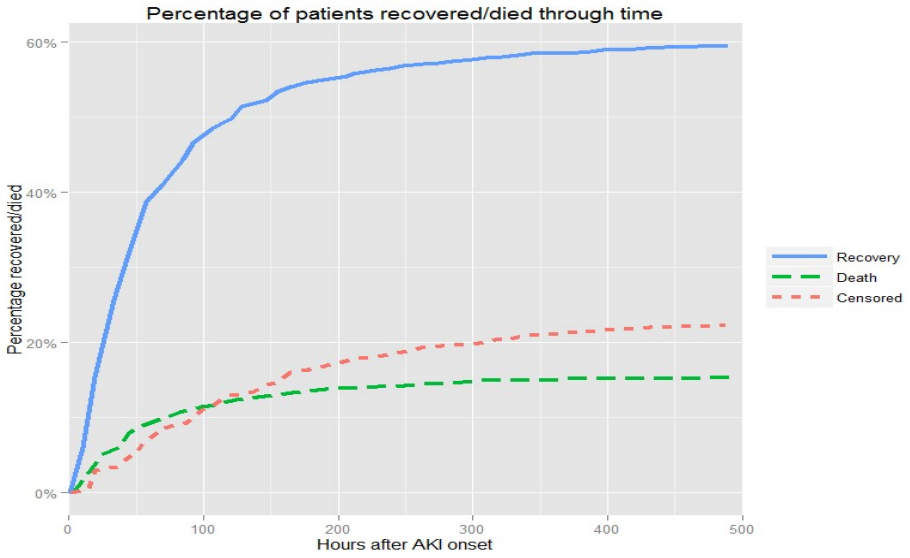


Figure 2.1: Percentages of patients that recovered, died or were censored

Creatinine Percentage Increase Definition2: The percentage increase of peak creatinine within 24 hours compared to admission value. (57.07% peak at AKI onset, 42.93% peak between 0-24 hours)

Creatinine Percentage Increase Definition3: The percentage increase of creatinine at AKI onset time compared to admission value.

Predictors:

Urine Volume: Total amount in 12 hours just before AKI onset, divided by 12.

Systolic Blood Pressure: Average systolic blood pressurein in 12 hours just before AKI.

Age as a continuous predictor

ICU type: 1-CCU(Coronary Care Unit) 2-CSRU (Cardiac Surgery Recovery Unit) 3-FICU (Finard Medical Surgical ICU)/MICU (Medical Surgical ICU) 4-SICU (Surgical ICU).

Creatinine Admission: Creatinine at ICU admission.

Creatinine Percentage Increase: Three definitions as shown previously.

In MIMIC-II, 3599 out of 22136 (approximately 16.26%) of ICU patients developed AKI after ICU admission, and Figure 2.1 shows the percentages of recovery, death or discharge from hospital for AKI patients. Previous studies showed that AKI is associated with increased risk of both the short term and long term mortality among patients [34, 33, 16]. A subset of the patients recovered from AKI during ICU or hospital stay, while others do not recover. Clinicians are interested in predicting which patient will recover from the disease and which ones will not, and in developing personalized treatment plans for each type of patient. We want to identify the variables that have good predictive power for predicting the outcome of the patient diagnosed with AKI.

It is interesting to see that while some patients recover very quickly, others take a much longer time to recover, as shown in Figure 2.1. Thus, we want to consider predicting time to recovery from AKI as opposed to merely predicting a single outcome whether they recovered or not. And we would like to use a similarity approach to improve the time-to-event prediction.

Another challenge is that the nature of the urgency in the ICU requires us to start predicting as soon as possible, in order for medical staff to act quickly. For longitudinal predictors, it is important to know how long should the follow up time be so that we can predict as soon as possible without losing potentially critical follow-up information and deteriorating the accuracy of the prediction. This motivates the method in Chapter 5.

Chapter 3

Similarity-based Patient Outcome Prediction for Continuous Data

3.1 Introduction

Traditionally, a global model is fitted to the entire dataset to optimize the average model performance. It is quite possible that the prediction of some subsets of data under the global model does not perform as well. Moreover, if the size of the subset is small, the parameter estimates are dominated by the majority cases. It is possible that this group can be identified using some of the available covariates. If we can use information from the available features to identify subgroups, and build separate models for them, the resulting model should be more useful for that group, and should improve prediction performance of the overall model as well.

The idea of using similarity to improve prediction performance can be seen in a number of papers based on how similarity is defined, these methods can be generally divided into

two categories, unsupervised and supervised, and are discussed in Chapter 1.

In addition to the similarity-based methods, mixture models are commonly used to represent the heterogeneity in the data as well. For example, FlexMix [28, 18] is a general framework for fitting discrete mixtures of regression models using EM algorithm, such that the M-step allows a general specification. Posterior probabilities are used as weights in the mixture of regression models, and need to be estimated. Mixture of experts models have also been used in the literature. These models have experts such as regression functions or classifiers and a gate to soft partition the data into different regions, so that individual experts will specialize on a smaller problem. The experts and the gate are then combined by a probabilistic model [59].

The rest of the section is organized as follows. In Section 3.2.1, we will explore the rationale behind our similarity-based modeling approach. A novel prediction algorithm is proposed in Section 3.2.2 We will demonstrate the prediction performance of the proposed method through simulations in Section 3.3. The impact of model misspecification is also studied.

In Section 3.4, we apply the method to an intensive care unit (ICU) dataset. In Section 3.5, we will summarize our findings and provide discussions on possible future work.

3.2 Methods

It is quite possible that the mixtures can be identified using a portion of the available covariates. If we can utilize that available information, and build separate models for different subgroups, the resulting model could very well be more useful for that group, and improve prediction performance of the overall model as well. That motivates our idea to

define **similarity** based on the relationship between the covariates and outcomes instead of just the closeness of the feature vectors or the outcomes.

3.2.1 Problem Set-up

We will first show how the similarity-based modeling is conducted and the reason that it generally has more accurate prediction than modeling using all data points.

Without loss of generality, assume there is a mixture of two linear models in the dataset. Denote the dataset $[Y, \boldsymbol{X}^*] = \{(Y_i, \boldsymbol{X_i}, \boldsymbol{S_i}), i = 1, 2, ..., n\}$, where Y is a continuous outcome. The columns of $\boldsymbol{X}_i^*$ consist of columns of $\boldsymbol{X}_i$ and $\boldsymbol{S}_i$, and might have common elements. The elements in $\boldsymbol{S}_i$ are called the similarity variables here and the elements in $\boldsymbol{X}_i$ are called the regression variables. Consider the following linear regression model: $Y_i = \boldsymbol{X}_i\boldsymbol{\beta}+\epsilon_i$ if $f(\boldsymbol{S_i}) \in (a, b)$ and $Y_i = \boldsymbol{X}_i\boldsymbol{\alpha}+\epsilon_i$ otherwise, $i = 1, 2, ..., n$. Y_i is the response variable, and $\boldsymbol{X}_i = (1, X_{i1}, X_{i2}, ..., X_{ip})$ is a $1 \times (p + 1)$ vector. $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)^T$ and $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, ..., \alpha_p)^T$ are both vectors of $(p + 1) \times 1$ regression coefficients. f is assumed to be a smooth function of the similarity variables, and maps the similarity variables to a real number. (a, b) is any open interval on the real line. ϵ_i is independently and normally distributed with mean 0 and variance σ^2.

For each case i, $i = 1, 2, ...n$, using some suitable distance measure, such as Euclidean distance or Mahalanobis distance, a function D, the distance between i, and any other case j, $j = 1, 2, ...n$, j not equal to i, is denoted as $D(S_i, S_j)$. The k cases corresponding to the smallest distances with i will be called the neighborhood of i. Suppose case i corresponds to the first linear model, that is, $Y_i = \boldsymbol{X}_i\boldsymbol{\beta} + \epsilon_i$, with suitable f, it is fair to assume that when k is sufficiently small, the neighborhood of i also corresponds to the same model. So if an ordinary linear model is fit to the data using a sufficiently small enough neighborhood

of i, which consists only of cases from the same distribution, the model estimates will be unbiased. As we widen the neighborhood, at some point, cases from the second model $Y_i = \boldsymbol{X}_i\boldsymbol{\alpha} + \epsilon$ will be included. Fitting an ordinary linear model to all the data in the widened neighborhood will in general lead to increased mean squared error. Fitting an ordinary linear model to all the data in the widened neighborhood will lead to biased estimates of $\hat{Y}$ and in general increased mean squared error. We will briefly show the proof as follows.

Depending on $f(\mathbf{S_i})$, $i = 1, 2, ..., n$, partition $\boldsymbol{X}_{n\times(p+1)}$ into two sets, so that $\boldsymbol{X}_i = \binom{W}{Z}$, and $\boldsymbol{Y} = \binom{W\alpha+\epsilon}{Z\beta+\epsilon}$, where W is an $n_1 \times (p+1)$ matrix and Z is an $n_2 \times (p+1)$ matrix with $n_1 + n_2 = n$. By using all the data in the sample, the OLS estimate of $\boldsymbol{Y}$, is $\widehat{\boldsymbol{Y}} = X(X^TX)^{-1}X^TY$. It is easy to show that $X^TX = W^TW + Z^TZ$. Then:

$$\begin{aligned} E(\hat{\boldsymbol{Y}}) &= \binom{W}{Z}(W^TW + Z^TZ)^{-1}(W^TW\alpha + Z^TZ\beta) \\ &= \binom{W}{Z}(W^TW + Z^TZ)^{-1}(W^TW\alpha + Z^TZ\alpha - Z^TZ\alpha + Z^TZ\beta) \\ &= \binom{W}{Z}[\alpha - (W^TW + Z^TZ)^{-1}Z^TZ(\alpha - \beta)]. \end{aligned}$$

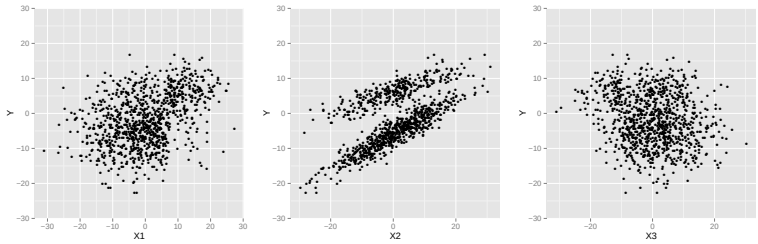
Thus:

$$\begin{aligned} E(\hat{\boldsymbol{Y}}) - \boldsymbol{E(Y)} &= \binom{0}{Z\alpha - Z\beta} - \binom{W}{Z}(W^TW + Z^TZ)^{-1}Z^TZ(\alpha - \beta) \\ &= \binom{0}{Z}(\alpha - \beta) - X(X^TX)^{-1}Z^TZ(\alpha - \beta) \\ &= [\binom{0}{Z} - X(X^TX)^{-1}Z^TZ](\alpha - \beta) \end{aligned}$$

We can see that $E(\hat{\boldsymbol{Y}})$ generally does not equal to $\boldsymbol{E(Y)}$, i.e., the estimated $\boldsymbol{Y}$ is not unbiased unless the coefficients in the two models are the same.

We will further use simulations from two models, to provide more rationale for similarity-based predictions. For simplicity, suppose we only have three variables, X_1, X_2 and X_3. In each of the scenarios, X_1 and X_3 jointly define the similarity in this model.

Figure 3.1: Case 1 data, Y vs. X_1, X_2, X_3

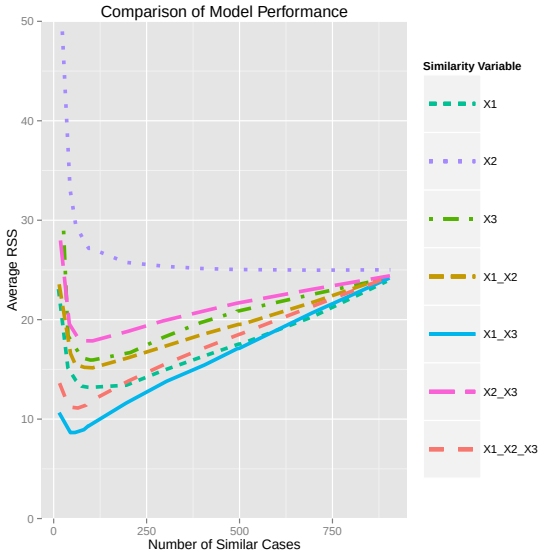


$A = \{(a,b)|(a-7)(b+10) > 0\}$ and $I_A(X_{i1}, X_{i3}) = 1$ if $(X_{i1}, X_{i3}) \in A$ and 0 otherwise. The following set of simulation cases are for proof of concept purposes, as certainly for big datasets, the scope of the problem, both for sample size, and covariate space, will be much larger, in general. Consider the first model:

$$Y_i = (1 - I_A(X_{i1}, X_{i3}))(0.5X_{i2} - 6) + I_A(X_{i1}, X_{i3})(0.3X_{i2} + 6) + \epsilon_i,$$

Thus, Y is a continuous mixture, linearly dependent on continuous variable X_2 and the resulting multiplicative interaction between continuous variables X_1 and X_3. We independently generate (X_{i1}, X_{i2}, X_{i3}) from a normal distribution with mean 0 and variance 10^2, and the error terms ϵ_i are generated from a normal distribution with mean 0 and variance 2^2. Since pairwise distances need to be calculated, to reduce the computational burden in the simulation study, and without loss of generality, the sample size n is set to be 1000. Figure 3.1 shows the univariate association between the outcome Y and each variable X_1, X_2 and X_3. Just viewing the plots, it appears that X_1 and X_3 are not very correlated with the outcome Y. But in fact, they jointly affect the relationship between X_2 and Y. For each sample case j, the distance between j and all other $(n-1)$ cases are calculated and the k closest cases are selected to train the case j specific model $\hat{Y} = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$; this model is then used to make a prediction for case j. After iterating through all cases, n fitted values will be obtained. The residual sum of squares is calculated to measure the

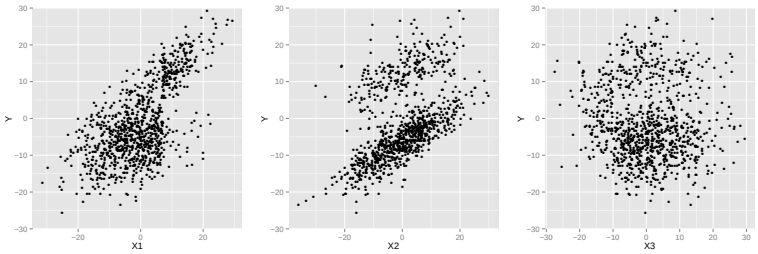
Figure 3.2: Case 1 results, average residual sum of squares as the number of similar cases increases



performance of the prediction.

The combination of X_1 and X_3 is the correct similarity measure in this model. But suppose it is unknown which variable(s) define the similarity, and we use Euclidean distances based on different combinations of X_1, X_2, X_3 to calculate the distance. For example, if we choose X_1 and X_2 as the similarity variables, then the distance between case i and j is $Distance(i,j) = D_{\{X1,X2\}}(i,j) = \sqrt{(X_{i1} - X_{j1})^2 + (X_{i2} - X_{j2})^2}$. With three variables, there are seven ways to define distance, i.e., $D_{\{X1\}}(i,j)$, $D_{\{X2\}}(i,j)$, $D_{\{X3\}}(i,j)$, $D_{\{X1,X2\}}(i,j)$, $D_{\{X1,X3\}}(i,j)$, $D_{\{X2,X3\}}(i,j)$, and $D_{\{X1,X2,X3\}}(i,j)$. Pairwise distances are calculated for each case, and the closest k neighbors are then selected for model training. The performance of seven models using different definitions of similarity will be compared. By increasing the neighborhood size, i.e., increasing k, the performance also varies. As k approaches n, the trained individual models will get closer to the global model, that is the

Figure 3.3: Case 2 data, Y vs. X_1, X_2, X_3



model using all the sample points. The above simulation is repeated 100 times to obtain an averaged residual sum of squares.

As shown in Figure 3.2, as k increases, the performances of all seven models converge. When k is very small, the model performance deteriorates due to small sample size and resulting variability issues. The RSS curve shows a skewed U-shape except when X_2 was chosen. The residual sum of squares is the smallest when both correct variables for similarity are chosen, that is, when both X_1 and X_3 are used in calculation of the distance, and the neighborhood is sufficiently small to exclude less relevant cases. The second best case is when all of the variables are used. If the wrong similarity variable X_2 is the only similarity variable used, the performance is the worst among all other scenarios. Also, if only one of the correct similarity variables is picked, the performance is not much improved, since the two similarity variables jointly affect the neighborhood.

For the second model,

$$\begin{aligned} Y_i =& (1 - I_A(X_{i1}, X_{i3}))(0.3X_{i1} + 0.5X_{i2} + 0.1X_{i3} - 6) \\ &+ I_A(X_{i1}, X_{i3})(0.7X_{i1} + 0.3X_{i2} - 0.1X_{i3} + 6) + \epsilon_i, \end{aligned}$$

where $A = \{(a, b)|(a - 7)(b + 10) > 0\}$. X_1, X_2, X_3 are normally distributed with mean 0 and variance 10^2, and the error term is normally distributed with mean 0 and variance 2^2. Again, X_1 and X_3 jointly define the similarity. But a difference from the first model where

only X_2 is in the linear component is that in this case all three variables are in the linear component. Figure 3.3 shows the plots of Y against each of the three variables. Unlike in Case 1 where X_1 and X_3 appear to be uncorrelated with Y, here there is a linear trend between the two variables and Y. This simulation tries to investigate the behavior of the similarity-based modeling when the similarity variable is also in the regression component. Patterns analogous to those seen in Case 1 can also been seen here. Figure 3.4 shows that when the correct similarity variables are selected, the lowest averaged residual sum of squares is achieved, when compared both horizontally and vertically. Using all the variables as similarity variables, the model performance is better than choosing only one correct similarity variable, but worse than when choosing two correct ones. Again we can see the U-shape of RSS curve just as in Case 1, which we speculate is due to when the sample size is very small, increasing the neighborhood includes more cases of the same class without introducing too many dissimilar cases, and the increase of similar cases reduces the additional variability from sample size that could take the prediction off target.

3.2.2 Algorithm

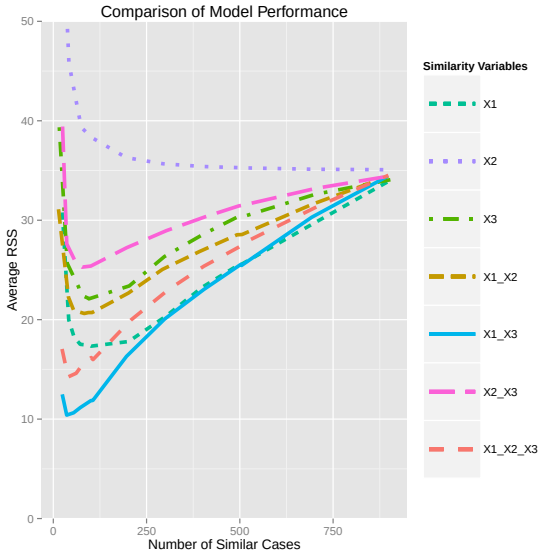
The previous two examples showed the importance of identifying similarity variables in localized prediction.

In this section, we will propose a new method to find similarity variables, and utilize that information to improve prediction.

We will call it *similarityMix*. Suppose there are n individuals and K mixture components. Let

$$Y_i = \sum_{j=1}^{K} I_{f(\boldsymbol{S_i})=j}(\boldsymbol{X_i}\boldsymbol{\beta_j}) + \epsilon_i, i = 1, 2, ..., n$$

Figure 3.4: Case 2 results, average residual sum of squares as the number of similar cases increases



where $\epsilon_i, i = 1, 2, ..., n$ are i.i.d Gaussian $(0, \sigma^2)$. $f(\boldsymbol{S_i})$ takes value in $1, 2, ..., K$ and $I_{f(\boldsymbol{S_i})=j}$ equals 1 if $f(\boldsymbol{S_i}) = j$ and 0 otherwise. $\boldsymbol{X_i}$ stands for regression variables and $\boldsymbol{S_i}$ stands for similarity variables for individual i. $\boldsymbol{\beta}_j$ are the regression coefficients for mixture component j, $j = 1, ..., K$. Let $\boldsymbol{Y} = [Y_1, Y_2, ...Y_n]'$, $\boldsymbol{X} = [\boldsymbol{X_1}', \boldsymbol{X_2}', ...\boldsymbol{X_n}']'$, and $\boldsymbol{S} = [\boldsymbol{S_1}', \boldsymbol{S_2}', ...\boldsymbol{S_n}']'$. Let $Z = X \cup S$ be the set of all variables. Then the likelihood is,

$$L(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K, f, \boldsymbol{S}) = \prod_{j=1}^{K} \prod_{i=1}^{n} N(Y_i|\boldsymbol{X}_i, \boldsymbol{\beta}_j, \sigma^2)^{I_{f(\boldsymbol{S})=j}},$$

where $N(Y_i|\boldsymbol{X}_i, \boldsymbol{\beta}_j, \sigma^2)$ is the density for a Gaussian distribution with mean $\boldsymbol{X}_i\boldsymbol{\beta}_j$ and variance σ^2. And the log-likelihood is

$$logL(Y|\boldsymbol{X}, \boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K, f, \boldsymbol{S}) = \sum_{j=1}^{K} \sum_{i=1}^{n} I_{f(\boldsymbol{S})=j} log(N(Y_i|\boldsymbol{X}_i, \boldsymbol{\beta}_j, \sigma^2))$$

.

The main difference between this model and the mixture of experts model is that a hard threshold, represented by an indicator function, is used here, and the algorithm is a combination of an EM type algorithm and an additional step of fitting similarity variables (Step 5).

The algorithm is as follows,

1. $f(\boldsymbol{S_i}), i = 1, 2, ..., n$ is the initial guess for mixture assignment.

For t=0 to T,

2. Given $f(\boldsymbol{S_i})^{t=0}, i = 1, 2, ..., n$, maximize the likelihood for each mixture j to get the regression coefficients $\hat{\boldsymbol{\beta}}_j^{\ t}$. In this step, we will use all variables in $\boldsymbol{Z}$ as $\boldsymbol{X}$ for now.

3. Given $\hat{\boldsymbol{\beta}}_j^{\ t}$, for i=1,2,...,n, update $f(\boldsymbol{S_i})^t$ to $f(\boldsymbol{S_i})^{t+1}$ to maximize the likelihood. That is, $\hat{f}(\boldsymbol{S_i})^{t+1} = \arg\min_j (Y_i - \hat{\boldsymbol{\beta}}_j^{\ t} X_i)^2$.

4. Repeat Step 2 and 3 until convergence of the respective coefficients.

5. After getting estimates for $\hat{\boldsymbol{\beta}}_j, j = 1, 2, ..., K$, and $\hat{f}(\boldsymbol{S_i}), i = 1, 2, ..., n$, the next step is to determine similarity variables $\boldsymbol{S}$ and the relationship between the similarity variables and group assignments f. We first fit a multinomial logistic regression with $\hat{f}(\boldsymbol{S_i}), i = 1, 2, ..., n$ as the outcome and all variables in $\boldsymbol{Z}$ as covariates. To select the similarity variables, standard variable selecting techniques for logistic regressions can be used. Tree structures such as classification/decision trees can be used to fit the similarity component as well. For prediction, a new case will be assigned to one of the components using similarity variables, then the model for that component will be used for prediction.

Table 3.1: Estimated coefficients for mixture 1 in Case 1

	β_0	β_1	β_2	β_3
iteration 2	-0.92	0.78	0.21	-0.41
iteration 5	2.02	0.89	0.24	-0.35
iteration 10	5.05	0.77	0.28	-0.15
Converged at iteration 15	5.74	0.72	0.28	-0.11
True Coefficents	6	0.7	0.3	-0.1

3.3 Simulation Studies

3.3.1 Mixture of Two First-order Polynomials

In the first example, I will look into the convergence of similarityMix when Y is a mixture of two first-order polynomials, i.e.

$$\begin{aligned} Y_i =& (1 - I_A(X_{i1}, X_{i3}))(0.3X_{i1} + 0.5X_{i2} + 0.1X_{i3} - 6) \\ &+ I_A(X_{i1}, X_{i3})(0.7X_{i1} + 0.3X_{i2} - 0.1X_{i3} + 6) + \epsilon_i, \end{aligned} \tag{3.1}$$

where $A = \{(a, b)|(a - 7)(b + 10) > 0\}$. $n = 1000$ sample points are generated where X_1, X_2, $X_3 \sim$ i.i.d. Gaussian(0,10), and the error terms $\epsilon \sim$ i.i.d Gaussian(0,2). The updates of class assignments and the fitted regression curves at each iteration are studied when we apply the first step of the proposed algorithm. Figure 3.5 plots the class assignments of each data point from X_2 perspective at iteration 2, 5, 10 and 15. Table 3.1 and 3.2 show that the regression curves converges quickly to the true regression curves. The algorithm converges at the 15th iteration.

Figure 3.5: Mixture of two first-order polynomials: Blue and red dots correspond to different class assignments at each iteration

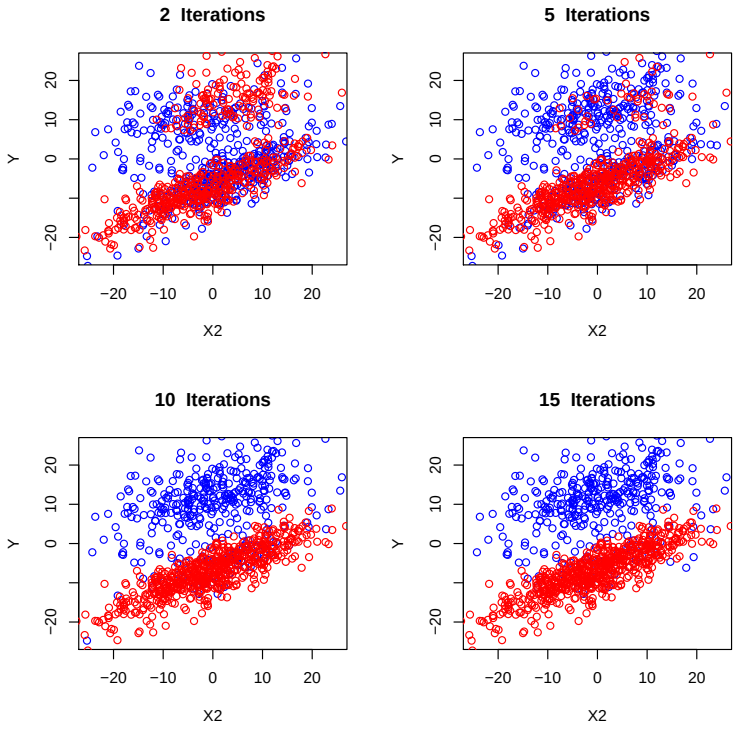


Table 3.2: Estimated coefficients for mixture 2 in Case 1

	β_0	β_1	β_2	β_3
iteration 2	-1.70	0.78	0.58	-0.35
iteration 5	-4.50	0.56	0.54	0.36
iteration 10	-6.03	0.34	0.49	0.09
Converged at iteration 15	-5.96	0.34	0.49	0.09
True Coefficents	-6	0.3	0.5	0.1

Figure 3.6: Mixture of first-order polynomial and second-order polynomial: Blue and red dots corresponds to different class assignments at each iteration

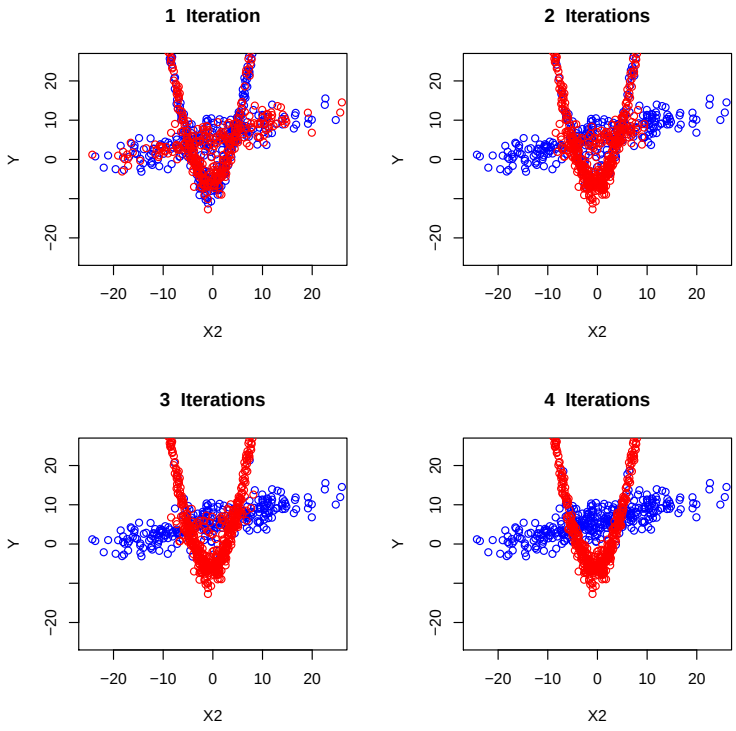


Table 3.3: Estimated coefficients for mixture 1 in Case 2

	β_0	β_1	β_2	β_3	β_4
iteration 1	-5.86	-0.67	0.34	0.32	0.40
iteration 2	-6.38	-0.04	0.47	-0.07	0.50
iteration 3	-5.77	0.01	0.50	-0.04	0.50
Converged at iteration 4	-5.96	0.02	0.50	0	0.50
True Coefficents	-6	0	0.5	0	0.50

3.3.2 Mixture of first-order and second-order polynomials

For the second example, the mixture of a linear and a quadratic function is considered, with

$$\begin{aligned} Y_i =&(1 - I_A(X_{i1}, X_{i3}))(0.5X_{i2}^2 + 0.5X_{i2} - 6) \\ &+ I_A(X_{i1}, X_{i3})(0.3X_{i2} + 0.1X_{i3} + 6) + \epsilon_i. \end{aligned} \tag{3.2}$$

where $A = \{(a, b)|(a - 7)(b + 10) > 0\}$. Again, $n = 1000$ sample points are generated with X_1, X_2, $X_3 \sim$ i.i.d. Gaussian(0,10), and the error terms $\epsilon \sim$ i.i.d Gaussian(0,2). Two quadratic functions of X_1, X_2, X_3 are fitted. Table 3.3 and 3.4, and Figure 3.6 show that the two fitted regression curves at each iteration converge quickly to the true components, faster than in the first example. To check for the robustness of the algorithm with respect to intial assignments, different starting values of the regression coefficients are chosen. In these examples, the coefficients converge to the same values regardless of the choice of the initial values.

Table 3.4: Estimated coefficients for mixture 2 in Case 2

	β_0	β_1	β_2	β_3	β_4
iteration 1	-0.41	-1.04	0.51	0.67	0.31
iteration 2	15.17	-1.06	0.51	0.97	0.01
iteration 3	9.20	-0.25	0.30	0.33	0.00
Converged at iteration 4	6.10	0	0.28	0.12	0.00
True Coefficients	6	0	0.3	0.1	0

3.3.3 Prediction Performance

I will illustrate the prediction performance of similarityMix using simulated data with three variables and 2, 3 or 5 mixture components. For each case, $n = 1000$ samples are generated according to (3.3)-(3.7) and Table 3.5, and we will use 5-fold cross-validation. For the first step, correct number of components are fitted to the data to obtain the class assignment for each training sample. At the second step, a classification tree with class assignments as the outcome and all three variables is used to find the similarity variables. The test data are assigned into one of the components via the classification tree for prediction. For each simulation, the residual sum of squares of the predicted test data are averaged over the 200 test samples. The simulation is repeated 100 times, and the averaged residual sum of squares for 100 simulations are calculated. In applications, the true number of mixture components are usually unknown, therefore we will also investigate the scenario where robustness of the prediction performance when incorrect number of mixtures are fitted. The results are compared with KNN regression with K=1, 2, ..., 500, and FlexMix [28, 18], and presented in Table 3.6. R package Flexmix is used for the comparison. The result of KNN regression are shown only at optimum K (i.e. K-opt) that corresponds to

Table 3.5: Simulation functions

X_{1i}, X_{2i}, X_{3i} *i.i.d.* $\sim Gaussian(0,10)$; ϵ_i *i.i.d.* $\sim Gaussian(0,2)$; $i = 1, ..., 1000$					
Number of Components	2		3		
Conditions	$X_{1i} \leq 0$, and $X_{2i} \leq 0$	else	$X_{1i} \leq 0$ and $X_{2i} \leq 0$	$X_{1i} \leq 0$ and $X_{2i} > 0$	else
Function	(3.3)	(3.4)	(3.3)	(3.4)	(3.5)
Number of Components	5				
Conditions	$X_{1i} \leq -5$ and $X_{2i} \leq -5$	$X_{1i} \leq -5$ and $-5 < X_{2i} \leq 5$	$X_{1i} \leq -5$ and $X_{2i} > 5$	$X_{1i} > -5$ and $X_{2i} \leq -5$	else
Function	(3.3)	(3.4)	(3.5)	(3.6)	(3.7)

the lowest residual sum of squares for KNN

$$Y_i = -0.2X_{i1} - 0.8X_{i2} - 0.4X_{i3} + 6 + \epsilon_i \tag{3.3}$$

$$Y_i = 0.2X_{i1} - 0.2X_{i2} + 0.1X_{i3} - 7 + \epsilon_i \tag{3.4}$$

$$Y_i = -0.8X_{i1} - 0.8X_{i2} - 0.9X_{i3} + 6 + \epsilon_i \tag{3.5}$$

$$Y_i = 0.7X_{i1} - 0.2X_{i2} + X_{i3} - 7 + \epsilon_i \tag{3.6}$$

$$Y_i = 0.6X_{i1} + 0.2X_{i2} - 0.8X_{i3} - 9 + \epsilon_i \tag{3.7}$$

When model is correctly specified, similarityMix outperforms KNN at K-opt for 2, 3, and 5 mixtures, and slightly better than FlexMix. When insufficient number of mixture

Table 3.6: Simulation results

True number of components	2		3			5	
Fitted # of components	2	3	2	3	5	3	5
RSS: KNN with K-opt	12.21		5.7			11.97	
RSS: FlexMix	9.146	8.145	5.054	3.793	2.285	6.734	3.943
RSS: similarityMix	8.895	8.094	7.230	2.514	2.316	6.981	3.861

components are fitted, the residual sum of squares is higher, but less than KNN in the five components case. The only case where it has higher residual sum of squares is when fitting 2 components to a 3 component model, but not much higher. The performance is similar to that of FlexMix. Unlike FlexMix (or mixture of experts model), a hard threshold is used to assign cases into different groups. The advantage of the hard threshold approach is that for each case, we are able to identify its similar cases, and use information from the similar cases to make predictions. Specifying more components than needed does seem to decrease the residual sum of squares of the prediction, but not by too much. In fact, the decrease in residual sum of squares when we increase from insufficient to correct number of components is much larger than that when we increase from correct to excess number of components. This could potentially be used to find the optimum number of components to balance between parsimony and goodness-of-fit, that is, an elbow in RSS vs. fitted number of components plot would indicate the true number of component for the data. We will give an example of finding the optimum number of components using this heuristic in Section 3.4.

3.4 Application to an ICU dataset

3.4.1 MIMIC-III

MIMIC-III (Medical Information Mart for Intensive Care III) is a freely accessible critical care database for 53423 distinct hospital admissions for adult patients (aged 16 and above). Data includes vital signs, medications, diagnostic code, survival data and high resolution data including lab results and bedside monitoring data [23]. The goal is to predict patient hospital length of stay (with a log transformation) with their age, gender, ICU type, admission type, SAPS-II score. ICU type includes CCU (Coronary Care Unit), CSRU (Cardiovascular Intensive Care Unit), MICU (Medical Intensive Care Unit), SICU (Surgical Intensive Care Unit) and TSICU (Trauma Surgical Intensive Care Unit). Admission type includes elective, emergency and urgent. Only the first hospital admission of adult patients (older than 15 years of age) are included in our study.

3.4.2 Methods

We compared the residual sum of squares of similarityMix with FlexMix [28, 18] for specified number of mixture components varying from 1 to 15. The maximum number of components for this analysis is chosen to be 15 because of computational considerations. For similarityMix, we use linear regression for the regression component. Multinomial logistic regression, or classification/decision trees are used for the similarity part. In FlexMix, mixtures of multivariate Gaussian distributions is specified, so the standard M-step is used in the EM algorithm. Then, as a demonstration of the graphical method of determining the optimum number of components mentioned in Section 3.3, we stratify data into more homogeneous subsets of the population based on ICU type, namely CCU, CSRU, MICU,

SICU and TSICU patients. This is because there seems to be more than 15 mixtures in the entire dataset. Then we run similarityMix and FlexMix within each stratum but with only age, gender, admission type and SAPS-II score as predictors, and the results are shown in Figure 3.8-.

3.4.3 Results and interpretation

Figure 3.7 shows that the residual sum of squares decreases as we increase the specified number of mixtures for both methods, and our method yields close results as the FlexMix in terms of prediction performance. This is also observed in Figure 3.8-. Compared to FlexMix, our method fitted both the regression component as well as the similarity component. In Figure 3.7, we did not see the elbow in RSS as we increase the number of mixtures up to 15, which is possibly because of the heterogeneity of the data, that is, there might be more than 15 mixtures in the data. In Figure 3.8-, we can see the drop in RSS flattens out. For example in Figure 3.8, the biggest drop in RSS is seen at number of mixtures from 1 to 4. This possibly indicates that the optimum number of components is 4.

3.5 Discussion

In this section, a meaningful definition of neighborhood based on the relationship between the outcome and the covariates is provided. A method of uncovering that difference in that relationship is presented. Simulations showed that our algorithm converges quickly to the true relationships when the number of features is three and there are two similarity variables. The proposed algorithm improves prediction performance and outperforms KNN

Figure 3.7: Real data analysis: Comparison between similarityMix and FlexMix

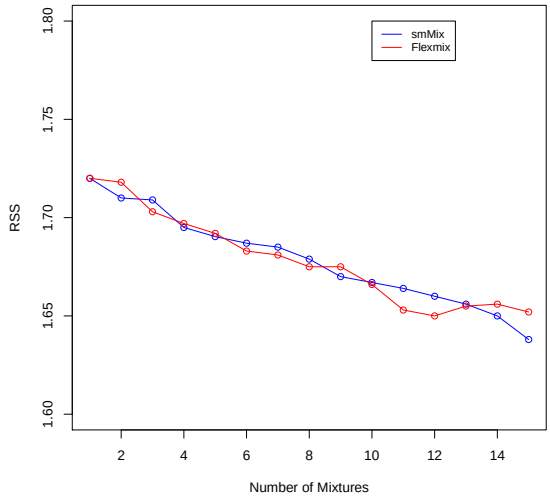


Figure 3.8: Real data analysis: CCU patients

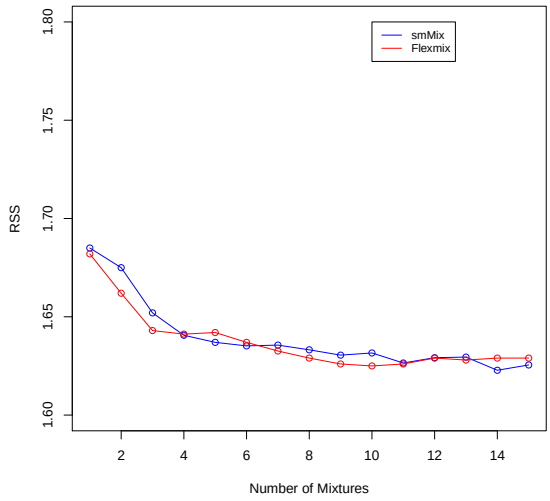


Figure 3.9: Real data analysis: CSRU patients

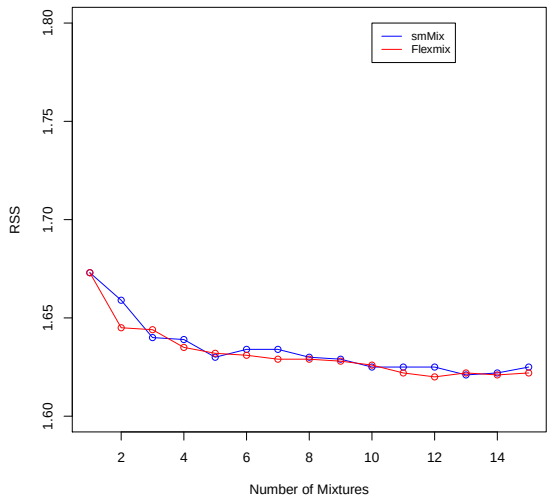


Figure 3.10: Real data analysis: MICU patients

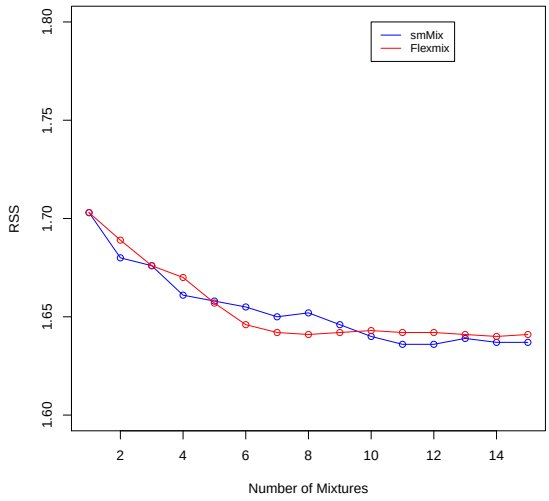


Figure 3.11: Real data analysis: SICU patients

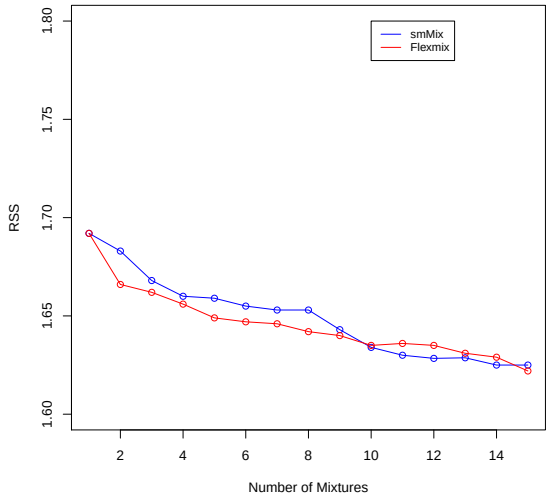
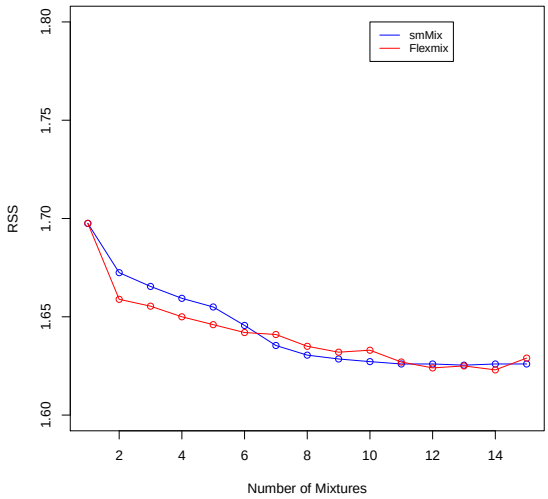


Figure 3.12: Real data analysis: TSICU patients



regression when the true number of mixtures is used in the model fitting, and that it is robust to excess number of mixtures specified.

The result is also consistent with the findings in the empirical study of Lee, Maslove & Dubin (2015) [26] which showed that using a patient similarity metric (PSM) to only include a subset of similar patients improves mortality prediction in the MIMIC-II database.

Similar results can be seen when the number of variables increases to 20, with sample size 1000, but further investigation is still needed in high dimensional cases, that is, when the number of covariates p, is close to, or even greater than the number of samples n in which dimension reduction is required. Localized methods such as KNN are known to suffer from curse of dimensionality since the local points in lower dimensions are no longer local in higher dimensions [15]. A definition of "local" such as the one discussed in this section might be more meaningful than the distance of the covariates. But problems such as over-fitting on the training dataset might be pronounced, since a large number of covariates will be used in both the first step of fitting the model and in the second step of finding similarity variables. Therefore, under the high dimensional setting, an effective variable selection method for both the first and the second step of the algorithm is needed.

In Section 3.3, I briefly mentioned a possible heuristic for selecting the number of mixtures by finding an elbow in the RSS vs. number of components plot and demonstrated this method in real data analysis in Section 3.4. It might be advantageous to have a large number of specified mixture components. Then the components can be examined, and the similar ones can be combined into one component. Finding more comprehensive ways of selecting the number of mixture components in this context is an area of future work.

Although our similarity-based prediction method is developed on mixture of linear models, it can be generalized to mixture of generalized linear models and regression trees.

Mixture of survival models can be considered too, as people have studied the problem of mixture models in the context of survival data. For one example, Farewell (1982) [12] proposed to model both "long-term survivors" and "short-term survivors " in the population. In the ideal case with independent censoring and with parametric assumption (such as Weibull distributions), the algorithm can be adapted to a survival outcome by replacing the mixture of Gaussian likelihood with the (partial) likelihood function of the mixture of survival model. One of the challenges with survival data is, with localized models, especially with the existence of censoring, there might be an insufficient number of events of interest to get a reasonable estimate for the model. Therefore, the modified algorithm needs to balance between the number of events occurring and the homogeneity of the data.

Our work has connected similarity-based prediction with mixture models, and we have showed that localizing using a meaningful definition of similarity based on the covariates and the outcome could improve prediction performance when there is a mixture of regression functions in the dataset.

Chapter 4

Extending Similarity-based Prediction to Time-to-event Data

4.1 Introduction

Electronic Health Records (EHRs) have generated health data sets that provide rich and diverse information for modeling and prediction. Survival analysis has been essential in clinical and epidemiological studies, and both parametric and semiparametric modeling have been utilized in the literature. With big dataset, patients can be heterogeneous, which pose challenges to accurate prediction of outcome of interest. Conditioning on a more relevant subset where the cases are more similar to the point of prediction might improve prediction. Similarity-based prediction in other prediction context has been focused upon by Lee, Maslove & Dubin (2015) [26]. The concept of similarity using random forests idea is seen in Xu, Nettleton & Nordman (2016) [57] paper for regression and classification. Lee (2017) [25] applied the case-specific random forests of Xu, Nettleton & Nordman (2016)

[57] to MIMIC data.

In survival analysis, the idea of similarity is seen in cure models. These models assume that while some cases will die from a disease or experimental stress, a sub-population will survive for a long time. Although the term similarity is not specifically mentioned in the literature, the sub-population of long-term survivors can be considered as a group of similar cases. Early studies on such models include Boag (1949) [5], Berkson and Gage (1952) [3] and Haybittle (1965) [19]. Pierce, Stewart & Kopecky (1979) [40] suggested a computationally easy method to deal with grouped survival data based on Cox proportional-hazard model. V. T. Farewell (1982) [12] and Kuk & Chen (1992) [24] used a mixture model representation for the two populations, which models the probability of being a long term survivor with a logistic regression and the time to event for those that would experience the event with survival models. Many variations of mixture cure model can be seen in literature. Tsodikov, Ibrahim and Yakovlev (2003) [54] provided an alternative to two-component mixture models in estimating cure rate by using bounded cumulative hazard function. These models focus on modeling rather than prediction.

We take a rather different approach to model and predict survival data when there is one or more sub-populations in the dataset, that is, when the relationship between the time-to-event outcome and the explanatory variables are the same within groups and different between groups. This is a more general case than the cure model as there can be more than two groups in the population, and the number of groups is unknown. Note that the similarity is not just based on the grouping of the survival time, or the closeness of the explanatory variables, but depends on the relationship between the two. Tree-based method such as random forests [7] are a natural way of incorporating both outcome and covariate information, and can be utilized to characterize similarity as cases in the same terminal node can be considered as similar to each other. Random forests methods have

been extended to survival data as well, as in Ishwaran et al. [22], and our approach is essentially combining the case-specific random forests model in Xu, Nettleton & Nordman (2016) [57] with random survival forests model [22]. Methods for dealing with dependent right censoring will be discussed as well.

In Section 4.2.1, we will first review the basic concepts and techniques in survival analysis. Then we introduce the similarity-based prediction in survival analysis by an illustration in Section 4.3. In Section 4.4.1, we will discuss the similarity-based random survival forests algorithm with independent right censoring, and methods to adjust for dependent censoring. Time-varying AUC is used as the criterion for prediction performance. Section 4.5 and 4.6 are applications of the algorithm to both simulated and a real dataset. In Section 4.7, we will summarize our methodology and findings from simulated and real data analysis.

4.2 Introduction to Survival Analysis

This section will provide an overview of the basic concepts of the analysis of survival data. Let T_i denote the non-negative random variable of survival time under study of subject i, where $i = 1, 2, ..., n$. The survival function describes the probability that the survival time is greater than t, i.e., the distribution of T_i. Assuming that survival time is continuous, the survival function is defined as

$$S(t) = Pr(T_i > t) = \int_t^\infty p(s)ds$$

where $p(s) = dF(s)/ds$ is the corresponding probability density function. $S(t)$ must be non-increasing with $S(t = 0) = 1$. Another important function is the hazard function $\lambda(t)$, which is the instantaneous risk for an event in the interval $[t, t + \delta t]$ provided that the individual did not experience an event prior to time t. It is defined as

$$\lambda(t) = \frac{\lim_{dt \to 0} Pr(t \leq T_i < t + dt | T_i \geq t)}{dt}, t > 0$$

The survival function can also be written in terms of the hazard function as

$$S(t) = exp\{-\Lambda(t)\} = exp\{-\int_0^t \lambda(s)ds\}$$

where $\Lambda(t)$ is the cumulative hazard function, which describes the accumulated hazard up to time t.

Survival times are often subject to censoring. In follow-up studies for example, participants might drop out of study before experiencing the event of interest. Generally speaking, censoring occurs when the event is only known to have occurred inside a time interval instead of at a known time. With regard to the relative position of the time of censoring and the time of the occurrence of the event, censoring can be classified into three categories, right censoring, left censoring and interval censoring. For right censoring, the event is only known to have occurred after a certain time point. If we denote the censoring time by C_i, then the observed time X_i will be the minimum of the event time and the censoring time, i.e., $X_i = min(C_i, T_i)$. Denote the event indicator by δ, which indicates the observed time corresponds to the true event time. Then $\delta = I(T_i \leq C_i)$, which is 1 if the event occurs before censoring, and 0 otherwise. For left censoring, the event is only known to have occurred before a certain time point. That is, $X_i = max(C_i, T_i)$, and $\delta = I(T_i > C_i)$. For interval censoring, the event time is only known to be inside a time interval $(C_{il}, C_{ir}]$. The event indicator $\delta = I(C_{il} < T_i \leq C_{ir})$.

Censoring can also be classified into informative censoring and non-informative censoring according to whether or not the censoring depends on the event process. In informative censoring, whether the censoring happens directly relates to the expected time to event.

On the other hand, for non-informative censoring, the censoring does not directly depend on the time to event process, although it can depend on some covariates. In this chapter we will mainly focus on right censoring and non-informative censoring.

In survival analysis, one is interested in estimating the survival function or the hazard function with available information. The data is usually of the form $\{T_i, \delta_i\}$. The most well-known nonparametric methods include the Kaplan-Meier estimator for estimating the survival function

$$\hat{S}_{KM} = \prod_{i:t_i \leq t} \frac{r_i - d_i}{r_i},$$

where r_i denotes the number of subjects in the risk set at time t_i, and d_i denotes the number of events at t_i; and the Nelson-Aalen estimator for estimating the cumulative hazard function

$$\hat{H}_{NA} = \prod_{i:t_i \leq t} \frac{d_i}{r_i}.$$

If we assume a parametric form for the survival function, then maximum likelihood methods can be used, and the log-likelihood function is

$$l(\theta) = \sum_{i=1}^{n} [\delta_i \log h(T_i; \delta) + (1 - \delta_i) \log S(T_i; \delta)].$$

We are generally interested in modeling the effect of covariates on the risk as well. If we assume a multiplicative effect of the covariates on the hazard for an event, we will have a Cox proportional hazard model. The hazard function is formulated as

$$\lambda_i(t|w_i) = \lambda_0(t) exp(\gamma^T w_i),$$

where w_i denotes the p-dimensional vector of covariates that are assumed to be associated with the risk, and function $\lambda_0(t)$ is called the baseline hazard function. If the baseline hazard assumes a parametric form, then a parametric PH model can be obtained. For

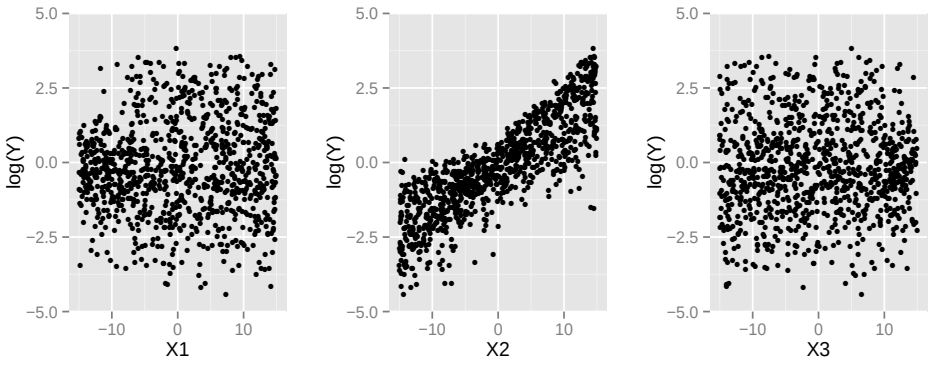


Figure 4.1: Under Survival Analysis Setting

example, if $\lambda_0(t) = \kappa\rho(\rho t)^{\kappa-1}$, where κ is the shape parameter and ρ is the scale parameter of a Weibull distribution, we have a Weibull proportional hazards model.

4.3 An illustration

We will first use a simulation example to illustrate the similarity-based approach in survival data.

We consider Y_i to be a time-to-event outcome that follows a Weibull distribution with $shape = 2$ and

$$\begin{aligned} log(scale) =& I_A(X_1, X_3)(0.2X_2 + 0.2) \\ &+ (1 - I_A(X_1, X_3))(0.07X_2 + 0.04). \end{aligned}$$

X_1, X_2, X_3 are independently and uniformly generated from $(-15, 15)$, and uniform and independent right-censoring is implemented. Figure 5.1 shows the plots of the time-to-event outcome Y vs each predictor. Similarly, we fit different survival models for each case j using only the k neighborhood cases and look at the residuals between the true

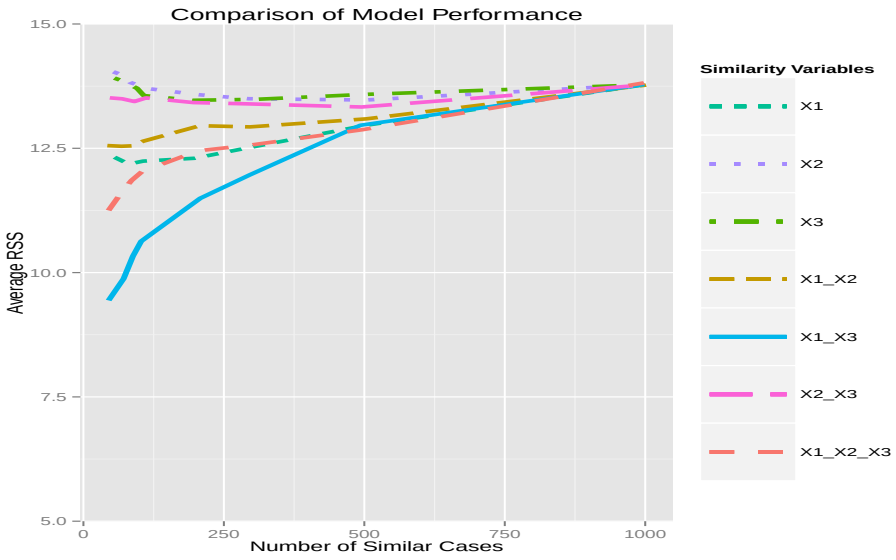


Figure 4.2: Survival Model Results

event time and the predicted. We want to see if similar patterns can be seen when Y is a time-to-event outcome.

Figure 5.2 shows the averaged residual sum of squares for a time-to-event outcome. In this case, we have to choose a bigger k than that used in the previous two cases simply because when k is too small, there are an insufficient number of events to get a reasonable estimate of the model. As a result, we do not see the U-shape in the averaged residual sum of squares for smaller k, as we did in Cases 1 and 2 in Chapter 3 (see Figures 3.2 and 3.4, respectively). Similar to the previous two cases, if we choose the correct combination of similarity variables the model performance is significantly better, and consideration of a lower number of similar cases (i.e., smaller neighborhood) leads to better results than for a larger neighborhood.

4.4 Similarity-based Random Survival Forest

In this section, we will introduce the algorithm for the similarity-based random survival forests. The idea is to build different random survival forests for prediction for each test case, giving more weights to the training cases that are in more close proximity of the test case, and using less information from those that merely add more noise to prediction. We will discuss independent censoring case in Section 4.2.1 and dependent censoring in Section 4.4.2. In Section 4.4.3, we will talk about using time-varying AUC for model comparison.

4.4.1 With Independent Censoring

We will assume independent censoring for now. Methods to incorporate dependent censoring will be discussed in Section 4.4.2.

- 1. Construct a regular random survival forests model for a training dataset that has sample size N_{train}.
  - (a) Draw B bootstrap samples from the training data. Uniform sampling is used.
  - (b) Grow a survival tree for each bootstrap sample under the constraint that it should have $d_0 > 0$ unique deaths.
- 2. For each point in the test dataset of size N_{test}, obtain a weight vector based on the random survival forest in the first step.
  - (a) Pass a test data point down each tree in the random survival forest, and keep track of how many terminal nodes group a training data point with the test point.

  - (b) Assign a weight vector of length N_{train} to each test data point based on how many terminal nodes group a training data with that test data point.
  - (c) Iterate through each test data point, and obtain a weight matrix of size $N_{train} \times N_{test}$. Normalize each row of the weight matrix so that each row sums to 1.

- 3. Build different similarity-based random survival forest for each test data.
  - (a) For a test data, build a random survival forest model with the weight vector as the sampling probability vector in the bootstrap.
  - (b) Pass down the test data in each tree, and calculate the CHF of the the terminal node to which the test data point belongs.
  - (c) Average among all trees to get an ensemble CHF for that test data.
  - (d) Repeat it for each test data. Note that a different survival forest is built for each test data.

4.4.2 Adjusting for Dependent Censoring

Dependent censoring for right censored data is common in epidemiological studies. For right censoring, the event is only known to have occurred after a certain time point. Denoting the censoring time by C_i, the observed time X_i will be the minimum of the event time and the censoring time, i.e., $X_i = min(C_i, T_i)$. Denote the event indicator by δ, which indicates the observed time corresponds to the true event time, then $\delta = I(T_i \leq C_i)$, which is 1 if the event occurs before censoring, and 0 otherwise. For non-informative censoring, the censoring process does not directly depend on the event process, although it can depend on some covariates. With informative censoring, the censoring happens directly

relates to the expected time to event. Inverse probability-of-censoring weights (IPCW) has been shown to account for the bias [21, 35]. The algorithm is modified as follows.

- 1. Use K-M estimator with censoring time as the event time to get the probability $P_i(C)$ of getting censored.
- 2. Calculate the IPC weights for each training case as $IPCW_i = 1/(1 - P_i(C))$, ie. the weights are equal to the inverse probability of not getting censored.
- 3. Calculate the similarity weights for a training case i and test case j as SWi, j as described in Section 4.4.1. The sampling weights for use in the similarity-based random survival forests for i and j will be proportional to $IPCW_i * SW_{i,j}$.

The intuition behind the multiplication of the weights is that, the algorithm now gives greater sampling weight to those data points that are more likely to be censored.

4.4.3 Prediction Accuracy

We will be using time-varying area-under-the receiver operating characteristic curve ROC curve (AUC) for model comparison. For binary outcomes, the prediction accuracy can be characterized by ROC, which plots the $sensitivity$ against $(1 - specificity)$ for all different threshod values. And the area under ROC (AUC) represents a measure of prediction accuracy.

For time-to-event outcome, there are a few proposals to generalize the concept of sensitivity and specificity [20]. One way is to look at sensitivity and specificity at each time of interest t. The survival probability up to t_k of a test case i, i.e., $S_i(t)$ can be derived from its cumulative hazard $\hat{H}_i(t)$. Then, $AUC(t)$ can be calculated at each t. In this chapter, we will consider the AUC over a dense grid of times.

4.5 Simulations

We use two simulated examples to further explain what similarity means in the model and demonstrate the prediction performance of the algorithm.

4.5.1 Example 1

In a simple example, each case has a 3-dimensional covariate $\{x_1, x_2, x_3\}$ that links directly to the survival outcome. Two of the covariates are linked to similarity as well. In this case, S is a survival outcome that follows a Weibull distribution with shape=2, and log(scale) mapped to linear predictor Y.

$$\begin{cases} Y = 0.2X_1 - 0.1X_2 + 0.5X_3, & \text{if } (X_1 + 7) * (X_3 - 10) > 0 \\ Y = 0.3X_1 + 0.1X_2 - 0.3X_3, & \text{otherwise} \end{cases} \tag{4.1}$$

Note that $(X_1 + 7) * (X_3 - 10) <= 0$ and $(X_1 + 7) * (X_3 - 10) > 0$ describes a binary tree structure that clusters cases into two subspaces. Within each subspace, the relationship between the survival outcome and the covariates are the same. 1000 cases are generated, where X_1, X_2, X_3 are independently and uniformly generated from (-15,15). Uniform right censoring (independent for now) is considered. Tuning parameters are chosen from a grid of parameter values, and are determined from the entire dataset. Figure 1 summarizes the comparison between the prediction performance of similarity-based random survival forests and the regular random survival forest. The red dots represents the time-varying AUC for the similarity-based random survival forests and the black dots are for the regular random survival forests. The AUCs are evaluated at each day from day 1 to day 20. At each day, the AUC of the similarity-based method exceeds the regular random survival forests by around 0.02.

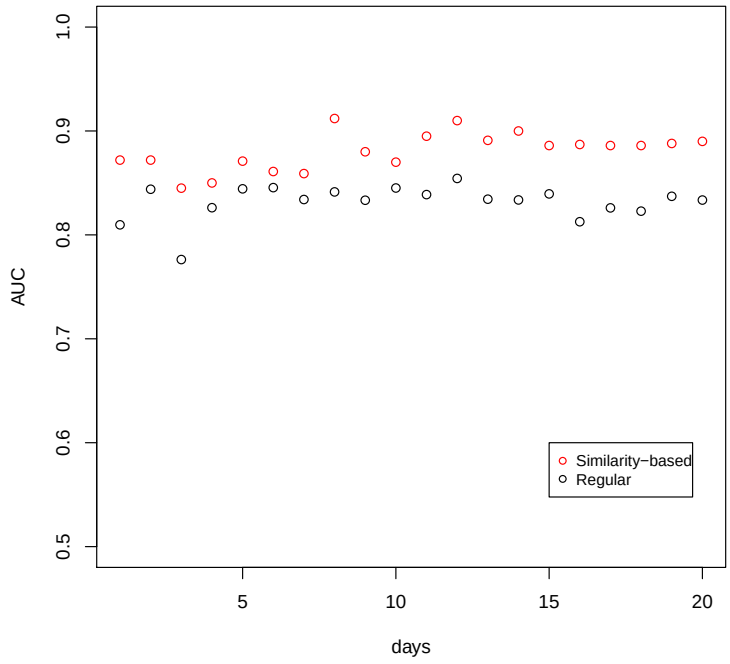


Figure 4.3: Time-varying AUC for simulated data in Example 1

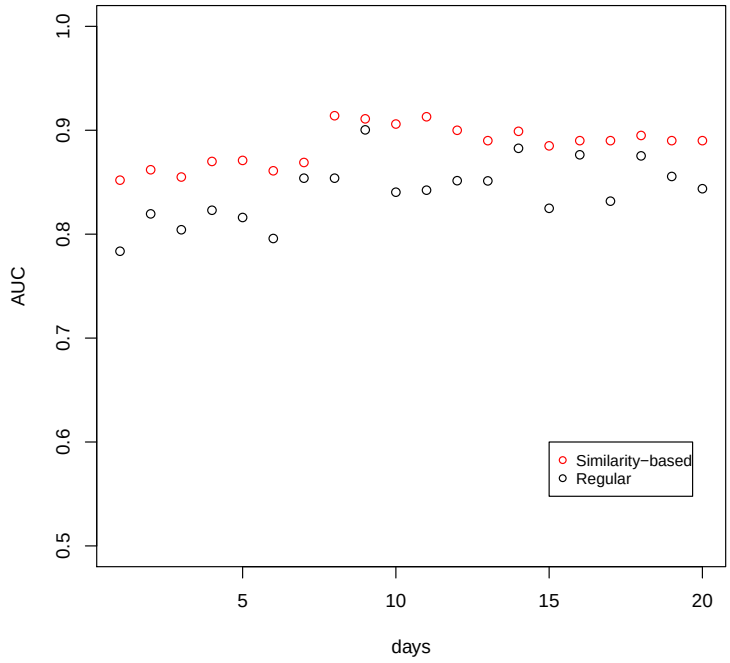


Figure 4.4: Time-varying AUC for simulated data in Example 2

4.5.2 Example 2

In the second model, each case has a 5-dimensional covariate $\{x_1, x_2, x_3, x_4, x_5\}$, where three of them also explains similarity. Again we will use a binary tree structure to define subspaces. In this case, we will prune the tree until there are four terminal nodes, i.e., four subspaces. Within each subspace, the relationship between Y and the covariates are the same.

The result in Figure 2 is similar to the first case. Giving more weights in the sampling to similar cases yields better prediction performance in the random survival forests framework.

4.6 Application to an ICU dataset

4.6.1 MIMIC-III

MIMIC-III (Medical Information Mart for Intensive Care III) is a freely accessible critical care database for 53423 distinct hospital admissions for adult patients (aged 16 and above). Data includes vital signs, medications, diagnostic code, survival data and high resolution data including lab results and bedside monitoring data [23]. This large dataset provides rich information for modeling and prediction, but the diversity of the patients also poses challenges to accurate prediction of outcome of interest. To illustrate the the goal is to predict ICU patient survival with their age, gender, ICU type, admission type, SAPSII score as predictors. ICU type includes CCU (Coronary Care Unit), CSRU (Cardiovascular Intensive Care Unit), MICU (Medical Intensive Care Unit), SICU (Surgical Intensive Care Unit) and TSICU. And admission type includes Elective, Emergency, Urgent. Only the first hospital admission of adult patients (older than 15 years of age) are included in our study. Excluding cases with missing data in one or more of the variables or outcome, the sample size is 38604. In this dataset , 80% of the cases are right censored at 90 days after for de-identification purposes.

4.6.2 Results

Figure 4.5(a) compares the time-varying AUC for the algorithm in Section 2.1 with the random survival forests method. The time-varying AUC from our proposed method outperforms that of the regular random survival forest at the beginning of the prediction and after day 20, the gap between the two lines increases as we predict further into the future.

Figure 4.5(b) shows the result when considering possible dependency in the censoring.

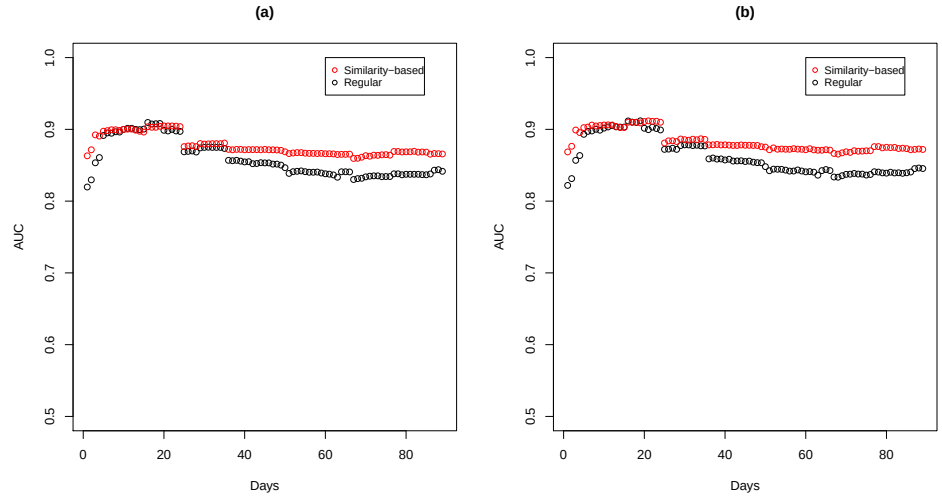


Figure 4.5: Time-varying AUC for application to MIMIC III dataset. (a) Ignoring the dependency in censoring (b) Adjusted for dependent censoring

The result is much similar to that in Figure 4.5(a). It is possible that for this dataset, there is not much dependency in the censoring and thus the calculation of the weights did not have a big impact on the result.

4.7 Discussion

In this study we proposed to improve the random survival forests by incorporating the similarity structure between a test data point and training data point. Instead of building a global random survival forests for each test case, we construct similarity-based random survival forests for each one of them, by giving more weights to the training cases that are in closer proximity to the test case. Proximity is measured using a regular random survival forests model. We also developed algorithm to account for dependent censoring which is

common in survival data.

Both simulations and a real data example shows promising result that in general, that the similarity-based prediction improves prediction performance of random survival forests in terms of time-varying AUC. This result is also consistent with other findings using similarity structure outside of the random forests model [26].

This method requires building a random survival forest for every test data point and requires a few tuning parameters. Specifically, the tuning parameters are the depth of the tree (represented by the number of unique deaths in the terminal nodes), and the number of trees in the forests. This can be computationally intensive when the size of the test data size is large. Future work is necessary to investigate the robustness of their choices. One way of reducing computation time is to use a hard threshold for sampling, that is, giving 0 weights to cases that are too far away from the test case. The tuning parameters for the simulations are selected based on the entire dataset. However, if they are determined from a smaller subset of the dataset, the computational time might be greatly reduced.

For future work, methods other than random forests may be utilized for similarity-based prediction for survival outcomes. One possible extension is the joint modeling of longitudinal covariates and time-to-event outcome [42]. One might be able to identify similar cases based on longitudinal covariates as well as time-fixed covariates.

Chapter 5

Determining the length of monitoring window for longitudinal covariates in prediction models from follow-up studies

5.1 Introduction

In biomedical health studies, covariates, such as blood pressure, blood glucose, pulse, etc. are often measured repeatedly over time. Predicting a future outcome with such longitudinal covariates usually requires a monitoring window on covariate trajectories. Generally speaking, the values of longitudinal covariates that are observed more recently are more correlated with the outcome, and therefore monitoring the covariates long enough up until the most recent time point provides additional, possibly valuable, information to

make more accurate predictions of near-future outcomes.

In addition, the dependent variable can be a survival outcome, in which case joint models of longitudinal covariates and survival outcome are usually preferred. In joint modeling, the longitudinal covariates are usually followed up to the time when prediction of the survival outcome is needed. For example, Rizopoulos [41] proposed a Monte Carlo approach to estimate risk of a target event and shows how it can be dynamically updated. Taylor et al. (2013) [51] developed a Bayesian method with a Markov chain Monte Carlo (MCMC) algorithm to dynamically predict both the longitudinal PSA measures and the recurrence rate. Blanche et al.(2015) extended the survival submodel to account for competing events [4]. Rizopoulos et al. (2013) [43] compared joint models for longitudinal and time-to-event data and landmark analysis (van Houwelingen, 2007 [55]), an alternative approach for dynamically updating survival probabilities. In these papers, the predictor can be dynamically updated as new longitudinal measurements become available for the target subjects, which provides updated risk assessment.

However, there are some scenarios where prediction would be more useful to be made at an earlier point, before the full trajectories of the longitudinal covariates are observed. A simple example is that in weather forecasting, monitoring relevant covariates up to an hour before the time of prediction will give quite accurate weather predictions at the end of the hour, but that result is not that useful because people generally would like to know the weather a few days in advance. On the other hand, predicting weather a month from now is usually not that accurate because of the lack of recent covariate information.

This can also be "early warnings" of a "tipping point" in some complex dynamic systems [47]. For example, in the financial market, one would like to have an early warning of systematic market crashes [9], or in Earth science, an early warning system that detects some tipping point of climate change is useful[29].

In health/medical data analysis, patients get censored due to a variety of reasons, such as hospital discharge as a result of recovery from a disease, or death. Waiting too long before making a prediction will exclude these patients from the study and the result might not be generalized to the patient population. One example comes from the AKI study introduced in Chapter 2. After AKI onset, patients can either recover from the disease, die, or get discharged from the hospital without recovering. In that study, among the 3599 patients that developed AKI, 189 died, 531 recovered from AKI, and 109 were discharged from hospital within 24 hours after AKI onset. Monitoring the peak creatinine value after AKI onset(which is an important predictor of the recovery from AKI) for 48 hours yields better predictive power than monitoring it for only 24 hours. But the downside of the approach is that a prediction can only be made after 48 hours after AKI onset, and will exclude all the patients that experienced an event (Recovery/Death/Censoring) in this 48-hour observation window. More importantly, physicians logically like to know, sooner than later, if a patient is at elevated risk of a future event, so that an appropriate treatment course can be provided as soon as deemed necessary.

In this chapter, predicting a continuous outcome at future time is considered. I will quantify the trade-off between prediction accuracy and early prediction, and identify scenarios where an early prediction can be made, particularly through a comprehensive simulation study. In Section 5.2.1, I will define the problem and introduce all notation. In Section 5.2.2, I propose an algorithm to select the length of the monitoring window. A comprehensive simulation study is conducted in Section 5.3 to further explain how this method works and demonstrate the performance of the algorithm in various scenarios. In Section 5.4, a detailed discussion is provided of the methodology.

5.2 Methods and Algorithm

5.2.1 Problem set-up and notations

The model considered here is similar to a finite distributed-lag model, where both the current and lagged values of explanatory variables are being used in a linear regression model to predict the outcome. I will focus on the case of one explanatory variable. It is assumed that the variables for different individuals are taken at the same time, and are independent of each other. For N individuals with up to a maximum of p lagged values of the explanatory variables, at a given time J where a prediction is needed,

$$Y_i(J) = \beta_0 + \sum_{p=1}^{P} \beta_p X_i(J-p+1) + \epsilon_i(J), i = 1, 2, ..., N \tag{5.1}$$

where $P \leq J$, $\epsilon_i(j) \sim i.i.d.N(0, \sigma_e^2)$, $X_i(j) \sim N(0, \sigma^2)$ and, in general, $X_i(j)$ and $X_i(j')$ are independent for all time points $0 \leq j, j' \leq J$, when $i \neq i'$. In the finite distributed-lag model, the regression coefficients are usually either truncated to a maximum lag if the lag distribution is effectively 0 after a certain number of lags, or using a functional form to allow the coefficients to gradually reduce to 0. Unlike that, the set-up here is that $X_i(J-P+1)$ is the first available variable value for individual i, and $X_i(J-p+1), p = 1, 2, ..., P$ are stochastic. To predict the outcome at a future time J, one has to decide how long to monitor the value of X. Ideally, one would like to make that prediction at time $J-P+1$, i.e. $P-1$ steps ahead, while having good prediction accuracy. That means setting $\hat{\beta}_1, \hat{\beta}_2, ..., \hat{\beta}_P$ to zero. Making such restrictions on the coefficients when the true coefficients are not close to zero will lead to biased estimation and likely diminished prediction performance. On the other hand, monitoring the values of X up to time J will give more accurate predictions, but the prediction might not be useful because it is generated too late to be practically useful. While sometimes there is just not enough information to make predictions ahead of

time, there are scenarios where an early prediction can be made with adequate accuracy. This is essentially a variable selection problem with an additional objective, that is to use as little longitudinal information as possible as we get closer to the time of the future predicted event. In other words, the goal is to estimate $\beta_p, p = 0, ..., P$ to get accurate predictions while also maximizing $max(p|p \in [0, P], \hat{\beta}_p = 0)$.

Generally speaking, as discussed above, there is trade-off between how early one makes prediction and how accurate that predicted value is. Sometimes monitoring the covariate for an additional time period does not improve the prediction performance by much, or that early prediction is needed (for example if one knows that a prediction must be made several days ahead). In these cases, OLS estimates of the coefficients are not optimal.

In the next section, an algorithm for selecting the monitoring window length will be introduced.

5.2.2 Algorithm

In order to select variables that are recorded earlier in time and more relevant to the outcome, a LASSO type penalty (Tibshirani 1996 [52]) with modified weights is used. For estimating the coefficients for model (5.1), the objective function to minimize is as follows,

$$\min_{\beta_0, \boldsymbol{\beta}} \{ \frac{1}{N} \sum_{i=1}^{N} [y_i(J) - \beta_0 - \sum_{p=1}^{P} \beta_p x_i(J - p + 1)]^2 \} \text{ subject to } \boldsymbol{\omega}^T |\boldsymbol{\beta}| \leq l. \tag{5.2}$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, ...\beta_P)$, and $\boldsymbol{\omega} = (\omega_1, \omega_2, ..., \omega_P)$ is a vector of penalty weights. In Lagrangian form, the objective function can be rewritten as

$$\min_{\beta_0, \boldsymbol{\beta}} \{ \frac{1}{N} \sum_{i=1}^{N} [y_i(J) - \beta_0 - \sum_{p=1}^{P} \beta_p x_i(J - p + 1)]^2 \ + \lambda \boldsymbol{\omega}^T |\boldsymbol{\beta}|. \tag{5.3}$$

For a given λ, denote the solution to (5.3) as

$$\hat{\boldsymbol{\beta}}_\lambda = \{\hat{\beta}_{0,\lambda}, \hat{\beta}_{1,\lambda}...\hat{\beta}_{P,\lambda}\} = \arg\min_{\beta_0,\boldsymbol{\beta}}\{\frac{1}{N}\sum_{i=1}^{N}[y_i(J) - \beta_0 - \sum_{p=1}^{P}\beta_p x_i(J-p+1)]^2 + \lambda\boldsymbol{\omega}^T|\boldsymbol{\beta}| \quad (5.4)$$

and predicted outcome at time J corresponding to λ to be $\hat{\boldsymbol{Y}}_\lambda(J) = \hat{\boldsymbol{Y}}_\lambda = \{\hat{y_1}, \hat{y_2}, ..., \hat{y_N}\}$. Denote the true outcome as $\boldsymbol{Y} = (y_1, y_2, ..., y_N)$. Then the mean squared error of the prediction is $MSE_\lambda = ||\hat{\boldsymbol{Y}}_\lambda - \boldsymbol{Y}||^2$. In order to make predictions earlier, the coefficients of the variables that are recorded more recently will be penalized more compared to that of the variables that are farther ahead of time. With suitable penalty weights, and by varying λ, the time when an adequate prediction can be made is effectively controlled. One choice of the weights that satisfies our objective is

$$\omega_p = 1/p, p = 1, 2, ...P.$$

The weights are then normalized to sum up to 1. The algorithm is presented in Algorithm 1.

Algorithm 1:

For a given set of the tuning parameter values Λ, partition the dataset into K subsets with equal sample size for K-fold cross-validation;
for *each* $\lambda \in \Lambda$ **do**
    On the entire dataset, solve $\hat{\boldsymbol{\beta}}_\lambda$ with $\omega_p = 1/p, p = 1, 2, ...P$;
    Calculate $Monitoring\_window\_length_\lambda = P - max(p|p \in [0, P], \hat{\beta}_p = 0)$;
    for *each pair of training and test dataset* **do**
        Solve $\hat{\boldsymbol{\beta}}_\lambda$ with $\omega_p = 1/p, p = 1, 2, ...P$ on the training dataset;
        Calculate the mean squared error of prediction on the corresponding test dataset.
    end
    Average the mean squared error from the K subsets to get MSE_λ.
end
Plot MSE_λ against $Monitoring\ window\ length_\lambda$ for each value of λ.

For fitting this linear model with penalties on the coefficients, R package glmnet [13] is used. glmnet fits generalized linear models via penalized maximum likelihood, which is equivalent to (5.2) under the linear setting. $Monitoring\ window\ length_\lambda = P - max(p|p \in [0, P], \hat{\beta}_p = 0)$ indicates how late a prediction is made (longer monitoring window). For example, $Monitoring\ window\ length_\lambda = 1$ means one makes a prediction at time 1, with the first available covariate value only.

From the plot generated from Algorithm 1, one will be able to see how prediction accuracy (in terms of K-fold cross validation MSE) changes with when a prediction is made. The simulations in the next section show that when there is a trade-off between prediction accuracy and how early a prediction is made (i.e. shorter monitoring window), one can decide where a decrease in the length of the monitoring window results in an acceptable deduction in prediction accuracy.

In addition, one could also specify a cost function, with C_1 as the cost of a unit decrease in prediction accuracy and C_2 as the cost of a unit increase in making late predictions (longer monitoring window). The total cost C is then $C = f(C_1 \times MSE_\lambda, C_2 \times length_\lambda)$. Suppose $MSE = h(length)$ from the plot, then the total cost is

$$C = f(C_1 \times h(length), C_2 \times length),$$

and

$$Optimum\ monitoring\ window\ length = \arg\min_{length}(f(C_1 \times h(length), C_2 \times length)).$$

In the special case where C_2 is zero, early prediction is not important, and one will select the monitoring window length that corresponds to the lowest MSE. If C_2 is very large compared to C_1, one would choose the monitoring window to be 1. We will not discuss further in details how C_1 and C_2 are chosen as the values of C_1 and C_2 are problem-specific.

5.3 Simulations

In this section, simulations are conducted to investigate the performance of the proposed method in selecting the length of the monitoring window. The results from weight function $\omega_p = 1/p, p = 1, 2, ...P$ (normalized to 1) and the uniform weight function $\omega_p = 1/P, p = 1, 2, ...P$ are compared.

Although one would expect that early prediction is more realistic in the case where there is some memory in the covariates instead of the covariates being independent, we first assume (in Section 5.3.1) that the covariates for the same individual are independent just for comparison purposes. Of course, it is much more reasonable to consider dependency for the covariate across time points within a person, as these lagged values are usually repeated measures of the same covariate for the same person. This more realistic case will be considered in Section 5.3.2.

5.3.1 Independent covariates

In the first case, 1000 cases are generated from a linear model.

$$Y_i(J) = \beta_0 + \sum_{p=1}^{6} \beta_p X_i(J - p + 1) + \epsilon_i(J), i = 1, 2, ..., 1000 \tag{5.5}$$

$X_i(J - p + 1), p = 1, ..., 6$ are measures of the covariate for individual i from $J - 5$ to time J. $X_i(J) \overset{i.i.d.}{\sim} N(0, 1)$ and $\epsilon_i(J) \overset{i.i.d.}{\sim} N(0, 1)$, $i = 1, 2, ..., 1000$. The coefficients used for simulation are found in Table 5.1, Scenario 1 through Scenario 5. These coefficients are selected so that the relationship between the outcome and the lagged terms of the explanatory variable X varies from highly dependent of recent values of X (as in Scenario 1) on the outcome to similarly dependent on all 5 lagged terms of X (as seen in Scenario

4), and Scenario 5 is a special case where the dependency increases and then decreases. Intuitively, one would not be able to make meaningful predictions three steps ahead (monitoring window length has to be longer than 3) for Scenario 1, but might be able to make early predictions for Scenario 3, 4 and 5.

We will show partial results here and the remainder of the results in the Appendix. Figure 5.1 shows the change in each estimated coefficient with varying λ, and Figure 5.2 plots the change in 10-fold CV MSE when the length of the monitoring windows varies. The black line in Figure 5.2 denotes the results from the proposed weight function and the red line shows the results of applying uniform weights on all coefficients (excluding intercept). The lowest value of both the black and red line are above 3. This means that early prediction is not available at time $j = 3$ or before. For monitoring window length from 4 to 6, the black line is always lower than or equal to the red line, meaning that if one makes a prediction at time 4, 5 or 6, the proposed method yields better prediction performance than using a regular LASSO regression. Another purpose of this approach is to show that given an acceptable prediction accuracy (in terms of MSE), one is able to make earlier predictions with the proposed weights than using the uniform weights.

For Scenario 4, in Figure 5.6, the black line remains lower than the red line given any MSE. This means that one can have a shorter monitoring window for making a prediction at time J without losing prediction accuracy. In this case, one can choose to monitor the covariates up to time 4 or 5.

Similar results can be seen for Scenarios 2, 3 and 5 in Figure 5.3, 5.4 and 5.7 respectively, i.e., that the proposed method gives higher prediciton accuracy given a certain length of the monitoring window.

Table 5.1: Coefficients for simulation

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
Scenario 1	5	1	1	1	0	0	0
Scenario 2	5	1	1	1	0.2	0.2	0.2
Scenario 3	5	1	0.8	0.5	0.3	0.2	0.1
Scenario 4	5	0.3	0.3	0.3	0.3	0.3	0.3
Scenario 5	5	0.1	0.1	0.5	0.4	0.3	0.1

Table 5.2: Penalty weight functions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
Penalty factor	0	1	1/2	1/3	1/4	1/5	1/6
Uniform penalty(regular Lasso)	0	1	1	1	1	1	1

5.3.2 Dependent covariates

In real applications, the longitudinal covariates are seldom independent of their lagged values. This is especially true for repeated measurements within the same individual. That is, the covariates will usually have some memory of the previous values. AR and MA models for covariates are popular in modeling such relationships, and we will use the ARMA(p,q) notation to specify such models; e.g., ARMA(1,0) is AR(1), ARMA(0,1) is MA(1), and so on.. The simulations scenarios are the same as in the independent case, except that for each individual, $X_i(J)$ follows an ARMA(2,1) process with coefficients for AR to be (0.5, 0.2) and MA coefficient being 0.3. Figure A.1 in the Appendix shows the ACF plot for an ARMA(2,1) process up to lag 5.

Figure 5.8-5.12 show similar results to the independent case, where the proposed method

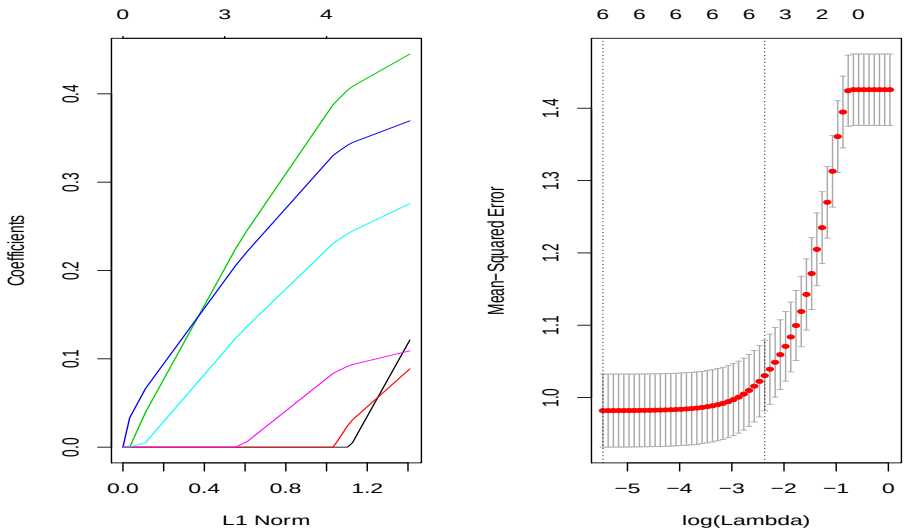


Figure 5.1: Scenario 1 with independent covariates: left figure plots the change in estimated coefficients with tuning parameter λ; right figure plots the 10-fold CV MSE vs.λ, the top is showing the number of coefficients that are non-zero.

is capable of making earlier predictions given a specific prediction accuracy. For example, in Figure 5.11, monitoring window length can be 4 with adequate prediction accuracy.

In addition, an ARMA(3,0) with AR coefficients (0.3, 0.2, 0.1) and an ARMA(0,3) with MA coefficients (0.3, 0.2, 0.1) for the covariates are simulated as well, and the results are shown in the Appendix, Figure A.1-A.13.

5.4 Discussion and future work

Early prediction is quite useful in many cases. However, the earlier one makes a prediction, the more uncertainty there will be in the true outcome and hence generally leads to

decreased prediction performance. When the trade-off between the need/desire to make an early prediction and the level of prediction accuracy exists, quantifying this relationship helps one to decide how long to monitor the longitudinal covariates and how early a prediction can be made without sacrificing too much accuracy. In this chapter, a graphical method utilizing the LASSO regularization (Tibshirani 1996 [52]) to select the length of monitoring window is introduced. This is further compared with a regular LASSO regression and simulations showed that the proposed method seems to have better prediction performance given any length of the monitoring window. That is, for example, if both methods are to make a prediction four days ahead, the proposed method yields a better accuracy. Also, shown through simulation, the proposed method works well whether the covariates are independent or dependent, the latter the more realistic scenario for real data applications. In addition, a cost function can be specified to calculate the optimum monitoring window length.

The simulations are limited to five lagged terms of one covariate. However, when there exist more lagged terms, the penalty weight function will decrease to 0 quickly with additional lags. Therefore, it requires a modification to the weight function. A decreasing function with respect to p that decreases to 0 much slower is needed. This can be problem specific and needs more investigation. Group lasso [58] can be applied here as well, where groups of lagged terms of covariates, as opposed to individual variables, can be selected out of the model for early prediction.

Prediction of a future continuous response can be made from a Bayesian point of view as well [60, 10, 6], using the predictive density that follows from the posterior distribution. Though this will lead to more computational complexity, it will also lead to the advantage of being able to quantify the uncertainty in our predictions.

The proposed method only focuses on modeling a continuous response measured at a

specific index time point in the future. However, predicting a future trend over time or a time-to event outcome is often of interest to physicians as well. Future work includes extending the method to a longitudinal or a survival outcome.

In spite of these limitations, I have developed methods that quantify the relationship between when a prediction is made and the prediction performance, and allows for a selection of the length of monitoring window given a pre-determined prediction accuracy.

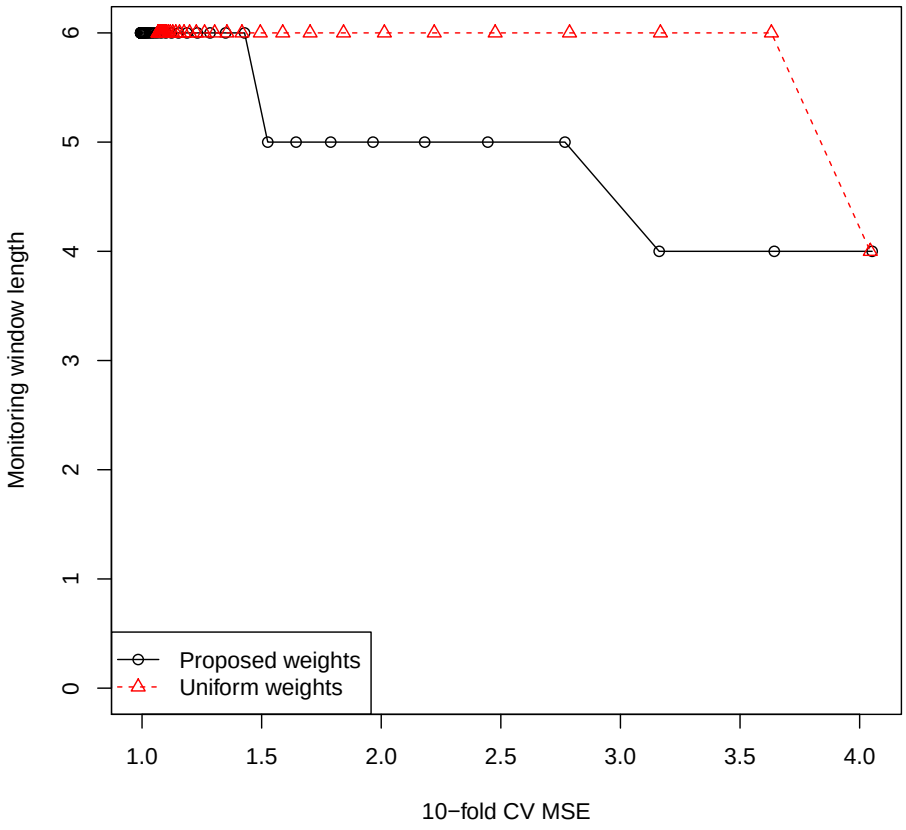


Figure 5.2: Scenario 1 with independent covariates: Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

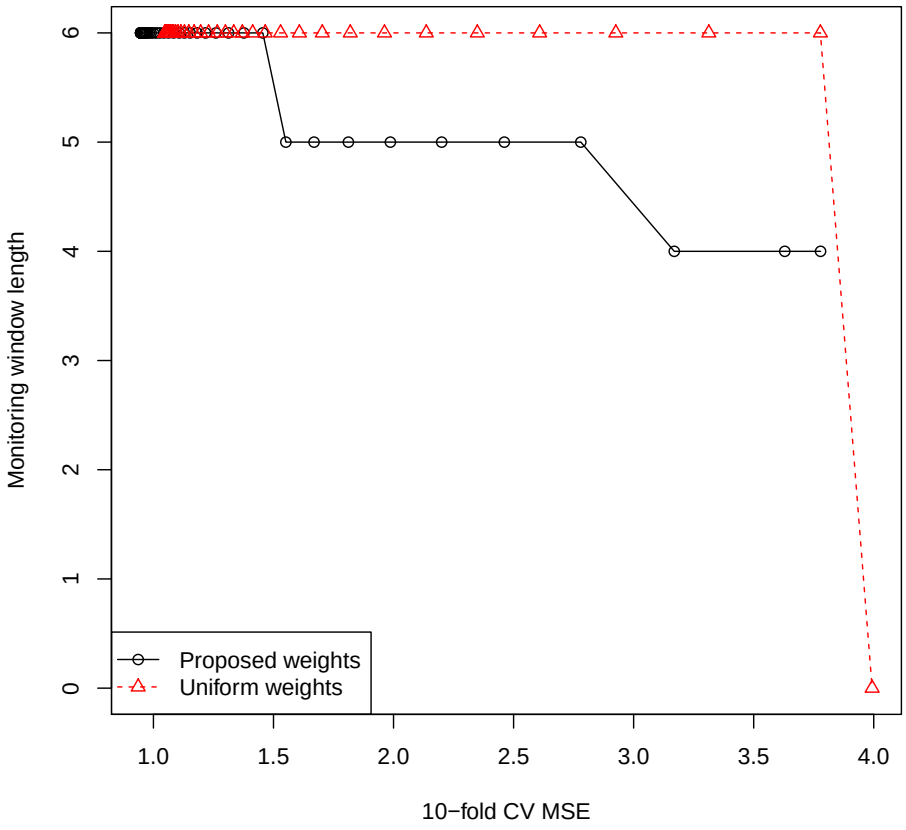


Figure 5.3: Scenario 2 with independent covariates: Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

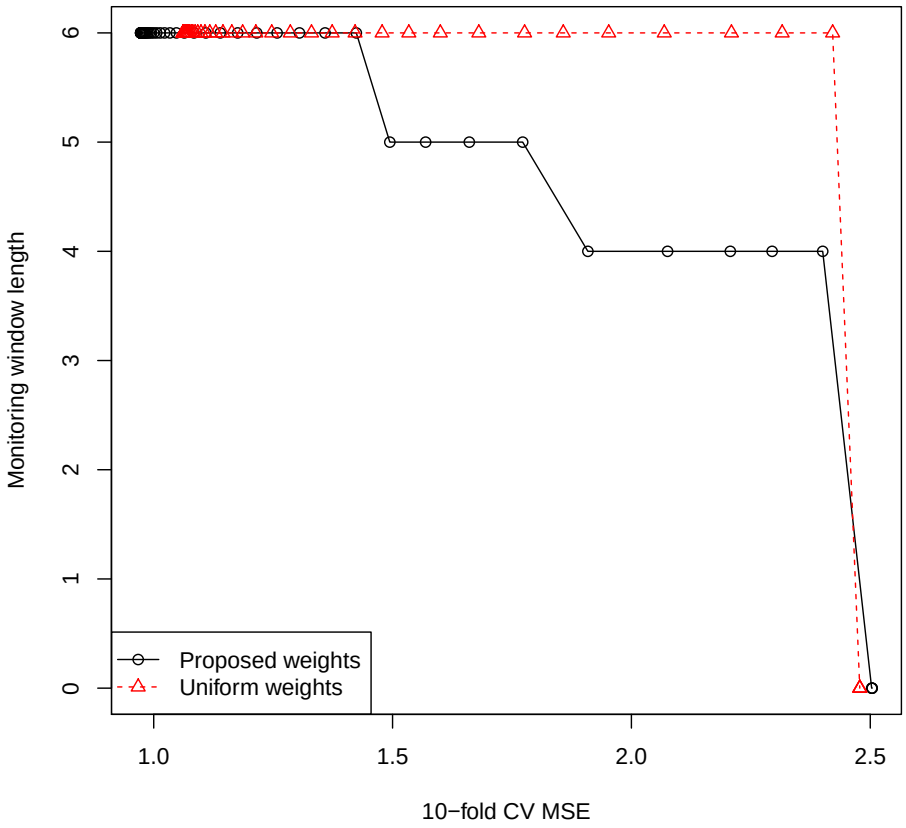


Figure 5.4: Scenario 3 with independent covariates: Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

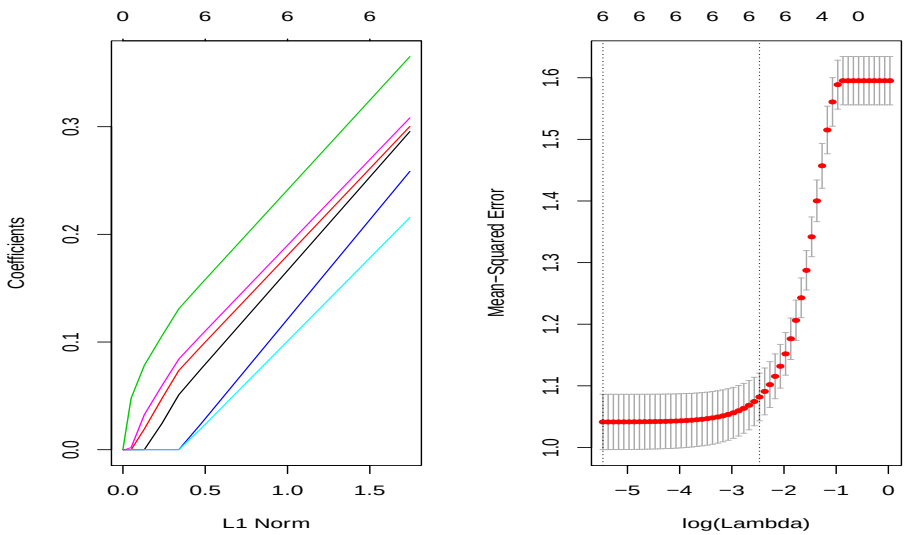


Figure 5.5: Scenario 4 with independent covariates: left figure plots the change in estimated coefficients with tuning parameterλ; right figure plots the 10-fold CV MSE vs.λ, the top is showing the number of coefficients that are non-zero.

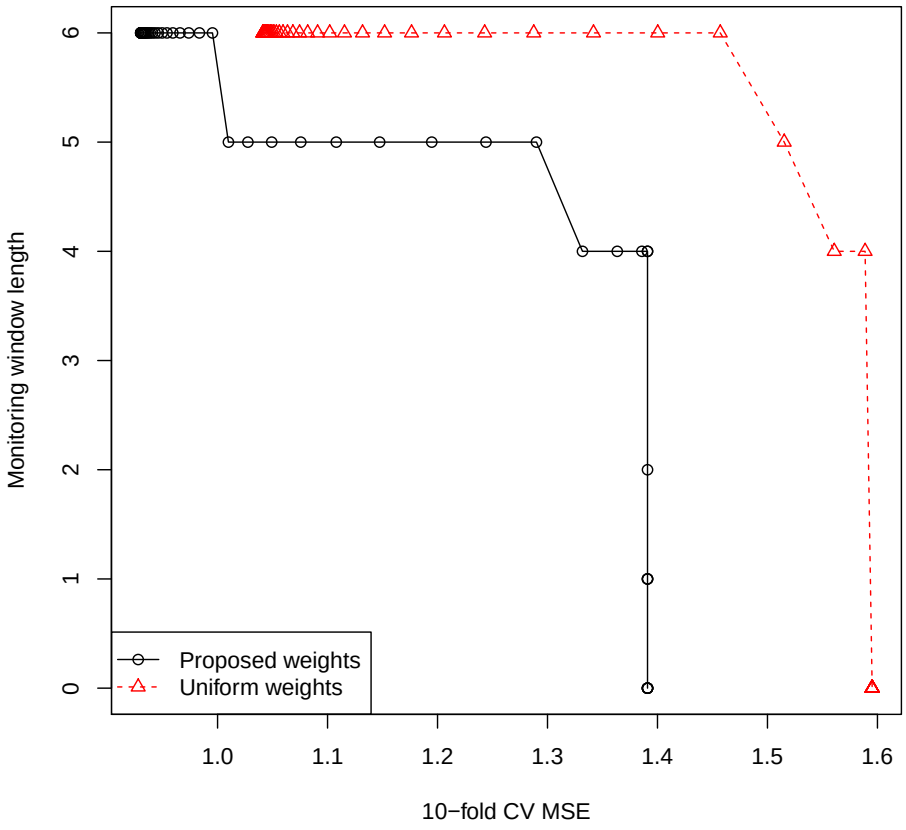


Figure 5.6: Scenario 4 with independent covariates: Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

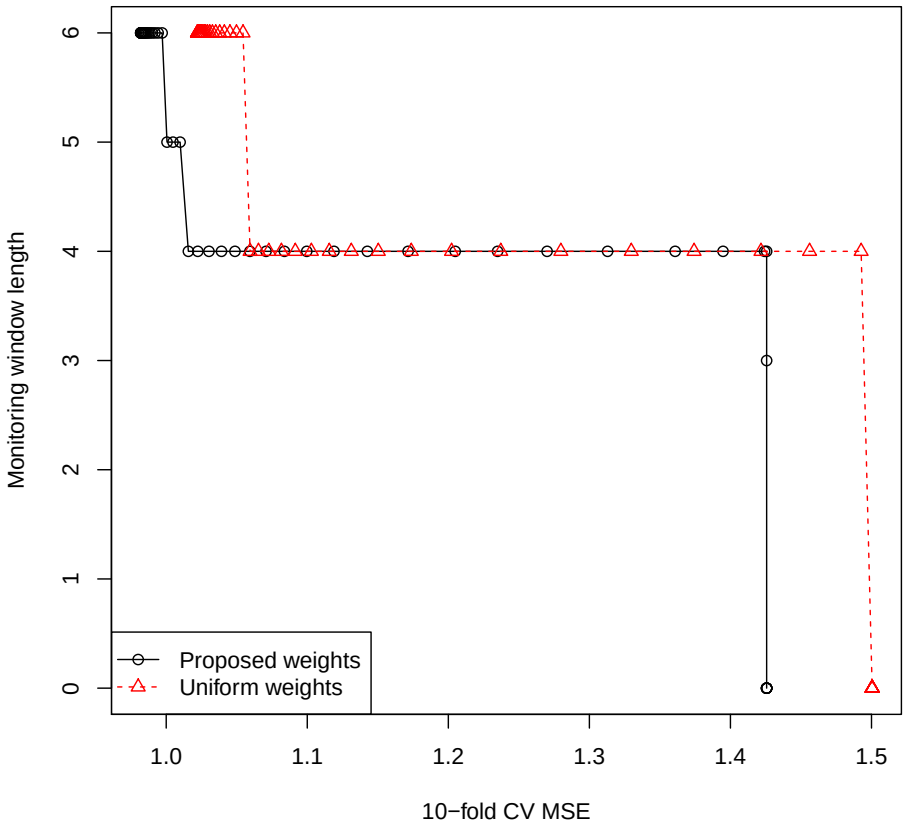


Figure 5.7: Scenario 5 with independent covariates: Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

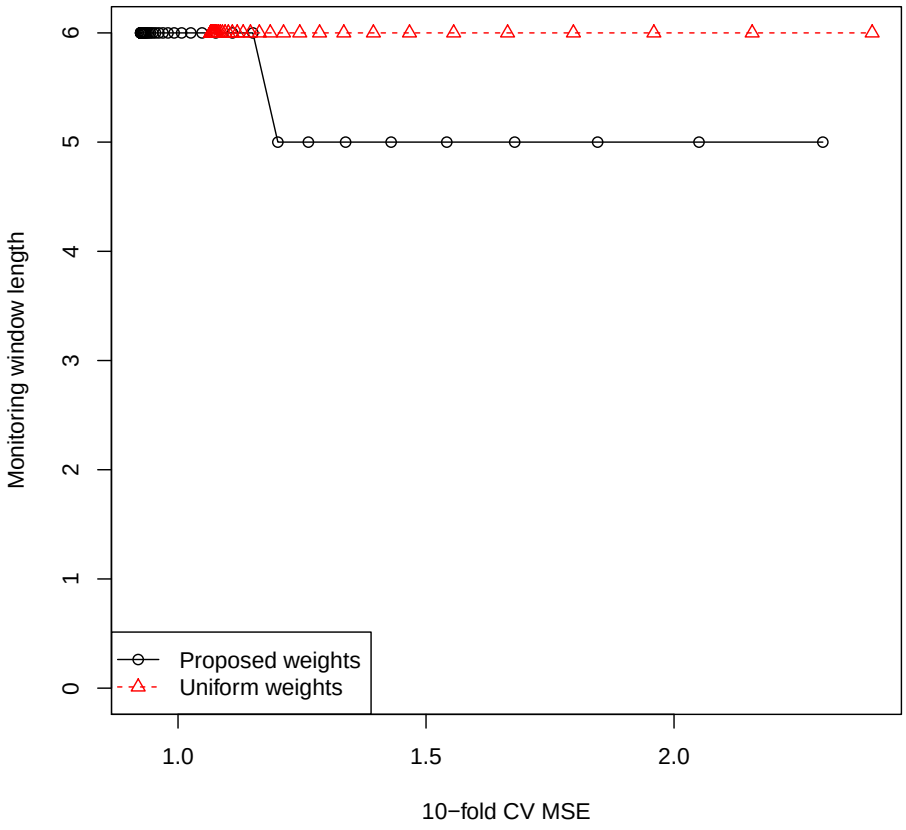


Figure 5.8: Scenario 1 with covariates following ARMA (2,1): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

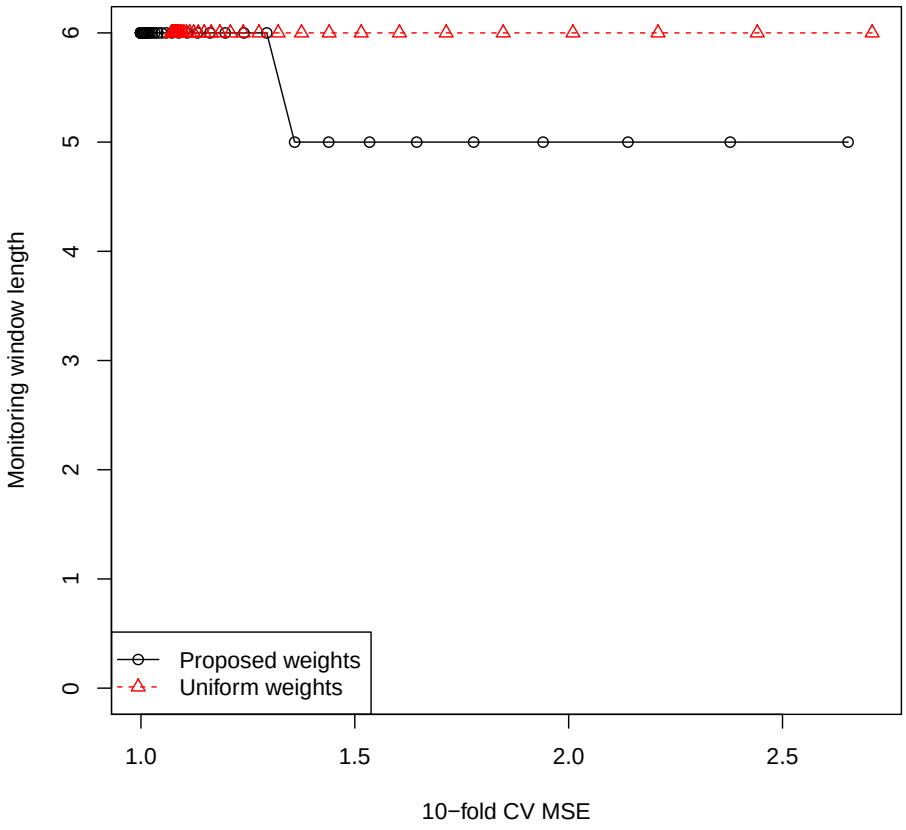


Figure 5.9: Scenario 2 with covariates following ARMA (2,1): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

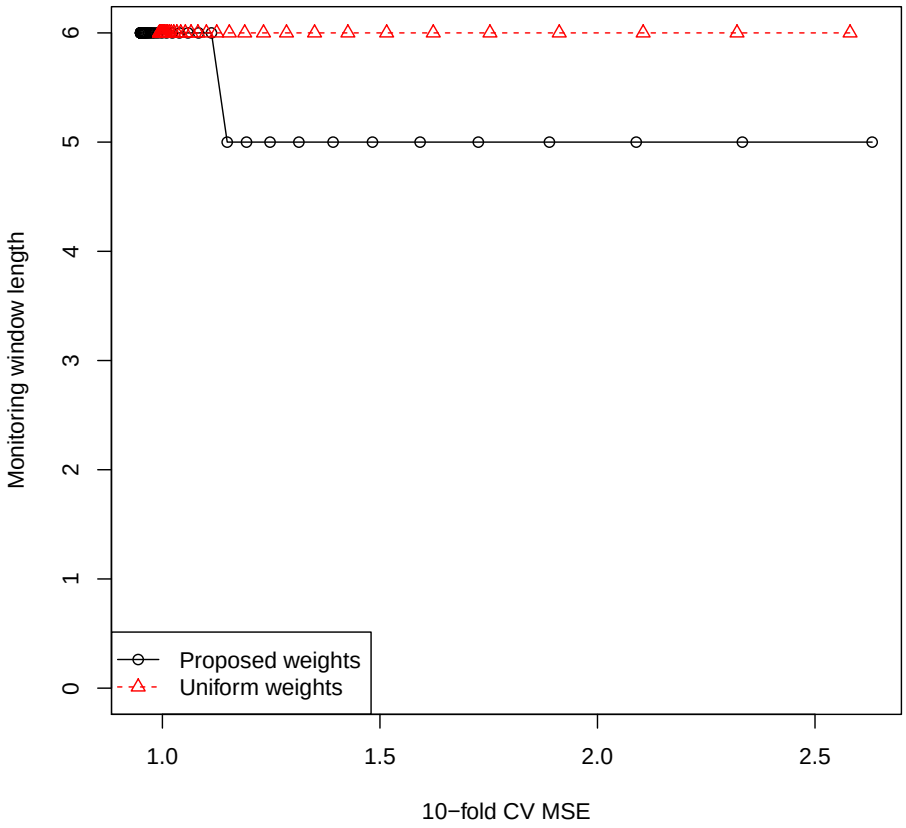


Figure 5.10: Scenario 3 with covariates following ARMA (2,1): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

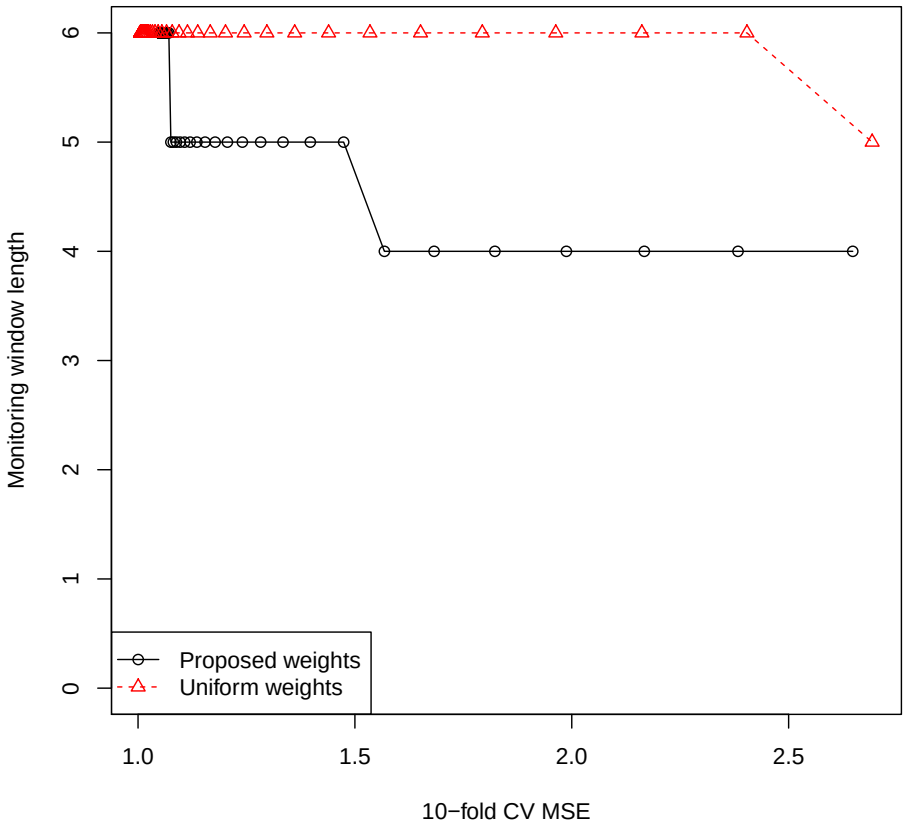


Figure 5.11: Scenario 4 with covariates following ARMA (2,1): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

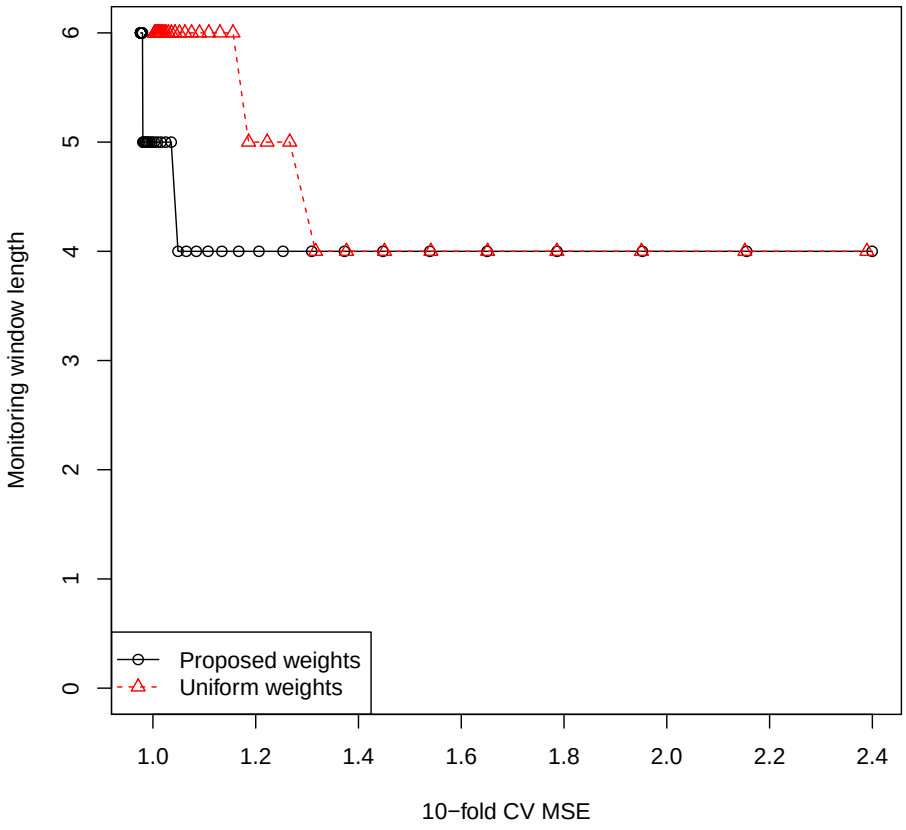


Figure 5.12: Scenario 5 with covariates following ARMA (2,1): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

Chapter 6

Discussion and Future Research

6.1 Summary

In this thesis, I have proposed methods to improve prediction accuracy for three predictive modeling problems. In Chapter 3, a similarity-based algorithm for improving prediction performance for continuous outcome, when there is heterogeneity in the relationship between the outcome and the covariates in the data, is developed. Existing methods that utilize similarity to improve prediction can be seen in a number of papers, and can be summarized into unsupervised methods, such as in Liu et al. (2007) [31], Lowsky et al. (2013) [32], Trivedi, Pardos & Heffernan (2015) [53] and Panahiazar et al. (2015) [37], and supervised/semi-supervised methods, such as in Ishwaran et al. (2008) [22], Xu, Nettleton, & Nordman (2016) [57] for survival outcomes. Mixture models are commonly used to represent the heterogeneity as well. In this thesis, a different approach is taken. The concepts of similarity and similarity variable are introduced. Similarity is defined on the relationship between the outcome and the covariate. The first set of simulations show that

localizing on the similarity variables instead of on all of the covariates and use only the similar cases for regression leads to better prediction accuracy, and provides rationale for the proposed algorithm, similarityMix, that determines similarity cases and models within each subset of similar ones. The second set of simulations demonstrated the prediction performance for similarityMix. A real data analysis is provided as well.

In Chapter 4, prediction for a survival outcome is considered. In survival analysis, the idea of similarity is seen in cure models such as V. T. Farewell (1982) [12], Kuk & Chen (1992) and Tsodikov, Ibrahim and Yakovlev (2003) [54]. This chapter deals with a more general case than the cure model, that heterogeneity is not just based on the grouping of the survival time, as with cure models, but on the relationship between the outcome and the explanatory variables. Therefore, a random forest model is considered as trees in the forest group similar cases together with both the outcome and covariate information. I have improved the random survival forest model by incorporating the similarity structure between a test data point and training point. The algorithm first considers independent censoring data, then is extended to account for independence censoring which is common in survival data, the latter a more general setting. Both simulations and a real data example shows that in general, the similarity-based prediction improves prediction performance of a random survival forest in terms of time-varying AUC.

In Chapter 5, I investigate a different aspect of improving prediction when one or more covariates are longitudinal, that is, how long should one monitor these covariates before an accurate prediction can be made. The main motivation is to balance the common need to make an early prediction (e.g., for a future outcome of a patient in the ICU) with the desire to have inaccurate predictions. The relationship between prediction accuracy and early prediction is quantified, and graphical method is proposed that allows for the selecting of a monitoring window given a specific prediction accuracy.

6.2 Future work

The current work of this thesis can be extended in several directions.

In Chapter 3, a heuristic for finding optimum number of components is mentioned. Finding more comprehensive ways of selecting the number of mixture components in this context is an area of interest. In addition, further investigation is needed in high dimensional cases, that is, when the number of covariates p, is close to, or even greater than the number of samples n. In this scenario, problems such as over-fitting on the training dataset might be pronounced, since a large number of covariates will be used in both the first step of fitting the model and in the second step of finding similarity variables. Therefore, under the high dimensional setting, an effective variable selection method for both the first and the second step of the algorithm is needed.

In Chapter 4, the computational time for the similarity-based random survival forest can be large as the algorithm is building a random survival forest for each test data point, and there are a number of tuning parameters to choose from. This time complexity may be reduced by determining the tuning parameters from a subset of the entire population, however, the robustness of this approach requires further investigation. Using a hard threshold in the sampling may also reduce computation time. In addition, similarity-based approach on other types of models instead of the random forest model may be considered. One interesting area would be joint modeling of longitudinal covariates and survival outcome, where the longitudinal information for the subjects can be utilized to define similarity and improve survival prediction.

In Chapter 5, predicting a continuous outcome at a specific future time point is considered. Predictions can also be made from a Bayesian point of view [60, 10, 6], utilizing the predictive density that follows from the posterior distribution. Although inference is

not focused upon in this chapter, there are many potential possibilities. For example, one might be able to specify how confident a prediction is, or construct prediction intervals, and then choose when to make a prediction based on that. This problem can possibly be set up as an optimum stopping problem [48] as well. In addition, there are many cases, in biomedical studies or other domains, where predicting a future trend, or a time-to-event outcome is of interest. An interesting future area of research is developing methods that finds an optimum monitoring window for these different types of outcome. For predicting future trend, it would be helpful if the prediction can be dynamically updated as more information becomes available while still keeping the monitoring window as short as possible. In addition, similarity-based approach in Chapter 3 and 4 can be extended to the early prediction problem in Chapter 5 as well. By utilizing similar cases' information, one might have a better idea of how early a prediction for a specific case can be made.

References

[1] https://www.kidney.org/atoz/content/acutekidneyinjury.

[2] Kjell Benson and Arthur J Hartz. A comparison of observational studies and randomized, controlled trials. *New England Journal of Medicine*, 342(25):1878–1886, 2000.

[3] Joseph Berkson and Robert P Gage. Survival curve for cancer patients following treatment. *Journal of the American Statistical Association*, 47(259):501–515, 1952.

[4] Paul Blanche, Cécile Proust-Lima, Lucie Loubère, Claudine Berr, Jean-François Dartigues, and Hélène Jacqmin-Gadda. Quantifying and comparing dynamic predictive accuracy of joint models for longitudinal marker and time-to-event in presence of censoring and competing risks. *Biometrics*, 71(1):102–113, 2015.

[5] John W Boag. Maximum likelihood estimates of the proportion of patients cured by cancer therapy. *Journal of the Royal Statistical Society. Series B (Methodological)*, 11(1):15–53, 1949.

[6] George EP Box and George C Tiao. *Bayesian inference in statistical analysis*, volume 40. John Wiley & Sons, 2011.

[7] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[8] John Concato, Nirav Shah, and Ralph I Horwitz. Randomized, controlled trials, observational studies, and the hierarchy of research designs. *New England Journal of Medicine*, 342(25):1887–1892, 2000.

[9] National Research Council et al. *New directions for understanding systemic risk: a report on a conference cosponsored by the Federal Reserve Bank of New York and the National Academy of Sciences*. National Academies Press, 2007.

[10] Carla Currin, Toby Mitchell, Max Morris, and Don Ylvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86(416):953–963, 1991.

[11] Philipp Dahm, Scott M Gilbert, Robert A Zlotecki, and Gordon H Guyatt. How to use an article about prognosis. *The Journal of Urology*, 183(4):1303–1308, 2010.

[12] Vern T Farewell. The use of mixture models for the analysis of survival data with long-term survivors. *Biometrics*, pages 1041–1046, 1982.

[13] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, 1(4), 2009.

[14] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[15] Jerome H Friedman. On bias, variance, 0/1loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1):55–77, 1997.

[16] Lior Fuchs, Joon Lee, Victor Novack, Yael Baumfeld, Daniel Scott, Leo Celi, Tal Mandelbaum, Michael Howell, and Daniel Talmor. Severity of acute kidney injury

and two-year outcomes in critically ill patients. *CHEST Journal*, 144(3):866–875, 2013.

[17] Assaf Gottlieb, Gideon Y Stein, Eytan Ruppin, Russ B Altman, and Roded Sharan. A method for inferring medical diagnoses from patient similarities. *BMC medicine*, 11(1):194, 2013.

[18] Bettina Grun and Friedrich Leisch. Flexmix version 2: finite mixtures with concomitant variables and varying and constant parameters, 2008.

[19] JL Haybittle. A two-parameter model for the survival curve of treated cancer patients. *Journal of the American Statistical Association*, 60(309):16–26, 1965.

[20] Patrick J Heagerty and Yingye Zheng. Survival model predictive accuracy and roc curves. *Biometrics*, 61(1):92–105, 2005.

[21] Torsten Hothorn, Kurt Hornik, and Achim Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics*, 15(3):651–674, 2006.

[22] Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. Random survival forests. *The Annals of Applied Statistics*, pages 841–860, 2008.

[23] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3, 2016.

[24] Anthony YC Kuk and Chen-Hsin Chen. A mixture model combining logistic regression with proportional hazards regression. *Biometrika*, 79(3):531–541, 1992.

[25] Joon Lee. Patient-specific predictive modeling using random forests: An observational study for the critically ill. *JMIR medical informatics*, 5(1), 2017.

[26] Joon Lee, David M Maslove, and Joel A Dubin. Personalized mortality prediction driven by electronic medical data and a patient similarity metric. *PloS ONE*, 10(5):e0127428, 2015.

[27] Joon Lee, Daniel J Scott, Mauricio Villarroel, Gari D Clifford, Mohammed Saeed, and Roger G Mark. Open-access mimic-ii database for intensive care research. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 8315–8318. IEEE, 2011.

[28] Friedrich Leisch. Flexmix: A general framework for finite mixture models and latent glass regression in r, 2004.

[29] Timothy M Lenton, Hermann Held, Elmar Kriegler, Jim W Hall, Wolfgang Lucht, Stefan Rahmstorf, and Hans Joachim Schellnhuber. Tipping elements in the earth's climate system. *Proceedings of the national Academy of Sciences*, 105(6):1786–1793, 2008.

[30] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[31] Feng Liu, Peng Du, Fangfei Weng, and Jun Qu. Use clustering to improve neural network in financial time series prediction. In *Third International Conference on Natural Computation (ICNC 2007)*, volume 2, pages 89–93. IEEE, 2007.

[32] DJ Lowsky, Y Ding, DKK Lee, CE McCulloch, LF Ross, JR Thistlethwaite, and SA Zenios. Ak-nearest neighbors survival probability prediction method. *Statistics in Medicine*, 32(12):2062–2069, 2013.

[33] Tal Mandelbaum, Joon Lee, Daniel J Scott, Roger G Mark, Atul Malhotra, Michael D Howell, and Daniel Talmor. Empirical relationships among oliguria, creatinine, mortality, and renal replacement therapy in the critically ill. *Intensive care medicine*, 39(3):414–419, 2013.

[34] Tal Mandelbaum, Daniel J Scott, Joon Lee, Roger G Mark, Atul Malhotra, Sushrut S Waikar, Michael D Howell, and Daniel Talmor. Outcome of critically ill patients with acute kidney injury using the akin criteria. *Critical Care Medicine*, 39(12), 2011.

[35] Annette M Molinaro, Sandrine Dudoit, and Mark J Van der Laan. Tree-based multivariate regression and density estimation with right-censored data. *Journal of Multivariate Analysis*, 90(1):154–177, 2004.

[36] Xiao-Xiao Niu and Ching Y Suen. A novel hybrid cnn–svm classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4):1318–1325, 2012.

[37] Maryam Panahiazar, Vahid Taslimitehrani, Naveen L Pereira, and Jyotishman Pathak. Using ehrs for heart failure therapy recommendation using multidimensional patient similarity analytics. In *IOS Press*, 2015.

[38] Margaret S Pepe, Holly Janes, and Christopher I Li. Net risk reclassification p values: valid or misleading? *Journal of the National Cancer Institute*, 106(4):dju041, 2014.

[39] Margaret Sullivan Pepe. An interpretation for the roc curve and inference using glm procedures. *Biometrics*, 56(2):352–359, 2000.

[40] Donald A Pierce, William H Stewart, and Kenneth J Kopecky. Distribution-free regression analysis of grouped survival data. *Biometrics*, pages 785–793, 1979.

[41] Dimitris Rizopoulos. Dynamic predictions and prospective accuracy in joint models for longitudinal and time-to-event data. *Biometrics*, 67(3):819–829, 2011.

[42] Dimitris Rizopoulos. *Joint models for longitudinal and time-to-event data: With applications in R*. CRC Press, 2012.

[43] Dimitris Rizopoulos, Magdalena Murawska, Eleni-Rosalina Andrinopoulou, Geert Molenberghs, Johanna JM Takkenberg, and Emmanuel Lesaffre. Dynamic predictions with time-dependent covariates in survival analysis using joint modeling and landmarking. *arXiv preprint arXiv:1306.6479*, 2013.

[44] David Ruppert, Matt P Wand, and Raymond J Carroll. Semiparametric regression. cambridge series in statistical and probabilistic mathematics 12. *Cambridge: Cambridge Univ. Press. Mathematical Reviews (MathSciNet): MR1998720*, 2003.

[45] Mohammed Saeed, Christine Lieu, Greg Raber, and Roger G Mark. Mimic ii: a massive temporal icu patient database to support research in intelligent patient monitoring. In *Computers in Cardiology, 2002*, pages 641–644. IEEE, 2002.

[46] Mohammed Saeed, Mauricio Villarroel, Andrew T Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H Kyaw, Benjamin Moody, and Roger G Mark. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Critical Care Medicine*, 39(5):952, 2011.

[47] Marten Scheffer, Jordi Bascompte, William A Brock, Victor Brovkin, Stephen R Carpenter, Vasilis Dakos, Hermann Held, Egbert H Van Nes, Max Rietkerk, and George Sugihara. Early-warning signals for critical transitions. *Nature*, 461(7260):53, 2009.

[48] Albert N Shiryaev. *Optimal stopping rules*, volume 8. Springer Science & Business Media, 2007.

[49] Ewout Steyerberg. *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*. Springer Science & Business Media, 2008.

[50] Jimeng Sun, Daby Sow, Jianying Hu, and Shahram Ebadollahi. Localized supervised metric learning on temporal physiological data. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 4149–4152. IEEE, 2010.

[51] Jeremy MG Taylor, Yongseok Park, Donna P Ankerst, Cecile Proust-Lima, Scott Williams, Larry Kestin, Kyoungwha Bae, Tom Pickles, and Howard Sandler. Real-time individual predictions of prostate cancer recurrence using joint models. *Biometrics*, 69(1):206–213, 2013.

[52] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[53] Shubhendu Trivedi, Zachary A Pardos, and Neil T Heffernan. The utility of clustering in prediction tasks. *arXiv preprint arXiv:1509.06163*, 2015.

[54] AD Tsodikov, JG Ibrahim, and AY Yakovlev. Estimating cure rates from survival data: an alternative to two-component mixture models. *Journal of the American Statistical Association*, 98(464):1063–1078, 2003.

[55] Hans C Van Houwelingen. Dynamic prediction by landmarking in event history analysis. *Scandinavian Journal of Statistics*, 34(1):70–85, 2007.

[56] Ning Wang, Qiaoling Zhang, Liejun Yang, and Mingming Chen. *A Novel E-commerce Recommendation Algorithm based on Neural Network and Collaborative Analysis*, 2016.

[57] Ruo Xu, Dan Nettleton, and Daniel J Nordman. Case-specific random forests. *Journal of Computational and Graphical Statistics*, 25(1):49–65, 2016.

[58] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[59] Seniha Esen Yuksel, Joseph N Wilson, and Paul D Gader. Twenty years of mixture of experts. *IEEE transactions on neural networks and learning systems*, 23(8):1177–1193, 2012.

[60] Arnold Zellner and V Karuppan Chetty. Prediction and decision problems in regression models from the bayesian point of view. *Journal of the American Statistical Association*, 60(310):608–616, 1965.

*

APPENDIX

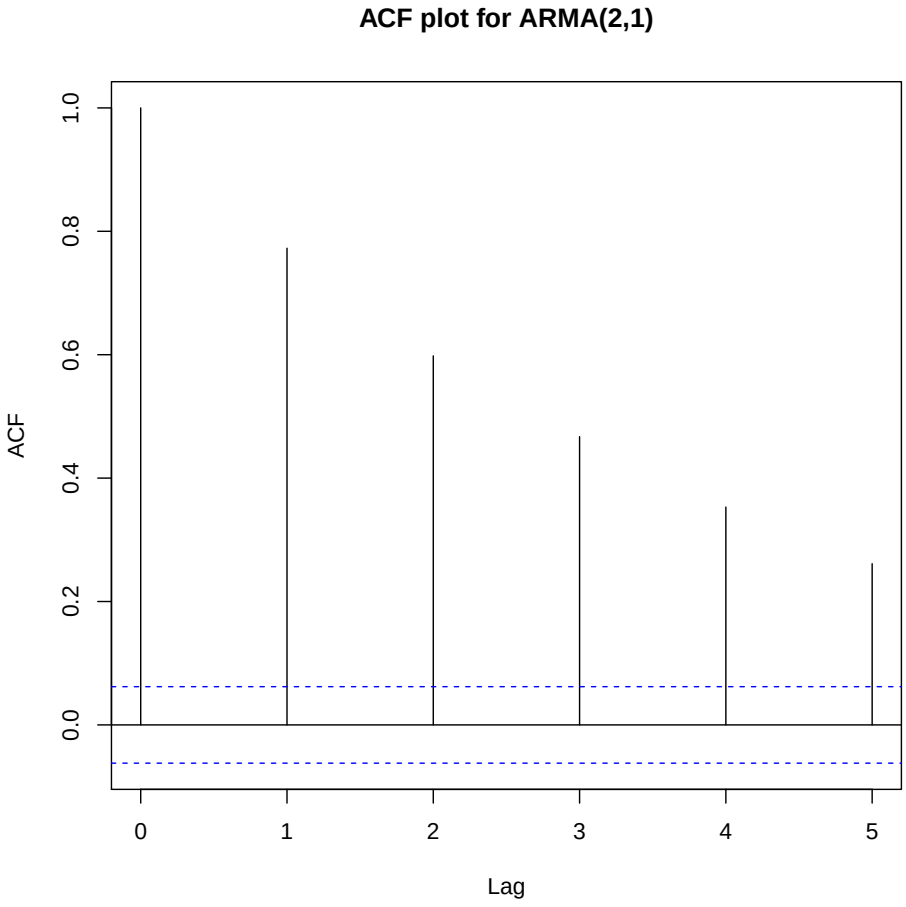


Figure A.1: ACF plot for ARMA (2,1)

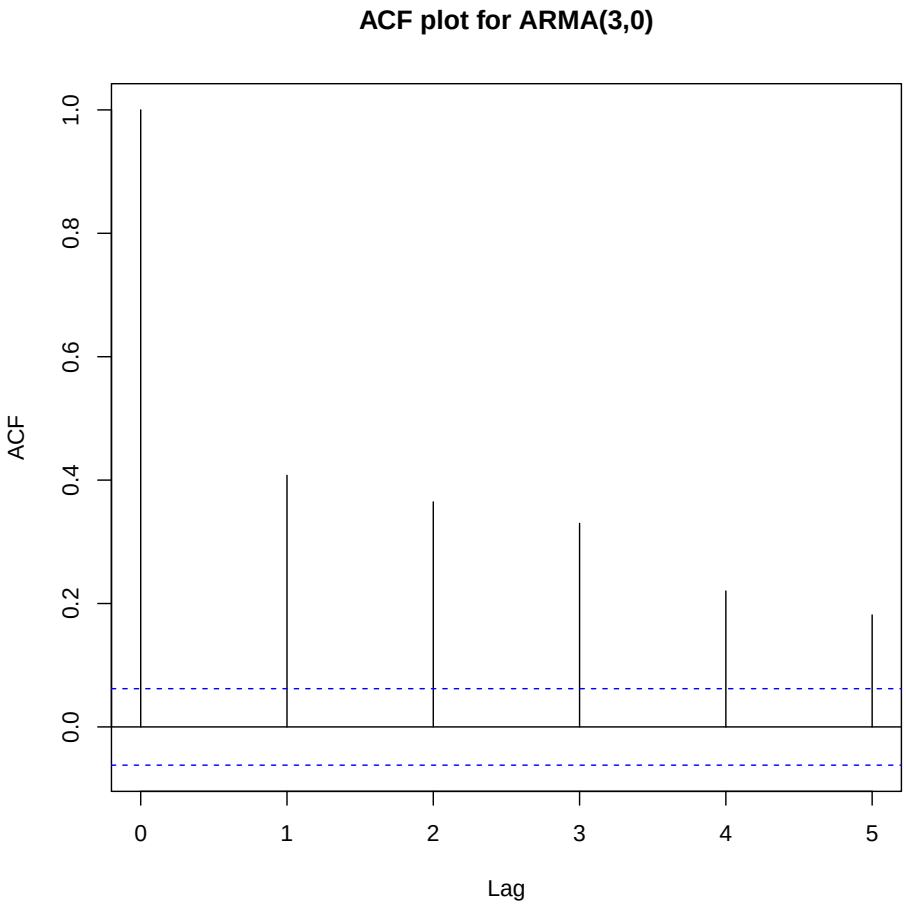


Figure A.2: ACF plot for an ARMA (3,0)

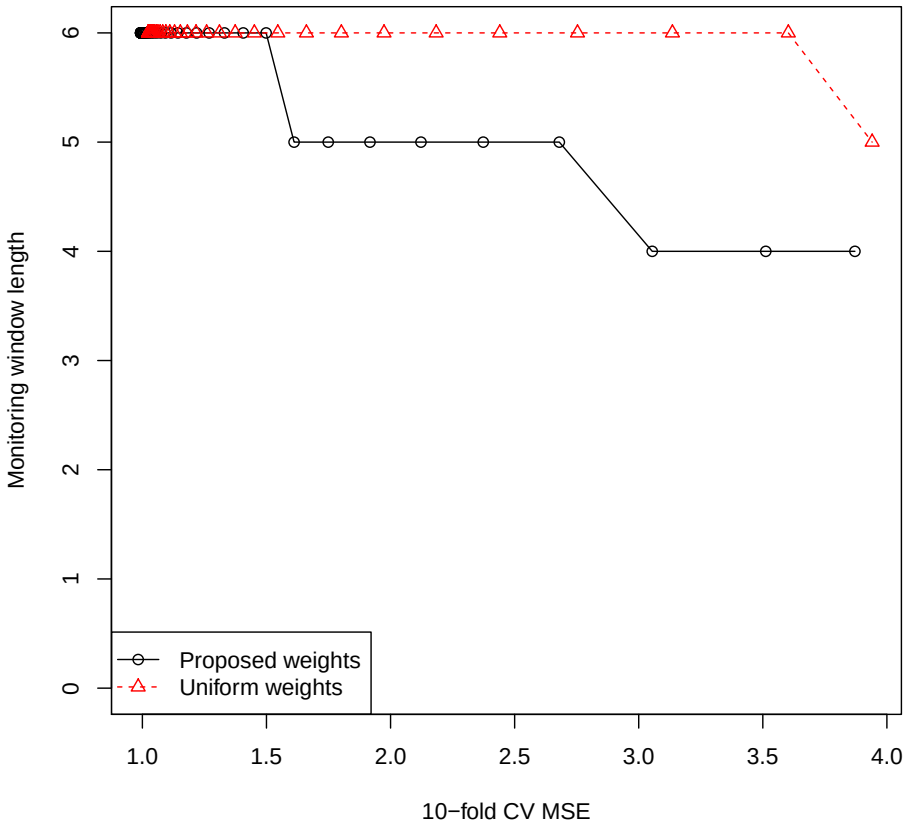


Figure A.3: Scenario 1 with covariates following ARMA (3,0): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

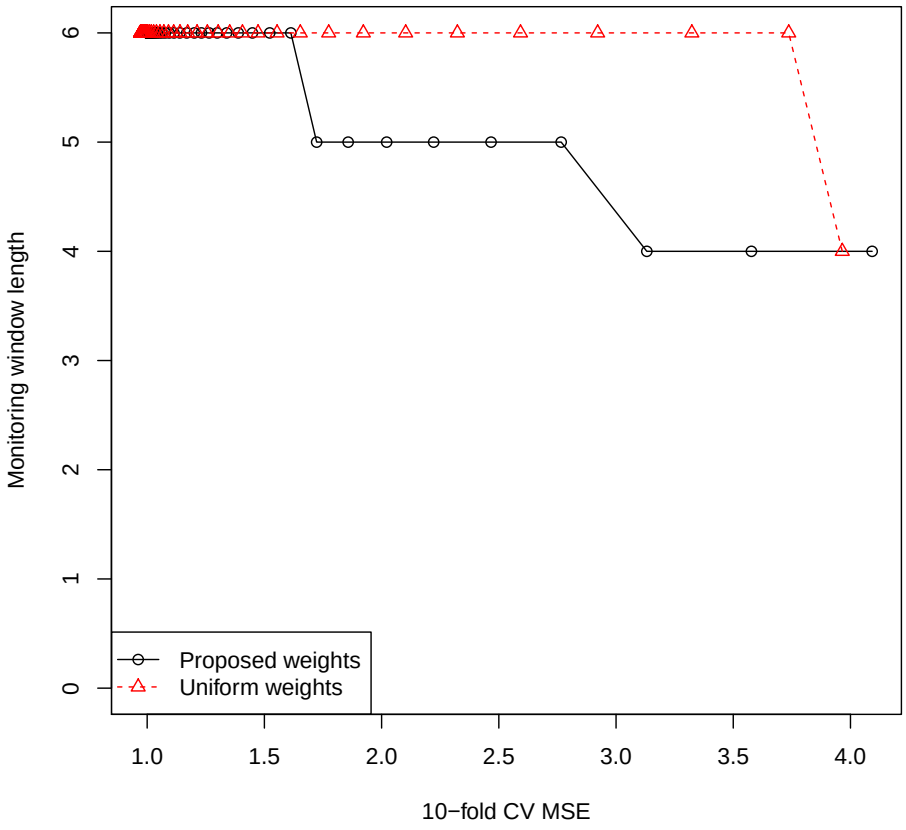


Figure A.4: Scenario 2 with covariates following ARMA (3,0): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

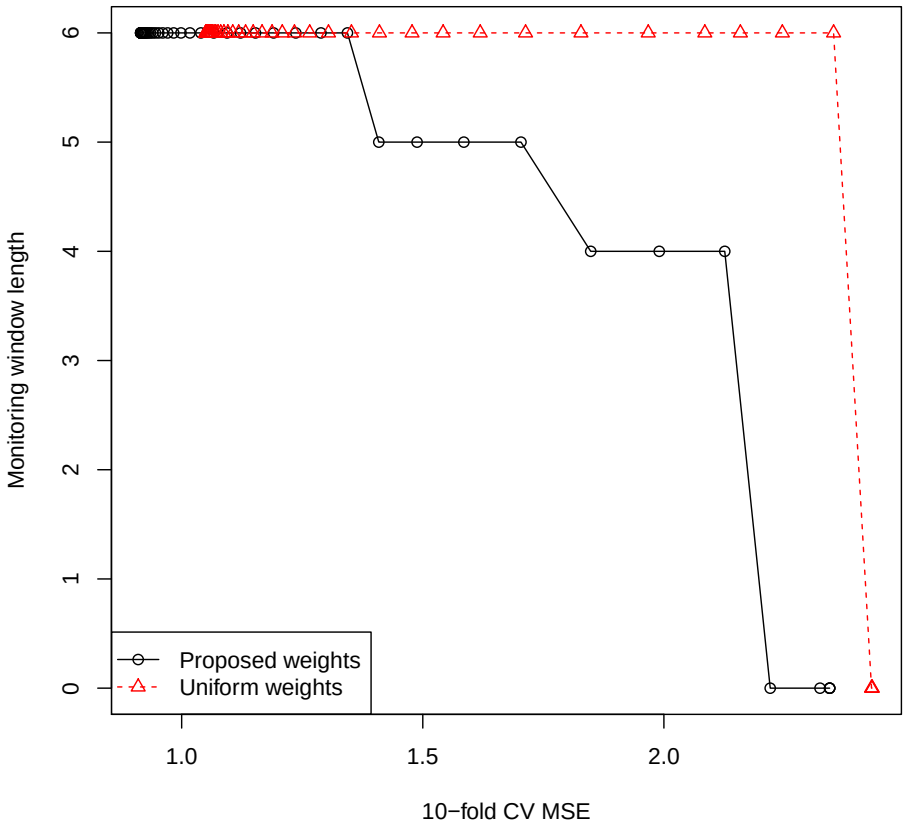


Figure A.5: Scenario 3 with covariates following ARMA (3,0): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

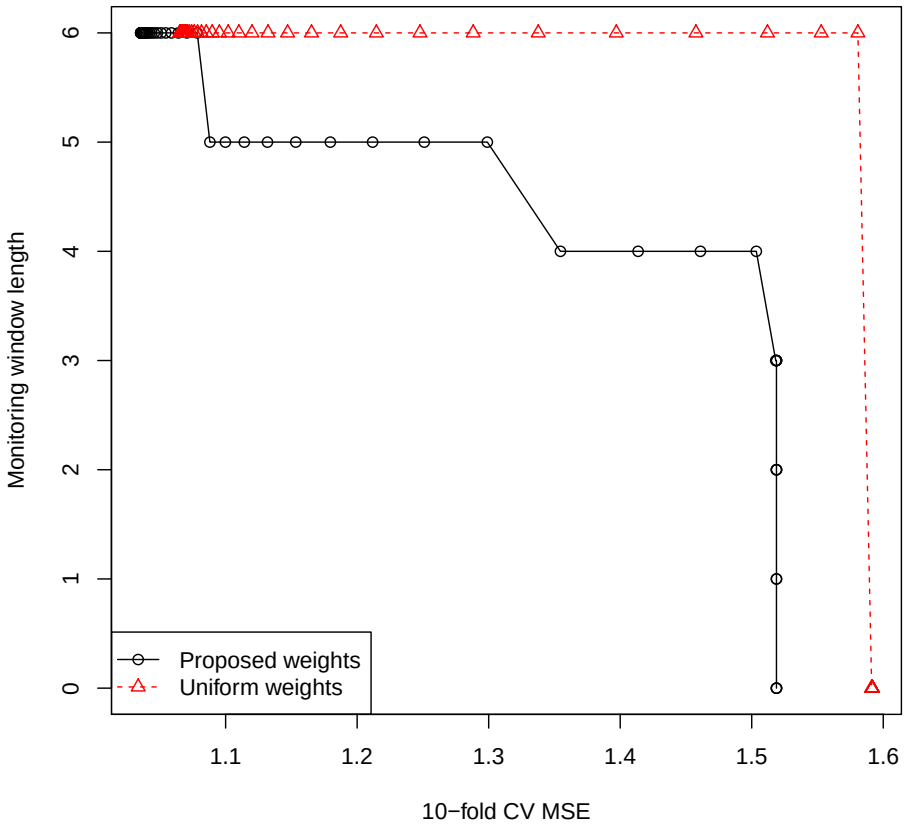


Figure A.6: Scenario 4 with covariates following ARMA (3,0): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

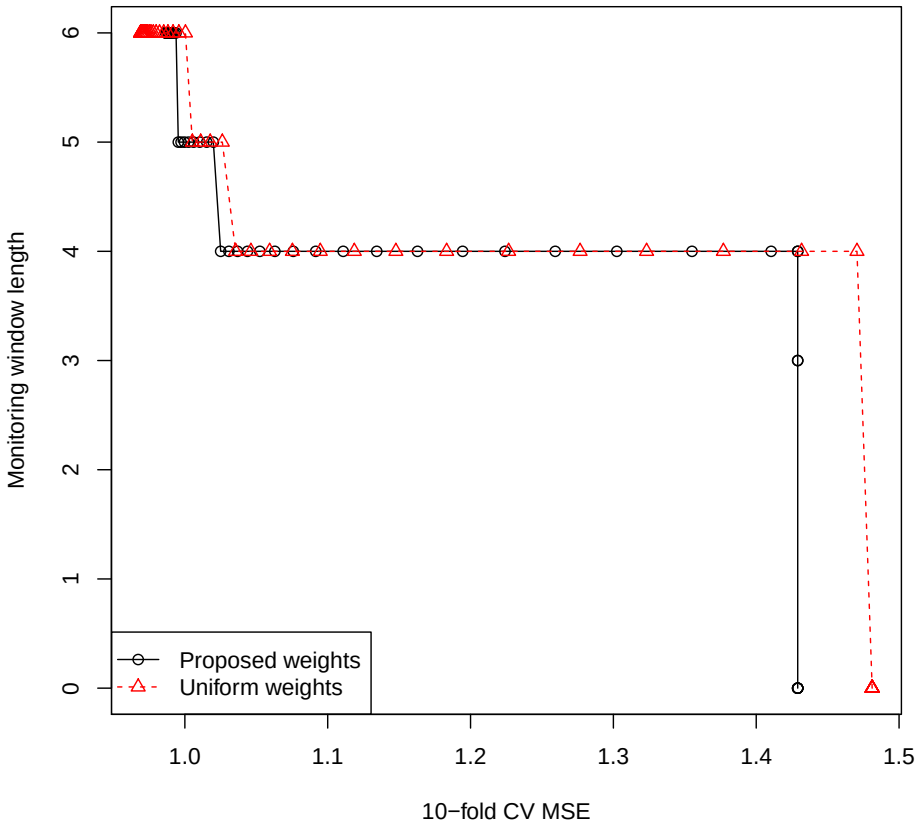


Figure A.7: Scenario 5 with covariates following ARMA (3,0): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

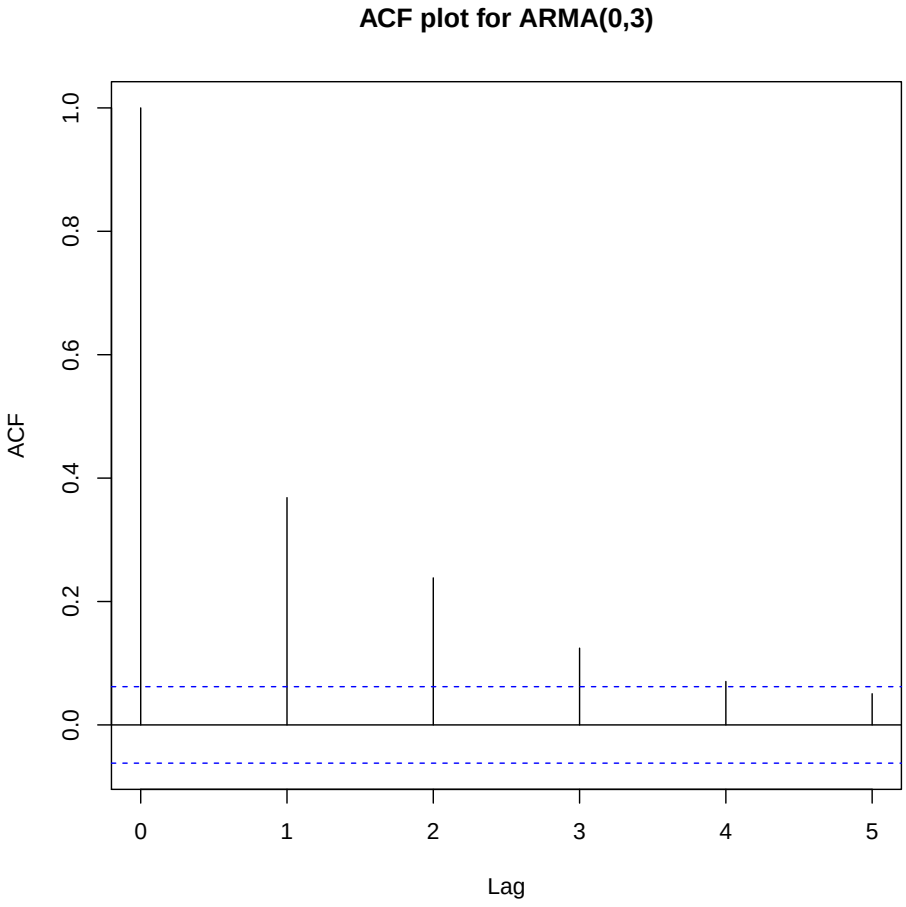


Figure A.8: ACF plot for ARMA (0,3)

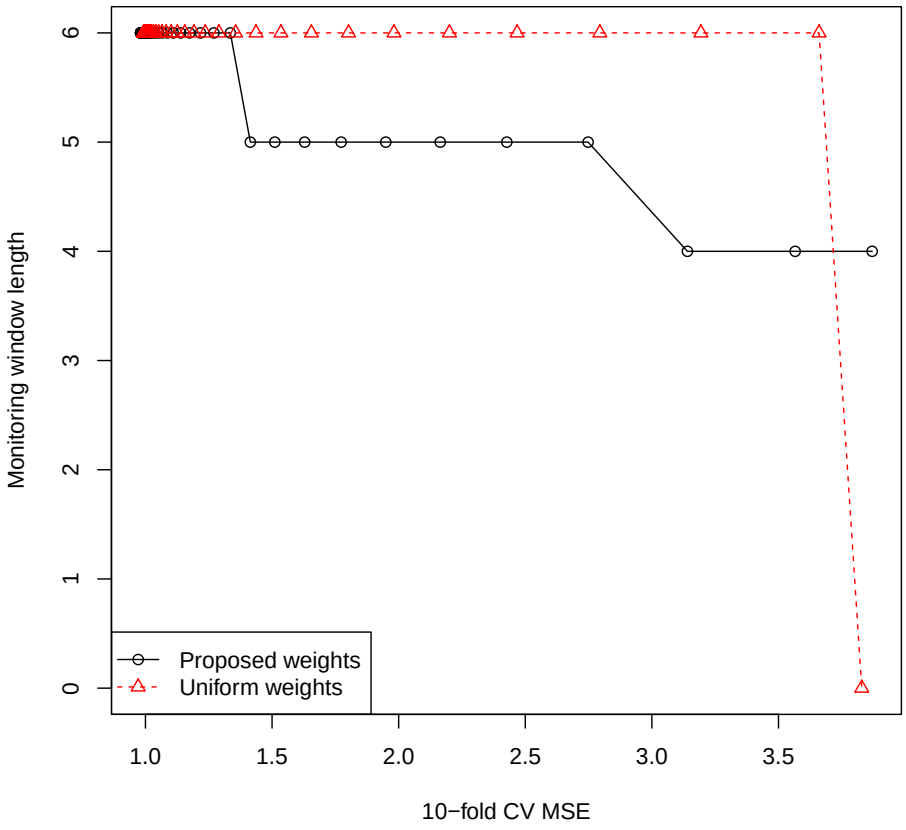


Figure A.9: Scenario 1 with covariates following ARMA (0,3): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

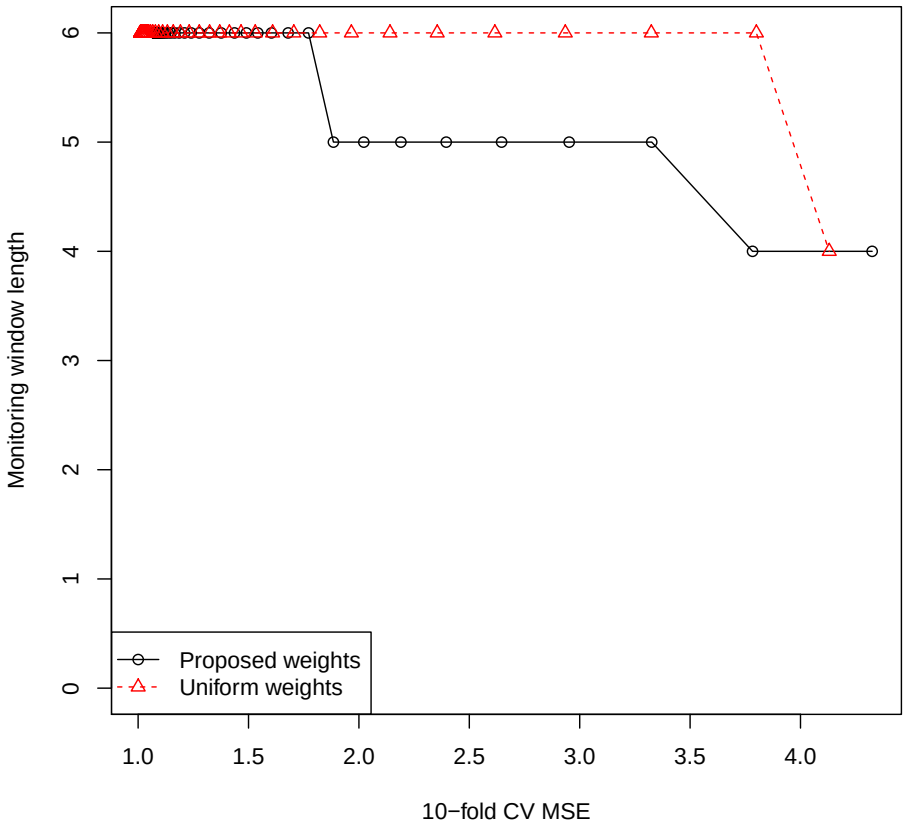


Figure A.10: Scenario 2 with covariates following ARMA (0,3): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

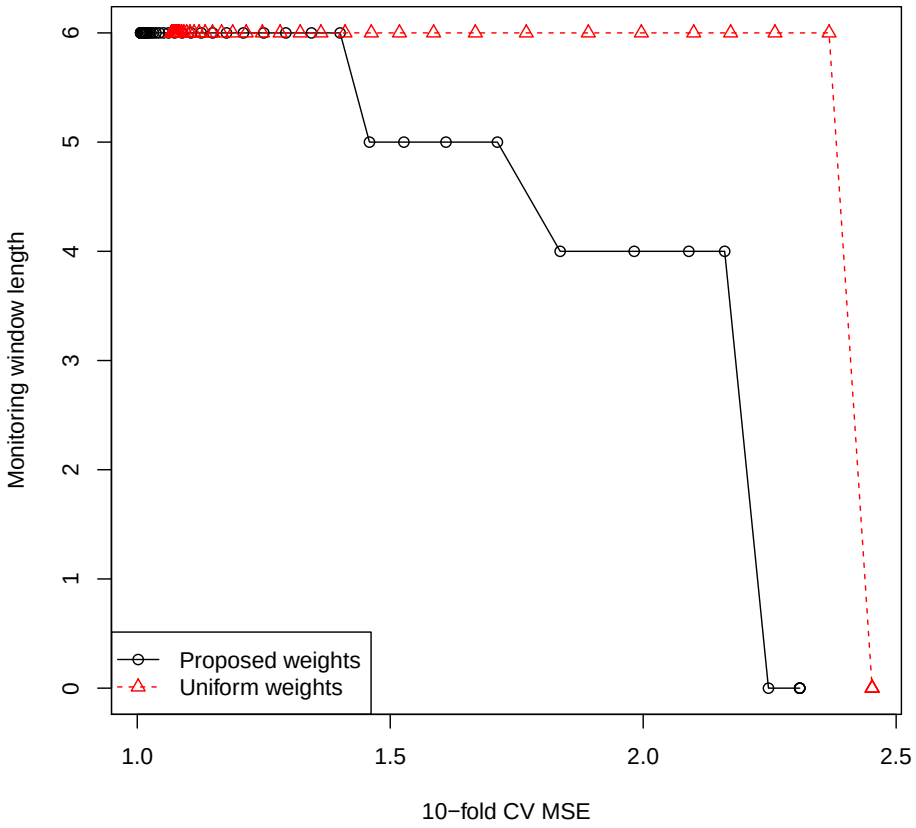


Figure A.11: Scenario 3 with covariates following ARMA (0,3): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

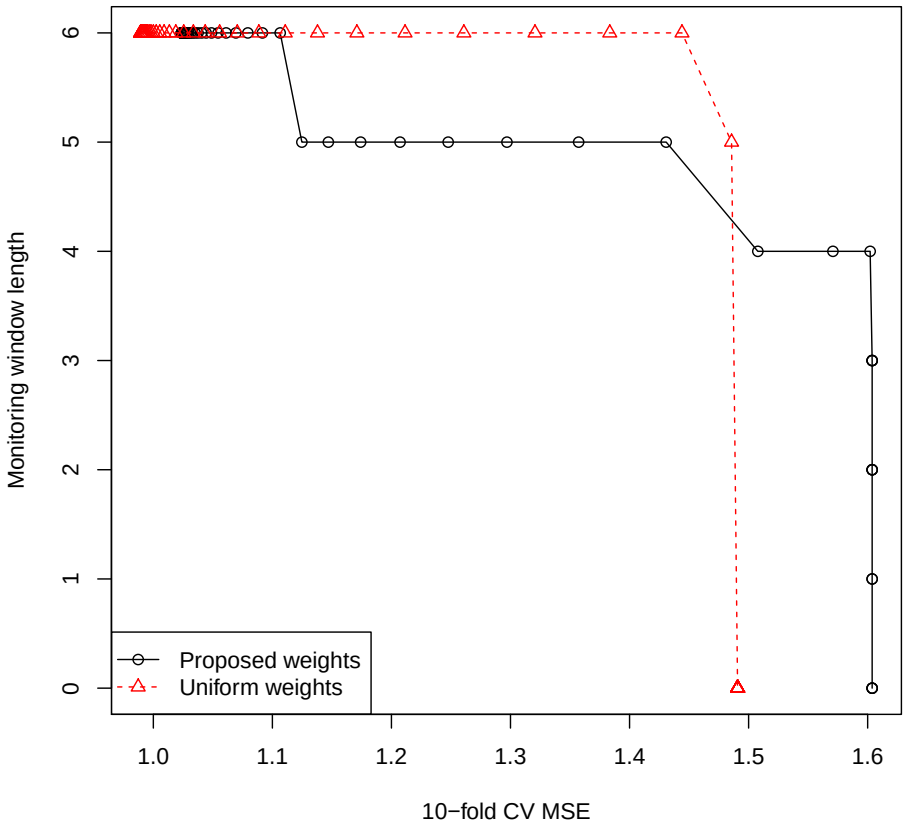


Figure A.12: Scenario 4 with covariates following ARMA (0,3): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies

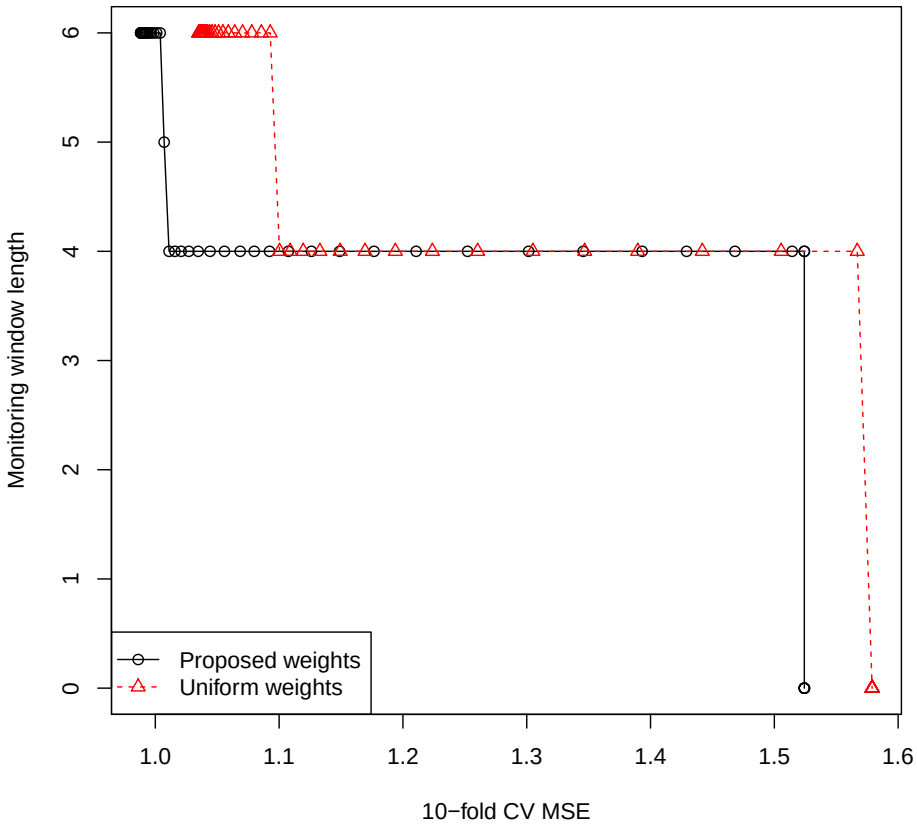


Figure A.13: Scenario 5 with covariates following ARMA (0,3): Shows the change in 10-fold CV MSE when the length of the monitoring windows varies