4-26-2021

Variable Selection and Subsequent Case Prediction in Semi-parametric Models

Jiasong Duan

VARIABLE SELECTION AND SUBSEQUENT CASE PREDICTION IN SEMI-PARAMETRIC MODELS

by

Jiasong Duan

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Biostatistics

The University of Memphis

May 2021

ACKNOWLEDGMENTS

I am forever grateful to my major professor, Dr. Zhang, who gives me the precious opportunity, continuous help and great patience to finish my thesis project. From analyzing data to writing the thesis, this project would not have been possible without her persistent support and nurturing.

I am fully thankful for my committee members Dr. Jiang and Dr. Ray for their time of review, valuable suggestions, and encouragement on this project.

I would like to express my deepest appreciation to my family, girl friend, and friends. They give me great love and endless support. They keep me going on and never let me down.

Abstract

Prediction of health status is a novel technique of forecasting the future health conditions with existing knowledge and available data. A reliable statistical model can lead to high performance of health status prediction. Built upon a semi-parametric variable selection approach, an algorithm to predict health conditions is developed and assessed. This algorithm is compared with three competing prediction methods based on logistic regressions, random forest, and support vector machines. Four statistics, accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC), are used to compare the performance across different approaches. The proposed approach, based on the simulation findings, does not perform as expected with respect to its ability to handle complex joint effects. The methods are then implemented and compared on the prediction of asthma status at the age of 10 years based on variables selected from 45 candidate variables available in the Isle of Wight birth cohort.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Health prediction is an emerging and developing field of forecasting the future health conditions. It constructs a system to inform the possible forward health situations so that the demands for health services can be addressed for health management [1]. For individuals, health prediction informs the risk of having certain health conditions in the future, and calls specific precautions to lower the risk. Before the onesite of a probable disease, a reliable prediction gives human beings chances to avoid or delay the speed of health deterioration, and reduce the cost of possible medical bills. Hence, forecasting health has the potential of leading to a improved life quality for mankind. For health service providers, predicting health enables them to improve the health prevention, health service delivery, and health surveillance [1]. In this level, health prediction is beneficial for monitoring public health, and it allows the limited health sources to be distributed more efficiently. A number of previous explorations have applied the methods of health prediction to practical health events for diverse purposes, such as a practice for elders to enhance the allocation of health service provision [2], an application for congenital anomalies with aim to provide assistances for gestational women and physicians [3], and an experiment for obesity prediction to reveal genetic factors' influence on fitness [4].

Basically, health prediction requires a predictive model built upon the comprehensive understanding of the interested health event, adequate usable data, and an analytical tool. A number of methods have been proposed to predict health conditions [1]. In addition to regular regression-based approaches, e.g., logistic regressions, supervised machine learning method is widely used in predicting disease. The machine learning algorithms integrate different techniques

from statistics, probability, and optimization to learn the nature and discover the pattern of an event from a given dataset, then the learned information is used to predict case status for a homogeneous group [5, 6]. Among various machine learning approaches, random forest and support vector machine are commonly used. The logistic regression method relies on the regression relationship between independent variables and dependent variable, which is in a classical statistical manner. Rather than primarily using statistical skills, the random forest and support vector machine adopt inductive methods to make prediction. The prediction performance of the three approaches have widespreadly been tested and validated [5, 7, 8].

Built upon a semi-parametric variable selection approach, an algorithm to predict health conditions is developed and assessed. According to a report, accuracy is a limit in conducting health prediction. Furthermore, the strength of association between the risk factors and the outcome, and the potential interactions between predictors can be principal reasons restricting the prediction accuracy [9]. In fact, health is determined by a number of factors [10], and the differing risks may have synergy or antagonism effects on the disease development [11]. Moreover, an addition of noninformative variables can cause various problems while studying the relationship of the factors with a target disease, including potential collinearity in the variables, inflated variance of prediction, and reduced generalization of the tested model [12, 13, 14]. Therefore, selecting the informative features is essential in order to deal with those problems. Besides, variable selection can also enhance the prediction performance, produce more efficient model, reduce cost of data arrangement, and foster data understanding [14, 15]. The variable selection approach by Zhang et al [16] deployed a Bayesian framework to select important features in semi-parametric models. The advantage of this technique is the joint effects among the predictors are considered in the calculation, which increase the probability of selecting the important variables from the model when the response may be influenced by some complicated interactions between the predictors. Following this variable selection procedure, a subsequent prediction method applied into forecasting health status is proposed and discussed in this thesis.

In the main text, we start with an introduction to Bayesian inference, which is the basic

structure of the semi-parametric variable selection approach. The variable selection is briefly discussed in sequence. Then the algorithm for the proposed prediction method is presented. Other three predictive methods, logistic regression, random forest, and support vector machine are also briefly described. To assess the performance of prediction algorithm, simulation studies are executed. Under the context of subsequent health condition prediction, we focus on binary dependent variables. Additionally, an application to a real dataset is also conducted with focus on predicting asthma status. In the end, the testing results are summarized and discussed.

Chapter 2

Methods

2.1 Introduction to Bayesian inference

2.1.1 Bayesian framework

Bayesian inference is developed from the Bayes' theorem. Suppose we have two different random events $\boldsymbol{A}$ and $\boldsymbol{B}$, and we let $p(\cdot)$ denote the probability of an event. The Bayes' theorem can be expressed as $p(\boldsymbol{A}|\boldsymbol{B}) = \frac{p(\boldsymbol{A})p(\boldsymbol{B})}{p(\boldsymbol{B})}$, where $p(\boldsymbol{A}|\boldsymbol{B})$ is the conditional probability of event A given the event B, $p(\boldsymbol{A})p(\boldsymbol{B})$ is the joint probability of events A and B, and $p(\boldsymbol{B})$ is the probability of event B which is not equal to 0. According to the formula of conditional probability, $p(\boldsymbol{B}|\boldsymbol{A}) = \frac{p(\boldsymbol{A})p(\boldsymbol{B})}{p(\boldsymbol{A})}$, where $p(\boldsymbol{A})$ is the probability of event A, $p(\boldsymbol{B}|\boldsymbol{A})$ is the conditional probability of event A given event B. Thus, the $p(\boldsymbol{A}|\boldsymbol{B})$ can be rewritten as $p(\boldsymbol{A}|\boldsymbol{B}) = \frac{p(\boldsymbol{B}|\boldsymbol{A})p(\boldsymbol{A})}{p(\boldsymbol{B})}$.

In the context of Bayesian inference, we assume a data vector $\boldsymbol{y}$ is randomly sampled from a probability distribution with an unknown parameter $\boldsymbol{\theta}$. The likelihood function of $\boldsymbol{\theta}$ is $L(\boldsymbol{\theta};\boldsymbol{y}) = f(y;\theta) = \prod_{i=1}^{n} f(y_i|\theta)$, where $f(\boldsymbol{y_i}|\boldsymbol{\theta})$ represents the probability density function of $\boldsymbol{y_i}$ given the parameter $\boldsymbol{\theta}$, and $i = 1,\ldots,n$ denoting one random trial. The unknown parameter $\boldsymbol{\theta}$ is a random vector, so we assume its probability distribution is $p(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})$ while discovering its properties. $\pi(\boldsymbol{\theta})$ is noted as the prior distribution of $\boldsymbol{\theta}$. Based on the data vector $\boldsymbol{y}$, the posterior distribution of $\boldsymbol{\theta}$ can be inferred. Because $\boldsymbol{\theta}$ and $\boldsymbol{y}$ are both random, they can be incorporated into the above Bayes' theorem. That is, $p(\boldsymbol{\theta};\boldsymbol{y}) = \frac{p(y|\theta)p(\theta)}{p(y)} = \frac{p(y;\theta)p(\theta)}{p(y)}$. In the inference, the vector $\boldsymbol{y}$ is unchanging, so its effect on the parameter $\boldsymbol{\theta}$ is fixed. In other words, the denominator in the last equation, $p(\boldsymbol{y})$, is a constant within the range of $(0,1)$, which means $p(\boldsymbol{\theta};\boldsymbol{y})$ is proportional to

$p(\boldsymbol{y};\boldsymbol{\theta})p(\boldsymbol{\theta})$ with a specific ratio. Hence, the last equation can be updated as $p(\boldsymbol{\theta};\boldsymbol{y}) \propto p(y;\theta)p(\theta) = L(\theta;y)p(\theta)$. In this equation, $p(\boldsymbol{\theta};\boldsymbol{y})$ is the posterior probability of parameter $\boldsymbol{\theta}$ given data $\boldsymbol{y}$, and is the inferred information of the interested parameter $\boldsymbol{\theta}$. Thus, by observing and summarizing the posterior probability, the estimated properties of parameter $\boldsymbol{\theta}$ can be drawn.

2.1.2 Selecting prior information

One important feature of Bayesian statistics is the selection of prior probability for the parameter of interest. The prior information is the initially fixed statistics of the inferred parameter. It can be assigned by the explorers based on their knowledge and beliefs about the parameter. The freedom of choosing the prior probability gives flexibility of outcomes in Bayesian inference. Also, the cooperation of original understanding of the parameter and the observed data could make a vital influence on the inference. The combined two factors leads to synergy effect with an accurate prior distribution and antagonism effect with an unreliable prior distribution. A desirable Bayesian model meets the balance between the prior probability and the existing data. That is, the posterior distribution is not independent or only dependent on the prior distribution.

2.1.3 Estimating posterior probability

In most cases, posterior distribution needs to be estimated by simulations in order to avoid the sophisticated process of integral, which is feasible with the rapidly developed high-throughout computation technologies. There are ample methods that can be used to summarize the target distribution. One of the most frequently used method is MCMC (Markov chain Monte Carlo) simulations. MCMC simulates data from the target distribution and the data eventually converges to a stationary distribution in Markov chains. In general, a Markov chain is the sequence consisting of generated values from a specified probability distribution. Each value in the Markov chain has the Markov property, which is also called memorylessness. That is, every value in the sequence is only dependent on the previous one. Monte Carlo is the idea of simulating data

repeatedly to approximate the parameter by taking the average of calculated probabilities from the simulations. Metropolis Hastings algorithm is a classical MCMC method. Given a probability distribution, the Metropolis Hastings generator starts replicating data with a fixed pioneer parameter. Except for the Markov property, every value in the generated Markov chains must pass the defined level of a calculated acceptance probability, which is a judging measurement to decide whether the simulated value needed to be accepted or rejected. Each component in the Makov chain sequence is selected by the iterative generating procedures, the simulating process is then called MH update. When more than two parameters are needed to estimate, Gibbs sampling, an updated variant of MH algorithm, can be used to generate the data. By using the Gibbs sampler, the probability of one parameter is fully dependent on the rest parameters.

2.2 The semi-parametric variable selection approach

Let $\boldsymbol{Z}_{n\times 1}$ be a binary variable, e.g., status of health condition, with n denoting sample size, and $\boldsymbol{X}_{n\times p}$ a set of p independent variables such that their contribution to $\boldsymbol{Z}$ is of particular interest, but its association with $\boldsymbol{Z}$ is likely to be non-linear and unknown. In the context of our application, $\boldsymbol{X}$ represents the level of DNA methylation, a continuous measure. Furthermore, let $C_{n\times p_0}$ be a set of p_0 covariates. The relationship between $\boldsymbol{Z}$ and $\boldsymbol{X}$ with $\boldsymbol{C}_0$ as covariates can be described as

$$E(\boldsymbol{Z}) = \phi\{\boldsymbol{C}_0\boldsymbol{\beta} + h(\boldsymbol{X})\}, \tag{2.1}$$

where $\boldsymbol{\beta}$ describes the additive linear effects of $\boldsymbol{C}_0$ on $\boldsymbol{Z}$, and $h(\boldsymbol{X})$ is an unknown function evaluating the joint effects of $\boldsymbol{X}$ on $\boldsymbol{Z}$, which is possibly non-linear and may involve complex interactions between $\boldsymbol{X}$. Function $\phi(\cdot)$ is the cumulative standard normal distribution function and used to describe a probit regression through data augementation via Gaussian latent variables. Define a latent variable Y_i with $\boldsymbol{Y} = (Y_1, \cdots, Y_n)^T$ such that

$$Z_i \;=\; 1_{\{Y_i>0\}}, \;\; \boldsymbol{Y} = \boldsymbol{C}_0\boldsymbol{\beta} + \boldsymbol{h}(\boldsymbol{X}) + \boldsymbol{\varepsilon}, \tag{2.2}$$

where 1_A denotes the indicator function of the event A, and $\boldsymbol{\varepsilon}$ is a vector of random error with dimension $n \times 1$ and its distribution is assumed to be $\boldsymbol{\varepsilon} \sim N(0, \sigma^2 I)$ with I denoting the identity matrix.

In [16], the unknown function $h(\cdot)$ is represented using a kernel function $K(\cdot,\cdot)$. Following the Mercer's theorem [17, 18], any function $h(\cdot)$ in the function space $\mathscr{H}$ can be represented as a linear combination of reproducing kernels [18, 19], $h(\boldsymbol{X}_i) = \sum_{k=1}^{n} K(\boldsymbol{X}_i, \boldsymbol{X}_k)\alpha_k = \boldsymbol{K}_i'\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = (\alpha_k, k = 1, \cdots, n)'$ is a vector of unknown parameters and $\boldsymbol{K}_i'$ is the ith row of kernel matrix $\boldsymbol{K}$. In this thesis, $K(\boldsymbol{X}_i, \boldsymbol{X}_k)$ is taken as a Gaussian kernel with a defined smoothness parameter ρ. To select important variables from $\boldsymbol{X}$, [16] introduced a vector of indicator variables $\boldsymbol{\delta}$ into the kernel to indicate the inclusion/exclusion of a variable. Consequently, the (i, j)th entry of the kernel matrix becomes a function of two parameters, ρ and $\boldsymbol{\delta}$, $K(\rho, \boldsymbol{\delta})$, defined as, $K_{i,j}(\rho, \delta) = exp\left\{ -\sum_m \left\| \delta_m (X_{im} - X_{jm}) \right\|^2 \Big/ \rho \right\}$, where δ_m denotes the inclusion/exclusion of variable m.

Under this setting, as shown in an earlier study [20], $\boldsymbol{h}(\boldsymbol{X})$ can be treated as random effects with $\boldsymbol{h} \sim N(\boldsymbol{0}, \tau \boldsymbol{K}(\rho_0, \boldsymbol{\delta}))$, where τ is an unknown variance component. The probability of $\boldsymbol{Z} = \boldsymbol{z}$ is then described as

$$\begin{aligned} Pr(\boldsymbol{Z} = \boldsymbol{z} | \beta, \tau, \sigma^2) &= \int_{\boldsymbol{A}(\boldsymbol{Z})} \left(\frac{1}{\sigma^2 + \tau}\right)^{n/2} |\Sigma_0(\rho_0, \boldsymbol{\delta})|^{-1/2} \\ &\quad \times \exp\left\{ -\frac{(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})' \Sigma_0(\rho_0, \boldsymbol{\delta})^{-1} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})}{2(\sigma^2 + \tau)} \right\} d\boldsymbol{Y}, \end{aligned} \tag{2.3}$$

where $\boldsymbol{A}(\boldsymbol{Z}) = \{A(Z_1), \cdots, A(Z_n)\}$ with

$$A(Z_i) = \begin{cases} (-\infty, 0], & \text{if } Z_i = 0, \\ (0, \infty), & \text{if } Z_i = 1, \end{cases}$$

and $\Sigma_0(\rho_0, \boldsymbol{\delta})$ is an $n \times n$ matrix with (i, j)-th entry being $\tau/(\tau + \sigma^2) k_{ij}(\rho_0, \boldsymbol{\delta})$ when $i \neq j$, and 1 when $i = j$. To avoid the problem of unidentifiability, Zhang et al[16] fixed σ^2 and τ at σ_0^2=0.2 and τ_0=0.8.

Under the above settings, the likelihood of $\boldsymbol{\beta}, \boldsymbol{\delta}$ is then given as

$$\begin{aligned} L(\boldsymbol{\beta}, \boldsymbol{\delta}|\boldsymbol{Z}) &\propto p(\boldsymbol{Z}|\boldsymbol{\beta}, \boldsymbol{\delta}) = \int p(\boldsymbol{Z}, \boldsymbol{Y}|\boldsymbol{\beta}, \boldsymbol{\delta}) d\boldsymbol{Y} \\ &\propto \int_{\boldsymbol{A}(\boldsymbol{Z})} |\Sigma_0|^{-1/2} \exp\left\{-(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})' \{\Sigma_0(\rho_0, \boldsymbol{\delta})\}^{-1} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})/4\right\} d\boldsymbol{Y}. \end{aligned} \quad (2.4)$$

The parameters including $\boldsymbol{\beta}$ (regression coefficients) and $\boldsymbol{\delta}$ (vector of indicators for variable inclusion/exclusion) are estimated using a fully Bayesian approach. Details for parameter inferences are in [16]. In the following, we focus on the algorithm for prediction.

2.3 Algorithm for prediction via the semi-parametric approach

We use training data to build prediction models. Assuming the sample size of a training data is n_1 and the sample size for a new data set is n_0. The relationship between the dependent variable and independent variables is learned using training data from n_1 subjects. Based on equation 2.2, the inferred parameters β and h describe this relationship. The testing data is used to assess the quality of prediction based on the inferred parameters. The steps of a prediction process are outlined as the following, in which we focus on the stage of prediction.

Assume $\boldsymbol{p_s}$ variables are selected from the training data, based on which we predict the status for each of the n_0 subjects in the new data. Two situations are consider, with and without covariates $\boldsymbol{C}_0$ of size p_0.

Step a. Estimate effects of covariates. Regression coefficients of the p_0 covariates, $\boldsymbol{\beta}$, are extracted from the converged MCMC sequences using the training data. After burn-in iterations, samples of $\boldsymbol{\beta}$ are kept and used to estimate $\boldsymbol{\beta}$. In this thesis, $\boldsymbol{\beta}$ is estimated using the median. If no covariates are needed for computation, $p_0 = 0$.

Step b. Estimate effects of important variables. A kernel matrix $\boldsymbol{K}_0$ is used to assess the distance of the $\boldsymbol{p_s}$ variables. The (i, j)th entry of an updated kernel matrix $\boldsymbol{K}_0$ for a variable m can be written as $K_{i,j}(\rho, \delta) = exp\left\{-\sum_m \left\|\delta_m(X_{im} - X_{jm})\right\|^2 \Big/ \rho\right\}$. We randomly generate a vector $\boldsymbol{\alpha_0}$ with length of 100 from the multivariate normal distribution $N(\boldsymbol{0}, \tau \boldsymbol{K}^{-1}(\rho_0, \boldsymbol{\delta}))$, where τ is still

the same value as that chosen in the process of variable selections. For a new data point X_0 in the new data with measures on the selected $\boldsymbol{p_s}$ variables, we calculate its distance to the samples in the training data using kernel matrix $\boldsymbol{K}(X_0, X)$, where X denotes the measures of $\boldsymbol{p_s}$ variables for n_1 subjects. In consequence, by utilizing $\boldsymbol{\alpha_0}$ and $\boldsymbol{K}(X_0, X)$, the predicted value for a subject with data X_0 can be estimated as $g(\hat{\boldsymbol{Y}}_{X_0}) = C_0\hat{\boldsymbol{\beta}} + \boldsymbol{K}'(X_0, X)\boldsymbol{\alpha}_0$.

Step c. Decide the status for subjects. Recall that 100 $\alpha_0's$ are estimated in step b, which indicates that in total 100 values of $g(\hat{\boldsymbol{Y}}_{X_0})$ are calculated. In this case, if a value is larger than 0, we assign the status for the corresponding subject to 1, otherwise 0. This will give us in total 100 estimated statuses for a given subject. The final status of the subject is based on proportion of allocations between 1 and 0. As a result, we predict the case as 1 if more than 50 of the estimated values of $\hat{\boldsymbol{Y}}_{X_0}$ are 1, and 0 otherwise.

2.4 Competing methods

To demonstrate and assess the proposed algorithm with respect to the quality of prediction, we consider the following three competing methods. The first approach is to do prediction based on logistic regressions, i.e., $\log(Z_i/(1-Z_i)) = \boldsymbol{C}_{0i}^T\boldsymbol{\beta} + \boldsymbol{X}_i^T\boldsymbol{\Gamma}$. A prediction of $\boldsymbol{Y}$ by this method is based on the estimates of $\boldsymbol{\beta}$ and $\boldsymbol{\Gamma}$ from a training data set. The $\boldsymbol{X}$ variables in the logistic regression models are the same as those in the proposed algorithm built upon reproducing kernels with a feature of variable selections.

The second approach is built upon the random forest algorithm [21] built upon classification trees. It is a classification mechanism by concluding the pattern of response variable with respect to the properties of explanatory variables. Instead of using all predictors and all samples, every classification tree randomly picks part predictors and utilizes training and testing samples aiming to avoid overfitting. After congregating all decisions from the trees, the final prediction is made using the strategy of majority vote.

The last competing method is the support vector machine technique. It conducts classifications by applying a support vector classifier to the transformed observations via kernel

functions [22]. To deal with high dimensional data which might be in non-linear form, kernel functions trigger the support vector classifier by mapping the observations to required form to complete classifications. The support vector classifier distinguishes an observation by judging which side of a built hyperplane it lies on. A hyperplane is a $p-1$ flat affine subspace of a feature space with dimension of p, which tries to maximize the margin between itself and training observations, while the observations have the maximum margin to the hyperplane which are hence called support vectors. Dependent on the training data, a hyperplane that separates the observations to two classes is constructed, based on which the of each object in the testing data is determined. In practice, to keep away from overfitting, the hyperplane aims to classify most training observations correctly with a small misclassification rate.

2.5 Numerical assessment of the prediction algorithm

Through intensive simulations, we assess the proposed prediction method and compare it with commonly used approaches.

2.5.1 Simulation study

Simulation scenarios

Each simulated data are generated based on 12 candidate explanatory variables $\boldsymbol{X}$ with each randomly generated from uniform distributions bounded between 0.0001 and $12/(2m)$, $U(0.0001, 12\Big/(2m))$, where $m = 1, \ldots, 12$. One covariate variable as an adjusting factor is also included in the model drawn from a normal distribution with mean 0 and variance of 4, $N(0,4)$. Random error is from a normal distribution $N(0, 1.9^2)$. Two underlying models are considered:

$$Linear: E(Y_i|C_{0i}, g_i) = f^{-1}\left\{C_{0i} + ag_{i1} + bg_{i2} + cg_{i3}\right\}.$$

$$Non-linear: E(Y_i|C_{0i}, g_i) = f^{-1}\left\{C_{0i} + d\cos(g_{i1} \times g_{i2}) + eg_{i3}\right\}.$$

In the above, C_0 is the covariate variable, g_{i1}, g_{i2}, g_{i3} are measures of three important independent variables, and a, b, c, d, e are coefficients denoting the effects of informative variables.

To assess the variation of prediction due to sample sizes and prevalence, two different sample sizes are used to simulate data, $n = 300$, 600, and three distinct prevalence levels are considered for both models, 31%, 50%, 85% for linear model and 40%, 53%, 64% for non-linear model with corresponding coefficients, $\{a, \cdots, e\}$=$\{-1, -1.5, 2.5, 3, -2\}$, $\{0.5, -2, 1.5, 3, 0.2\}$, and $\{3, -2, 3.5, 3, 2\}$, respectively. For each combination of prevalence level and sample size, 50 Monte Carlo (MC) replicates are generated. These settings plus the two different models give in total 12 scenarios in data simulations. For convenience, we denote this set of scenarios as "generation simulations".

Follow the above setting, the covariate and independent variables are created as continuous. To test the generalization of the predictive methods, various combinations of predictors consisting of both binary variables and continuous variables are considered. To examine the performance of the method under these considerations, we take sample size of 300 and use the same coefficients as in "general simulations" to simulate data. Four situations are considered: 1) binary covariate; 2) no covariate; 3) one of the independent variables being binary; and 4) larger variance in the distribution of random errors, i.e., $N(0, 3.8^2)$ instead of $N(0, 1.9^2)$. The four extended scenarios are designed to match the potential complexity of independent variables in practice. Like the definition of general simulations, we call these four simulation scenarios as derivative simulation, and by order, specifically call them as derivative simulations 1 to 4, respectively.

To assess the quality of prediction, we split each simulated data (MC replicate) into a training set and a testing set at a ratio of 4 : 1 with respect to sample size. The results are assessed by four statistics: accuracy, sensitivity, specificity, and area under the curve (AUC). The accuracy is the ability to discriminate positive and negative cases, which is defined as the fraction of the truly predicted positives and negatives over the number of predictions in total. Sensitivity refers to the extent of classifying positive cases correctly. It is calculated as the proportion of the truly predicted positives over underlying true positives. Specificity is the probability to correctly

identify true negative cases. It is estimated as the percentage of the truly predicted negatives over true underlying negatives. The statistic AUC is the area under a Receiver Operating Characteristic (ROC) curve. The first three measurements have been widely used to assess quality of prediction, AUC is applied extensively in machine learning area, and multiple studies have used and validated AUC's ability to evaluate the effectiveness of classification [23, 24, 25]. Larger values of AUC indicate higher prediction quality [26]. For each of the four statistics, median and a 95% empirical interval (95% EI) are recorded.

In the process of prediction, informative variables are selected using the approach discussed in Section 2.2 and these selected variables are then used to predict outcomes using the proposed approach as well as the competing methods to ensure fair comparisons among these methods. For each subject, predictions were performed 50 times aiming to diminish the influence of random sampling errors. All the programs for variable selections and prediction are written in R [27]. For random forest and support vector machine, the R packages are [28] and [29], respectively, and in our analyses, the default settings in the packages are implemented.

Results of general simulation

The results are assembled in Tables 1 and 2.

For data with linear associations, regardless of sample sizes, the proposed method normally gives high sensitivity and relatively low specificity with the prevalence of 31%, or high specificity and relatively low sensitivity with the prevalence of 85%. When the prevalence is 50%, comparable sensitivity and specificity are observed (Figure 1). In the two situations that the sensitivity and specificity are not commensurate with each other, the risk of higher false positive rates and false negative rates would arise because of the low specificity and sensitivity, respectively. In contrast, the patterns of sensitivity and specificity for the three competing approaches are all better than the proposed approach for both models and all settings.

For data with non-linear associations, the observed patterns from the linear models still exist. That is, the three competing methods all outperform the proposed methods, although the statistics

(sensitivities, specificities, accuracy, and AUC) overall all lower than those when the model was linear. We had expected that the proposed approach had the ability to handle non-linear associations and, at least, should have outperformed the method based on logistic regressions. Due to the inclusion of a linear association covariate in the model, it is possible that the prediction was dominated by the covariate rather than the important variables. The further assessment, denoted as ”derivative simulations”, has focus on this and findings of this assessment are discussed int he section below.

Results of derivative simulations

For derivative simulation 1, that is, we use a binary covariate rather than a continuous variable. To ease the simulation, we revised the prevalence levels to 24%, 57%, 92% and 36%, 63%, 80% for data from linear model and non-linear model, respectively. For both linear and non-linear models, the results of proposed method are in the contrary way as in general simulation (Table S1 and Figure S1). The proposed method shows high specificity and relatively low sensitivity with the prevalence of 24% or 36%, or high sensitivity and relatively low specificity with the prevalence of 80% in non-linear model. Furthermore, when the prevalence is in a medium level of 57% or 63%, a sharp distinction is noticed between a high sensitivity and a low specificity. Compared to the numerical measurements in general simulation, almost all statistics decrease in varying degrees. For the three competing methods, all the statistics also decrease and drop in larger extents than the values from proposed method. Especially when the data with prevalence of 92% or 80%, nearly half of total 50 models for each approach had a specificity value of 0, which means most models have no ability to detect the real negative cases. Under this setting, the proposed method overall performs better than the competing methods, especially for the situation of non-linear associations.

For derivative simulation 2 (i.e., covariates removed), the generated data from linear model and non-linear model have separate prevalence levels of 21%, 50%, 92% and 30%, 57%, 76%. Two noticeable patterns of the results are found in terms of the two types of data relationships

(Table S2 and Figure S2). For the data with linear association, the proposed method is still inferior compared to the competing approaches with overall low sensitivity and specificity, unless the prevalence is high. When the underlying model is non-linear, the proposed approach overall is more steady, compared to the competing approaches, but both sensitivity and specificity are low with low accuracy and AUC.

For derivative simulation 3 (i.e., the first independent variable being binary), the prevalence levels are 52%, 42%, 57% for data from linear model and 48%, 61%, 71% for data from non-linear model. The results from developed approach are in the same trend as in general simulation (Table S3 and Figure S3). One additional phenomenon is discovered. The evaluation values get improved predominantly for data with linear relationship, and get diminished for data with non-linear relationship. Unlike the derivative simulations 1 and 2, the distinction of each prevalence level between the current simulation scenario and general simulation is large. The distinctions could be the possible reason for the changes of assessment values. In addition, the three competitive approaches do not produce dissimilar results as in general simulation.

For derivative simulation 4 (i.e., larger variance in the distribution of random errors), the prevalence levels are 35%, 50%, 81% for data from linear model and 42%, 52%, 61% for data from non-linear model. For both developed method and competitive methods, their results have no perceptible variations compared to the results in general simulations (Table S4 and Figure S4).

2.5.2 Summary of simulation results

Built upon all the simulations, the proposed approach has stable performance across diverse situations, but in most cases it is inferior to the competing approaches. However, by design, reproducing kernel based approach has the potential to estimate complex associations, and thus is expected to improve prediction quality. However, so far, this is not strongly supported by the simulation results, except for several situations, e.g., high prevalence. Further investigations are needed to diagnose the model fitting and prediction.

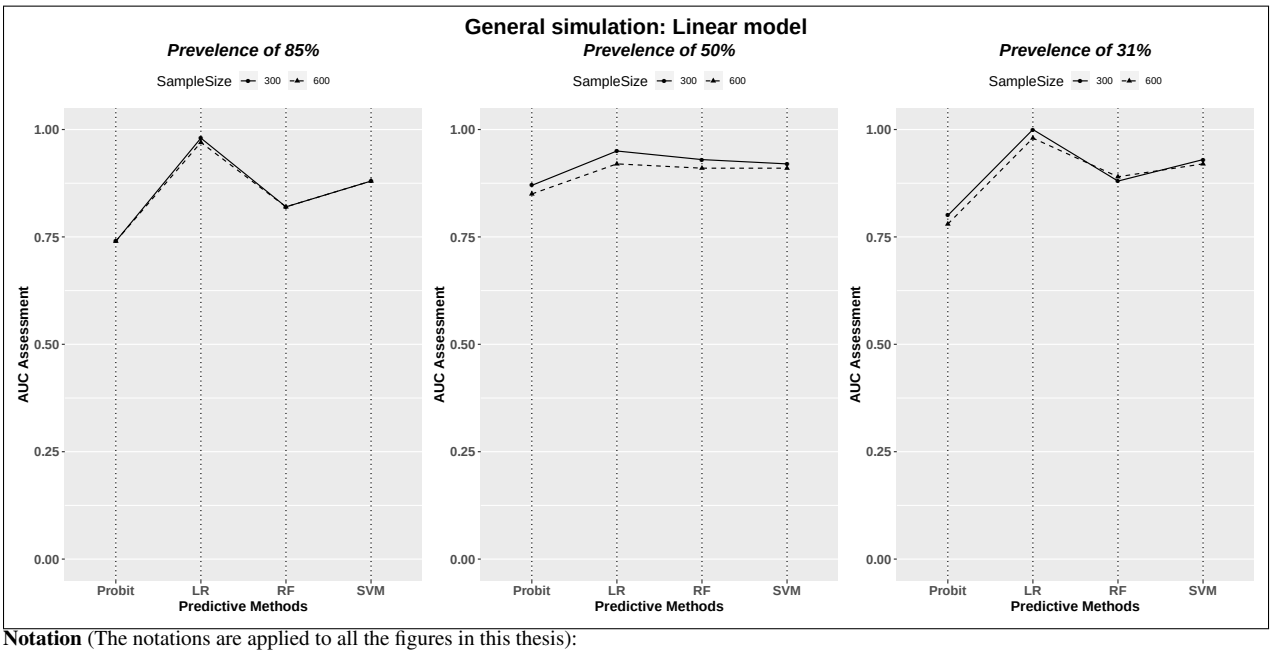
Table 1 Results of general simulation with N=300

Results of general simulation (N=300)						
Model Type	**Mean prevalence of sample (N=300)**	**Method#**	**Accuracy$; 95% EI&**	**Sensitivity$; 95% EI&**	**Specificity$; 95% EI&**	**AUC$; 95% EI&**
Linear Model	**85%**	**Probit**	0.62 (0.42, 0.80)	0.57 (0.35, 0.78)	1 (0.67, 1)	0.74 (0.59, 0.84)
		LR	0.98 (0.88, 1)	0.98 (0.92, 1)	1 (0.64, 1)	0.98 (0.79, 1)
		RF	0.93 (0.88, 0.98)	0.98 (0.82, 1)	0.68 (0.36, 1)	0.82 (0.66, 0.99)
		SVM	0.96 (0.88, 1)	1 (0.94, 1)	0.79 (0.43, 1)	0.88 (0.71, 1)
	50%	**Probit**	0.88 (0.49, 0.94)	0.90 (0.54, 1)	0.87 (0, 1)	0.87 (0.50, 0.94)
		LR	0.95 (0.83, 1)	0.96 (0.81, 1)	0.95 (0.86, 1)	0.95 (0.81, 1)
		RF	0.93 (0.85, 0.98)	0.94 (0.82, 1)	0.92 (0.82, 1)	0.93 (0.85, 0.98)
		SVM	0.92 (0.86, 0.98)	0.92 (0.81, 1)	0.92 (0.84, 1)	0.92 (0.85, 0.99)
	31%	**Probit**	0.77 (0.60, 0.89)	0.92 (0.69, 1)	0.70 (0.46, 0.88)	0.80 (0.65, 0.88)
		LR	1 (0.86, 1)	1 (0.75, 1)	1 (0.90, 1)	1 (0.82, 1)
		RF	0.91 (0.84, 0.98)	0.81 (0.60, 1)	0.95 (0.85, 1)	0.88 (0.79, 0.97)
		SVM	0.95 (0.85, 1)	0.89 (0.62, 1)	0.98 (0.92, 1)	0.93 (0.81, 1)
Non-linear Model	**64%**	**Probit**	0.81 (0.44, 0.92)	0.75 (0.08, 0.91)	0.95 (0.76, 1)	0.84 (0.50, 0.92)
		LR	0.88 (0.79, 0.94)	0.91 (0.82, 0.98)	0.83 (0.62, 0.95)	0.87 (0.75, 0.94)
		RF	0.87 (0.79, 0.93)	0.90 (0.82, 0.98)	0.80 (0.62, 0.95)	0.86 (0.76, 0.92)
		SVM	0.87 (0.79, 0.94)	0.92 (0.83, 0.98)	0.79 (0.57, 0.93)	0.85 (0.75, 0.92)
	53%	**Probit**	0.85 (0.51, 0.95)	0.84 (0.17, 0.98)	0.92 (0.73, 1)	0.85 (0.50, 0.95)
		LR	0.87 (0.80, 0.96)	0.88 (0.76, 0.97)	0.88 (0.75, 0.97)	0.88 (0.79, 0.96)
		RF	0.87 (0.81, 0.95)	0.90 (0.77, 1)	0.88 (0.71, 0.97)	0.87 (0.81, 0.96)
		SVM	0.87 (0.79, 0.96)	0.87 (0.74, 0.97)	0.88 (0.70, 0.96)	0.86 (0.79, 0.96)
	40%	**Probit**	0.84 (0.72, 0.92)	0.94 (0.85, 1)	0.80 (0.59, 0.92)	0.85 (0.77, 0.94)
		LR	0.89 (0.81, 0.95)	0.87 (0.72, 0.96)	0.90 (0.77, 0.98)	0.89 (0.80, 0.94)
		RF	0.89 (0.80, 0.94)	0.88 (0.71, 1)	0.90 (0.77, 0.97)	0.89 (0.80, 0.95)
		SVM	0.88 (0.78, 0.94)	0.85 (0.74, 0.96)	0.91 (0.77, 0.97)	0.88 (0.78, 0.93)

Notation (The notations are applied to all the results tables in this thesis):
#: Method:
Probit: The proposed method which employs probit regression models.
LR: Logistic regression.
RF: Random forest.
SVM: Support vector machine.
$: Criteria:
Accuracy: Median values of accuracy measurements of total iterative predictions.
AUC: Median values of AUC measurements of total iterative predictions.
&: $95\%EI$:
95% Empirical intervals of the measurements of total iterative predictions.

Table 2 Results of general simulation with N=600

Results of general simulation (N=600)						
Model Type	**Mean prevalence of sample (N=600)**	**Method#**	**Accuracy$; 95% EI&**	**Sensitivity$; 95% EI&**	**Specificity$; 95% EI&**	**AUC$; 95% EI&**
Linear Model	**85%**	**Probit**	0.61 (0.27, 0.83)	0.54 (0.16, 0.87)	0.95 (0.71, 1)	0.74 (0.58, 0.84)
		LR	0.99 (0.82, 1)	0.99 (0.94, 1)	0.95 (0.21, 1)	0.97 (0.58, 1)
		RF	0.94 (0.81, 0.97)	0.99 (0.95, 1)	0.64 (0.14, 0.87)	0.82 (0.55, 0.93)
		SVM	0.96 (0.81, 0.99)	1 (0.97, 1)	0.76 (0.05, 0.95)	0.88 (0.52, 0.97)
	50%	**Probit**	0.85 (0.55, 0.91)	0.89 (0.51, 1)	0.87 (0.29, 1)	0.85 (0.50, 0.91)
		LR	0.92 (0.84, 1)	0.93 (0.78, 1)	0.94 (0.83, 1)	0.92 (0.84, 1)
		RF	0.91 (0.85, 0.98)	0.91 (0.82, 0.98)	0.91 (0.83, 0.99)	0.91 (0.85, 0.98)
		SVM	0.91 (0.83, 0.97)	0.91 (0.77, 0.98)	0.91 (0.80, 0.98)	0.91 (0.83, 0.97)
	31%	**Probit**	0.73 (0.57, 0.84)	0.90 (0.73, 1)	0.66 (0.43, 0.90)	0.78 (0.50, 0.84)
		LR	0.98 (0.79, 1)	0.97 (0.64, 1)	0.99 (0.86, 1)	0.98 (0.75, 1)
		RF	0.92 (0.79, 0.97)	0.82 (0.65, 0.95)	0.96 (0.86, 0.99)	0.89 (0.74, 0.97)
		SVM	0.95 (0.79, 0.98)	0.87 (0.53, 0.98)	0.98 (0.87, 1)	0.92 (0.72, 0.98)
Non-linear Model	**64%**	**Probit**	0.80 (0.39, 0.90)	0.71 (0, 0.90)	0.97 (0.83, 1)	0.83 (0.50, 0.90)
		LR	0.88 (0.81, 0.93)	0.92 (0.86, 0.95)	0.83 (0.69, 0.91)	0.87 (0.80, 0.92)
		RF	0.88 (0.81, 0.93)	0.91 (0.87, 0.99)	0.80 (0.67, 0.94)	0.86 (0.79, 0.92)
		SVM	0.87 (0.80, 0.93)	0.91 (0.87, 0.97)	0.77 (0.64, 0.92)	0.85 (0.77, 0.92)
	53%	**Probit**	0.86 (0.75, 0.92)	0.83 (0.60, 0.98)	0.92 (0.73, 0.98)	0.87 (0.76, 0.92)
		LR	0.88 (0.81, 0.94)	0.88 (0.79, 0.95)	0.87 (0.81, 0.94)	0.88 (0.81, 0.94)
		RF	0.88 (0.81, 0.94)	0.90 (0.80, 0.97)	0.87 (0.78, 0.97)	0.88 (0.81, 0.94)
		SVM	0.86 (0.80, 0.92)	0.87 (0.75, 0.95)	0.87 (0.76, 0.93)	0.86 (0.80, 0.92)
	40%	**Probit**	0.83 (0.63, 0.90)	0.94 (0.68, 1)	0.78 (0.56, 0.94)	0.84 (0.50, 0.89)
		LR	0.88 (0.82, 0.92)	0.83 (0.76, 0.91)	0.91 (0.83, 0.97)	0.88 (0.81, 0.91)
		RF	0.87 (0.83, 0.92)	0.84 (0.75, 0.92)	0.90 (0.83, 0.96)	0.87 (0.82, 0.92)
		SVM	0.87 (0.82, 0.92)	0.84 (0.70, 0.91)	0.91 (0.83, 0.96)	0.86 (0.81, 0.91)



Notation (The notations are applied to all the figures in this thesis):
Probit: The proposed method.
LR: Logistic regression.
RF: Random forest.
SVM: Support vector machine.

Figure 1: Median AUC values for linear models in general simulation

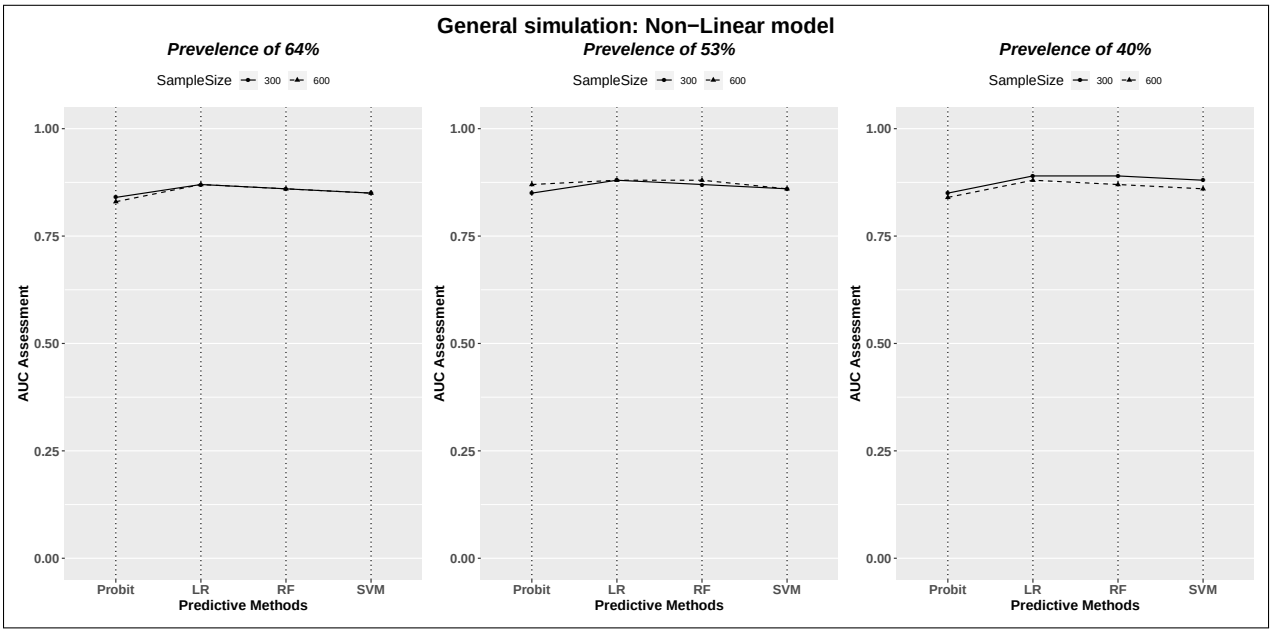


Figure 2: Median AUC values for non-linear models in general simulation

2.6 Real data application

We apply the proposed approach along with the competing methods to a real data set. The data was collected from a birth cohort study established on the Isle of Wight (IoW) in the United Kingdom [30]. In total, $1,456$ newborns were included in the study and subsequently followed up at ages 1, 2, 4, 10, 18, and 26 years of age.

In this application, asthma status (Yes/No) at 10 years of age is used in the analyses as the health outcome. Asthma status is defined as the diagnosis of asthma and wheezing in the last 12 months which is confirmed by doctor and/or use of asthma medication. The asthma prevalence is 14.69%. A pool of possible features which might be associated with the status of asthma are included as potential predictors, including maternal smoking status during pregnancy and a set of potential risk factors before age 4 of each subject. In total, the independent variables are consisted of 5 continuous variables and 40 categorical variables. Table S5 shows the descriptive statistics for all the variables.

2.6.1 Application settings

At the first stage, the IoW data is split to training part and testing part by the same ratio used in the simulation analysis, $4:1$. Then the inferences from the training part are used to make prediction on testing part. Following the setting in simulation study, for every predictive approach, we conduct 50 models by using 50 different training data. However, due to the limit of time, the three competing approaches make prediction based on their own selected important variables. The results are evaluated and summarized using the same assessment criteria used in the simulation.

2.6.2 Application results

The proposed prediction method shows low accuracy, mediocre AUC, high sensitivity and meanwhile low specificity (Table 3 and Figure 3). Although the proposed approach has dominance in identifying positive asthma cases, it is overall inferior to the three competing

approaches. In view of that the asthma prevalence is 14.69% in the real data, the results follow the same pattern as in simulation studies.

Table 3 Results of real data application

Method#	Accuracy$; 95% EI&	Sensitivity$; 95% EI&	Specificity$; 95% EI&	AUC$; 95% EI&
Probit	0.18 (0.12, 0.33)	0.95 (0.72, 1)	0.04 (0, 0.24)	0.50 (0.47, 0.55)
LR	0.86 (0.83, 0.89)	0.12 (0, 0.26)	0.99 (0.96, 1)	0.55 (0.50, 0.63)
RF	0.83 (0.79, 0.86)	0.03 (0, 0.11)	0.96 (0.94, 0.99)	0.50 (0.48, 0.53)
SVM	0.86 (0.84, 0.89)	0.11 (0.02, 0.20)	1 (0.98, 1)	0.55 (0.51, 0.60)

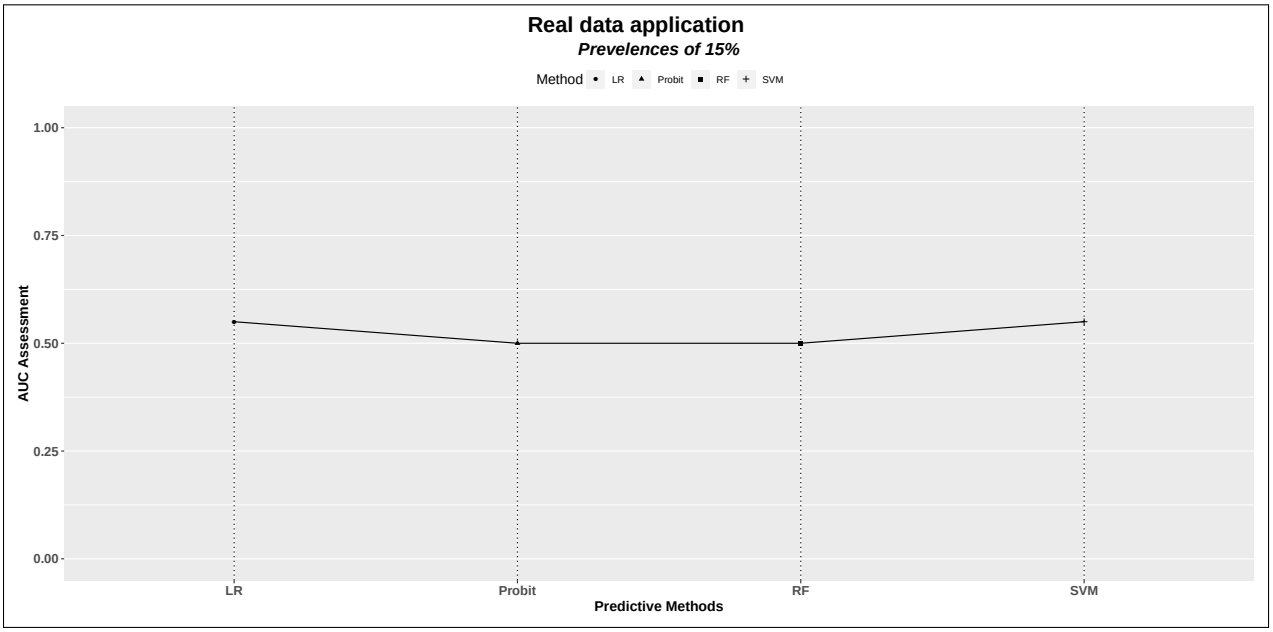


Figure 3: Median AUC values for real data application

Chapter 3

Conclusion

In this thesis, a prediction method for classifying health status is developed based on a published variable selection method. The variable selection approach adopts a bayesian framework to draw inference of the features by applying a reproducing Gaussian kernel to a probit regression. The basis for the subsequent prediction is dependent on the inferred regression coefficients (β) and selected variable (δ). With β and δ estimated, and information on $h(X)$, we are able to estimate or predict the response, Z, using the probit model 2.1. To study the developed method's forecasting facility, three commonly used machine learning approaches, logistic regression, random forest, and support vector machine are used as the reference of the proposed methods. These methods are briefly introduced and discussed.

By using a group of simulation studies, the prediction ability of the developed method is assessed. We discovered multiple combinations of different types of covariate variable and independent variables. Due to limit of time and source, we did not cover all the possible variable combinations, like using few continuous variable and lots of categorical variables to perform the prediction. In addition, by conducting an application to a real dataset with focus on predicting asthma status, we explored the feasibility of the developed method in forecasting health conditions in real-life situations. In most time, the developed approach generally has a stable prediction performance. In comparison to the three comparable methods, the proposed approach has lower accuracy and AUC values. However, it overall has strong competence in finding rare cases. For instance, when there are a small number of positive cases in a dataset, the developed method has excellent ability to identify those positive cases and it has advantage over the other

three competitive methods.

In conclusion, we developed and introduced a predictive method relying on a published variable selection technique. A cluster of simulation studies and a application are applied to examine the approach on its ability of prediction. Observing the highly developing skills in machine learning and prediction fields. This method is developed in a novel way and it mainly uses statistical skills. We hope it could bring new thoughts into the areas of statistical learning and health prediction. Simulations and real data applications indicated that the proposed approach overall did not perform well compared to the competing approaches. The approach in Zhang et al [16] was designed for complex interactions between and among variables. However, such a design was not supported by findings from the simulations presented in this thesis. Additional explorations are warranted to analyze model fitting and improve prediction performance. Because of the time limit, in the real data application, the three competing methods did not perform predictions based on important variables selected from approach by Zhang et al [16] and thus the comparison results tend to be biased. Further and in-depth assessment of the proposed methods are certainly needed.

References

[1] Ireneous N Soyiri and Daniel D Reidpath. An overview of health forecasting. *Environmental health and preventive medicine*, 18(1):1–9, 2013.

[2] Fang-Yu Qin, Zhe-Qi Lv, Dan-Ni Wang, Bo Hu, and Chao Wu. Health status prediction for the elderly based on machine learning. *Archives of Gerontology and Geriatrics*, 90:104121, 2020.

[3] Akhan Akbulut, Egemen Ertugrul, and Varol Topcu. Fetal health status prediction based on maternal clinical history using machine learning techniques. *Computer methods and programs in biomedicine*, 163:87–100, 2018.

[4] Andrew J Walley, Alexandra IF Blakemore, and Philippe Froguel. Genetics of obesity and the prediction of risk for health. *Human molecular genetics*, 15(suppl_2):R124–R130, 2006.

[5] Shahadat Uddin, Arif Khan, Md Ekramul Hossain, and Mohammad Ali Moni. Comparing different supervised machine learning algorithms for disease prediction. *BMC medical informatics and decision making*, 19(1):1–16, 2019.

[6] R Mitchell, J Michalski, and T Carbonell. *An artificial intelligence approach*. Springer, 2013.

[7] Xiaowei Song, Arnold Mitnitski, Jafna Cox, and Kenneth Rockwood. Comparison of machine learning techniques with classical statistical models in predicting health outcomes. In *Medinfo*, pages 736–740, 2004.

[8] Ch Anwar Ul Hassan, Muhammad Sufyan Khan, and Munam Ali Shah. Comparison of machine learning algorithms in data classification. In *2018 24th International Conference on Automation and Computing (ICAC)*, pages 1–6. IEEE, 2018.

[9] Sandro Galea and Katherine M Keyes. Population health science and the challenges of prediction. *Annals of internal medicine*, 167(7):511–512, 2017.

[10] Diana Kuh, Yoav Ben-Shlomo, John Lynch, Johan Hallqvist, and Chris Power. Life course epidemiology. *Journal of epidemiology and community health*, 57(10):778, 2003.

[11] Kenneth J Rothman. Synergy and antagonism in cause-effect relationships. *American journal of epidemiology*, 99(6):385–388, 1974.

[12] Carsten F Dormann, Jane Elith, Sven Bacher, Carsten Buchmann, Gudrun Carl, Gabriel Carré, Jaime R García Marquéz, Bernd Gruber, Bruno Lafourcade, Pedro J Leitao, et al. Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, 36(1):27–46, 2013.

[13] David M Allen. The relationship between variable selection and data agumentation and a method for prediction. *technometrics*, 16(1):125–127, 1974.

[14] Alain Rakotomamonjy. Variable selection using svm-based criteria. *Journal of machine learning research*, 3(Mar):1357–1370, 2003.

[15] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[16] Hongmei Zhang, Arnab Maity, Hasan Arshad, John Holloway, and Wilfried Karmaus. Variable selection in semi-parametric models. *Statistical methods in medical research*, 25(4):1736–1752, 2016.

[17] James Mercer. Xvi. functions of positive and negative type, and their connection the theory

of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209(441-458):415–446, 1909.

[18] Nello Cristianini, John Shawe-Taylor, et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[19] Daniel Gianola and Johannes BCHM Van Kaam. Reproducing kernel hilbert spaces regression methods for genomic assisted prediction of quantitative traits. *Genetics*, 178(4):2289–2303, 2008.

[20] Dawei Liu, Xihong Lin, and Debashis Ghosh. Semiparametric regression of multidimensional genetic pathway data: Least-squares kernel machines and linear mixed models. *Biometrics*, 63(4):1079–1088, 2007.

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

[22] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[23] Foster Provost and Tom Fawcett. Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, KDD'97, page 43¡§C48. AAAI Press, 1997.

[24] Foster J. Provost, Tom Fawcett, and Ron Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 445¡§C453, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[25] Jin Huang and Charles X Ling. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on knowledge and Data Engineering*, 17(3):299–310, 2005.

[26] Charles X Ling, Jin Huang, Harry Zhang, et al. Auc: a statistically consistent and more discriminating measure than accuracy. In *Ijcai*, volume 3, pages 519–524, 2003.

[27] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.

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

[29] David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, and Friedrich Leisch. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2019. R package version 1.7-3.

[30] S Hasan Arshad, Veeresh Patil, Frances Mitchell, Stephen Potter, Hongmei Zhang, Susan Ewart, Linda Mansfield, Carina Venter, John W Holloway, and Wilfried J Karmaus. Cohort profile update: The isle of wight whole population birth cohort (iowbc). *International Journal of Epidemiology*, 49(4):1083–1084, 2020.

Appendix A

Table S1 Results of derivative simulation 1

Results of derivative simulation 1						
Model Type	**Mean prevalence of sample (N=300)**	**Method#**	**Accuracy$; 95% EI&**	**Sensitivity$; 95% EI&**	**Specificity$; 95% EI&**	**AUC$; 95% EI&**
Linear Model	**92%**	**Probit**	0.69 (0.18, 0.92)	0.68 (0.14, 1)	0.76 (0, 1)	0.66 (0.40, 0.87)
		LR	0.90 (0.83, 0.97)	0.98 (0.92, 1)	0.15 (0, 0.60)	0.56 (0.48, 0.77)
		RF	0.91 (0.85, 0.97)	0.98 (0.95, 1)	0 (0, 0.50)	0.50 (0.48, 0.75)
		SVM	0.91 (0.84, 0.98)	1 (0.96, 1)	0 (0, 0.50)	0.50 (0.48, 0.75)
	57%	**Probit**	0.58 (0.42, 0.75)	0.85 (0.08, 1)	0.39 (0, 0.93)	0.54 (0.48, 0.72)
		LR	0.79 (0.67, 0.89)	0.83 (0.73, 0.93)	0.74 (0.59, 0.91)	0.78 (0.66, 0.89)
		RF	0.76 (0.69, 0.88)	0.81 (0.66, 0.93)	0.73 (0.53, 0.88)	0.76 (0.67, 0.88)
		SVM	0.77 (0.69, 0.85)	0.83 (0.67, 0.95)	0.69 (0.55, 0.83)	0.77 (0.68, 0.85)
	24%	**Probit**	0.72 (0.33, 0.82)	0.19 (0, 0.95)	0.93 (0.07, 1)	0.53 (0.48, 0.72)
		LR	0.85 (0.76, 0.93)	0.63 (0.31, 0.82)	0.93 (0.85, 0.98)	0.76 (0.61, 0.88)
		RF	0.83 (0.75, 0.90)	0.50 (0.21, 0.78)	0.94 (0.87, 1)	0.72 (0.61, 0.85)
		SVM	0.83 (0.74, 0.90)	0.49 (0.25, 0.82)	0.93 (0.85, 1)	0.71 (0.57, 0.85)
Non-linear Model	**80%**	**Probit**	0.64 (0.18, 0.85)	0.69 (0, 1)	0.38 (0, 1)	0.50 (0.44, 0.61)
		LR	0.80 (0.72, 0.88)	1 (0.92, 1)	0 (0, 0.22)	0.50 (0.48, 0.59)
		RF	0.80 (0.71, 0.89)	0.98 (0.92, 1)	0 (0, 0.56)	0.50 (0.49, 0.70)
		SVM	0.80 (0.72, 0.87)	1 (0.94, 1)	0 (0, 0.25)	0.50 (0.50, 0.60)
	63%	**Probit**	0.53 (0.36, 0.70)	0.68 (0.02, 1)	0.28 (0, 1)	0.51 (0.39, 0.58)
		LR	0.61 (0.53, 0.74)	0.89 (0.74, 1)	0.14 (0, 0.36)	0.50 (0.47, 0.60)
		RF	0.69 (0.61, 0.76)	0.88 (0.71, 1)	0.37 (0, 0.62)	0.62 (0.50, 0.74)
		SVM	0.66 (0.58, 0.75)	0.89 (0.76, 1)	0.31 (0, 0.50)	0.58 (0.50, 0.68)
	36%	**Probit**	0.53 (0.32, 0.71)	0.43 (0, 1)	0.59 (0, 1)	0.50 (0.42, 0.58)
		LR	0.68 (0.58, 0.74)	0.33 (0.04, 0.56)	0.86 (0.71, 1)	0.58 (0.49, 0.69)
		RF	0.68 (053., 0.78)	0.31 (0.05, 0.58)	0.86 (0.74, 1)	0.58 (0.48, 0.71)
		SVM	0.67 (0.55, 0.76)	0.33 (0.05, 0.55)	0.85 (0.71, 0.97)	0.58 (0.50, 0.69)

Table S2 Results of derivative simulation 2

Results of derivative simulation 2						
Model Type	**Mean prevalence of sample (N=300)**	**Method#**	**Accuracy$; 95% EI&**	**Sensitivity$; 95% EI&**	**Specificity$; 95% EI&**	**AUC$; 95% EI&**
Linear Model	**92%**	**Probit**	0.54 (0.08, 0.92)	0.55 (0, 1)	0.24 (0, 1)	0.50 (0.37, 0.78)
		LR	0.92 (0.85, 0.98)	1 (0.93, 1)	0.20 (0, 0.80)	0.60 (0.49, 0.88)
		RF	0.91 (0.85, 0.97)	0.98 (0.91, 1)	0.20 (0, 0.60)	0.59 (0.47, 0.79)
		SVM	0.91 (0.84, 0.98)	1 (0.96, 1)	0 (0, 0.57)	0.50 (0.50, 0.77)
	50%	**Probit**	0.51 (0.37, 0.62)	0.72 (0, 1)	0.22 (0, 1)	0.50 (0.43, 0.63)
		LR	0.75 (0.67, 0.85)	0.80 (0.62, 0.89)	0.76 (0.63, 0.87)	0.76 (0.66, 0.85)
		RF	0.74 (0.64, 0.82)	0.74 (0.60, 91)	0.73 (0.57, 86)	0.74 (0.64, 0.83)
		SVM	0.75 (0.65, 0.84)	0.78 (0.61, 0.88)	0.73 (0.64, 0.88)	0.76 (0.63, 0.84)
	21%	**Probit**	0.43 (0.14, 0.78)	0.52 (0, 1)	0.44 (0, 1)	0.50 (0.41, 0.74)
		LR	0.87 (0.73, 0.95)	0.55 (0.19, 0.91)	0.93 (0.88, 1)	0.74 (0.53, 0.94)
		RF	0.84 (0.73, 0.92)	0.46 (0.12, 0.80)	0.94 (0.87, 1)	0.70 (0.54, 0.86)
		SVM	0.86 (0.73, 0.93)	0.48 (0.11, 0.82)	0.96 (0.88, 1)	0.71 (0.53, 0.89)
Non-linear Model	**76%**	**Probit**	0.49 (0.20, 0.77)	0.47 (0, 1)	0.60 (0, 1)	0.50 (0.38, 0.64)
		LR	0.76 (0.67, 0.83)	1 (0.87, 1)	0 (0, 0.33)	0.50 (0.50, 0.64)
		RF	0.71 (0.61, 0.83)	0.89 (0.75, 1)	0.20 (0, 0.50)	0.55 (0.45, 0.67)
		SVM	0.75 (0.68, 0.82)	1 (0.91, 1)	0 (0, 0.24)	0.50 (0.48, 0.58)
	57%	**Probit**	0.49 (0.31, 0.66)	0.41 (0, 1)	0.52 (0, 1)	0.50 (0.41, 0.61)
		LR	0.57 (0.45, 0.67)	0.78 (0.60, 0.97)	0.29 (0.04, 0.50)	0.53 (0.44, 0.62)
		RF	0.68 (0.57, 0.80)	0.76 (0.58, 0.87)	0.59 (0.38, 0.76)	0.66 (0.56, 0.77)
		SVM	0.60 (0.51, 0.75)	0.71 (0.47, 0.94)	0.48 (0.15, 0.74)	0.58 (0.48, 0.74)
	30%	**Probit**	0.53 (0.26, 0.75)	0.44 (0, 1)	0.57 (0, 1)	0.50 (0.42, 0.61)
		LR	0.72 (0.62, 0.78)	0.22 (0, 0.53)	0.90 (0.81, 1)	0.57 (0.49, 0.68)
		RF	0.71 (0.52, 0.80)	0.33 (0.13, 0.64)	0.84 (0.66, 0.96)	0.61 (0.43, 0.74)
		SVM	0.72 (0.62, 0.79)	0.16 (0, 0.46)	0.94 (0.85, 1)	0.56 (0.50, 0.68)

Table S3 Results of derivative simulation 3

Results of derivative simulation 3						
Model Type	**Mean prevalence of sample (N=300)**	**Method#**	**Accuracy$; 95% EI&**	**Sensitivity$; 95% EI&**	**Specificity$; 95% EI&**	**AUC$; 95% EI&**
Linear Model	**57%**	**Probit**	0.82 (0.71, 0.90)	0.78 (0.64, 0.88)	0.88 (0.72, 1)	0.83 (0.72, 0.90)
		LR	0.98 (0.85, 1)	0.98 (0.87, 1)	1 (0.79, 1)	0.99 (0.83, 1)
		RF	0.92 (0.86, 0.97)	0.93 (0.84, 1)	0.89 (0.79, 1)	0.91 (0.85, 0.97)
		SVM	0.94 (0.84, 1)	0.94 (0.85, 1)	0.93 (0.78, 1)	0.94 (0.84, 1)
	52%	**Probit**	0.83 (0.46, 0.90)	0.83 (0.38, 0.96)	0.85 (0.37, 0.96)	0.83 (0.53, 0.91)
		LR	0.96 (0.87, 1)	0.96 (0.88, 1)	0.97 (0.83, 1)	0.96 (0.87, 1)
		RF	0.93 (0.85, 0.98)	0.93 (0.85, 1)	0.94 (0.79, 1)	0.94 (0.84, 0.98)
		SVM	0.94 (0.83, 1)	0.93 (0.85, 1)	0.96 (0.80, 1)	0.94 (0.83, 1)
	42%	**Probit**	0.80 (0.38, 0.93)	0.90 (0, 1)	0.80 (0, 1)	0.79 (0.49, 0.93)
		LR	0.97 (0.88, 1)	0.97 (0.81, 1)	0.97 (0.88, 1)	0.97 (0.88, 1)
		RF	0.96 (0.88, 1)	0.95 (0.78, 1)	0.97 (0.87, 1)	0.95 (0.86, 1)
		SVM	0.94 (0.85, 0.98)	0.92 (0.78, 1)	0.96 (0.85, 1)	0.94 (0.84, 0.99)
Non-linear Model	**71%**	**Probit**	0.75 (0.38, 0.90)	0.65 (0.28, 0.87)	1 (0.45, 1)	0.81 (0.49, 0.92)
		LR	0.95 (0.88, 1)	0.95 (0.89, 1)	0.91 (0.77, 1)	0.94 (0.84, 1)
		RF	0.94 (0.83, 1)	0.98 (0.90, 1)	0.87 (0.61, 1)	0.92 (0.78, 1)
		SVM	0.94 (0.84, 1)	0.97 (0.90, 1)	0.86 (0.67, 1)	0.91 (0.80, 1)
	61%	**Probit**	0.77 (0.39, 0.92)	0.77 (0.14, 0.98)	0.88 (0, 1)	0.79 (0.44, 0.93)
		LR	0.93 (0.89, 0.98)	0.95 (0.86, 1)	0.90 (0.82, 1)	0.93 (0.88, 0.98)
		RF	0.96 (0.88, 1)	1 (0.91, 1)	0.90 (0.82, 1)	0.95 (0.88, 1)
		SVM	0.95 (0.87, 1)	0.97 (0.89, 1)	0.93 (0.79, 1)	0.94 (0.87, 1)
	48%	**Probit**	0.88 (0.56, 0.94)	0.90 (0.22, 1)	0.87 (0.62, 1)	0.88 (0.50, 0.94)
		LR	0.92 (0.84, 0.97)	0.91 (0.80, 1)	0.92 (0.83, 1)	0.92 (0.85, 0.96)
		RF	0.93 (0.85, 0.98)	0.93 (0.82, 1)	0.93 (0.83, 1)	0.93 (0.86, 0.98)
		SVM	0.93 (0.84, 0.98)	0.93 (0.78, 1)	0.92 (0.82, 1)	0.93 (0.84, 0.98)

Table S4 Results of derivative simulation 4

Results of derivative simulation 4						
Model Type	**Mean prevalence of sample (N=300)**	**Method#**	**Accuracy$; 95% EI&**	**Sensitivity$; 95% EI&**	**Specificity$; 95% EI&**	**AUC$; 95% EI&**
Linear Model	**81%**	**Probit**	0.65 (0.50, 0.78)	0.58 (0.42, 0.73)	1 (0.83, 1)	0.77 (0.67, 0.83)
		LR	0.98 (0.88, 1)	0.98 (0.94, 1)	1 (0.54, 1)	0.98 (0.74, 1)
		RF	0.94 (0.86, 0.98)	0.98 (0.92, 1)	0.77 (0.33, 1)	0.87 (0.67, 0.99)
		SVM	0.95 (0.85, 0.98)	0.98 (0.95, 1)	0.83 (0.33, 1)	0.91 (0.67, 0.99)
	50%	**Probit**	0.89 (0.43, 0.96)	0.92 (0, 1)	0.91 (0.18, 1)	0.89 (0.47, 0.97)
		LR	0.96 (0.87, 1)	0.96 (0.85, 1)	0.97 (0.86, 1)	0.96 (0.87, 1)
		RF	0.95 (0.89, 1)	0.94 (0.87, 1)	0.95 (0.87, 1)	0.95 (0.88, 1)
		SVM	0.95 (0.84, 1)	0.94 (0.79, 1)	0.94 (0.84, 1)	0.95 (0.84, 1)
	35%	**Probit**	0.80 (0.69, 0.89)	0.94 (0.67, 1)	0.73 (0.56, 0.92)	0.83 (0.72, 0.92)
		LR	0.97 (0.86, 1)	0.96 (0.78, 1)	0.98 (0.89, 1)	0.96 (0.82, 1)
		RF	0.92 (0.84, 0.97)	0.85 (0.67, 1)	0.95 (0.87, 1)	0.90 (0.82, 0.96)
		SVM	0.93 (0.83, 0.98)	0.88 (0.67, 1)	0.97 (0.88, 1)	0.92 (0.81, 0.99)
Non-linear Model	**61%**	**Probit**	0.85 (0.43, 0.95)	0.79 (0.11, 0.95)	0.96 (0.76, 1)	0.87 (0.48, 0.95)
		LR	0.90 (0.84, 0.96)	0.93 (0.85, 1)	0.87 (0.76, 0.97)	0.90 (0.84, 0.96)
		RF	0.90 (0.82, 0.97)	0.92 (0.84, 1)	0.87 (0.74, 1)	0.90 (0.79, 0.96)
		SVM	0.90 (0.81, 0.96)	0.94 (0.85, 1)	0.85 (0.67, 0.96)	0.89 (0.79, 0.96)
	52%	**Probit**	0.89 (0.52, 0.94)	0.86 (0.43, 1)	0.93 (0.73, 1)	0.89 (0.50, 0.94)
		LR	0.91 (0.84, 0.96)	0.90 (0.79, 0.97)	0.92 (0.81, 1)	0.91 (0.84, 0.96)
		RF	0.91 (0.85, 0.96)	0.93 (0.81, 1)	0.93 (0.76, 1)	0.92 (0.84, 0.96)
		SVM	0.90 (0.82, 0.96)	0.90 (0.79, 1)	0.89 (0.81, 0.97)	0.90 (0.82, 0.96)
	42%	**Probit**	0.87 (0.56, 0.94)	0.96 (0.19, 1)	0.84 (0.55, 1)	0.89 (0.50, 0.95)
		LR	0.92 (0.83, 0.98)	0.91 (0.78, 1)	0.92 (0.81, 1)	0.91 (0.83, 0.98)
		RF	0.91 (0.82, 0.97)	0.89 (0.76, 1)	0.91 (0.82, 1)	0.91 (0.82, 0.96)
		SVM	0.90 (0.79, 0.97)	0.89 (0.73, 0.96)	0.93 (0.82, 1)	0.90 (0.79, 0.97)

Table S5 Summary statistics of IoW data

Variable	Continuous variable (Yes/No)	Definition	Mean	SD	Median	Min	Max
Mat_age	Yes	Maternal age at booking	26.98	5.29	27	16	43
Birthweight	Yes	Birth weight	3.41	0.52	3.42	0.93	5.16
Solid_food	Yes	Child's BMI (z-score) at age 1	14.53	4.75	14	3	48
SDS_BMI_1	Yes	Child's BMI (z-score) at age 4	-0.11	1.56	-0.15	-15.13	9.91
SDS_BMI_4	Yes	Age at which solid foods (cereals/solids) were introduced to the child's diet	0.2	1.08	0.24	-6.77	3.91
			Frequency of "No"	**Frequency of "Yes"**	**Relative frequency of "No"**	**Relative frequency of "Yes"**	
Asthma_10YR	No	Doctor diagnosed asthma and wheezing in the last 12 months and/or use of asthma medication	1167	201	0.85	0.15	
Winter	No	Season of birth	922	446	0.67	0.33	
Spring	No	Season of birth	1043	325	0.76	0.24	
Summer	No	Season of birth	1052	316	0.77	0.23	
Mat_smoking_birth	No	Atopy status -sensitisation (+SPT) to one or more allergen by age 2	1038	330	0.76	0.24	
Pat_smoking_birth	No	Atopy status - sensitisation (+SPT) to one or more allergens at age 4	846	522	0.62	0.38	
Dog_birth	No	Household pet cat by age 2	971	397	0.71	0.29	
Cat_birth	No	Household pet cat at age 4	913	455	0.67	0.33	
Furry_pet_birth	No	Household pet cat during pregnancy	636	732	0.46	0.54	
Mat_asthma	No	Occurrence of chest infections before age 2	1224	144	0.89	0.11	
Mat_eczema	No	Occurrence of cough before age 2	1205	163	0.88	0.12	
Mat_hayfever	No	Occurrence of cough at age 4	1098	270	0.8	0.2	
Pat_asthma	No	Mode of delivery	1235	133	0.9	0.1	
Pat_eczema	No	Household pet dog by age 2	1278	90	0.93	0.07	
Pat_hayfever	No	Household pet dog at age 4	1163	205	0.85	0.15	
Sex	No	Household pet dog during pregnancy	672	696	0.49	0.51	
Delivery	No	Eczema status by age 2	1234	134	0.9	0.1	
Parity	No	Eczema status at age 4	594	774	0.43	0.57	
Wheeze_without_cold_2YR	No	Main residence on a farm in the first year of life	935	433	0.68	0.32	
Cough_2YR	No	Household furry pet (dog, cat or other animal) by age 2	855	513	0.62	0.38	
Nasal_symp_2YR	No	Household furry pet (dog, cat or other animal) at age 4	946	422	0.69	0.31	
Chest_infection_2YR	No	Household furry pet during pregnancy - dog, cat or other animal	1049	319	0.77	0.23	
Noct_symp_2YR	No	Hayfever status by age 2	868	500	0.63	0.37	
Eczema_2YR	No	Hayfever status at age 4	945	423	0.69	0.31	
Hayfever_2YR	No	Maternal asthma status (pregnancy)	1077	291	0.79	0.21	
Atopy_2YR	No	Maternal eczema status (pregnancy)	1025	343	0.75	0.25	
Monosensitisation_2YR	No	Maternal hayfever status (pregnancy)	1061	307	0.78	0.22	
Polysensitisation_2YR	No	Maternal smoking status during pregnancy	1237	131	0.9	0.1	
Dog_2YR	No	Sensitisation (+SPT) to one allergen by age 2	932	436	0.68	0.32	
Cat_2YR	No	Sensitisation (+SPT) to one allergen at age 4	271	1097	0.2	0.8	
Furry_pet_2YR	No	Occurrence of nasal symptoms before age 2	79	1289	0.06	0.94	
Wheeze_without_cold_4YR	No	Occurrence of nasal symptoms at age 4	1159	209	0.85	0.15	
Cough_4YR	No	Occurrence of nocturnal asthma symptoms before age 2	1047	321	0.77	0.23	
Nasal_symp_4YR	No	Occurrence of nocturnal asthma symptoms at age 4	1107	261	0.81	0.19	
Noct_Symp_4YR	No	Position of child in the family based on number of live births - were they the first born child or not	1058	310	0.77	0.23	
Eczema_4YR	No	Paternal asthma status (pregnancy)	1193	175	0.87	0.13	
Hayfever_4YR	No	Paternal eczema status (pregnancy)	1262	106	0.92	0.08	
Atopy_4YR	No	Paternal hayfever status (pregnancy)	832	536	0.61	0.39	
Monosensitisation_4YR	No	Paternal smoking status during pregnancy	923	445	0.67	0.33	
Polysensitisation_4YR	No	Sensitisation (+SPT) to two or more allergens by age 2	1241	127	0.91	0.09	
Dog_4YR	No	Sensitisation (+SPT) to two or more allergens by age 4	987	381	0.72	0.28	
Cat_4YR	No	Child's gender	858	510	0.63	0.37	
Furry_pet_4YR	No	Likely occurence of wheezing in the absence of a cold before age 2	565	803	0.41	0.59	
Farm_early_life	No	Likely occurence of wheezing in the absence of a cold at age 4	1301	67	0.95	0.05	

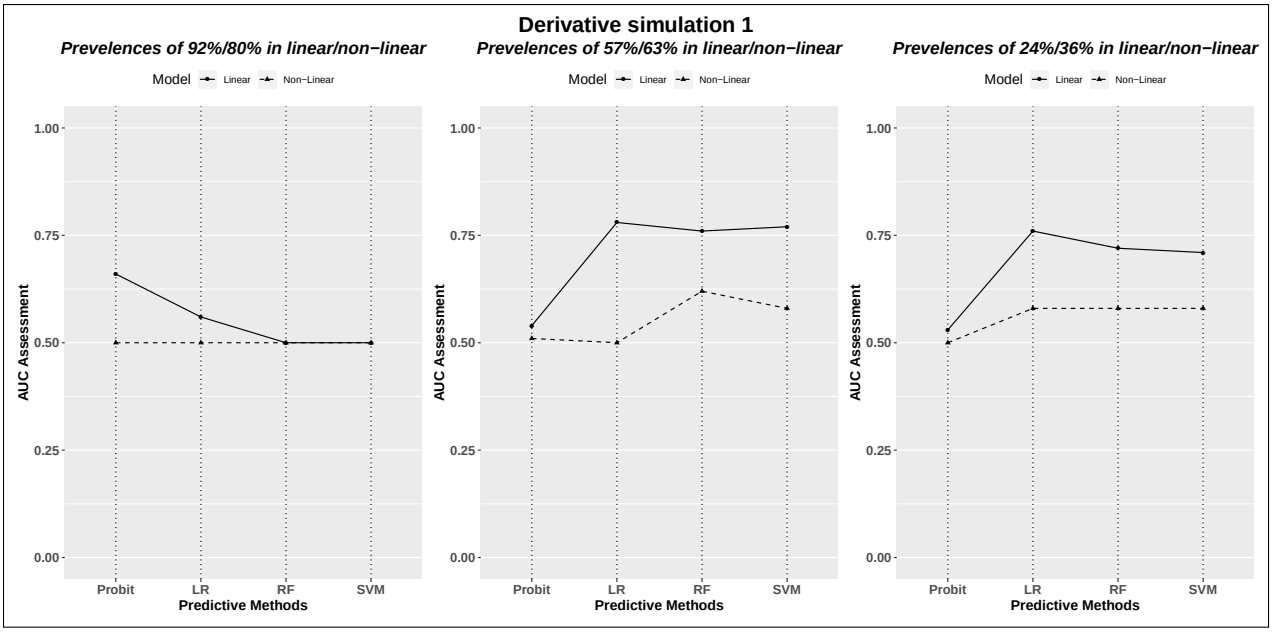


Figure S1: Median AUC values for derivative Simulation 1

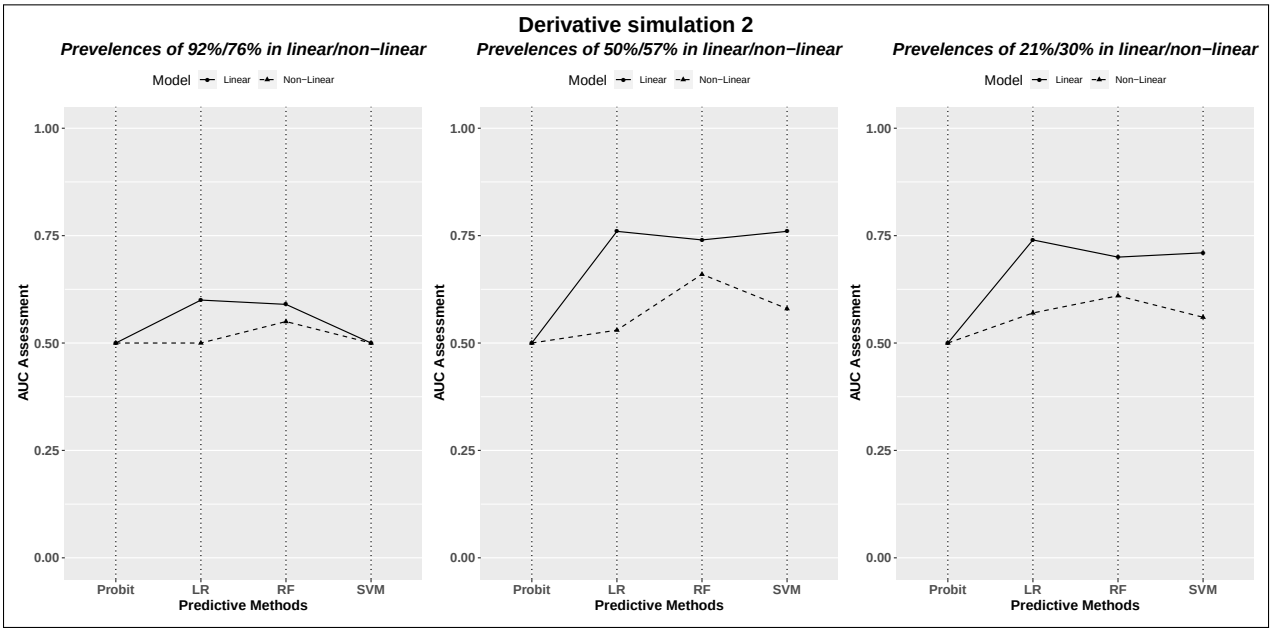


Figure S2: Median AUC values for derivative Simulation 2

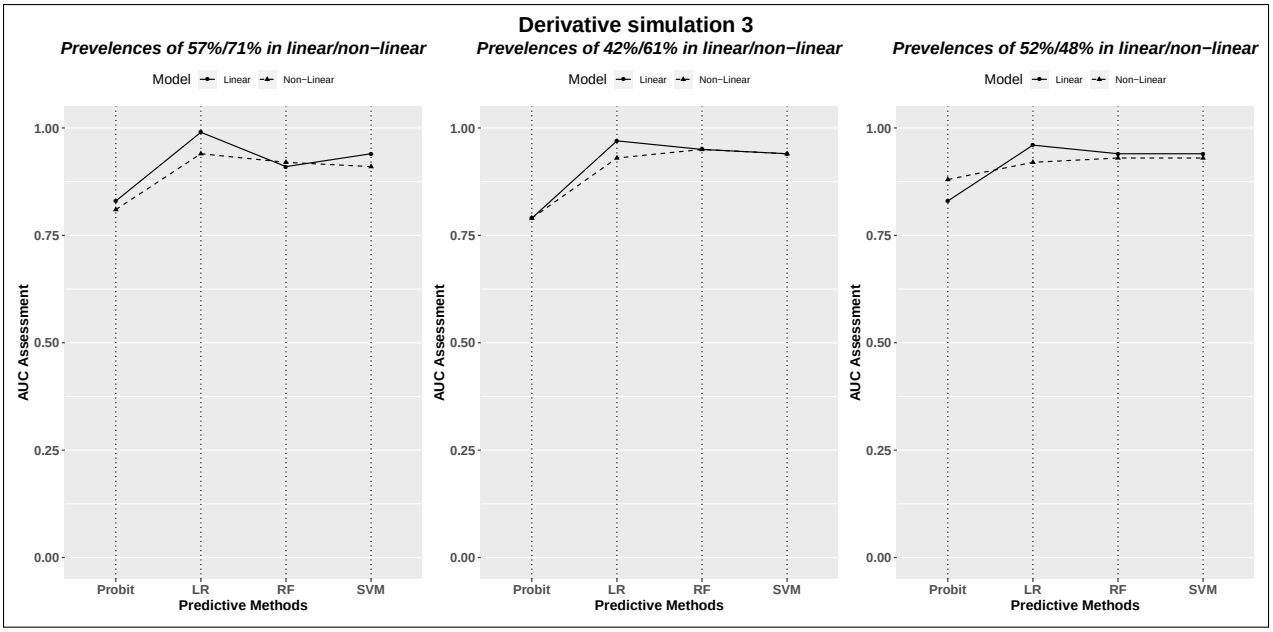


Figure S3: Median AUC values for derivative Simulation 3

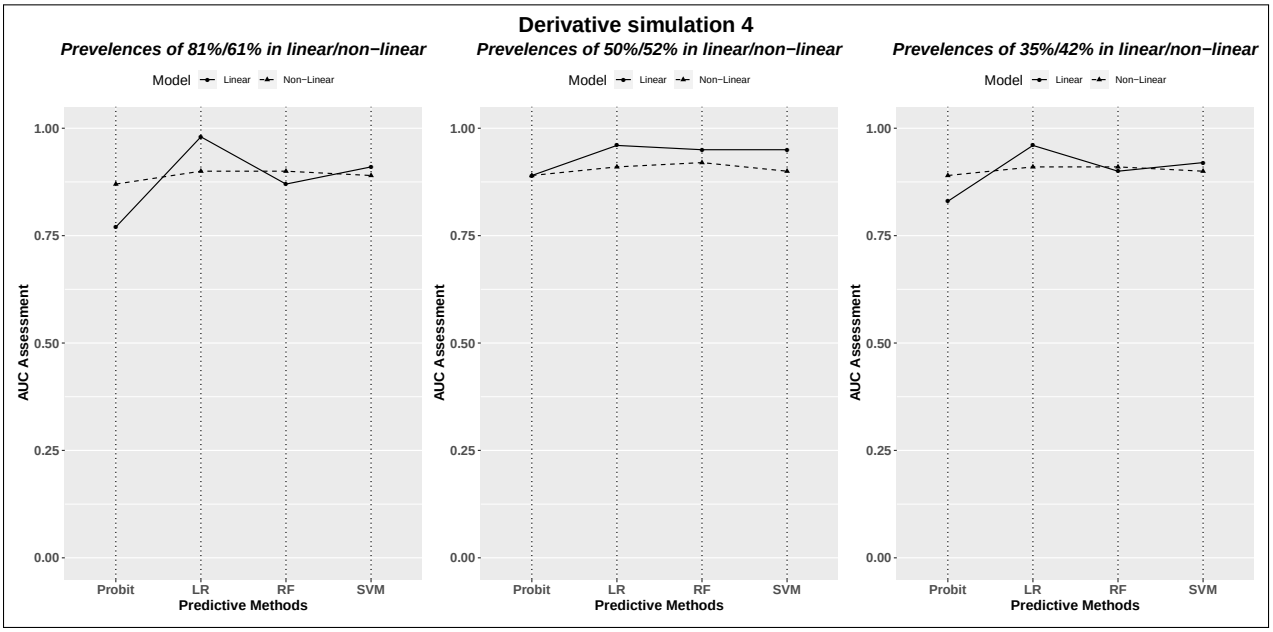


Figure S4: Median AUC values for derivative Simulation 4

Appendix B

```
########################################Example R code#####################################

########################################Simulation#########################################
# n: sample size, numVov: Number of covariate, numGene: Number of independent variables.
# numData: Number of MC replicate
# Seed for high prevelence in linear/non-linear model: set.seed(30013*nr)/set.seed(30023*nr),
# Seed for medium prevelence in linear/non-linear model: set.seed(30012*nr)/set.seed(30022*nr),
# Seed for low prevelence in linear/non-linear model: set.seed(30011nr)/set.seed(30021*nr),
# Coefficients for high prevelance level in linear model:
# gEff<-c(3, -2.5, 3.5, rep(0,length=numGene-3))*delta
# Coefficients for medium prevelance level in linear model:
# gEff<-c(0.5, -2, 1.5, rep(0,length=numGene-3))*delta
# Coefficients for low prevelance level in linear model:
# gEff<-c(-1.5, -2, 4, rep(0,length=numGene-3))*delta
# Coefficients for high prevelance level in non-linear model:
# gEff<-c(0,0,10,rep(0,length=numGene-3))*delta
# Coefficients for medium prevelance level in non-linear model:
# gEff<-c(0,0,1,rep(0,length=numGene-3))*delta
# Coefficients for low prevelance level in non-linear model:
# gEff<-c(0,0,-10,rep(0,length=numGene-3))*delta

library(mnormt)

n<-300
# Sample size is 600
#n<-600
numCov<-1
numGene<-12
numData<-50
```

```
for (nr in 1:numData)
{
#generate covariates
x<-matrix(rep(0,numCov*n),ncol=numCov)
# Seed for high prevelence in linear model
set.seed(30013*nr)
for (i in 1:numCov)
{
        x[,i]<-rnorm(n,0,4)
}
################ this is for extended simulation 1 #################
#for (i in 1:numCov)                                                #
#{                                                                  #
#       x[,i]<-rbinom(n,1,.5)                                       #
#}                                                                  #
####################################################################

################ this is for extended simulation 2 #################
#Do not simulate x                                                  #
####################################################################

#generate independent variables
g<-matrix(rep(0,numGene*n),ncol=numGene)
# Seed for high prevelence in linear model
set.seed(30013*nr)
for (i in 1:numGene)
{
        g[,i]<-runif(n,0.0001, numGene/(2*i))
}

################ this is for extended simulation 3 #################
# Binary variable                                                  #
#g[,1]<-rbinom(n,1,.5)                                             #
# Continuous variable                                              #
#for (i in 2:numGene)                                              #
#{                                                                 #
#       g[,i]<-runif(n,0.0001, numGene/(2*i))                      #
#}                                                                 #
####################################################################

# generate responses
```

```
# this is for linear
beta<-seq(1,1,length=numCov)
# Let the first independent variables be important
delta<-c(1,1,1, rep(0,numGene-3))
# Coefficient for high prevelance level
gEff<-c(3, -2.5, 3.5, rep(0,length=numGene-3))*delta

######################## this is for non-linear ####################
#an interaction term                                               #
#inter<-10*cos(g[,1]*g[,2])                                        #
####################################################################

# Variance is 1.9^2
var<-diag(1.9^2,n,n)
################ this is for extended simulation 4 #################
#var<-diag(3.8^2,n,n)                                              #
####################################################################

cat("Data ",nr,"\n")
set.seed(30013*nr)
mean<-rep(0,n)
# Error term
        err<-t(rmnorm(1,mean,var))

#       this is for linear
        z<-x%*%beta+g%*%gEff+t(err)

######################## this is for non-linear ####################
#z<-x%*%beta+0.2*g%*%gEff+t(0.3*inter+err)                         #
####################################################################

#Make data matrix
        y<-rep(0,n)
        loca1<-which(z>0)
        y[loca1]<-1
        index<-rep(nr,n)

        dataSIMU<-cbind(index,y,x,g)
# write data out for data with high prevelance level from linear model
# write(t(data),paste0("HL300r",nr,".txt"),ncol=2+numGene+numCov)
}
```

```
dim(dataSIMU)



########################################Variable selection#################################################
# Details in Hongmei Zhang, Arnab Maity, Hasan Arshad, John Holloway, and Wilfried Karmaus. Variable#
# selection in semi-parametric models.Statistical methods in medical research,25(4):1736 1752, 2016.#
###########################################################################################################
library(mnormt)
library(msm)
library(MASS)
library(caret)
library(e1071)
library(randomForest)
library(pROC)
library(Matrix)

# Function VarSel() is used to select important variables in a reproducing kernel
# Input:
#       Y: response variable (nx1)
#       X: matrix of covariates (nxnumCov)
#       Candid: matrix of candidate variables (nxp)
#       numCov: number of covariates
#       rho: the scale parameter in the Gaussian kernel
#       tau: the overall set effect
#       numChain: number of MCMC chains
#       numItr: number of iterations per chain
#       sigBeta is the variance in the prior of beta

# Output: this includes three pieces
#       a)  quantiles obtained by treating (n - n1) x numChain matrix as samples
#         from posterior of the first beta, where n1 is the number of iterations after burn-in.
#     b) Gelman and Rubin's convergence measure R and an upper confidence
#         limit for R (values near 1 are desirable) for each of beta
#       c) model size
#       d) indices of selected variables
#       e) posterior probability of selecting each variable
#       f) ordered posterior probabilities of each variable

##The selection function
VarSel<-function(Y, X, Candid, numCov=1, rho=1, tau=0.8, numItr=100, numChain=2,sigBeta=10)
```

```
{
        set.seed(12345)
        sampsize<-length(Y)
        numCandid<-ncol(Candid)
        paramArray<-array(0,dim=c(numItr,numChain,numCov+1))

        # Gaussian kernel is used
        distance<-array(0,dim=c(numCandid,sampsize,sampsize))
        set.seed(12345)
        delta<-rbinom(numCandid,1,0.5)
Setbeta<-NULL
Setdelta<-NULL
        for (i in 1:(sampsize))
        {
                distance[,,i]<-t((Candid - matrix(1, ncol=1, nrow=sampsize)%*%(as.matrix(Candid[i,])))^2)
        }
        K<-updateK(delta,distance,sampsize,rho)
        error<-0
        for (i in 1:numChain)
        {
                beta<-rnorm(numCov)
                meanz<-rep(0,length(Y))
                varz<-diag(1,length(Y),length(Y))
                Z<-rmnorm(1,meanz,varz)
                delta<-rbinom(numCandid,1,0.5)
                K<-updateK(delta,distance,sampsize,rho)
                error<-0

                for (j in 1:numItr)
                {
                        cat("iter␣",j,"\n")
                        var<-tau*K+diag(-tau+1,nrow(K),ncol(K))
                        invVar<-try(solve(var))
                        if (class(invVar)=="try-error")
                        {
                                error<-1 &
                                break
                        }
                        else
                        {
                                invVar<-solve(var)
```

```
}

#update Z (latent normal)
for (jj in 1:length(X))
{
        rowvarnoi<-var[jj,]
        rowvarnoi<-rowvarnoi[-jj]

        rowinvVarnoi<-invVar[jj,]
        rowinvVarnoi<-rowinvVarnoi[-jj]
        invVarnoi<-invVar[-jj,-jj]-(rowinvVarnoi)%*%t(rowinvVarnoi)/invVar[jj,jj]

        meanZi<-rowvarnoi%*%invVarnoi%*%(Z[-jj]-X[-jj]%*%t(beta))+X[jj]%*%beta
        varZi<-1-rowvarnoi%*%invVarnoi%*%(rowvarnoi)
        if (varZi<0)
        {
                varZi<-10^(-40)
        }
        sdvarZi<-sqrt(varZi)
        if (Y[jj]==1)
        {
                if (sdvarZi<10^(-15) && meanZi<0)
                {
                        Z[jj]<-10^(-10)
                }
                else
                {
                        Z[jj]<-rtnorm(1,meanZi,sdvarZi,lower=0)
                }
        }
        else
        {
                if (sdvarZi<10^(-15) && meanZi>0)
                {
                        Z[jj]<-0
                }
                else
                {
                        Z[jj]<-rtnorm(1,meanZi,sdvarZi,upper=0)
                }
        }
```

```
                }

                #update beta
                Vbeta<-try(solve(diag(sigBeta^(-1),numCov,numCov)+t(X)%*%invVar%*%X))
                if (class(Vbeta)=="try-error")
                {
                        error<-1 &
                        break
                }
                else
                {
                        Vbeta<-solve(diag(sigBeta^(-1),numCov,numCov)+t(X)%*%invVar%*%X)
                        Mbeta<-Vbeta%*%(t(X)%*%invVar%*%Z)
                        beta<-rmnorm(1,Mbeta,Vbeta)
                }
                cat("beta",beta,"\n")
        Setbeta<-rbind(Setbeta,beta)
                paramArray[j,i,1:numCov]<-beta

                # update delta
                randIndex<-sample(numCandid,numCandid)
                for (kk in 1:numCandid)
                {
                        s<-randIndex[kk]
                        var<-tau*K+diag(-tau+1,nrow(K),ncol(K))

                        invVar<-try(solve(var))
                        if (class(invVar)=="try-error")
                        {
                                error<-1 &
                                break
                        }
                        else
                        {
                                invVar<-solve(var)
                        }

a<-0.5*log(det(invVar))-0.5*(t(Z-as.matrix(X)%*%beta)%*%invVar%*%(Z-as.matrix(X)%*%beta))

                        deltaTmp<-delta
                        deltaTmp[s]<-1-delta[s]
```

```
                                newK<-updateK(deltaTmp,distance,sampsize,rho)
                                newvar<-tau*newK+diag(-tau+1,nrow(newK),ncol(newK))

                                newinvVar<-try(solve(newvar))
                                if (class(newinvVar)=="try-error")
                                {
                                        error<-1 &
                                        break
                                }
                                else
                                {
                                        newinvVar<-solve(newvar)
                                }

b<-0.5*log(det(newinvVar))-0.5*(t(Z-as.matrix(X)%*%beta)%*%newinvVar%*%(Z-as.matrix(X)%*%beta))

                                maxab<-max(a,b)
                                if (a==Inf && b<Inf) {prob<-delta[s]}
                                else if (a<Inf && b==Inf) {prob<-1-delta[s]}
else if (a<Inf && b<Inf) {prob<-exp(a*delta[s]+b*(1-delta[s])-maxab)/(exp(a-maxab)+exp(b-maxab))}
                                else if (a==Inf && b==Inf) {prob<-0.5}


                                deltaTmp[s]<-rbinom(1,1,prob)
                                if (deltaTmp[s]!=delta[s])
                                {
                                        delta[s]<-deltaTmp[s]
                                        K<-updateK(deltaTmp,distance,sampsize,rho)
                                }
                        }
                        if (error==1) break
                        cat("delta␣",delta,"\n")
                Setdelta<-rbind(Setdelta,delta)
                        if (j==ceiling(numItr/2+1) && i==1)
                        {
                                sumDelta<-rep(0,length(delta))
                                kkk<-0
                        }
                        else if (j>ceiling(numItr/2+1) && i==1)
                        {
                                sumDelta<-sumDelta+delta
```

```
                                cat("sumDelta ",sumDelta,"\n")
                                kkk<-kkk+1
                        }
                }#end numItr
                if (error==1) break

                sumDelta<-sumDelta/kkk

                propDelta<-cbind(seq(1,numCandid),sumDelta)
                orderProp<-propDelta[order(propDelta[,2],decreasing=T),]
                maxDiff<-max(abs(diff(orderProp[,2])))
                VarIndex<-which(abs(diff(orderProp[,2]))==maxDiff)
                VarSelect<-orderProp[1:VarIndex,1]
                VarSelect<-VarSelect[order(VarSelect)]

                modelSize<-length(VarSelect)
        }#end chain
        diagnosis1<-gandr.conv(paramArray[,,1])
        list(diagB1=diagnosis1,modelSize=modelSize,propDelta=propDelta,orderProp=orderProp,
                VarSelect=VarSelect,K=K,Setbeta=Setbeta,Setdelta=Setdelta)
}

# calculate the kernel matrix (Gaussian kernel)
updateK<-function(delta, distance,sampsize,rho)
{
        K<-matrix(rep(0,sampsize*sampsize),nrow=sampsize)
        for (i in 1:sampsize)
        {
                temp<-(t(distance[,,i])%*%delta)/rho
                K[i,]<-exp(-temp)
        }
        return(K)
}

# The grandr.conv function is to obtain posterior inferences and convergence diagnosis.
# The method for convergence diagnosis is from Bayesian Data Analysis by Gelman, Carlin,
# Stern, and Rubin. 2003. Taylor & Francis.
# The codes are adapted from the program written by Professor Hal Stern.
# It takes two inputs:
#        n x m matrix of posterior draws for a single parameter of interest
#              (n = number of iterations per chain; m = number of chains)
```

```
#        n1 = number of iterations to ignore as transient
#             (defaults to 0.5*n if no number supplied)
# Output contains 3 pieces:
#      1. an approximate posterior 2.5,50,97.5% points (t-approximation)
#      2. quantiles obtained by treating (n - n1) x m matrix as samples
#         from posterior
#      3. Gelman and Rubin's convergence measure R and an upper confidence
#         limit for R (values near 1, say less than 1.1 are desirable)
#

gandr.conv<-function(r, n1 = nrow(r)/2) {
#
#  r: matrix of simulated sequences
#  n1: length of initial transient to ignore (default = 0.5)
#
        alpha <- 0.05   # 95% intervals
        m <- ncol(r)
        x <- r[(n1 + 1):nrow(r),  ]     # part of simulated sequences to process
        n <- nrow(x)    # We compute the following statistics:
#
#  xdot:  vector of sequence means
#  s2:  vector of sequence sample variances (dividing by n-1)
#  W = mean(s2):  within MS
#  B = n*var(xdot):  between MS.
#  muhat = mean(xdot):  grand mean; unbiased under strong stationarity
#  varW = var(s2)/m:  estimated sampling var of W
#  varB = B^2 * 2/(m+1):  estimated sampling var of B
#  covWB = (n/m)*(cov(s2,xdot^2) - 2*muhat*cov(s^2,xdot)):
#                                                 estimated sampling cov(W,B)
#  sig2hat = ((n-1)/n))*W + (1/n)*B:  estimate of sig2; unbiased under
#                                                 strong stationarity
#  quantiles:  emipirical quantiles from last half of simulated sequences
#
        xdot <- as.vector(col.means(x))
        s2 <- as.vector(col.vars(x))
        W <- mean(s2)
        B <- n * var(xdot)
        muhat <- mean(xdot)
        varW <- var(s2)/m
        varB <- (B^2 * 2)/(m - 1)
        covWB <- (n/m) * (cov(s2, xdot^2) - 2 * muhat * cov(s2, xdot))
```

```
        sig2hat <- ((n - 1) * W + B)/n
        quantiles <- quantile(as.vector(x), probs = c(0.025, 0.25, 0.5, 0.75,
                0.975))
        if(W > 1e-08) {
#
# non-degenerate case
# Posterior interval post.range combines all uncertainties
# in a t interval with center muhat, scale sqrt(postvar),
# and postvar.df degrees of freedom.
#
#       postvar = sig2hat + B/(mn):  variance for the posterior interval
#                                The B/(mn) term is there because of the
#                                sampling variance of muhat.
#       varpostvar:  estimated sampling variance of postvar
#
                postvar <- sig2hat + B/(m * n)
                varpostvar <- (((n - 1)^2) * varW + (1 + 1/m)^2 * varB +
                        2 * (n - 1) * (1 + 1/m) * covWB)/n^2
                post.df <- chisqdf(postvar, varpostvar)
                if (post.df<2) post.df=2.0001
                post.range <- muhat +
                              sqrt(postvar) * qt(1-alpha/2, post.df) * c(-1,0,1)
#
# Estimated potential scale reduction (that would be achieved by
# continuing simulations forever) has two components:  an estimate and
# an approx. 97.5% upper bound.
#
# confshrink = sqrt(postvar/W),
#     multiplied by sqrt(df/(df-2)) as an adjustment for the
#     width of the t-interval with df degrees of freedom.
#
# postvar/W = (n-1)/n + (1+1/m)(1/n)(B/W); we approximate the sampling dist.
# of (B/W) by an F distribution, with degrees of freedom estimated
# from the approximate chi-squared sampling dists for B and W.  (The
# F approximation assumes that the sampling dists of B and W are independent;
# if they are positively correlated, the approximation is conservative.)
#
                varlo.df <- chisqdf(W, varW)
                confshrink.range <- sqrt((c(postvar/W, (n - 1)/n + (1 + 1/m) *
                       (1/n) * (B/W) * qf(0.975, m - 1, varlo.df)) * post.df)/
                       (post.df - 2))
```

```
                list(quantiles = quantiles, confshrink =
                        confshrink.range)
        }
        else {
#
# degenerate case:  all entries in "data matrix" are identical
#
                list(post = muhat * c(1, 1, 1), quantiles = quantiles,
                        confshrink = c(1, 1)) }
}
#
# some functions needed by the above
#
col.vars<-function(mat) {
        means <- col.means(mat)
        col.means(mat * mat) - means * means }
#
col.means<-function(mat) {
        ones <- matrix(1, nrow = 1, ncol = nrow(mat))
        ones %*% mat/nrow(mat) }
#
cov<-function(a, b) {
        m <- length(a)
        ((mean((a - mean(a)) * (b - mean(b)))) * m)/(m - 1) }
#
chisqdf<-function(A, varA) {
        2 * (A^2/varA) }

###################################Proposed predictive model#####################################
#########################Training data and Test data#########################
#Read data from simulation
# For example
#dataOrig<- read.table("HL300r1.txt",header=F)
# Covariate
head(dataOrig)
X_all<-dataOrig[,3]
# Seed is defined by user. In my thesis, a loop with sign "a" is used to test 50 MC replicate,
# so my seed is set.seed(1000*a)
# Here, for one dataset, seed assigned to 1000 for example
set.seed(1000)
ind <- sample(2, nrow(dataOrig), replace = TRUE, prob=c(0.8, 0.2))
```

```
data<- dataOrig[ind == 1,]
data_test <-dataOrig[ind == 2,]
X<-as.matrix(X_all[ind == 1])
X_test<-as.matrix(X_all[ind == 2])
dim(X)
dim(data)

Y<-data[,2]

Candid<-data[,4:ncol(data)]
dim(Candid)
colnames(Candid)

# Variable selection
VarSelRst<-VarSel(Y,X,Candid,numCov=1,rho=1,tau=0.8,numItr=200,numChain=2,sigBeta=10)
# Selected variables
VarSelRst$VarSelect

##step a: Estimate effects of covariates
Sbeta<-median(VarSelRst$Setbeta)

##Step b: Estimate effects of important variables
# Kernel K0
rho=1
tau=0.8
sampsize<-length(data_test[,1])
Candid<-data_test[,4:ncol(data_test)]
numCandid<-ncol(Candid)
distance<-array(0,dim=c(numCandid,sampsize,sampsize))
set.seed(1000)
for (i in 1:(sampsize))
{
        distance[,,i]<-t((Candid - matrix(1, ncol=1, nrow=sampsize)%*%(as.matrix(Candid[i,])))^2)
}
delta<-rep(0,12)
delta[VarSelRst$VarSelect]<-rep(1,length(VarSelRst$VarSelect))
K<-updateK(delta,distance,sampsize,rho)

InverK<-try(solve(K,tol = 1e-100))
if (class(InverK)=="try-error")
                        {
```

```
                 Bay<-rbind(a,"singular")
                        }else{
                InverK<-solve(K,tol = 1e-100)

## Generate alpha
MND<-tau* InverK
mu<-c(rep.int(0, nrow(data_test)))
set.seed(seed=1234)
alpha<-matrix(,nrow=nrow(data_test),ncol=100)
ALA<-try(for ( i in 1: 100)
{
alpha[,i] <- mvrnorm(1, mu = mu, Sigma = MND,tol = 1e-100)
})
## MND needs to be positive definite
if (class(ALA)=="try-error")
                        {
class(MND)
NMND=as.matrix(nearPD(MND)$mat)
dim(NMND)
class(NMND)
set.seed(seed=1234)
for ( i in 1: 100)
{
alpha[,i] <- mvrnorm(1, mu = mu, Sigma = NMND,tol = 1e-100)
}
}else{
set.seed(seed=1234)
for ( i in 1: 100)
{
alpha[,i] <- mvrnorm(1, mu = mu, Sigma = MND,tol = 1e-100)
}
}

## Testing data with selected important variables
Varsel<-as.matrix(data_test[,c(VarSelRst$VarSelect+3)])
delta=1

## Kernel K(X0,X)
Rts<-NULL
RZ<-NULL
for (r in 1:nrow(Varsel))
```

```
   {
   g0<-data_test[r,c(VarSelRst$VarSelect+3)]
   g0measure<-NULL
       for (v in 1:nrow(Varsel))
       {
       int<-Varsel[v,]-g0
       g0measure<-rbind(g0measure,int)
       }
 dim(g0measure)
 sum<-g0measure*g0measure
temp<-0
for (e in ncol(sum))
{
 temp<-temp+sum[,e]
}
 Kg0<-exp(-temp)
 class(Kg0)

## estimated h
 h_m<-Kg0%*%alpha

## estimated g(.)
Sbeta<-median(VarSelRst$Setbeta)
g_m<-as.numeric(X_test[r,]%*%t(Sbeta))+h_m

##Step c: Decide the status for subjects
frequencies<- as.data.frame(table(g_m>0))
z<-ifelse (frequencies[which.max(frequencies[,2]),1]=="TRUE",z<-1,z<-0)
RZ<-rbind(RZ,z )
}
## Numerical assessment
#roc
Xroc=data.frame(pred=RZ,observed=data_test[,2])
colnames(Xroc)=c("pred","observed")
length(which(Xroc$observed==Xroc$pred))
RocX=roc(Xroc$observed, Xroc$pred,auc=TRUE)
xAUC=as.numeric(RocX$auc)

##accuracy
frequenciesfinal<- as.data.frame(table(RZ))
length(which(data_test[,2]==RZ))
```

```
accur<-as.data.frame(table(data_test[,2]-RZ))
accuracyB<-accur[which(accur[,1]==0),2]/nrow(data_test)
table(data_test[,2],RZ)

##Sensitivity
Test.Positive<- which(as.matrix(RZ)== "1")
Orig.Positive<-which(as.matrix(data_test[,2])== "1")
QuantiP<-length(intersect(Test.Positive,Orig.Positive))
p<-length(Orig.Positive)/length(data_test[,2])
SeB<-QuantiP/length(Orig.Positive)

##Specificity
Test.Negtive<- which(as.matrix(RZ)== "0")
Orig.Negtive<- which(data_test[,2]== "0")
QuantiN<-length(intersect(Test.Negtive,Orig.Negtive))
SpB<-QuantiN/length(Orig.Negtive)

ppvB<-SeB*p/(SeB*p+(1-SpB)*(1-p))
NPVB<-SpB*(1-p)/(SpB*(1-p)+(1-SeB)*p)

##Assemble results
Bay<-rbind(accuracyB,SeB,SpB,xAUC)
}


########################################Competing methods#######################################
# Selected variables
VarSelRst$VarSelect
head(dataOrig)
# Data with response, covariate, and selected important variables
DataSel<-dataOrig[,c(2,3,VarSelRst$VarSelect+3)]
head(DataSel)
# If no covariate
#DataSel<-dataOrig[,c(2,VarSelRst$VarSelect+3)]
# Split data
set.seed(1000)
ind <- sample(2, nrow(DataSel), replace = TRUE, prob=c(0.8, 0.2))
data<- DataSel[ind == 1,]
data_test <-DataSel[ind == 2,]
dim(data)
dim(data_test)
```

```
################Logistic regression##############
dependent=data[,1]
dependent=as.factor(dependent)
Candid<-data[,2:ncol(data)]
indep=as.data.frame(Candid)
TrainLR=as.data.frame(cbind(dependent,indep))
TrainLR$dependent=as.factor(TrainLR$dependent)
##Fit model
fmla <- as.formula(paste("dependent ~ ", paste(colnames(indep), collapse= "+")))
mylogit <- glm(fmla, data=TrainLR,family = "binomial",maxit=100)

## Prediction
dependents=data_test[,1]
indeps=as.data.frame(data_test[,2:ncol(data_test)])
Varsele=as.data.frame(cbind(dependent=dependents,Candid=indeps))
Varsele$dependent=as.factor(Varsele$dependent)
colnames(Varsele)=colnames(TrainLR)
fitted.results <- predict(mylogit ,newdata=Varsele,type='response')
fitted.resultss <- ifelse(fitted.results >= 0.5,1,0)
table(fitted.resultss )
tableLR=table(data_test[,1],fitted.resultss)

## Numerical assessment
#roc#
broc=data.frame(pred=fitted.resultss,actual=data_test[,1])
TRY=roc(broc$actual, broc$pred,auc=TRUE)
LRauc=as.numeric(TRY$auc)

#Accuracy
overtableLR=as.data.frame(table(data_test[,1]==fitted.resultss))
overlapLR=overtableLR[which(overtableLR[,1]=="TRUE"),2]
AccuracyLR=overlapLR/length(data_test[,1])

#Sensitivity
Test.PositiveLR<- which(as.matrix(fitted.resultss)== "1")
Orig.PositiveLR<-which(as.matrix(data_test[,1])== "1")
QuantiPLR<-length(intersect(Test.PositiveLR,Orig.PositiveLR))
pLR<-length(Orig.PositiveLR)/length(data_test[,1])
SeLR<-QuantiPLR/length(Orig.PositiveLR)
```

```
#Specificity
Test.NegtiveLR<- which(as.matrix(fitted.resultss)== "0")
Orig.NegtiveLR<- which(data_test[,1]== "0")
QuantiNLR<-length(intersect(Test.NegtiveLR,Orig.NegtiveLR))
SpLR<-QuantiNLR/length(Orig.NegtiveLR)

ppvLR<-SeLR*pLR/(SeLR*pLR+(1-SpLR)*(1-pLR))
NPVLR<-SpLR*(1-pLR)/(SpLR*(1-pLR)+(1-SeLR)*pLR)

##Assemble results
LRstatistics<-rbind(AccuracyLR,SeLR,SpLR,LRauc)



#########################Random Forest#########################
set.seed(1000)
ind <- sample(2, nrow(DataSel), replace = TRUE, prob=c(0.8, 0.2))
data<- DataSel[ind == 1,]
data_test <-DataSel[ind == 2,]
dim(data)
dim(data_test)
head(data)

Traindata<-data
Testdata<-data_test
Traindata[,1]<-as.factor(Traindata[,1])
Testdata[,1]<-as.factor(Testdata[,1])
colnames(Traindata)[1]="V2"
colnames(Testdata)[1]="V2"

###Training and prediction
set.seed(1000)
RF  <- randomForest(V2 ~ ., data=Traindata, importance=TRUE, proximity=TRUE)
predRF <- predict(RF , Testdata)

## Numerical assessment
table(predRF,Testdata[,1])
#roc#
RFroc=data.frame(pred=predRF,actual=Testdata[,1])
RFTRY = roc(as.integer(as.character(RFroc$actual)),
as.integer(as.character(RFroc$pred)),auc=TRUE)
RFauc=as.numeric(RFTRY$auc)
```

```
#Accuracy
overtableRF=as.data.frame(table(Testdata$V2==predRF ))
overlapRF=overtableRF[which(overtableRF[,1]=="TRUE"),2]
AccuracyRF=overlapRF/length(Testdata$V2)

#Sensitivity
Test.PositiveRF<- which(as.matrix(predRF)== "1")
Orig.PositiveRF<-which(as.matrix(Testdata$V2)== "1")
QuantiPRF<-length(intersect(Test.PositiveRF,Orig.PositiveRF))
pRF<-length(Orig.PositiveRF)/length(Testdata$V2)
SeRF<-QuantiPRF/length(Orig.PositiveRF)

#Specificity
Test.NegtiveRF<- which(as.matrix(predRF)== "0")
Orig.NegtiveRF<- which(Testdata$V2== "0")
QuantiNRF<-length(intersect(Test.NegtiveRF,Orig.NegtiveRF))
SpRF<-QuantiNRF/length(Orig.NegtiveRF)

ppvRF<-SeRF*pRF/(SeRF*pRF+(1-SpRF)*(1-pRF))
NPVRF<-SpRF*(1-pRF)/(SpRF*(1-pRF)+(1-SeRF)*pRF)

##Assemble results
BFstatistics<-rbind(AccuracyRF,SeRF,SpRF,RFauc)



########################----------SVM-------########################
set.seed(1000)
ind <- sample(2, nrow(DataSel), replace = TRUE, prob=c(0.8, 0.2))
data<- DataSel[ind == 1,]
data_test <-DataSel[ind == 2,]
dim(data)
dim(data_test)
head(data)

TraindataSVM<-data
TestdataSVM<-data_test
TraindataSVM[,1]<-as.factor(TraindataSVM[,1])
TestdataSVM[,1]<-as.factor(TestdataSVM[,1])
colnames(TraindataSVM)[1]="V2"
colnames(TestdataSVM)[1]="V2"
```

```
## svm, model fitting and prediction
set.seed(1000)
svm.model <- svm(V2 ~ ., data =TraindataSVM,probability=TRUE)
svm.pred <- predict(svm.model, TestdataSVM)

## Numerical assessment
table(svm.pred,Testdata[,1])
#roc#
SVMroc=data.frame(pred=svm.pred,actual=TestdataSVM[,1])
SVMTRY = roc(as.integer(as.character(SVMroc$actual)),
as.integer(as.character(SVMroc$pred)),auc=TRUE)
SVMauc=as.numeric(SVMTRY$auc)

#Accuracy
overtableSVM=as.data.frame(table(TestdataSVM$V2==svm.pred))
overlapSVM=overtableSVM[which(overtableSVM[,1]=="TRUE"),2]
AccuracySVM=overlapSVM/length(TestdataSVM$V2)

#Sensitivity
Test.PositiveSVM<- which(as.matrix(svm.pred)== "1")
Orig.PositiveSVM<-which(as.matrix(TestdataSVM$V2)== "1")
QuantiPSVM<-length(intersect(Test.PositiveSVM,Orig.PositiveSVM))
pSVM<-length(Orig.PositiveSVM)/length(TestdataSVM$V2)
SeSVM<-QuantiPSVM/length(Orig.PositiveSVM)

#Specificity
Test.NegtiveSVM<- which(as.matrix(svm.pred)== "0")
Orig.NegtiveSVM<- which(TestdataSVM$V2== "0")
QuantiNSVM<-length(intersect(Test.NegtiveSVM,Orig.NegtiveSVM))
SpSVM<-QuantiNSVM/length(Orig.NegtiveSVM)

ppvSVM<-SeSVM*pSVM/(SeSVM*pSVM+(1-SpSVM)*(1-pSVM))
NPVSVM<-SpSVM*(1-pSVM)/(SpSVM*(1-pSVM)+(1-SeSVM)*pSVM)

##Assemble results
SVMstatistics<-rbind(AccuracySVM,SeSVM,SpSVM,SVMauc)
```