BOOTSTRAP-ADJUSTED QUASI-LIKELIHOOD INFORMATION CRITERIA FOR MIXED MODEL SELECTION

Wentao Ge

A Dissertation

Submitted to the Graduate College of Bowling Green
State University in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2019

Committee:

Junfeng Shang, Advisor

Eric Worch,
Graduate Faculty Representative

Hanfeng Chen

John Chen

ABSTRACT

Junfeng Shang, Advisor

Model selection has received much attention and significantly developed in the recent decades. When statistical modeling is utilized to analyze the data set and make predictions, it is always natural to ask whether the candidate model fitted is a good model or not. This question is to be investigated and answered by the process of model selection. The challenge of model selection lies in the choice of a suitable selection criterion which sets the guideline to choose the most appropriate model among a set of possible candidate models. To some extent, the consistence and efficiency of the selection criterion determine the performance of the final model. When longitudinal data are involved, numerous selection criteria have been developed recently but few of them are both consistent and efficient. Furthermore, most of the existing selection criteria are based on the original sample alone. As a result, the uncertainty of the data has not been taken into consideration.

In this dissertation, to achieve consistent and efficient model selection performance and address the issue of uncertainty, we focus on the construction of two model selection criteria, named by QAICb1 and QAICb2, as modifications of Akaike information criterion (AIC, Akaike, 1973) based on the quasi-likelihood function and the bootstrap approach. Similar to some of the quasi-likelihood-based model selection criteria, quasi-likelihood independence criterion (QIC(R), Pan, 2001) for example, QAICb1 and QAICb2 consist of a log quasi-likelihood function and an estimation of bias correction term. We are able to make our two proposed criteria more widely applicable because the use of quasi-likelihood and bootstrap does not rely on distribution assumptions.

We utilize the quasi-likelihood to extend the Kullback-Leibler (K-L, Kullback and Leibler, 1951) discrepancy from which QAICb1 and QAICb2 are derived and proven to be asymptotically equivalent and consistent estimators. In addition to the linear mixed model, we extend QAICb1 and QAICb2 to generalized linear models with random effects because of the simplicity of constructing the corresponding quasi-likelihood functions.

To implement QAICb1 and QAICb2 in the context of the longitudinal data, we employ the generalized estimating equations (GEE, Pan, 2001b) to obtain the estimated parameters of candidate models. We first compute the quasi-likelihood component of the original sample using GEE with a pre-specified working correlation matrix. Then we estimate the bias correction terms of QAICb1 and QAICb2 using the bootstrap approach and GEE as well. By combining the calculated quasi-likelihood function and the estimation of the bias correction term, we compute the values of QAICb1 and QAICb2 for each of the candidate models, with which we are able to select the most appropriate model with the minimum.

Simulation studies are conducted using different settings including both the linear mixed models and generalized linear models with random effects under the non-parametric and semi-parametric bootstrap methods. The results demonstrate that both the two proposed model selection criteria are, in general, more consistent and efficient than existing model selection criteria across various scenarios. Two applications are given at the end of the dissertation to evaluate the model selection performance of QAICb1 and QAICb2 using data sets from clinical trials.

ACKNOWLEDGMENTS

Upon the completion my dissertation, I would like to appreciate everyone that has helped me during my 4-year Ph.D study. First and foremost, I would express my sincere gratitude to my advisor, Dr. Junfeng Shang. She led me into the research field and showed the beauty of statistics. Dr. Shang guided me with her passion, patience and valuable experience. She also provided me with sound advice on both my academic research and career development.

Next I would extend my gratitude to my other committee members: Dr. Hanfeng Chen, Dr. John Chen, and Dr. Rick Worch for their timely feedback and valuable suggestions to improve my dissertation work.

I am also grateful to all the staff and students in the Department of Math and Statistics. I have been offered a great learning environment to become a qualified researcher for the last 6 years. Moreover, I appreciate the opportunity to work as a graduate teaching associate, which has greatly benefited my Ph.D study.

Lastly, I would like to thank my parents for their consistent support across all these years. Special thanks should also go to my wife, Meiling Wu. Her understanding, encouragement and love helped me finally make this dissertation possible.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

CHAPTER 1 INTRODUCTION

1.1 Motivation of Dissertation

Model selection is one of the most important branches in the development of statistics. It refers to the procedure of selecting a single model from a set of candidate models based on the given date for statistical inference, including parameter estimation and predictions. It is natural to believe that there exists a true model that generates the data in the population level and captures all the information we have in hand. However, finding this correct model is too hard to achieve in reality because sometimes not all the covariates that characterize the population are identified so that the true model may not even be among the candidate models (Box, 1976). As a result, the final model chosen in the procedure of model selection is the best available model that can be relied on (Bancroft and Han, 1977, Müller *et al.*, 2013). Moreover, models are always estimated from the original sample as well as the selection procedure where the true model is not guaranteed to better than other candidate models. Therefore, prior to applying any selection criteria for model selection and statistical analysis, the attention must be paid to the philosophy of models (Burnham and Anderson, 2004). Given the point of view that all the models are approximations, the best model is chosen as the most suitable model of the given data under certain assumptions.

A proper model should have the ability to extract as much information as possible from the sample. In other words, a model is considered good if it is a well fit to the data and we could make reasonable inference about the population using this model (Lindley, 1981). However, blindly pursuing the goodness of fit is not practical and even makes the model useless (Anderson and Burnham, 2001). For example, in the case of a simple non-linear model, increasing the order of predictor variables also improves the fit to the original data. If there is a data set containing 100 data points, a polynomial with highest order of 99 is a perfect fit that goes through every point. While this "perfect" model, which is actually overfitted, does contain all the given information of the data set without any loss, it is useless to make predictions for new data points coming from

the same source because it unknowingly treats the effect of data variation as that from the model structure.

The appropriate model is not only a well fit to the original dataset but also easy to implement and interpret (McQuarrie and Tsai, 1998). There is a tendency in model selection that a complex model with a larger number of predictor variables are more capable of describing the data than a relatively simple model, which will cause the confusion that the more predictor variables we build the model with, the better results we may acquire. In the era of big data, there are numerous data sets with hundreds or thousands predictors. It is very common to run the model by including all the predictor variables regardless of whether some are redundant or not. Containing highly correlated variables will increase the difficulty of parameter estimation, reduce the efficiency of the model and lead to unreliable conclusions. Besides, fitting a complex model requires time, which may be affordable with the development of technologies but is not appealing in terms of statistical modeling because such a complicated model is difficult to make interpretations. Therefore, an ideal model should come with the ability to provide a good fit to the sample while maintaining a reasonable size. When it comes to model selection, the goal is to choose the most appropriate model among all the candidate models that are of various sizes and formats. The best model is the one balances the goodness of fit and simplicity, which are two eternal and antithetical topics in the fields of model selection.

During the process of model selection, the model selection criteria, established upon the philosophy of models, play the vital role to choose the final model out of all available models. They act like rules that rank each candidate model from important perspectives and set the guideline for model selection. For instance, we can estimate how far a candidate model is away from the true model and the model which minimizes such distance is taken as the most appropriate one. One of certain model selection criterion is the well-known Akaike information criterion (AIC, Akaike, 1973). The candidate model with the smallest AIC value is chosen, meaning that it is one closest to the true model. Furthermore, considering the Bayesian posterior probability, the Bayesian information criterion (BIC, Schwarz, 1978) offers the probabilities of candidate models being the

most appropriate model. In other words, BIC finds the final model that has the highest probability to be the true model. When the prediction ability of the model is of high interest, cross-validation (CV) evaluates each candidate model using mean square predictive error (MSPE) (Kohavi, 1995). It can be seen that most of the selection criteria assign scores to each candidate model, making the selection process easy to implement and understand.

Of all the model selection criteria, perhaps AIC is the most widely used across numerous research fields due to its simple form and high efficiency. Furthermore, AIC, originated from the information theory, assesses the model exactly through the two important aspects as the goodness of fit and simplicity, which corresponds to the two components of AIC: a likelihood function and an estimation of the bias correction term.

The likelihood or log-likelihood function of a candidate model is used to gain information from the sample and indicates how well the model fits the data set. The bias correction term comes from the feature that this item compensates for the bias from the likelihood function to make AIC an unbiased estimator of the distance between the candidate model and the true model. The bias correction term estimation of AIC directly is associated with the dimension of the corresponding model. As the amount of knowledge obtained from the data set is always limited and similar for different candidate models, the addition of the bias correction term will make a noticeable distinction for the AIC values among all the candidate models. In other words, by penalizing the models with more predictors and assigning higher weights for smaller ones, AIC tries to reach the balance between the data fit and model simplicity.

Unfortunately, each coin has two sides and AIC is likely to choose more complex models rather than simpler ones especially in the small sample scenario due to its simple form of the bias correction term (Hurvish and Tsai, 1989). As discussed, a model with more predictor variables would be a better fit than others, which means the corresponding value of the likelihood function tends to dominate the corresponding AIC value. If the best model is relatively simpler among the candidate models, making the right move to select the most appropriate model will not be easy. As a result, modifications based on the estimation of the bias correction term are proposed to increase

the selection efficiency of AIC. For example, AICc (Sugiura, 1978) was proposed as an extension of AIC by including the sample size into the bias correction term and it brings down the chance of overfitting when the sample size is small. However, AICc requires strong assumption of candidate models that narrows its application to univariate models. More importantly, when dealing with correlated date fitted by quasi-likelihood functions, AIC-based model selection criterion should be extended accordingly.

Model selection has been through remarkable progress over the years in various scientific fields. When it comes to the area such as sociology and medical research, it involves models characterizing correlated data sets. For example, the longitudinal data take the form of repeated measurements on individuals over a period of time and the observations within the same individual are correlated with each other even though different individuals are independent with each other. To handle the correlated data sets, a typical linear model which only measures random noise in addition to the predictor variables, can no longer be utilized.

The linear mixed model (Laird and Ware, 1982) was introduced to fit the correlated data by introducing new parameters to capture and estimate the correlation structure so there are two different kinds of parameters in a mixed model: fixed effects and random effects. The fixed effects are taken directly from the linear model to measure the features at the population level where the corresponding data points should be independent with each other. The random effects, on the other hand, measure the individual level characteristics. As for the longitudinal data, the covariates that distinguish different subjects are considered to be the fixed effects while the correlation within the repeated measurements is estimated by the random effect.

To estimate both the fixed and random effects, the linear mixed model usually utilizes the maximum likelihood estimation (MLE) or restricted maximized likelihood (REML) where both cases the likelihood functions of the data are to be computed (Thompson, 1962). Unlike the independent data sets, the addition of random effects brings the estimation of within individual correlation, which makes the computation of the corresponding likelihood complex especially when the correlation is large. Furthermore, the usage of likelihood-based approaches requires the assumption

of normal distributions for the random effects as well as the error term. The MLE or REML will provide unbiased fixed effects estimations even though the normal assumption is violated. However, when the random effects are of high interest, the normal assumption should be seriously considered. Besides, we are not able to estimate the corresponding correlation structure and confidence intervals without the normal assumption. It is also likely that the correlation structure of the data is complex and could not be fully captured by the random effects. Based on these concerns, other than the likelihood function, an alternative estimation approach that depends less on the distribution assumptions should be considered to deal with correlated data sets.

The quasi-likelihood function (Wedderburn 1974) was introduced to take care of the situation where the distribution assumption is not guaranteed and a complex correlation is involved. The quasi-likelihood shares the similar properties of the traditional likelihood functions but can be defined as long as the first two moments of the distribution for the data set are specified. In other words, the quasi-likelihood is constructed by the mean and variance of the response variable. One advantage of using quasi-likelihood is the lack of distribution assumptions so it is more flexible for model fitting under different kinds of data sets. Secondly, with the introduction of an over-dispersion parameter, the quasi-likelihood function is capable of reducing the effect brought by over-dispersion. In the case when the variance is a known function of the mean such as the Poisson model, the issue of over-dispersion is very likely to occur because the variance is not estimated as a free parameter independent of the mean. When there is no concern over-dispersion like the normal model, the corresponding over-dispersion parameter estimation can be simply ignored.

When quasi-likelihood functions are applied to correlated data, the method of generalized estimating equations (GEE) (Liang and Zeger, 1986) is commonly used to estimate the model parameters. The GEE employs a working correlation matrix to measure the correlation structure. The working correlation matrix has a pre-specified form whose parameters are estimated empirically from the data set. The GEE approach has the advantage that under the circumstances where the true correlation is misspecified, the model parameter estimation is consistent as well as the as well as the corresponding variance covariance estimator, called the sandwich estimator (White,

1980). Ideally, when the working correlation correctly captures the correlation structure of the data, GEE provides the most efficient estimators. Another attractive aspect of GEE is the free of distribution assumptions by taking advantage of the quasi-likelihood function. Therefore, without correctly specifying the distribution of the data, the GEE procedure provide consistent and efficient estimators.

The model selection criteria that originated from the likelihood functions such as AIC or BIC will not be applicable if quasi-likelihood functions are involved due to the lack of distribution assumptions. By mimicking the construction of AIC and modifying the K-L discrepancy (Kullback and Leibler, 1951) using the quasi-likelihood function, QIC(R) (Pan, 2001) was proposed as a model selection criterion using GEE estimators. QIC(R) utilizes an independence assumption when computing the quasi-likelihood function, which assumes the data points are independently distributed, and exhibits high efficiency when the correlation structure of the data is not complicated.

There are two main drawbacks of the model selection criteria QIC(R). The first comes from the derivation of QIC(R) where one term is dropped due to the difficulty of estimation. Although simulation studies were given to support that the efficiency of QIC(R) was not greatly influenced (Pan, 2001), a certain amount of information would be lost because of this simplification. The second is the use of the independence assumption. As an AIC-type variable selection criterion, QIC(R) consists of a quasi-likelihood term and the estimation of the bias correction term as well. The quasi-likelihood part is computed through independent correlation matrix and one of the sandwich variance estimators is also updated through GEE using independent working correlation. This makes QIC(R) efficient when the true correlation of the data is relatively small but can not guarantee the performance of QIC(R) in larger correlation situations. In such case, this selection criterion tends to underestimate the corresponding QIC(R) values for relatively larger candidate models. As a result, a candidate model with more predictor variables is more likely to be selected by QIC(R) as the final model. Following is a simulation study showing how the selection performance of QIC(R) would be affected by different true correlation structures.

Table 1.1: Selection results of QIC(R) with different correlation matrix.

Correlation	M1	M2	M3	TRUE	M5	M6	M7	M8
0.2	0	0	0	602	142	103	78	75
0.4	0	0	0	560	143	95	90	112
0.6	0	0	0	543	129	99	104	125
0.8	0	0	0	469	134	115	129	153

Consider a set of 8 candidate models with first one being $Y = X_1$, second one being $Y = X_1 + X_2$, ..., and the last one being $Y = X_1 + X_2 + ... + X_7 + X_8$. To simplify the notations, vectors are used and the error term is not included in the model setting. Let the fourth model $Y = X_1 + X_2 + X_3 + X_4$ be the true model. Here all of the X's are independently generated from a standard normal distribution and Y is constructed by the way that the within subject correlation structure is exchangeable (EX) with parameter 0.2, 0.4, 0.6 and 0.8. The larger the parameter, the higher the correlation. The sample size is 50 and the results are based on 1000 iterations.

From Table 1.1 we can see that the true model selection rates of QIC(R) reduces noticeably as the within subject correlation increases. When the correlation is 0.2 or 0.4 that is close to the independent structure, QIC(R) is relatively more efficient while in the case of 0.6, there is roughly 54% probability we could reach the correct conclusion. When the correlation continues raising to 0.8, there is a small chance we are able to find the best model and QIC(R) tends to select the most complicated model. Therefore, a more efficient and consistent selection criteria other than QIC(R) needs to be developed.

To improve the model selection quality, resampling approaches can be incorporated, of which the most influential technique is bootstrap (Efron, 1982, 1983, 1992) method. A typical bootstrap approach takes three forms: parametric bootstrap, semi-parametric bootstrap and non-parametric bootstrap, of with the non-parametric bootstrap is more widely used. During a non-parametric bootstrap procedure, resamples are generated by sampling with replacement from the original data set and each resample is treated as the original one to perform analysis on. By averaging estimations from all the bootstrap samples, it provides a more accurate result than that based on a single sample(Davison *et al.*, 1986). Similar to the quasi-likelihood function, the non-parametric

bootstrap method is free from the parametric distribution of the data by using the bootstrap distribution (Efron, 1992). The only assumption needed is that all of the data points are independent and identically distributed. To achieve certain accuracy of bootstrap estimators, a reasonable large number of resamples should be considered (Efron, 1992), which was a concern when it was first proposed. Nowadays, with the growing power of computational technologies, obtaining resamples for moderate or large sample sizes is not time consuming even with a personal computer.

The bootstrap method can be easily applied to improve the model selection efficiency. Rao and Tibshirani (1997) proposed an out-of-bootstrap method for model selection based on observations that are left out of bootstrap samples. The construction of bootstrap weights (Buckland, Burnham and Augustin, 1997) is a combination of selection criteria and bootstrap, which also takes the uncertainty of model selection into consideration. AIC could be improved by absorbing the bootstrap method because of its nature to be an estimator of the Kullback-Leibler discrepancy. Ishiguro and Sakamoto (1991) proposed WIC through bootstrap followed by a successful application to a practical problem in Ishiguro, Morita and Ishiguro (1991). Ishiguro, M., Sakamoto, Y., and Kitagawa, G (1997) proposed the extended information criterion (EIC) to extend the usage of AIC by estimating the bias correction term based on the bootstrap resamples. Cavanaugh and Shumway (1997) proposed AICb, which gave a different estimation of the penalty term by bootstrapping.The bootstrap-based WIC and AICb were adjusted to be asymptotically equivalent by Shibata (1997) under certain conditions, which were limited to the independent data. When it comes to the mixed model with dependent data, Shang and Cavanaugh (2008) brought out two bootstrap-based selection criteria, AICb1 and AICb2, which are efficient especially in the small sample scenarios. Unfortunately, these bootstrap modifications rely on the likelihood functions of the data sets which furthermore depend on the distribution assumption of the data. It will be attractive to combine the quasi-likelihood and the bootstrap method for a more effective selection criterion.

1.2 Objectives of Dissertation

The main objective of this dissertation is to propose two model selection criteria that are capable of efficiently identifying the best model from a pool of candidate models. To be more specific,

our selection criterion are constructed based on the quasi-likelihood function and the bootstrap method for correlated data. It is a modification of AICb1 and AICb2 (Shang and Cavanaugh, 2008) by including the quasi-likelihood functions and GEE estimators. Our proposed model selection criteria could overcome the disadvantages of some existing criteria in terms of longitudinal data and mixed model selection.

Our first goal is to extend AICb1 and AICb2 using quasi-likelihood functions under linear mixed models. We name our two proposed bootstrap-adjusted information criteria based on the quasi-likelihood function QAICb1 and QAICb2. Similar to AICb1 and AICb2, there are two components in our QAICb1 and QAICb2: a quasi-likelihood function and an estimation of the bias correction term. The quasi-likelihood part consists of estimators given by the GEE approach using the original sample with a pre-specified working correlation matrix. The bias correction term is estimated from the combination of the original sample and quasi-likelihood functions obtained from the bootstrap samples.

Secondly, we focus on the selection efficiency and computation cost when implementing QAICb1 and QAICb2. To obtain the quasi-likelihood function of the data set, we assume that all the data points, whether they are at population level or individual level, are independent with each other. The use of independence assumption when computing the quasi-likelihood makes QIC(R) easy to be implemented but not effective if the correlation structure of the data is far from independence. However, our proposed selection criterion, by taking advantaging of the bootstrap method, are more consistent than QIC(R) throughout different correlation structures even though the independent assumption is implanted as well. Furthermore, within each bootstrap sample, we update the fitted model with the independent working correlation to get the "independently updated" quasi-likelihood functions. In this way, we are able to increase the selection efficiency of the two proposed criteria and acquire more consistent performance.

The third objective is to widen the scope of use of the QAICb1 and QAICb2 by expanding them to generalized linear models with random effects including the logistic or Poisson mixed models. As there is a tendency to encounter over-dispersion for these models, we have to include the over-

dispersion parameter into both the estimation procedure and the computation of the corresponding quasi-likelihood functions. Due to the simple form of the quasi-likelihood, it is convenient to add the over-dispersion parameter without increasing the amount of calculation.

Theoretical results are provided to address the validity of our selection approaches with linear mixed model as well as the generalized linear models with random effects. We also establish asymptotic properties to prove the consistence of the two criteria. Furthermore, various numerical studies are conducted to evaluate the performance of the proposed selection criteria via comparing the selection outcomes with those of some other existing selection approaches including the QIC(R). We explore the efficiency our selection criteria over a sequence of candidate models under different correlation structures and several correlation parameters. The influence of the sample size is investigated as well. Along with the simulation studies, two applications with real data sets are demonstrated.

1.3 Outline of Dissertation

Chapter 1 provides the introduction to this dissertation research as well as the motivation and objectives.

Chapters 2 and 3 give the background materials that are fundamental to the rest of the dissertation. The beginning of chapter 2 includes the framework of quasi-likelihood functions for both the independent and dependent observations with corresponding parameters estimation. We will talk about the GEE estimating procedure in detail, which serves as the parameter estimation approach in our simulation studies in Chapter 6.

In Chapter 3, we will present the model settings of both the linear mixed model and generalized linear model with random effects that will be used throughout the remaining of the dissertation. The quasi-likelihood function with its extension to mixed models are discussed as well. Moreover, we also include the parameters estimation of certain models using the GEE method.

In Chapter 4, the model selection process is emphasized and several commonly used model selection criteria in the linear models are discussed. We bring the introduction of the K-L discrepancy which would be extended using quasi-likelihood functions in the next chapter. In addition,

we will introduce several model selection criteria for linear mixed models.

Chapter 5 can be divided into three parts. The first part focuses on some existing model selection criteria based on bootstrap or quasi-likelihood functions. We will also provide the definition and the common resampling approaches of the bootstrap within this part. In the second part, we propose our bootstrap-adjusted quasi-likelihood information criteria QAICb1 and QAICb2 based on the extended K-L discrepancy. Lastly, we will prove the two proposed QAICb1 and QAICb2 are equivalent and consistent estimators of the corresponding K-L discrepancy based on the quasi-likelihood function for the linear mixed model and generalized linear model as well.

Chapter 6 mainly provides simulation studies to demonstrate the efficiency of QAICb1 and QAICb2. Before going deep into the simulations, we discuss the potential drawback of the raw versions of the proposed criteria, which suggests a modification on the estimation of bias correction terms when it comes to implementations. The simulation results are compared with two of the existing model selection criteria QIC(R) and QIC$_u(R)$ in terms of the GEE estimation and longitudinal data. During the simulations, both the linear mixed model and generalized linear model with random effects are considered. Moreover, the non-parametric and semi-parametric bootstrap are utilized as well.

In Chapter 7, we will provide two applications to evaluate the model selection performance of our proposed QAICb1 and QAICb2.

Chapter 8 summarizes the performance of our proposed model selection criterion and includes the discussion of possible improvement in the future.

CHAPTER 2 QUASI-LIKELIHOOD FUNCTIONS

2.1 Introduction

One traditional way of performing parameter estimation is to utilize the maximum likelihood estimation (MLE) where we have to derive the likelihood function by fully specifying the distribution of the data. However, due to the lack of information of the data distribution, it is sometimes difficult to obtain the likelihood function. More importantly, if the corresponding distribution assumption is violated, the estimated parameters will not be trustworthy. To address this issue and broaden the use of the likelihood function for more practical circumstances, the quasi-likelihood methodology should be taken advantage of.

The quasi-likelihood function was first introduced by Wedderburn (1974) when fitting regression models without specifying the distribution of the data but by the relationship between the mean and the variance of the observations. Moreover, the quasi-likelihood function could be easily derived when the corresponding variance of the observations is connected to the mean by a known function and a possibly over-dispersion or nuisance parameter. McCullagh (1983) extended Wedderburn's definition of the quasi-likelihood function and showed that under mild regularity assumptions, the quasi-likelihood function perform very similar to the likelihood function. In addition to the distribution free property, the quasi-likelihood function would also be able to capture the over-dispersion originated from the data and certain models (McCullagh and Nelder, 1989). For example, in a Poisson or binomial model, the mean and the variance are dependent, under which circumstance the estimation of the over-dispersion parameter needs to be carefully considered (McCullagh, 1983).

In this chapter, we will provide the definition of the quasi-likelihood function as well as its properties for both independent and dependent observations. Moreover, different parameter estimation approaches are discussed including the quasi-score estimating equations.

2.2 Quasi-likelihood with Independent Observations

2.2.1 Construction and Properties of Quasi-likelihood

Let the observations $y_i (i = 1, ..., n)$ be independent with expectation $E(y_i) = \mu_i$ and variance $Var(y_i) = \phi v_i(\mu_i)$, where ϕ, may or may not be known, is the nuisance or over-dispersion parameter and $v_i(\cdot)$ is a known function only dependent on the μ_i. Moreover, the parameters of interest $\beta_j (j = 1, ..., k)$ and μ_i are also connected using a known function associated with $x_{ij} (j = 1, ..., k)$. For example, when the normal model is used, $\mu_i = \sum_{j=1}^{k} x_{ij}\beta_j$ and when the logistic model is applied, $\text{logit}(\mu_i) = \sum_{j=1}^{k} x_{ij}\beta_j$. In most cases, the $v_i(\cdot)(i = 1, ..., n)$ are chosen to have the same form, say $v(\cdot)$, regardless of the different values resulted from different μ_i (McCullagh and Nelder, 1989). So in the rest of this chapter, the notation $v(\cdot)$ or $V(\cdot)$ will be used for the function connecting the mean and the variance of the data.

For each observation y_i, Wedderburn (1974) defined the quasi-likelihood or log quasi-likelihood function $Q(\mu_i, \phi; y_i)$ using partial differential equation as

$$\frac{\partial Q(\mu_i, \phi; y_i)}{\partial \mu_i} = \frac{y_i - \mu_i}{\phi v(\mu_i)},$$

which, by integration and ignoring the constant term, implies the $Q(\mu_i, \phi; y_i)$ to be

$$Q(\mu_i, \phi; y_i) = \int_{y_i}^{\mu_i} \frac{y_i - t}{\phi v(t)} dt. \tag{2.1}$$

Wedderburn (1974) also showed that the quasi-likelihood function $Q(\mu_i, \phi; y_i)$, for each obser-

vation $i = 1, .., n$ and $j, p, q = 1, ..., k$, has the following properties:

$$
\begin{aligned}
&E\{\frac{\partial Q(\mu_i, \phi; y_i)}{\partial \mu_i}\} = 0 \\
&E\{\frac{\partial Q(\mu_i, \phi; y_i)}{\partial \beta_j}\} = 0 \\
&\text{Var}\{\frac{\partial Q(\mu_i, \phi; y_i)}{\partial \mu_i}\} = \frac{1}{\phi v(\mu_i)} \\
&E\{\frac{\partial Q(\mu_i, \phi; y_i)}{\partial \mu_i}\}^2 = -E\{\frac{\partial^2 Q(\mu_i, \phi; y_i)}{\partial \mu_i^2}\} = \frac{1}{\phi v(\mu_i)} \\
&E\{\frac{\partial Q(\mu_i, \phi; y_i)}{\partial \beta_p}\frac{\partial Q(\mu_i, \phi; y_i)}{\partial \beta_q}\} = -E\{\frac{\partial^2 Q(\mu_i, \phi; y_i)}{\partial \beta_p \partial \beta_q}\} \\
&\qquad = \frac{1}{\phi v(\mu_i)}\frac{\partial Q(\mu_i, \phi; y_i)}{\partial \beta_p}\frac{\partial Q(\mu_i, \phi; y_i)}{\partial \beta_q}.
\end{aligned}
\tag{2.2}
$$

It can be seen from the first two properties in equation (2.2) that both the traditional likelihood function and the quasi-likelihood function share the same property that the corresponding first derivative is 0. Moreover, based on the properties presented in equation (2.2), most of the first-order results of the likelihood are also shown in the quasi-likelihood case. So if the integrated quasi-likelihood function (2.1) exists, it should perform much like the likelihood function under mild conditions (McCullagh and Nelder, 1989). In fact, under the assumption that all the observations are independent, McCullagh (1983) gave several explicit solutions of the quasi-likelihood (2.1) and pointed out that if the distribution is in the exponential family and the log likelihood function is linear in y, then the likelihood and the quasi-likelihood function have the same form except for the over-dispersion parameter ϕ. For instance, in the cases where normal models are used when there exists no over-dispersion parameter, the likelihood and quasi-likelihood functions are identical.

Let Y be the vector consisting of all the responses with expectation μ, ϕ be the over-dispersion parameter and $V(\mu) = Var(Y)$, a function of μ, be the variance covariance matrix. The log quasi-likelihood function of the entire data set is defined through the following partial differential equation:

$$\frac{\partial Q(\mu, \phi; Y)}{\partial \mu} = \frac{Y - \mu}{\phi V(\mu)},$$

where $V(\mu)$ is a diagonal matrix with $V(\mu) = \text{diag}\{v(\mu_1), ..., v(\mu_n)\}$ because all the observations are independent, which also implies that the quasi-likelihood $Q(\mu, \phi; Y)$ can be written as the summation of the quasi-likelihood function for each individual observation y_i:

$$Q(\mu, \phi; Y) = \sum_{i=1}^{n} Q(\mu_i, \phi; y_i).$$

2.2.2 Parameters Estimation

As been stated in Section 2.2.1, under the circumstances that linear models or generalized linear models are used, μ is a function of the parameter β as $\mu(\beta)$. By introducing the term $D = \frac{\partial \mu}{\partial \beta'}$, the quasi-score function could be written as a function of β by

$$U(\beta) = \frac{\partial Q(\mu, \phi; Y)}{\partial \beta} = D' V^{-1}(\mu)(Y - \mu)/\phi, \tag{2.3}$$

where the elements of D, D_{ij} , are the derivatives of $\mu(\beta)$ with respect to β, denoted by $\partial \mu(\beta)/\partial \beta$. The equation $U(\beta) = 0$ is called the independent estimating equations.

Recall that in the method of MLE, the corresponding estimator is derived by solving the equations that the first derivative of the log likelihood functions equals 0. We can make an analogy to the MLE for the quasi-likelihood functions by solving the independent estimating equation to obtain the estimated parameter $\hat{\beta}$. However, unlike the MLE that it has explicit solutions in most cases, iterative approaches such as the Newton-Raphson method or weighted least squares should be utilized to find the $\hat{\beta}$ of the quasi-likelihood (McCullagh and Nelder, 1989).

To apply the Newton-Raphson method to obtain the estimator $\hat{\beta}$, first is to construct the variance covariance matrix of $U(\beta)$, $\boldsymbol{i}_\beta$ as

$$\boldsymbol{i}_\beta = D' V^{-1} D/\phi,$$

which is parallel to the Fisher information for the traditional likelihood function. Meanwhile, $\boldsymbol{i}_\beta$ is also the negative expected second order derivative of the quasi-likelihood $Q(\mu, \phi; Y)$. Asymptoti-

cally, the variance covariance matrix $\text{cov}(\hat{\beta})$ can be approximated by the reciprocal of $\boldsymbol{i}_\beta$ by

$$\text{Cov}(\hat{\beta}) \approx \boldsymbol{i}_{\hat{\beta}}^{-1} = \phi(D'V^{-1}D)^{-1}$$

under usual limiting conditions over the eigenvalues of $\boldsymbol{i}_\beta$ (McCullagh and Nelder, 1989).

By choosing a starting value $\hat{\beta}^{(0)}$ in the neighborhood of $\hat{\beta}$, the generated sequence of estimated parameters using the Newton-Raphson method starts with

$$\hat{\beta}^{(1)} = \hat{\beta}^{(0)} + (\hat{D}_0'\hat{V}_0^{-1}\hat{D}_0)^{-1}\hat{D}_0'\hat{V}_0^{-1}(Y - \hat{\mu}^{(0)}) \tag{2.4}$$

until the process converges. Note that the estimating of β is independent of the over-dispersion parameter ϕ which is cancelled in the equation (2.4). Under certain conditions, the estimated $\hat{\beta}$ also has the properties of approximate unbiasedness and asymptotic normality as the followings:

1. $E(\hat{\beta} - \beta) = O(n^{-1})$.
2. $n^{-1}U_\beta \sim \mathcal{N}_k(\mathbf{0}, \phi\boldsymbol{i}_\beta/n + O_k(n^{-\frac{1}{2}}))$.
3. $n^{\frac{1}{2}}(\hat{\beta} - \beta) \sim \mathcal{N}_k(\mathbf{0}, n\phi\boldsymbol{i}_\beta^{-1} + O_k(n^{-\frac{1}{2}}))$.

The $\mathcal{N}_k$ represents the k-variate normal distribution. Based on the above properties we see that the estimator from the quasi-likelihood behaves very similar to the one from the likelihood function.

When the over-dispersion parameter ϕ is not known, the corresponding estimation is based on the Pearson's chi-square statistic (McCullagh, 1983). Let $\hat{\mu}$ be the fitted values of μ and N be the total number of observations, the naive estimate $\hat{\phi}$ of the ϕ is denoted by

$$\hat{\phi}^{-1} = \frac{(Y - \hat{\mu})'V^{-1}(\hat{\mu})(Y - \hat{\mu})}{N - p} = \frac{\chi^2}{N - p} \tag{2.5}$$

and the above numerator χ^2 is the generalized Pearson's statistic.

McCullagh (1983) also pointed out that the naive estimator of ϕ could be improved if the distribution of the data comes from the exponential family, in which case the quasi-likelihood and

the log likelihood would have the same form. Besides, the maximum likelihood estimator of ϕ is a function of two log likelihood functions derived from the unrestricted parameter space and the candidate model. This difference, also known as the deviance, can be expressed as

$$d(Y, \hat{\mu}) = 2Q(Y; Y) - 2Q(\hat{\mu}; Y). \tag{2.6}$$

Then the $\hat{\phi}$ could be calculated by setting the deviance (2.6) equal to the its corresponding expectation. Note that the estimator $\hat{\phi}$ is usually different from the one obtained by the MLE. However, this modification is usually not preferred because of the difficulty in computing the expectation as well as the in consistency of the estimator.

2.3 Quasi-likelihood with Dependent Observations

2.3.1 Construction of Quasi-likelihood Function

Under the circumstances that the observations are no longer independent, for example the longitudinal data, a different form of the variance covariance matrix of Y should be considered. Let $\text{cov}(Y) = \phi V(\mu)$, where $V(\mu)$ is not a diagonal but a positive-definite matrix with elements $v_{ij}(\mu)$ and $i, j = 1, ..., n$. The quasi-score equation (2.3), with components $U_r(\beta)$, has the following properties which are parallel to the ones in the independent case:

1. $E(U_r(\beta)) = 0$.
2. $\text{Cov}(U(\beta)) = D'V^{-1}D/\phi = \boldsymbol{i}_\beta$.
3. $-E(\frac{\partial U_r(\beta)}{\partial \beta_s}) = D'V^{-1}D/\phi$.

for $r, s = 1, ...k$. The estimate $\hat{\beta}$, a root by solving the corresponding quasi-score equation, satisfying the property that

$$U(\hat{\beta}) = \hat{D}'\hat{V}^{-1}(Y - \hat{\mu}) = 0.$$

Under certain limiting conditions, $\hat{\beta}$ is an asymptotic unbiased estimator of β as well as approximately normal distributed with the limiting variance

$$\text{Cov}(\beta) \approx \phi(D'V^{-1}D)^{-1} = \boldsymbol{i}_{\beta}^{-1}.$$

Note that the exact conditions used to establish the consistency and asymptotic normality of the estimator β are vary complicated (McCullagh and Nelder, 1989). However, it is necessary to believe that when $n \to \infty$, the $U(\beta)$ is asymptotically normally distributed.

2.3.2 Parameters Estimation via Independent Estimating Equation

The optimal estimating function is one way to obtain parameter estimations under the quasi-likelihood functions. The main difference between the quasi-likelihood for independent and dependent observations lies in the variance covariance matrix of Y. When it comes to the longitudinal data where repeated measurements are usually made on the same subject, the variance matrix is block-diagonal, which means the correlation only exists within the same subject not between different subjects. As an analogy, the independent estimating equations constructed using the score function (2.3) can be applied in this case with the assumption that all the observations are independent. Apparently this would bring unreliable estimators when the within-subject correlation is so large that can not be ignored. However, the estimator obtained from the independent estimating equations, denoted by $\hat{\beta}_I$, is easy to compute and fairly consistent given the correct specification of the regression model. Besides, the $\hat{\beta}_I$ could be reasonably efficient under several simple scenarios (Liang and Zeger, 1986). To address the inefficiency of the independent estimating equations under correlated observations, a more commonly used way to estimate β is the generalized estimating equations (GEE) proposed by Liang and Zeger (1986). In the next section, the GEE approach and the corresponding properties of the estimators under the longitudinal data will be discussed.

2.3.3 Generalized Estimating Equation (GEE) and Working Correlation Matrix

The GEE, an extension of the generalized mixed models, combines the quasi-likelihood and the longitudinal data in a way that it would provide consistent estimators for regression covari-

ates and the corresponding variance components (White 1980). The estimating procedure of the GEE relies on a working correlation structure $R(\rho)$ to measure the correlation within the repeated measurements.

The use of the working correlation, which determines the shape of the true correlation structure, is very similar to putting assumptions on the variance covariance matrix of the random effects in the mixed-effects model. Besides greatly simplifying the calculations, it will estimate the parameters consistently even if the true correlation matrix is misspecified. That being said, only a little or no information of the correlation structure is needed to implement the GEE approach, which corresponds to one advantage of the quasi-likelihood that no distribution assumption is placed. When the true correlation is correctly identified, the GEE estimators gain the most efficiency in this ideal situation. During simulation studies in Chapter 6, it is demonstrated that our proposed model selection criteria, by implanting the GEE estimators, perform consistently across different scenarios. With the working correlation structure, the GEE utilizes the marginal model and focuses on the population averages.

There are several kinds of working correlation matrices suggested by Liang and Zeger (1986). IN is the independent working correlation matrix with $R(\rho) = I$. The independent estimating equations is a special case of the GEE because of the usage of the IN working correlation. EX is the exchangeable or compound symmetric working correlation with $R_{ij}(\rho) = \rho$ for $(i \neq j)$, where any pair of the repeated measurements within the same group have the same correlation. AR is the first order auto-regressive working correlation with $R_{ij}(\rho) = \rho^{|i-j|}$ for $(i \neq j)$, which applies to the situation where the correlation for the grouped responses decays as time goes on. UN is the unstructured working correlation with $R_{ij}(\rho) = \rho_{ij}$ $(i \neq j)$. Of all these working correlation structures, the EX or AR correlation matrix is more commonly used. Thanks to the consistency of the GEE procedure, the estimators are reliable as long as the working correlation is not too far away from the true correlation structure.

2.3.4 Parameters Estimation via GEE

The GEE and the quasi-likelihood functions are closely related in the way of the correlation matrix. We can borrow the model setting partly from that of the linear mixed model. Let $Y = (y_1, ..., y_n)$ be the vector of all the observations with $E(Y) = \mu$ where y_i represents the vector of the responses from the ith subject. For each response vector y_i, define A_i to be an $m_i \times m_i$ diagonal matrix with $v(\mu_{ij})$ as the jth diagonal element and $R_i(\rho)$ to be the $m_i \times m_i$ working correlation matrix for the ith group. Then the variance covariance matrix cov(y_i) for each subject i could be defined as

$$V_i = \phi A_i^{\frac{1}{2}} R_i(\rho) A_i^{\frac{1}{2}}. \tag{2.7}$$

Note that the V_i is not exactly the same as the variance covariance matrix used in the definition of the quasi-likelihood functions in Section 2.2.1 because it is a function of two parameters μ and ρ instead of just the μ. More specifically, the matrix V_i should be expressed as $V_i(\mu, \rho)$ but in order to simplify and be consistent with the notations from previous sections, we use V_i instead. When replacing the estimator $\hat{\rho}$ of ρ into the $V(\mu, \rho)$, it will reduce to the normal covariance matrix.

By utilizing the covariance matrix (2.7) and referring to the independent estimating equations, the generalized estimating equations (GEE, Liang and Zeger, 1986) are defined through

$$U(\beta, \rho) = \sum_{i=1}^{n} U_i(\beta, \rho) = \sum_{i=1}^{n} D_i' V_i^{-1} (y_i - \mu_i) = 0 \tag{2.8}$$

and $U_i(\beta, \rho)$ is the quasi-score function of the response vector y_i. By solving the equation (2.8), the resulting $\hat{\beta}_G$ is called the GEE estimator of β. Note that when $R(\rho)$ in the covariance matrix (2.7) is set to be the independent working correlation matrix I, the GEE equation (2.8) will reduce to the independent estimating equations.

2.3.4.1 Estimation of β

The quasi-score equation (2.8) has three parameters β, ρ and ϕ so in general there exist no closed forms for using the single equation along, which means a iterative method is preferred.

From Liang and Zeger (1986), if the correlation coefficient ρ of the working correlation matrix $R(\rho)$ in the variance covariance matrix (2.7) is replaced by a $\sqrt{n}$-consistent estimator of ρ, denoted by $\hat{\rho}(Y, \beta, \phi)$, the equation (2.8) could be written as a function only consisting the parameter β. Here the $\hat{\rho}(Y, \beta, \phi)$ is obtained when both the β and ϕ are known and satisfies that $\sqrt{n}(\hat{\phi} - \phi) = \mathcal{O}_p(1)$. Furthermore, the over-dispersion parameter ϕ is estimated by $\hat{\phi}(Y, \beta)$, which is also a $\sqrt{n}$-consistent estimator when β is known. Therefore, the quasi-score equation (2.8) could be re-expressed as

$$\sum_{i=1}^{n} U_i(\beta, \hat{\rho}(\beta)) = 0 \tag{2.9}$$

and the solution of the above equation (2.9) is the GEE estimator of β, denoted by $\hat{\beta}_G$. Under mild regularity conditions, the $\hat{\beta}_G$ has the following asymptotic properties:

1. $\hat{\rho}$ is $\sqrt{n}$-consistent given β and ϕ.

2. $\hat{\phi}$ is $\sqrt{n}$-consistent given β.

3. $|\frac{\partial \hat{\rho}(\beta,\phi)}{\partial \phi}| \leq H(Y, \beta)$, which is $\mathcal{O}_p(1)$.

Indicated by the above properties of $\hat{\beta}_G$, $\sqrt{n}(\hat{\beta}_G - \beta)$ is asymptotically multivariate Normal with mean zero and variance covariance matrix V_G defined by

$$V_G = \lim_{n\to\infty} n(\sum_{i=1}^{n} D_i' V_i^{-1} D_i)^{-1} \{\sum_{i=1}^{n} D_i' V_i^{-1} \text{Cov}(y_i) V_i^{-1} D_i\} (\sum_{i=1}^{n} D_i' V_i^{-1} D_i)^{-1}.$$

By replacing the $\text{Cov}(y_i)$ with $\{y_i - \mu_i(\hat{\beta}_G)\}\{y_i - \mu_i(\hat{\beta}_G)\}'$ and substituting β, ρ and ϕ with the corresponding $\sqrt{n}$-consistent estimators using the fitted model, the estimator of V_G, denoted by $\hat{V}_G$, is also consistent, which is known as the sandwich or robust variance covariance estimator (White, 1980).

Similar to the estimation of β in the dependent scenario, $\hat{\beta}_G$ is estimated iteratively using the Gauss-Newton method. Within each iteration t, given the current estimates $\hat{\rho}$ and $\hat{\phi}$, the following

iterative procedure for β is suggested by Liang and Zeger (1986):

$$\hat{\beta}_G^{(t+1)} = \hat{\beta}_G^{(t)} - \{\sum_{i=1}^{n} D_i'(\hat{\beta}_G^{(t)})\hat{V}_i^{-1}(\hat{\beta}_G^{(t)})D_i(\hat{\beta}_G^{(t)})\}^{-1}\{\sum_{i=1}^{n} D_i'(\hat{\beta}_G^{(t)})\hat{V}_i^{-1}(\hat{\beta}_G^{(t)})(y_i - X_i\hat{\beta}_G^{(t)})\}\,,$$

where $\hat{V}_i^{-1}(\beta) = V_i\{\beta, \hat{\rho}(\beta, \hat{\phi}(\beta))\}$. This procedure can also be treated as a modification of Fisher's scoring method. Besides, under certain transformations, the procedure of calculating $\hat{\beta}_G$ is equivalent to that of the iteratively reweighted linear regression.

2.3.4.2 Estimation of ρ and ϕ

As been discussed before, during each iteration, the estimation of ϕ depends on the current estimate of $\hat{\beta}$ and based on the estimated $\hat{\phi}$, we could obtain the estimate of ρ. The $\hat{\phi}$ is also calculated using the Pearson statistic defined by (2.5) as

$$\hat{\phi}^{-1} = \sum_{i=1}^{n} \frac{(y_i - X_i\hat{\beta})'v^{-1}(X_i\hat{\beta})(y_i - X_i\hat{\beta})}{N - p} \tag{2.10}$$

with N being the total number of observations. Please note that the $v(\cdot)$ is function used to connect the mean and the variance of the response vector y_i in section 2.2.1 and it is not the V_i that obtained through the working correlation matrix. Liang and Zeger (1986) showed that $\hat{\phi}$ is a consistent estimator of ϕ given the fourth moments of Y exist and are finite.

The estimation of the ρ depends on the choice of the working correlation structure given the $\hat{\phi}$ by (2.10). Let $\hat{r}_{ij} = (y_{ij} - x_{ij}\hat{\beta})/v(x_{ij}\hat{\beta})$ for $i = 1, ..., n$ and $j = 1, ..., m$, then a general way to estimate ρ is to use the following simple function

$$\hat{R}_{uw} = \sum_{i=1}^{n} \frac{\hat{r}_{iu}\hat{r}_{iw}}{N - p}\,,$$

where $\hat{R}_{uw}$ is the uwth element of the estimated working correlation matrix $\hat{R}$. Generally speaking, $\hat{\rho}$ is estimated from the Pearson residuals by utilizing the data from all the groups.

CHAPTER 3 MIXED MODELS AND ESTIMATION

During medical and public health researches, data are often formed by repeated measurements over the same individual during different time points in the experiment. This type of data is called the longitudinal data. The longitudinal data have complex correlation structures due to the fact that the observations within the same unit are correlated to some extent. The traditional linear model can only deal with data that the points are independent with each other while the linear mixed or mixed-effects model (Laird and Ware, 1982) were developed for the correlated data.

The linear mixed model differs from the linear model by introducing the random effects to deal with the correlation structure. There are two kinds of covariates in a typical linear mixed model, the fixed effects and the random effects. The fixed effects are population level covariates, which are the same as the ones in the linear model. However, the random effects are group level covariates and they are assumed to have same correlation structure across different groups. The addition of the random effects also makes the linear mixed model more complex in both the model fitting and parameter estimation.

To fit a linear mixed model requires the likelihood function obtained from the fix and random effects but this process will be challenging when there exists a complex correlation structure. Moreover, likelihood-based approaches depend heavily on distribution of the data, in which case if the distribution assumption is violated, the estimated model will not be reliable. Therefore, we are motivated to connect linear mixed models and quasi-likelihood functions. As mentioned in Chapter 2, the quasi-likelihood functions are valid as long as the first and second moments of the distribution of the data could be specified. This will give us more flexibility when constructing linear mixed models.

When quasi-likelihood functions are introduced, the generalized estimating equations (GEE) (Liang and Zeger, 1986) could be utilized as an alternative to deal with the mixed models. In this chapter, we will first provide the model setting for the linear mixed model in terms of the quasi-likelihood function and then discuss the GEE estimation of the linear mixed model. In addition,

we will cover the GEE estimation of generalized linear models with random effects, which is an extension of the linear mixed model by including a link function.

3.1 Linear Mixed Models and Estimation

3.1.1 Linear Mixed Models

Let N be the number of total observations with $N = \sum_{i=1}^{n} n_i$, the linear mixed model for n clusters takes the form of

$$y_i = X_i\beta + Z_i b_i + \epsilon_i, \quad i = 1, ..., n\ , \tag{3.1}$$

where i is the index for the clusters; β is a $(p+1) \times 1$ vector of fixed coefficients; b_i is a $q \times 1$ vector of random effects following a normal distribution with $E(b_i) = 0$ and $Var(b_i) = \Delta$; ϵ_i is a normally distributed $n_i \times 1$ vector of error term with $E(\epsilon_i) = 0$ and $Var(\epsilon_i) = \sigma^2 I_{n_i}$; y_i is the response vector; X_i and Z_i are the covariate matrices for the fixed and random effects. The matrix Δ is positive definite and I_{n_i} is an $n_i \times n_i$ identity matrix. The corresponding matrix notation can be expressed as

$$Y = X\beta + Zb + \epsilon. \tag{3.2}$$

In equation (3.2), $Y = (y_1^{'}, ..., y_n^{'})^{'}$ is an $N \times 1$ vector of all the responses; $b = (b_1^{'}, ..., b_n^{'})$ is an $nq \times 1$ vector for random effects; $\epsilon = \epsilon_1^{'}, ..., \epsilon_n^{'}$ is an $N \times 1$ error vector. $X = (X_1^{'}, ..., X_n^{'})^{'}$ is a $N \times (p+1)$ matrix and assumed to be full rank; Z is a $N \times nq$ block diagonal matrix with diagonal elements $Z_1^{'}, ..., Z_n^{'}$.

3.1.2 Parameter Estimation via Quasi-likelihood

Note that a typical linear mixed model requires the random effects follow a multivariate normal distribution with mean $\mathbf{0}$ and variance covariance matrix Δ. As the quasi-likelihood functions only require the first two moments of distribution, more flexibility is given when it comes to the distribution of the random effects. So with quasi-likelihood functions, when the mean and variance of Y are specified, the corresponding quasi-score equation could be constructed to obtain the corre-

sponding parameter estimations. Furthermore, even when the variance matrix of Y is not correctly specified (Zeger, Liang and Albert, 1988), we would obtain consistent estimators of β based on a working correlation matrix, which is to be discussed shortly.

To further simplify the notations, it is assumed that $n_i = m$ for all $i = 1, ..., n$ and $Var(\epsilon_i) = \sigma^2 I$, where I is a $m \times m$ identity matrix. In a linear mixed model, let $\mu_i = E(y_i) = X_i\beta$ and $V_i = Var(y_i) = Z_i \Delta Z_i' + Var(\epsilon_i) = Z_i \Delta Z_i' + \sigma^2 I$, then the response vector y_i has mean μ_i and variance covariance matrix V_i. Thus the log quasi-likelihood $Q(\mu_i; y_i)$ is defined through the following differential equation:

$$\frac{\partial Q(\mu_i; y_i)}{\partial \mu_i} = V_i^{-1}(y_i - \mu_i). \tag{3.3}$$

Note that $D_i = \partial\mu/\partial\beta_i = X_i$, the equation (3.3) could be written in terms of β by:

$$\frac{\partial Q(\mu_i; y_i)}{\partial \beta} = D_i' V_i^{-1}(y_i - X_i\beta). \tag{3.4}$$

Moreover, the estimated $\hat{\beta}$ can be achieved by solving the follow quasi-score equation

$$U(\beta) = \sum_{i=1}^{n} D_i' V_i^{-1}(y_i - X_i\beta) = 0. \tag{3.5}$$

Note that

$$E(U(\beta)) = \sum_{i=1}^{n} D_i' V_i^{-1}(E(y_i) - X_i\beta) = 0,$$

which means when the first moment of the distribution is correctly specified, the root of the equation (3.5), $\hat{\beta}$, is consistent. Moreover, the robust variance estimate of $\hat{\beta}$ given by White (1982),

$$V_{\hat{\beta}} = (\sum_{i=1}^{n} X_i' V_i^{-1} X_i)^{-1} \{\sum_{i=1}^{n} X_i' V_i^{-1}(y_i - X_i\hat{\beta})(y_i - X_i\hat{\beta})' V_i^{-1} X_i\} (\sum_{i=1}^{n} X_i' V_i^{-1} X_i)^{-1},$$

is consistent as well provided that $E(y_i) = X_i\beta$ (Zeger, Liang and Albert, 1988). Please note that we do not include the over-dispersion parameter when constructing the quasi-likelihood function

like we have done in Chapter 2. As the mean and variance of Y are independently distributed, the parameter estimation of the linear mixed model will not encounter over-dispersion.

3.2 Generalized Linear Models with Random Effects and Estimation

3.2.1 Generalized Linear Models with Random Effects

Follow the notations in Section 3.1.1, a linear model is defined by

$$y_i = X_i'\beta + \epsilon_i. \tag{3.6}$$

As an extension, the generalized linear model distinguishes from the linear model (3.6) by the use of a link function $h(\cdot)$ through

$$h(\mu_i) = X_i'\beta \tag{3.7}$$

for $i = 1, ..., n$ so that

$$E(y_i) = \mu_i = h^{-1}(X_i'\beta) \quad \text{and} \quad \text{Var}(y_i|b_i) = \phi \cdot v(\mu_i).$$

The term $E(y_i)$ is also considered as the marginal expectation of the response vector. It can be seen that when the link function $h(\cdot)$ is the identity link, the generalized linear model reduces to the linear mixed model. When $h(\cdot)$ is the logit link, the link function (3.7) defines the logistic regression model. Moreover, when $h(\cdot)$ is the natural logarithmic (log) function, the link function (3.7) defines the Poisson regression model.

The generalized linear model with random effects is very commonly used when it comes to the longitudinal data with repeated measurements, in which case the effect of model parameters is directly revealed on the population averages when changing the covariates of individuals. As a result, the response vectors y_i are independent with each other for different values of i while the values y_{ij} within each vector y_i are highly correlated. The correlation within each vector y_i is measured by a correlation matrix without specifying the random effects like we did in linear mixed models. With quasi-likelihood functions, this correlation is estimated by the variance covariance

matrix V_i as defined in equation 3.3. Therefore, no estimation of random effects will be included during the quasi-likelihood parameter estimation procedure.

3.2.2 Parameter Estimation using GEE

We could utilize the generalized estimating equations (GEE) discussed in Section 2.3.4 to estimate the unknown parameter β (Zeger *at al.*, 1988). Let $R_i(\rho)$ be the working correlation matrix for the ith individual and $\phi \cdot A_i$ be the variance covariance of y_i, then β can be estimated by solving the following quasi-score equation

$$U(\beta) = \sum_{i=1}^{n} \frac{\partial \mu_i'}{\partial \beta} V_i^{-1}(\rho)(y_i - \mu_i) = 0, \tag{3.8}$$

where $V_i^{-1}(\rho) = A_i^{\frac{1}{2}} R_i(\rho) A_i^{\frac{1}{2}}$. In most cases, the $R_i(\rho)$ is chosen to be the same for all the individuals. In this case, the corresponding robust variance estimate of $\hat{\beta}_G$ can be calculated by

$$V_{\hat{\beta}_G} = M_0^{-1} M_1 M_0^{-1},$$

where

$$M_0 = \sum_{i=1}^{n} \frac{\partial \hat{\mu}_i'}{\partial \beta} \hat{V}_i^{-1} \frac{\partial \hat{\mu}_i}{\partial \beta}$$

and

$$M_0 = \sum_{i=1}^{n} \frac{\partial \hat{\mu}_i'}{\partial \beta} \hat{V}_i^{-1} \frac{\partial \hat{\mu}_i}{\partial \beta}$$

and

$$M_1 = \sum_{i=1}^{n} \frac{\partial \hat{\mu}_i'}{\partial \beta} \hat{V}_i^{-1} (y_i - \hat{\mu}_i)(y_i - \hat{\mu}_i)' \hat{V}_i^{-1} \frac{\partial \hat{\mu}_i}{\partial \beta}.$$

CHAPTER 4 METHODOLOGIES OF MODEL SELECTION

Model selection comes naturally when a most appropriate model should be chosen from a set of candidate models. Ideally, the goal of the model selection is always to find the true model that generates the data. However, the true model is difficult to identify in reality and does not necessarily exist especially when the covariates are not fully specified. Taking this into consideration, when there are some candidate models to work with, the process of model selection helps to find the best one among these models. In other words, the model selection procedure offers the most appropriate model to the sample for parameter estimation and predictions.

Over the years a large number of selection methods have been proposed for either the linear model or the linear mixed model. At the beginning of this chapter, the subset selection methods are introduced, which is based on the candidate models formed by the combination of all the possible predictor variables. Then traditional model selection criteria as well as those involving quasi-likelihood and bootstrap approach are discussed to provide us with the knowledge needed to propose our modified model selection criteria in the next chapter.

4.1 Subset and Stepwise Selection

4.1.1 Best Subset Selection

Model selection is sometimes called the subset selection because the final model is always a subset of the largest model containing all the possible covariates. One common model selection approach, in terms of computation and implementation, is the best subset selection that evaluates all the possible models. Suppose there are k predictor variables of interest, we will be able to form up to 2^k possible models by either adding or dropping a covariate and these candidate models contain covariates numbered from 0 to k. Then all these models are fit using the dataset and compared using certain model selection criterion, based on which the final model is chosen.

One advantage of the best subset selection is the applicability to all kinds of models and various model selection criterion. Unfortunately, the procedure of the subset selection is simple despite

that it is an exhaustive searching algorithm so the corresponding efficiency decreases dramatically when the number of covariates k increases. In the case of only 10 covariates, for example, there will be 1024 possible candidate models to choose from during subset selection, not to mention the number of covariates in the era of big data when the number of predictor variables is increasing incredibly. Thus the best subset selection is only recommended when a small number of covariates are involved.

4.1.2 Stepwise Model Selection

There are basically three stepwise model selection approaches: forward selection, backward elimination and bidirectional elimination.

The forward selection criterion begins with the model with no inputs and the predictor variables are gradually placed into the previous candidate model step by step. At one step, all the variables outside the current model are evaluated by being separately included into the current one. The variable, which is the most influential to the existing model, would be remained and the process keeps on until no other variables are important according to certain threshold. The backward selection/elimination, starting with full model, is on the opposite side of the forward selection. During the backward process, the covariate with the least influence is dropped from the current model one by one until the final model has all the significant covariates.

Both the forward and backward selection criteria would produce $\frac{k(k+1)}{2} + 1$ candidate models, which is more efficient than the best subset selection especially when k is large. However, it is worth noting that, by reducing the computational cost for not performing exhaustive searching, the best model is not guaranteed to be achieved in these two selection methods.

The bidirectional elimination can be treated as the integration of the forward and backward selection methods. It's essentially the same the forward selection by adding the most important variable to the current while retaining the possibility of dropping a variable that is of no importance. This approach is preferable to the other two as it's more likely to produce the best model while not raising the cost of computation.

One main issue of the stepwise method is the tendency to get overfitted models so that the

goal of parsimony is not reached (Roecher, 1991). Even though this drawback could be overcome by applying more strict rules when adding or dropping a variable, by doing so, some predictors carrying useful information would be excluded from the last model. Furthermore, due to the repeat use of the same data set during each step, biased results are provided (Copas, 1983), which will greatly increase the significance of the final model. To address the disadvantages of the stepwise approaches, more efficient selection criterion are to be discussed.

4.2 Model Selection Criteria

As mentioned in Section 4.1.1, a model selection criterion is of huge importance by serving as a measure to determine the best model among all the candidate models. In fact, different model selection criteria will assess the models from different perspectives in the way that different criteria would produce different best models. The choice of the selection criteria is affected by the model to be fitted, the size of the dataset, the purpose of the model fitting and so on. In this section, several model selection criteria are provided based on different scenarios.

4.2.1 Kullback-Leibler Discrepancy

Model selection criteria will capture some characteristic between the candidate model and the true model, which coincides with idea that the final model from model selection is the one that is closest to the true model. We start this section by introducing the K-L discrepancy which measures the distance between the true model and the candidate models.

To distinguish the parameters and the corresponding estimators involved with the likelihood function and quasi-likelihood function, we will use θ as the notation for parameter when likelihood function is used. Let θ_0 and θ be the parameters of the true model and a candidate model respectively. The K-L discrepancy (Kullback and Leibler, 1951) between this candidate model and the true model is defined as

$$d(\theta, \theta_0) = E_0\{-2 \log L(\theta|Y)\}, \tag{4.1}$$

where E_0 denotes the expectation taken under the true model and $L(\theta|Y)$ represents the likelihood

function of the candidate model.

Suppose $\hat{\theta}$ is set of the parameter estimators with the likelihood function of the corresponding model to be $L(\hat{\theta}|Y)$. The overall K-L discrepancy is written as

$$d(\hat{\theta}, \theta_0) = E_0\{-2 \log L(\theta|Y)\}|_{\theta=\hat{\theta}}, \tag{4.2}$$

which provides a measure of separation between the true model and the fitted candidate model.

The expected overall K-L discrepancy, with k being the dimension of the $\hat{\theta}$, is denoted by

$$\Delta(k, \theta_0) = E_0\{d(\hat{\theta}, \theta_0)\} = E_0\{E_0\{-2 \log L(\theta|Y)\}|_{\theta=\hat{\theta}}\}. \tag{4.3}$$

It serves as a statistic when it comes to the procedure of model selection. Moreover, the overall K-L discrepancy (4.3) can be decomposed as

$$\begin{aligned}\Delta(k, \theta_0) &= E_0\{E_0\{-2 \log L(\theta|Y)\}|_{\theta=\hat{\theta}}\} \\ &= E_0\{-2 \log L(\hat{\theta}|Y)\} \\ &\quad + [E_0\{E_0\{-2 \log L(\theta|Y)\}|_{\theta=\hat{\theta}}\} - E_0\{-2 \log L(\hat{\theta}|Y)].\end{aligned} \tag{4.4}$$

It should be noted that evaluating both the equations (4.2) and (4.3) is not possible in reality because the information of the true model is never known. Akaike (1973) pointed out that the likelihood function $-2 \log L(\hat{\theta}|Y)$ of the fitted model is a biased estimator of the equation (4.2) and the bias correction (4.4) could be asymptotically approximated by $2k$, which forms the AIC that will be talked over shortly in Section 4.2.2.

4.2.2 Akaike Information Criterion (AIC)

One of the most commonly used model selection criteria is the Akaike (1973) information criterion, which is also known as the AIC. The AIC, a likelihood-based criteria, is defined as

$$\text{AIC} = -2\log(L(\hat{\theta}|Y)) + 2k, \tag{4.5}$$

where $L(\hat{\theta}|Y)$ is the likelihood function, $\hat{\theta}$ is the maximum likelihood estimator and k is number of estimated covariates of the candidate model. $2k$ is called the penalty part or the bias correction term of AIC. Asymptotically, AIC is an unbiased estimator of the Kullback-Leibler discrepancy between the true model that generates the date and the candidate model used as an approximation. The model with smaller KL discrepancy is considered to be better so the final model is the one which has the smallest AIC value. It was noted by Sakamoto, Ishiguro, and Kitagawa (1986) that "AIC is not a criterion for the estimation of the true order but the one for the best model fit.". Moreover, the complexity of the model increases as the dimension of the model increases. (Buckland, Burnham and Augustin, 1997). Therefore the best model selected by AIC is the best approximating model to the given data set.

AIC works best if the sample size is large and candidate models have relatively small number of parameters. It was discussed by Hervich and Tsi (1989) that AIC would substantially underestimated the expected overall K-L discrepancy (4.3) due to the fact that the bias correction term $2k$ will be much larger than the bias (4.4) when the sample size is small. Therefore candidate models with relatively large k may have smaller AIC values than expected. In other words, the AIC is more likely to select the model with a larger number of estimated covariates as the the best model. To overcome this disadvantage, the corrected version of AIC, AICc was proposed by Siguira (1978) as

$$\text{AICc} = AIC + \frac{2k(k+1)}{N-k-1}. \tag{4.6}$$

When the sample size is large, the additional penalty added to AIC is negligible so that the AICc is asymptotically equivalent as AIC. The AICc has been extended by Hurvich and Tsai(1989), who also showed that the performance of AICc would be significantly better than that of the AIC in the small sample case. However, due to the fact that AICc is based on the specific family of candidate models, it is less widely used than the original AIC.

4.2.3 Bayesian Information Criterion (BIC)

Schwarz (1978) proposed the Bayesian information criterion, known as the BIC, from the Bayesian perspective. The BIC is defined as

$$\text{BIC} = -2\log(L(\hat{\theta}|Y)) + \log(N)k \tag{4.7}$$

with N being the number of total observations in the dataset. The BIC is similarly defined as AIC but it is based on the Bayes estimators whose leading term is the maximum likelihood estimator when the sample size N is large. Asymptotically, the BIC value of each candidate model can be viewed as the posterior probability of that model being the best model under certain choice of the prior distributions. In contrast to AIC, BIC assumes the existence of a low-dimension true model is consistent to the data dimension. (Schwarz, 1978). For example, in the case of the sample size N being 8 or more, the BIC penalizes more than the AIC does, which leads to the fact that the BIC tend not to choose models with a large K and is more consistent in the large sample situation. As a result, the consistence of BIC highly relies on the large sample assumption.

4.2.4 Cross-Validation

Cross-validation (CV) is used to account for the prediction error of candidate models and to some extent, reduce the chance of model overfitting. The general idea of K-fold CV is to randomly divide the original dataset into K different subsets, then use any $K-1$ of the partitions to fit the models and the rest one to evaluate the predictive performance of the model. After repeating this process for K times, the results are combined and averaged and the candidate model with the lowest predictive error is chosen as the most appropriate model. Besides the prediction ability of the candidate models, the cross-validation can also be used to estimate the bias correction (4.4) of the K-L discrepancy.

When compared to the bootstrap approach which will be discussed in the following Section 5.1.1, CV is not efficient when estimating the error rate although both of the two methods requires much more computation than the traditional ways (Efron, 1983). Another shortage of the CV is

the requirement of a relatively large dataset to work on. With a small dataset, the random partition of the original data provides insufficient data points, which would result in unstable or biased estimation of model parameters. Furthermore, the computation cost of CV will be high when the number of subsets K is small and the sample size is large.

CHAPTER 5 BOOTSTRAP-ADJUSTED QUASI-LIKELIHOOD INFORMATION CRITERIA FOR MIXED MODELS SELECTION

5.1 Selection Criteria Based on Bootstrap and Quasi-likelihood

5.1.1 Overview of the Bootstrap Selection Procedures

Bootstrap (Efron, 1992) is a resampling technique known for producing more accurate estimators by averaging estimators calculated from bootstrap samples. It is easy to implement and does not dependent on distribution assumptions of the data. By utilizing the empirical distribution to approximate the sampling distribution, the only assumption needed for bootstrap is that the data points are independently and identically distributed.

The bootstrap approach takes various ways when it comes to implementation. Besides the non-parametric and parametric bootstrap we have mentioned in Section 1.1, there is also a semi-parametric suggested by Efron(1992). We will discuss the three bootstrapping methods shortly in detail under the setting of linear mixed models.

Recall that for mixed models, the data is in the form of $(X_i, Z_i; y_i), i = 1, ..., n$, where i is the index at individual level, X_i and Z_i are matrices corresponding to the fixed and random effects and y_i is the response vector with size n_i. The total number of observations is $N = \sum_{i=1}^{n} n_i$ with each n_i is the observation number for the cluster i. For simplicity, we assume that all the groups are equal with same number of data points, that is, $n_i = m$ for all $i = 1, ..., n$. The desired number of bootstrap samples is B.

5.1.1.1 Non-parametric Bootstrap

As discussed in Section 1.1, a non-parametric bootstrap begins by repeatedly sampling from the given data until the size of the resample reaches the original sample size and each resample is analyzed the same as the original sample. By repeating this process for a sufficient number of times, the parameter of interest could be estimated through the bootstrap estimators from all the bootstrap resamples.

If there exit correlated observations within the data set, the non-parametric bootstrap should be done at the individual or group level to avoid violating the corresponding assumption of bootstrap. When longitudinal data is considered, for instance, the resampling is conducted over all the subjects including the corresponding repeated measurements. Therefore, the generated resamples of the correlated data are indexes for each subject and the actual resample values $(X_i^*, Z_i^*; y_i^*)$ are the date values in the original sample corresponding to indexes.

5.1.1.2 Parametric Bootstrap

When carrying out the parametric bootstrap method, a parametric model should be specified to generate the resamples $(X_i^*, Z_i^*; y_i^*)$. The equation (3.1.1) denotes a common linear mixed model setting:

$$y_i = X_i\beta + Z_i b_i + \epsilon_i, \quad i = 1, ..., n,$$

where $b_i \sim N(0, \Delta)$ and $\epsilon_i \sim N(0, \sigma^2)$, from which an algorithm of the parametric bootstrap can be defined as the following:

(1) Fit the candidate model (3.1.1) to the original data to acquire estimators $\hat{\beta}$, $\hat{\Delta}$ and $\hat{\sigma}$.

(2) Generate bootstrap samples y_i^* case-by-case using the fitted model:

$$y_i^* = X_i\hat{\beta} + Z_i\hat{b}_i^* + \hat{\epsilon}_i^*, \quad i = 1, ..., n, \tag{5.1}$$

where $\hat{b}_i^*$ and $\hat{\epsilon}_i^*$ are generated from $N(0, \hat{\Delta})$ and $N(0, \hat{\sigma}^2)$, respectively.

(3) Fit the candidate model (3.1.1) again to the bootstrap sample y_i^* to acquire estimators $\hat{\beta}^*$, $\hat{\Delta}^*$ and $\hat{\sigma}^*$.

(4) Repeat the previous steps (2) and (3) B times.

It can be seen that the parametric bootstrap approach relies so much on the model generating the residuals that if this model is not specified reasonably, there is a possibility of drawing biased conclusions.

5.1.1.3 Semi-parametric Bootstrap

The semi-parametric bootstrap differs from the parametric case in that the resampling is done over all the residuals (Efron, 1979, Freedman, 1981). Efron (1979) pointed out that in terms of linear regression models, the independently distributed error terms are the ones generating the bootstrap samples. Besides, the residuals of a linear mixed model are only associated with the fixed effects $X_i\hat{\beta}$. The corresponding algorithm is described as following:

(1) Fit the candidate model (3.1.1) to the original data for estimators $\hat{\beta}$ and $\hat{\sigma}$.

(2) Obtain the original sample of residuals $\hat{\epsilon}_i, i = 1, ..., n$ through the equation

$$\hat{\epsilon}_i = y_i - X_i\hat{\beta}. \tag{5.2}$$

(3) Generate the bootstrap sample of residuals $\hat{\epsilon}_i^*, i = 1, ..., n$ by sampling with replacement from the original sample of residuals.

(4) Construct the bootstrap sample of data values, for $i = 1, ..., n$, by the model

$$y_i^* = X_i\hat{\beta} + \hat{\epsilon}_i^*. \tag{5.3}$$

(4) Fit the candidate model (3.1.1) again to the bootstrap sample y_i^* to acquire estimators $\hat{\beta}^*$ and $\hat{\sigma}^*$.

(5) Repeat the previous steps (2), (3) and (4) B times.

The semi-parametric bootstrap depends less on the model specified than the parametric bootstrap but it is still sensitive to the candidate models (Simar, 1992). As a result, the non-parametric bootstrap is more widely used than the other two bootstrap approaches. Moreover, when GEE is chosen to analyze the longitudinal data, the non-parametric or semi-parametric bootstrap is used because there is no distribution assumed on the data as well as the random effects.

5.1.2 Selection Criteria Using Bootstrap

With the wide acceptance of the bootstrap approach due to the rapid development of technologies and computational software, the bootstrap-based selection criteria are more and more used in the field of model selection. In the rest part of this section, several model selection criteria involving the bootstrap are introduced. More specifically, these selection criteria are based on the K-L discrepancy but estimate the bias correction term (4.4) from different perspectives. Of these existing model selection criteria, AICb1 and AICb2 (Shang and Cavanaugh, 2008) have motivated us to propose the two bootstrap-adjusted quasi-likelihood information criteria QAICb1 and QAICb2 for mixed models.

5.1.2.1 Extended Information Criterion (EIC)

In Section 4.2.1, we see that the derivation of AIC is based on the Kullback-Leibler (K-L) discrepancy. The asymptotic property of AIC shows that it is an unbiased estimator of the K-L distance between the true model and a candidate model when the sample size is large enough. In addition, the AIC requires the data to follow certain distributions. The bootstrap method, based on the empirical distribution, is suitable in a wide range of model frameworks and would provide more consistent estimators over small samples compared to the approaches demanding large sample approximations. So with the purpose of improving the performance of AIC, that is, increasing the accuracy of estimating the K-L discrepancy from the perspectives of sample size and distribution assumptions, the bootstrap method has been thoroughly taken advantage of.

Ishiguro, M., Sakamoto, Y., and Kitagawa, G (1997) proposed the extended information criterion (EIC) as an extension of AIC. In contrast to AIC, the bias term (4.4) is not estimated by the dimension of the models but numerically using estimators from bootstrap samples. Recall that the bias term (4.4) of the expected K-L discrepancy consists of two parts

$$E_0\{E_0\{-2\log L(\theta|Y)\}|_{\theta=\hat{\theta}}\} \quad \text{and} \tag{5.4}$$

$$E_0\{-2\log L(\hat{\theta}|Y)\}. \tag{5.5}$$

These two different terms (5.4) and (5.5) will be estimated separately using the bootstrap method.

First is to establish the empirical distribution of a bootstrap sample following the definition by Efron (1992). To simplify the notations, we use uncorrelated data and let $Y = (y_1, ..., y_n)'$ be an $n \times 1$ vector of all the responses corresponding to a sample of size n. The empirical distribution $F_*(t)$ is given by

$$F_*(t) = \frac{1}{n} \sum_{i=1}^{n} I(t, y_i), \tag{5.6}$$

where $I(t, a)$ is an indicator function where $I(t, a) = 0$ if $t < a$ and $I(t, a) = 1$ otherwise. The bootstrap sample $Y^* = (y_1^*, ..., y_n^*)'$ is a random sample from the empirical distribution F_*.

Suppose the true cumulative distribution of the sample Y is F with parameter θ_0 and let $\hat{\theta}^*$ be the estimate of θ based on the bootstrap sample Y^*. The bootstrap method treats the empirical distribution of the sample to be the true distribution so We can replace each item associated with the true distribution F, which is unknown, by that from the empirical distribution F_* as

$$F \to F_*$$

$$Y \sim F \to Y^* \sim F_*$$

$$E_0 \to E_*$$

$$\hat{\theta} \to \hat{\theta}^*$$

where the E_* represents the expectation taken over the empirical distribution of the bootstrap sample Y^*. By applying the above replacement, the two terms (5.4) and (5.5) can be written as

$$E_*\{E_*\{-2 \log L(\theta|Y^*)\}|_{\theta=\hat{\theta}^*}\} \quad \text{and} \tag{5.7}$$

$$E_*\{-2 \log L(\hat{\theta}^*|Y^*)\}. \tag{5.8}$$

We are able to evaluate the two quantities (5.7) and (5.8) through the bootstrap method because the empirical likelihood function can be derived using the bootstrap sample.

Let $\{Y^*(i), i = 1, ..., B\}$ represent a set of B bootstrap samples and $\{\hat{\theta}^*(i), i = 1, ..., B\}$ be

the collection of MLE estimates of parameters θ obtained from these bootstrap samples. As the resamples are independently generated from the empirical distribution during a non-parametric bootstrapping, the first term (5.7) can be simplified as

$$\begin{aligned} E_*\{E_*\{-2\log L(\theta|Y^*)\}|_{\theta=\hat{\theta}^*}\} &= E_*\{\{-2\log L(\theta|Y)\}|_{\theta=\hat{\theta}^*}\} \\ &= E_*\{-2\log L(\hat{\theta}^*|Y)\}, \end{aligned}$$

which can be estimated by

$$\frac{1}{B}\sum_{i=1}^{B} -2\log L(\hat{\theta}^*(i)|Y). \tag{5.9}$$

Shibata (1997) showed that the bootstrap estimate (5.9) is valid for non-parametric bootstrap and are also applicable in terms of parametric or semi-parametric bootstrap settings when linear regression models are used. Similarly, the second quantity (5.8) can be estimated by

$$\frac{1}{B}\sum_{i=1}^{B} -2\log L(\hat{\theta}^*(i)|Y^*(i)). \tag{5.10}$$

Then we could obtain the estimation of the the bias term (4.4) using bootstrap through:

$$\begin{aligned} &\frac{1}{B}\sum_{i=1}^{B} -2\log L(\hat{\theta}^*(i)|Y) - \frac{1}{B}\sum_{i=1}^{B} -2\log L(\hat{\theta}^*(i)|Y^*(i)) \\ &= \frac{1}{B}\sum_{i=1}^{B} -2\{\log L(\hat{\theta}^*(i)|Y) - \log L(\hat{\theta}^*(i)|Y^*(i))\} \\ &= \frac{1}{B}\sum_{i=1}^{B} -2\log \frac{L(\hat{\theta}^*(i)|Y)}{L(\hat{\theta}^*(i)|Y^*(i))}. \end{aligned} \tag{5.11}$$

Therefore, the extended information criterion (EIC) is formed by replacing the bias correction term $2k$ using the bootstrap estimate (5.11) as

$$\text{EIC} = -2\log L(\hat{\theta}|Y) + \frac{1}{B}\sum_{i=1}^{B} -2\log \frac{L(\hat{\theta}^*(i)|Y)}{L(\hat{\theta}^*(i)|Y^*(i))}.$$

5.1.2.2 AICb

Another bootstrap-based modification of the AIC was proposed by Cavanaugh and Shumway (1997) for state-space model selection. They established the asymptotic result that, under certain conditions, when the sample size $N \to \infty$, the bias (4.4) can be asymptotically approximated by

$$2[E_*\{\ \log - 2L(\hat{\theta}^*|Y)\} -\ \log L(\hat{\theta}|Y)]. \tag{5.12}$$

Furthermore, according to the law of large numbers, as the number of bootstrap sample $B \to \infty$, the approximated quantity (5.12) would be asymptotically approached by

$$\begin{aligned} &\frac{1}{B}\sum_{i=1}^{B} -2\log L(\hat{\theta}^*(i)|Y) - \{-2\log L(\hat{\theta}|Y)\} \\ &= \frac{1}{B}\sum_{i=1}^{B} -2\{\ \log L(\hat{\theta}^*(i)|Y) -\ \log L(\hat{\theta}|Y)\} \\ &= \frac{1}{B}\sum_{i=1}^{B} -2\log\frac{L(\hat{\theta}^*(i)|Y)}{L(\hat{\theta}|Y)}. \end{aligned} \tag{5.13}$$

Then the model selection criterion AICb can be written as

$$\text{AICb} = -2\log L(\hat{\theta}|Y) + \frac{1}{B}\sum_{i=1}^{B} -2\log\frac{L(\hat{\theta}^*(i)|Y)}{L(\hat{\theta}|Y)}.$$

Shibata (1997) established asymptotic equivalence of the two selection criterion EIC and AICb. Moreover Shibata (1997) indicated that other bootstrap-based modification of the bias term (4.4) might exist. As the justification he provided was not under the correlated the data and mixed model setting. We will outline two other bootstrap-adjusted variants for estimating the bias (4.4) with respect to the mixed model in the following section.

5.1.2.3 AICb1 and AICb2 for Mixed Model Selection

When it comes to the mixed model with dependent data, Shang and Cavanaugh (2008) brought out a bootstrap adjustment to the bias correction term (4.4). Two bootstrap-based selection crite-

ria, AICb1 and AICb2, were discussed and shown to be efficient especially in the small sample scenario.

Let Y denote the sample of correlated date and the two consistent estimators of the bias (4.4) be defined as

$$\begin{aligned} \mathrm{b}_1 &= E_*[-2\log L(\hat{\theta}^*|Y) - \{-2\log L(\hat{\theta}^*|Y^*)] \quad \text{and} \\ \mathrm{b}_2 &= 2E_*[-2\log L(\hat{\theta}^*|Y) - \{-2\log L(\hat{\theta}|Y)]. \end{aligned}$$

By the strong law of large numbers, as the number of bootstrap samples $B \to \infty$, we have

$$\begin{aligned} &\frac{1}{B}\sum_{i=1}^{B}\{-2\log L(\hat{\theta}^*(i)|Y)\} \to E_*\{-2\log L(\hat{\theta}^*|Y)\} \text{ almost surely and} \\ &\frac{1}{B}\sum_{i=1}^{B}\{-2\log L(\hat{\theta}^*(i)|Y^*(i))\} \to E_*\{-2\log L(\hat{\theta}^*|Y^*)\} \text{ almost surely.} \end{aligned}$$

based on witch the selection criterion AICb1 and AICb2 are defined as

$$\begin{aligned} \text{AICb1} &= -2\log L(\hat{\theta}|Y) + \frac{1}{B}\sum_{i=1}^{B} -2\log\frac{L(\hat{\theta}^*(i)|Y)}{L(\hat{\theta}^*(i)|Y^*(i))} \text{and} \\ \text{AICb2} &= -2\log L(\hat{\theta}|Y) + \frac{1}{B}\sum_{i=1}^{B} -2\log\frac{L(\hat{\theta}^*(i)|Y)}{L(\hat{\theta}^*(i)|Y^*(i))}. \end{aligned}$$

Note that AICb1 and AICb2 follow the same expression as the EIC and AICb, respectively. However, it was shown by Shang and Cavanaugh (2008) that these two criteria are asymptotically equivalent when mixed models are used.

We can see that both EIC and AICb as well as AICb1 and AICb2 are based on the log likelihood function and MLE estimators, which means that these selection criteria depend on the distribution assumption of the data. This indicates that with the application of quasi-likelihood functions, we can extend AICb1 and AICb2 to more model selection scenarios. Before we introduce our two selection criteria QAICb1 and QAICb2 that combine quasi-likelihood function and bootstrap, we would like to discuss model selection criteria based on quasi-likelihood functions.

5.1.3 Selection Criteria Using Quasi-likelihood

The likelihood-based model selection criteria we talked about before are only valid when the distribution assumptions are met. In other words, under the circumstance that the likelihood function of the data is not available, they could not be simply modified for estimators or models based on quasi-likelihood functions. Further, when the problem of over-dispersion arises, the quasi-likelihood is preferred over the traditional likelihood and a selection criterion involving over-dispersion parameter should be utilized.

5.1.3.1 QAIC and QAICc

As discussed in Section 2.2.1, there is a tendency to encounter over-dispersion when the mean and the variance are not independently estimated. However, the traditional model selection criterion of mixed models hardly takes the over-dispersion into consideration. As a modification of the selection criterion AIC, by adding the over-dispersion estimator to the log-likelihood function, QAIC can be defined as

$$\text{QAIC} = -2 \log L(\hat{\theta}|Y)/\hat{\phi} + 2k,$$

where $\hat{\theta}$ is the MLE estimator of θ and $\hat{\phi}$ is the estimator for the over-dispersion parameter (Burnham and Anderson, 2002). Due to the additional parameter, one more dimension is added to the k but when the model experiences no over-dispersion, $\hat{\phi}$ is set to be 1 and QAIC reduces to AIC. Similarly, to address the issue overfitting due to small samples, QAICc (Burnham and Anderson, 2002)

$$\text{QAICc} = \text{QAIC} + \frac{2k(k+1)}{N-k-1}.$$

is utilized.

Both QAIC and QAICc make the adjustment for AIC in the cases where there exists the over-dispersion parameter while the traditional log likelihood function still plays an important role in these two AIC-type criteria. In other worlds, a model selection criteria fully incorporates the quasi-likelihood function and the corresponding estimators should be considered.

5.1.3.2 QAIC$_u(R)$ and QIC(R)

The quasi-likelihood and traditional likelihood functions share similar properties, which leads to one possible modification by replacing the likelihood part of the AIC using the quasi-likelihood of the model. According to the suggestion in Pan (2001), when incorporating GEE to obtain the estimated model coefficients, the model selection criterion QAIC$_u(R)$ can be defined by

$$\text{QAIC}_u(R) = -2Q(\hat{\beta}_G(R), \hat{\phi}; Y, I) + 2k,$$

where $\hat{\beta}_G(R)$ is the GEE estimator under any working correlation R and $\hat{\rho}$ is the estimator for the over-dispersion parameter ρ. However, Pan (2001) did not establish any asymptotic properties for QAIC$_u(R)$. In fact, during simulation studies in chapter 6, the corresponding results demonstrate that although not consistent across all the model settings, QAIC$_u(R)$ performs quite well when the correlation within individuals is large and the variance of the noise is small.

Pan (2001) also proposed a quasi-likelihood based information criterion (QIC(R)) partially under the independence assumption. In other words, QIC(R) treats all the within-individual observations to be independent with each other as the independent estimating equations do. The parameters estimated in QIC(R) are based on the GEE approach such that QIC(R) is distribution free compared to other AIC-type selection criteria. We will outline the details of constructing QIC(R) in the rest of this section.

Recall that AIC is created based on the K-L discrepancy which measures the expected distance between the true model and the candidate model. Following the construction of AIC, the quasi-likelihood function is used to define a similar discrepancy as

$$d(\beta, \beta_0, I) = E_0\{-2Q(\beta, \phi; Y, I)\}, \tag{5.14}$$

where β and β_0 are parameters of the candidate and true model respectively, $Q(\beta, \phi; Y, I)$ is the log quasi-likelihood of any candidate model under the independence working correlation I and

the expectation is taken with respect to the true model. Pan (2001) followed the discussion of McCullagh and Nelder (1989) that if a more general working correlation R other than I is used, the corresponding quasi-likelihood $Q(\beta, \phi; Y, R)$ is not guaranteed to exist unless certain conditions are met. Moreover, the computation of the quasi-likelihood will require too much work from derivation and integration over a complex correlation structure. Applying the independence assumption, the quasi-likelihood of a candidate model can be written as

$$Q(\beta, \phi; Y, I) = \sum_{i=1}^{n} \sum_{j=1}^{n_i} Q(\beta, \phi; y_{ij}), \tag{5.15}$$

where y_{ij} represents the jth observation from the ith individual.

Additionally, we have the following properties

$$E_0 \left\{ - \left. \frac{\partial Q(\beta, \phi; Y, I)}{\partial \beta} \right|_{\beta=\beta_0} \right\} = 0 \quad \text{and}$$
$$\Sigma_I = E_0 \left\{ - \left. \frac{\partial^2 Q(\beta, \phi; Y, I)}{\partial \beta \partial \beta'} \right|_{\beta=\beta_0} \right\} = \sum_{i=1}^{n} D_i' V_i D_i,$$

where the Σ_I is positive definite. Pan (2001) pointed out that the β_0 is a local minimizer of $d(\beta, \beta_0, I)$ with respect to β such that

$$d(\beta, \beta_0, I) \geq d(\beta_0, \beta_0, I) \tag{5.16}$$

as the integral and derivative can be exchanged under certain conditions. Lehmann (1983) verified that when the quasi-likelihood function and likelihood function are the same, the true model parameter β_0 is indeed a global minimizer of $d(\beta, \beta_0, I)$. Therefore, the equation (5.16) implies that $d(\beta, \beta_0, I)$ is well-defined discrepancy for the candidate models in the neighborhood of the true model.

Let $\hat{\beta} = \hat{\beta}_G(R)$ be the GEE estimator of the data under any working correlation R. According

to Linhart and Zucchini (1986), the expectation $E_0\{d(\hat{\beta}, \beta_0, I)\}$ can be approximated by

$$\begin{aligned} E_0\{d(\hat{\beta}, \beta_0, I)\} \approx & -2E_0\{Q(\hat{\beta}, \phi; I, Y)\} \\ & + 2E_0\{(\hat{\beta} - \beta_0)' S(\hat{\beta}; I, Y)\} \\ & + 2trace(\Sigma_I J), \end{aligned} \tag{5.17}$$

where $S(\hat{\beta}; I, Y)$ is the quasi-score equation under the independence working assumption and $J = \text{cov}(\hat{\beta})$ is the variance covariance matrix of the estimated parameter $\hat{\beta}$. The variance matrix J can be estimated consistently by the sandwich estimator $\hat{V}_r$ (Liang and Zeger, 1986) and the Σ_I is estimated by the naive estimator of the variance of $\hat{\beta}_G(I)$ which is the GEE estimator under the independence working correlation matrix. The quasi-score equation $S(\hat{\beta}; R, Y)$ is 0 when $\hat{\beta}$ is estimated by $\hat{\beta}_G(R)$ using a general working correlation R. However, $S(\hat{\beta}_G(R); I, Y)$ is not necessarily to be 0 unless the $\hat{\beta}_G(I)$ is considered. In other words, only when the independence working correlation is taken use of during the GEE estimation procedure, the term (5.17) can be treated as 0.

Under the independence working correlation, Pan (2001) proposed an information criterion based on the quasi-likelihood function, QIC(R), as

$$\text{QIC}(R) = -2Q(\hat{\beta}_G(R), \hat{\phi}; Y, I) + 2\text{trace}(\hat{\Sigma}_I \hat{V}_r),$$

where $\hat{\phi}$ is estimated based on the largest candidate model. QIC(R) can also be used to select the most appropriate working correlation structure by fitting the same model using different working correlation and selecting the one with the smallest QIC(R).

As mentioned before, the second term (5.17) of the expectation $E_0\{d(\hat{\beta}, \beta_0, I)\}$ is 0 under the independent working correlation matrix I. So QIC(I) is an unbiased estimator of the discrepancy (5.14). However, the term (5.17) is ignored in the construction of QIC(R) because it is hard to estimated under a general working correlation R. As a result, QIC(R) is not an unbiased estimator of the discrepancy (5.14). Although Pan (2001) showed that ignoring this term would not signifi-

cantly affect the performance of QIC(R) during simulation studies, QIC(R) tend to underestimate the quantity (5.14) when the true correlation of the data is relatively large. The simulation studies in Chapter 6 indicate that the performance of QIC(R) is much more affected by a larger correlation compared to other selection criterion.

Note that QIC(R) is defined under the independence model where the data is assumed to be uncorrelated. It seems that the independent working structure would not perform well if the true correlation of the data is relatively large, which is shown by QIC(R). However, when implementing our proposed QAICb1 and QAICb2, by updating the model coefficients using the independent working matrix to construct the bias correction term, our two model selection criteria have a significant performance improvement. Simulation results in Chapter 6 show that QAICb1 and QAICb2 are consistent and efficient under different correlation structures especially in scenarios with large correlations where QIC(R) can hardly work.

5.2 Bootstrap-adjusted Quasi-likelihood Selection Criteria

In this section, we propose two selection criteria QAICb1 and QAICb2 based on the quasi-likelihood function as well as the bootstrap method.

5.2.1 Extension of K-L Discrepancy via Quasi-likelihood Functions

The usual Kullback-Leibler discrepancy (4.1) defined over the likelihood function need to be extended using the quasi-likelihood function. Let β_0 and β be the parameters for the true model and the candidate model, respectively. The quasi-likelihood function corresponding to the parameter β for a candidate model is defined as $Q(\beta, \phi; Y)$. Following Pan's (2001) definition of the model discrepancy in the previous section, the quasi-likelihood based model discrepancy is

$$d(\beta, \beta_0, \phi; Y) = E_0[-2Q(\beta, \phi; Y)],$$

where the expectation E_0 is taken under the true model. As discussed before, the discrepancy $d(\beta, \beta_0, \phi; Y)$ is valid for β from all candidate models in the neighborhood of β_0 with β_0 being the local minimizer of $d(\beta, \beta_0, \phi; Y)$.

Let $\hat{\beta}$ be the estimator of β from a certain candidate model. Here $\hat{\beta}$ is defined the same as that in Chapter 2, which is derived by solving the quasi-score equations. Then the corresponding discrepancy can be written as

$$d(\hat{\beta}, \beta_0, \phi; Y) = E_0[-2Q(\beta, \phi; Y)]|_{\beta=\hat{\beta}}. \tag{5.18}$$

In reality, it is not possible to evaluate the quantity (5.18) because the parameter β_0 is usually unknown. Referring to the decomposition of the overall K-L discrepancy (4.4), let $d(\beta_0, \phi, k)$ be the expectation of the discrepancy (5.18) under the true model, then we have

$$\begin{aligned} d(\beta_0, \phi, k) &= E_0[d(\hat{\beta}, \beta_0, \phi; Y)] \\ &= E_0[E_0\{-2Q(\beta, \phi; Y)\}|_{\beta=\hat{\beta}}] \\ &= E_0[-2Q(\hat{\beta}, \phi; Y)] \\ &+ E_0[E_0\{-2Q(\beta, \phi; Y)\}|_{\beta=\hat{\beta}}] - E_0[-2Q(\hat{\beta}, \phi; Y)]. \end{aligned}$$

Moreover, the expected overall K-L distance $d(\beta_0, \phi, k, I)$ under the independent assumption I can be written as

$$\begin{aligned} d(\beta_0, \phi, k, I) &= E_0[-2Q(\hat{\beta}, \phi; Y, I)] \\ &+ E_0[E_0\{-2Q(\beta, \phi; Y, I)\}|_{\beta=\hat{\beta}}] - E_0[-2Q(\hat{\beta}, \phi; Y, I)]. \end{aligned}$$

McCullagh (1983) pointed out that when the data are uncorrelated and if the log likelihood is from the exponential family and linear in the response variable Y, the log likelihood and the log quasi-likelihood functions are the same expect for the over-dispersion parameter which is a scalar added to the corresponding quasi-likelihood. Thus, according to Akaike's (1973) construction of

AIC, the quantity $-2Q(\hat{\beta}, \phi; Y, I)$ is an biased estimator of $d(\beta_0, \phi, k, I)$ and the bias term

$$E_0[E_0\{-2Q(\beta, \phi; Y, I)\}|_{\beta=\hat{\beta}}] \tag{5.19}$$

$$- E_0[-2Q(\hat{\beta}, \phi; Y, I)] \tag{5.20}$$

could be estimated by $2k$, where k is the number of estimated covariates in the candidate model.

Following our discussion, a direct modification of AIC based on the quasi-likelihood function would have the following form by replacing the log likelihood component with the log quasi-likelihood function under the independence structure as

$$\text{QAIC}_u = -2Q(\hat{\beta}, \hat{\phi}; Y, I) + 2k \tag{5.21}$$

which is the general form of the selection criterion $\text{QIC}_u(R)$ that was discussed in Section 5.1.3.2. When the GEE estimator $\hat{\beta}_G(R)$ is used, QAIC_u will reduce to $\text{QAIC}_u(R)$. The performance of $\text{QAIC}_u(R)$ and our proposed two model selection criteria will be compared during simulation studies in Chapter 6.

5.2.2 Estimation of Bias Correction Term via Bootstrap

Next, We apply a bootstrap-based method to estimate the bias correlation term as an extension to the bootstrap correction of AIC proposed by Shang and Cavanaugh (2008). To simplify the notations, the over-dispersion parameter ϕ is not displayed in the construction of the model selection criterion because ϕ is a multiplier added to the variance covariance matrix and can be viewed as a constant with respect to the quasi-likelihood functions. Furthermore, under the circumstances that linear mixed models are used, this over-dispersion parameter is set to be 1 because the mean and the variance are independent.

Let the vectors Y and $\hat{\beta}$ denote the original sample and the corresponding estimator of the coefficient β, respectively. Let $\{Y^*(i), i = 1, ..., B\}$ be the B bootstrap samples obtained through Y by resampling on the individual level and $\{\hat{\beta}^*(i), i = 1, ..., B\}$ be the corresponding estimators

from the bootstrap samples. Note that if the resampling is done over all the observations, it will violate the independence assumption of the bootstrap method. Suppose G is the distribution of Y with parameter β and G_* is the corresponding empirical distribution of the bootstrap sample Y^*. As discussed in Section 5.1.2.1, we can replace the terms of the original sample by the relevant ones from the bootstrap samples by

$$G \to G_*$$

$$Y \sim G \to Y^* \sim G_*$$

$$E_0 \to E_*$$

$$\hat{\beta} \to \hat{\beta}^*$$

where the expectation E_* is taken with respect to the empirical distribution of the bootstrap sample Y^*. Thus, the two quantities (5.19) and (5.20) of the bias correction term could be expressed as

$$E_*[E_*\{-2Q(\beta; Y^*)\}|_{\beta=\hat{\beta}^*}] \tag{5.22}$$

$$- E_*[-2Q(\hat{\beta}^*; Y^*)] \tag{5.23}$$

where, fortunately, the two expectations (5.22) and (5.23) can be estimated numerically through estimators obtained using the bootstrap samples. It should be pointed out that the quasi-likelihood is not easy to compute if the sample has a complex correlation structure. In addition to greatly simplifying the calculations, to guarantee the validation of the corresponding quasi-likelihood function and the existence of the local minimum $\hat{\beta}^*$, the independent model structure is utilized. This means that we will evaluate the expectations (5.22) and (5.23) as the following:

$$E_*[E_*\{-2Q(\beta; Y^*, I)\}|_{\beta=\hat{\beta}^*}] \tag{5.24}$$

$$- E_*[-2Q(\hat{\beta}^*; Y^*, I)]. \tag{5.25}$$

It has been discussed in Section 5.2.1 that under the independent assumption and the corre-

sponding distribution belonging to the exponential family, the log quasi-likelihood and log likelihood functions share the common form except for the scalar over-dispersion parameter ϕ. The linear mixed model setting enables us to ignore the scalar and mimic the two model selection criterion AICb1 and AICb2 proposed by Shang and Cavanaugh (2008) using the log quasi-likelihood instead of the log likelihood. However, the application of the quasi-likelihood only requires the first two moments of the distribution are s pecified. In particular, due to the fact that the variance covariance used in the quasi-likelihood is defined as a function of the mean, given this function, only specifying the mean is enough to construct a valid quasi-likelihood function. Furthermore, when quasi-likelihood and the longitudinal data are considered, we can take advantage of the GEE procedure to produce more consistent estimators.

At the rest of this section, we will illustrate the construction of the two bootstrap-adjusted correction term based on the quasi-likelihood function and provide the proof and the asymptotic results of the proposed criteria in the last two sections of this chapter.

5.2.3 First Bootstrap-adjusted Quasi-likelihood Akaike Information Criterion (QAICb1)

Motivated by the construction of EIC (Ishiguro, Sakamoto and Kitagawa, 1997) and AICb1 (Shang and Cavanaugh, 2008), the bootstrap estimation of the expectation (5.24) replies on a crucial assumption that

$$E_*\{-2Q(\beta; Y^*, I)\} = -2Q(\beta; Y, I) \tag{5.26}$$

under the parametric, semi-parametric or non-parametric bootstrap approaches. The detailed proof of the assumption (5.26) will be displayed in Section 5.3. Taking advantage of the assumption (5.26), the expectation (5.24) can be expressed as

$$\begin{aligned} E_*[E_*\{-2Q(\beta; Y^*, I)\}|_{\beta=\hat{\beta}^*}] &= E_*[-2Q(\beta; Y, I)|_{\beta=\hat{\beta}^*}] \\ &= E_*[-2Q(\hat{\beta}^*; Y, I)] \end{aligned} \tag{5.27}$$

where the bootstrap expectation (5.27) can be estimated by

$$\frac{1}{B}\sum_{i=1}^{B} -2Q(\hat{\beta}^*(i); Y, I) \tag{5.28}$$

because when the number of the bootstrap samples $B \to \infty$, we have

$$\frac{1}{B}\sum_{i=1}^{B} -2Q(\hat{\beta}^*(i); Y, I) \longrightarrow E_*\{-2Q(\hat{\beta}^*; Y, I)\} \quad \text{a.s.}$$

according to the law of large numbers (LLN).

Similarly, we can employ the bootstrap approach to directly estimate the quantity (5.25) by

$$\frac{1}{B}\sum_{i=1}^{B} -2Q(\hat{\beta}(i)^*; Y(i)^*, I) \tag{5.29}$$

since when the number of the bootstrap samples $B \to \infty$, we have

$$\frac{1}{B}\sum_{i=1}^{B} -2Q(\hat{\beta}^*(i); Y_i^*, I) \longrightarrow E_*\{-2Q(\hat{\beta}^*; Y^*, I)\} \quad \text{a.s..}$$

Let the first bootstrap-adjusted QAIC variant defined as

$$\text{b1} = E_*[-2Q(\hat{\beta}^*(i); Y, I) - \{-2Q(\hat{\beta}^*(i); Y^*(i), I)\}], \tag{5.30}$$

then by utilizing the two bootstrap estimates (5.28) and (5.29), the variant b1 can be estimated as

$$\begin{aligned}
\text{b1} &= E_*\{-2Q(\hat{\beta}^*(i); Y, I)\} - E_*\{-2Q(\hat{\beta}^*(i); Y^*(i), I)\} \\
&\approx \frac{1}{B}\sum_{i=1}^{B} -2Q(\hat{\beta}^*(i); Y, I) - \frac{1}{B}\sum_{i=1}^{B} -2Q(\hat{\beta}^*(i); Y^*(i), I) \\
&\approx \frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}^*(i); Y, I)}{Q(\hat{\beta}^*(i); Y^*(i), I)},
\end{aligned}$$

from which we propose the first bootstrap-adjusted quasi-likelihood information criteria QAICb1

for the linear mixed model as

$$\text{QAICb1} = -2Q(\hat{\beta}; Y, I) + \frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}^*(i); Y, I)}{Q(\hat{\beta}^*(i); Y^*(i), I)}.$$

5.2.4 Second Bootstrap-adjusted Quasi-likelihood Akaike Information Criterion (QAICb2)

The second bootstrap-adjusted variant is constructed similarly as b1 following the development of the AICb (Cavanaugh and Shumway, 1997) and AICb2 (Shang and Cavanaugh, 2008). The difference of the two quantities (5.19) and (5.20), in terms of the independent assumption, can be written as

$$E_0[E_0\{-2Q(\beta; Y, I)\}|_{\beta=\hat{\beta}}] - E_0\{-2Q(\beta_0; Y, I)\} \tag{5.31}$$

$$+ E_0\{-2Q(\beta_0; Y, I)\} - E_0[-2Q(\hat{\beta}; Y, I)]. \tag{5.32}$$

By replacing the expectations in expressions (5.31) and (5.32) using the bootstrap expectation and applying the crucial assumption (5.26), we have

$$\begin{aligned} &E_0[E_0\{-2Q(\beta; Y, I)\}|_{\beta=\hat{\beta}}] - E_0\{-2Q(\beta_0, ; Y, I)\} \\ &= E_*[E_*\{-2Q(\beta; Y^*, I)\}|_{\beta=\hat{\beta}^*}] - E_*\{-2Q(\beta_0; Y^*, I)\} \\ &= E_*\{-2Q(\hat{\beta}^*; Y^*, I)\} - \{-2Q(\beta_0; Y, I)\} \end{aligned} \tag{5.33}$$

and

$$\begin{aligned} &E_0\{-2Q(\beta_0; Y, I)\} - E_0[-2Q(\hat{\beta}; Y, I)] \\ &= E_*\{-2Q(\beta_0; Y^*, I)\} - E_*[-2Q(\hat{\beta}; Y^*, I)] \\ &= -2Q(\beta_0; Y, I) - 2Q(\hat{\beta}; Y, I). \end{aligned} \tag{5.34}$$

Let the second bootstrap-adjusted QAIC variant defined as

$$\text{b2} = 2E_*[-2Q(\hat{\beta}^*; Y, I) - \{-2Q(\hat{\beta}; Y, I)\}] \tag{5.35}$$

and under certain conditions, we can show that both the two quantities (5.33) and (5.34) can be approximated by half of the b2 because when $n \to \infty$,

$$\begin{aligned} &E_*[-2Q(\hat{\beta}^*; Y, I) - \{-2Q(\hat{\beta}; Y, I)\}] \\ &\longrightarrow E_*\{-2Q(\hat{\beta}^*; Y^*, I)\} - \{-2Q(\beta_0; Y, I)\} \quad \text{a.s.} \end{aligned}$$

and

$$\begin{aligned} &E_*[-2Q(\hat{\beta}^*; Y, I) - \{-2Q(\hat{\beta}; Y, I)\}] \\ &\longrightarrow -2Q(\beta_0; Y, I) - 2Q(\hat{\beta}; Y, I) \quad \text{a.s..} \end{aligned}$$

Therefore,

$$\begin{aligned} \text{b2} \to\ &E_*\{-2Q(\hat{\beta}^*; Y^*, I)\} - \{-2Q(\beta_0; Y, I)\} \\ &- 2Q(\beta_0; Y, I) - 2Q(\hat{\beta}; Y, I) \quad \text{a.s..} \end{aligned}$$

Moreover, when the number of the bootstrap samples $B \to \infty$, we have

$$\frac{1}{B}\sum_{i=1}^{B} -2Q(\hat{\beta}^*(i); Y_i^*) \longrightarrow E_*\{-2Q(\hat{\beta}^*; Y^*)\} \quad \text{a.s.}$$

then the variant b2 can be estimated as

$$\begin{aligned}
\text{b2} &= 2E_*\{-2Q(\hat{\beta}^*(i);Y,I)\} - -2Q(\hat{\beta};Y,I)\} \\
&\approx 2\Big\{\frac{1}{B}\sum_{i=1}^{B} -2Q(\hat{\beta}^*(i);Y,I) - \frac{1}{B}\sum_{i=1}^{B} -2Q(\hat{\beta};Y,I)\Big\} \\
&\approx 2\Big\{\frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}^*(i);Y,I)}{Q(\hat{\beta};Y,I)}\Big\}
\end{aligned}$$

from which we propose the second bootstrap-adjusted quasi-likelihood information criteria QAICb2 for the linear mixed model

$$\text{QAICb2} = -2Q(\hat{\beta};Y,I) + 2\Big\{\frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}^*(i);Y,I)}{Q(\hat{\beta};Y,I)}\Big\}.$$

In the next section, we will show that the two variants b1 and b2 are asymptotically equivalent estimators of the two bias terms (5.19) and (5.20).

As a summary, we have proposed the following two bootstrap-adjusted model selection criteria based on the quasi-likelihood function:

$$\begin{aligned}
\text{QAICb1} &= -2Q(\hat{\beta};Y,I) + \frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}^*(i);Y,I)}{Q(\hat{\beta}^*(i);Y^*(i),I)}, \\
\text{QAICb2} &= -2Q(\hat{\beta};Y,I) + 2\Big\{\frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}^*(i);Y,I)}{Q(\hat{\beta};Y,I)}\Big\}.
\end{aligned} \tag{5.36}$$

Moreover, when a generalized linear model with random effects is considered, we can extend QAICb1 and QAICb2 in (5.36) by adding the estimator of over-dispersion parameter $\hat{\phi}$ as

$$\begin{aligned}
\text{QAICb1} &= -2Q(\hat{\beta},\hat{\phi};Y,I) + \frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}^*(i),\hat{\phi}^*(i);Y,I)}{Q(\hat{\beta}^*(i),\hat{\phi}^*(i);Y^*(i),I)}, \\
\text{QAICb2} &= -2Q(\hat{\beta},\hat{\phi};Y,I) + 2\Big\{\frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}^*(i),\hat{\phi}^*(i);Y,I)}{Q(\hat{\beta},\hat{\phi};Y,I)}\Big\}.
\end{aligned} \tag{5.37}$$

5.3 Proofs and Asymptotic Properties

In this section, the asymptotic properties of QAICb1 and QAICb2 are established under linear mixed models. We will prove that our proposed selection criteria are asymptotically equivalent when the number of individuals $n \to \infty$. In addition, we will show that QAICb1 and QAICb2 are consistent unbiased estimators of the expected K-L discrepancy $d(\beta_0, k, I)$.

Let $\mathcal{B}$ be the parameter space of the unknown parameter β. Let $\bar{\beta}$ be the pseudo true parameter of β, which is the estimator of β obtained by solving the corresponding quasi-score equation. It should be pointed out that $\bar{\beta}$ is not necessarily a global minimizer and is assumed to be existing and unique.

5.3.1 Asymptotic Equivalence of QAICb1 and QAICb2

In this section, we will focus on proving the asymptotic equivalence of QAICb1 and QAICb2. It is the same to prove the asymptotic equivalence of the two bias correction terms b1 and b2. Following Shang and Cavanaugh (2008), we will establish the consistency of the two estimators $\hat{\beta}$ and $\hat{\beta}^*$ from the original and bootstrap sample, respectively. We will show that as $n \to \infty$,

$$\hat{\beta} \to \bar{\beta} \text{ and } \hat{\beta}^* \to \bar{\beta}$$

based on **Assumptions 1** to **4**.

Assumption 1.

1. The parameter space $\mathcal{B}$ is a compact subset of k-dimensional Euclidean space.
2. The first, second and third derivatives of the log quasi-likelihood with respect to β exist, and are continuous and bounded over $\mathcal{B}$.
3. $\bar{\beta}$ is an interior point of $\mathcal{B}$.

With **Assumption 1**, let $Q_i(y_i|\beta)$ and $Q_i(y_i|\tilde{\beta})$ be the two marginal densities for the ith individual given β and $\tilde{\beta}$ in a neighborhood U of $\mathcal{B}$. The corresponding log quasi-likelihood ratio is defined by

$$R_i(y_i, \beta, U) = \inf_{\tilde{\beta} \in U} \{Q_i(y_i|\beta) - Q_i(y_i|\tilde{\beta})\} = \inf_{\tilde{\beta} \in U} \frac{Q_i(y_i|\beta)}{Q_i(y_i|\tilde{\beta})}.$$

Assume that the following limit

$$\bar{R}(\bar{\beta}, U) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} E_0\{R_i(y_i, \bar{\beta}, U)\}$$

exists and is finite in a neighborhood $U = U_\beta$ for any β in $\mathcal{B}$.

Note that the log quasi-likelihood function $Q(\beta; Y)$ is continuous with

$$\lim_{\tilde{\beta}\to\beta} Q(\tilde{\beta}; Y) = Q(\beta; Y)$$

for any $\tilde{\beta}$ in the space $\mathcal{B}$. Based on the Lebesgue monotone convergence theorem, we have the following limits

$$\lim_{q\to\infty} \bar{R}(\bar{\beta}, U_\beta^{(q)}) = \bar{R}(\bar{\beta}, \beta) = \lim_{n\to\infty} \frac{1}{n} E_0\{Q(\bar{\beta}; Y) - Q(\beta; Y)\}$$

hold true for a monotone decreasing sequence of neighborhoods $U_\beta^{(q)}$, $q = 1, 2, 3, ...$ converging to a parameter β.

For a bootstrap samples $Y^* = (y_1^{*'}, ..., y_n^{*'})'$, we can similarly define

$$\bar{R}_B(\bar{\beta}, U) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} E_0 E_*\{R_i(y_i^*, \bar{\beta}, U)\}$$

and

$$\lim_{q\to\infty} \bar{R}(\bar{\beta}, U_\beta^{(q)}) = \bar{R}_B(\bar{\beta}, \beta) = \lim_{n\to\infty} \frac{1}{n} E_0 E_*\{Q(\bar{\beta}; Y^*) - Q(\beta; Y^*)\}.$$

Then we provide the next assumption:

Assumption 2.

1. $\frac{1}{n} \sum_{i=1}^{n} R_i(y_i, \bar{\beta}, U_\beta)$ converges almost surely to $\bar{R}(\bar{\beta}, U_\beta)$ in a neighborhood U_β for any β in $\mathcal{B}$ and $\bar{R}(\bar{\beta}, U_\beta) > 0$ for β in $\mathcal{B}$, where $\beta \neq \bar{\beta}$.

2. $\frac{1}{n} \sum_{i=1}^{n} R_i(y_i^*, \bar{\beta}, U_\beta)$ converges almost surely to $\bar{R}_B(\bar{\beta}, U_\beta)$ in a neighborhood U_β for any β in $\mathcal{B}$ and $\bar{R}_B(\bar{\beta}, U_\beta) > 0$ for β in $\mathcal{B}$, where $\beta \neq \bar{\beta}$.

Next we will provide **Assumption 3** as well as the corresponding proof using the non-parametric and semi-parametric approaches under the linear mixed model. **Assumption 3** and its special case **Assumption 3b** lay the foundation of our proposed QAICb1 and QAICb2. In Section 5.3.3, we will extend **Assumption 3** and **Assumption 3b** to the generalized linear model with random effects.

Assumption 3. $E_* Q(\beta; Y^*, I) = Q(\beta; Y, I)$.

Proof. To prove the **Assumption 3**, we take advantage of the independent model structure by assuming the data are not correlated. Shang and Cavanaugh (2008) proved a assumption similar to **Assumption 3** under log likelihood and MLE estimators using parametric, semi-parametric and non-parametric settings. As both a MLE estimator and a quasi-score-based estimator are obtained by solving the first derivative of the corresponding log likelihood functions, **Assumption 3** can be extended using quasi-likelihood functions. Moreover, the log quasi-likelihood and log likelihood functions take the same form other than a constant term arisen from the integral when the linear mixed model is used.

For the linear mixed model, the variance covariance matrix $V = Z\Delta Z' + V_0$ is free and independent from the parameter β. When quasi-likelihood is employed to fit the linear mixed model, the estimate of the variance covariance V, $\hat{V}$, is a constant with respect to β. Taking this advantage will greatly simplify the computation of quasi-likelihood functions. We will first show the proof of **Assumption 3** using the semi-parametric approach.

Semi-parametric bootstrap

The semi-parametric bootstrap depends on resampling over the residuals obtained through the fitted model by

$$\hat{\xi}_i = y_i - X_i\hat{\beta}, \quad i = 1, ..., n$$

where $\hat{\xi}_i$ is the vector of residuals corresponding to the ith individual. Let $\hat{\xi} = (\hat{\xi}_1, ..., \hat{\xi}_n)'$ and the equation of obtaining the residuals $\hat{\xi}$ in the matrix form is denoted by

$$\hat{\xi} = Y - X\hat{\beta}.$$

Then a bootstrap sample Y^* using the semi-parametric bootstrapping is achieved by

$$Y^* = X\hat{\beta} + \hat{\xi}^* \tag{5.38}$$

where $\hat{\xi}^*$, denoted by $(\hat{\xi}_1^*, ..., \hat{\xi}_n^*)'$, is selected with replacement based on the empirical distribution of the residuals $\hat{\xi}$.

Let V be the variance covariance matrix of Y which is independent of β. With respect to β, X and Y, the first derivative of the quasi-likelihood function under the independent model setting can be written by

$$\frac{\partial Q(\beta; Y, I)}{\partial \beta} = D'V^{-1}(Y - X\beta). \tag{5.39}$$

Actually, the derivative (5.39) is the same as the quasi-score equation (2.3) defined under the uncorrelated observations. By integrating (5.39) and ignoring the constant, the log quasi-likelihood function of the original sample is calculated as

$$Q(\beta; X, Y, I) = \frac{1}{2}D'(Y - X\beta)'V^{-1}(Y - X\beta) \tag{5.40}$$

and the corresponding log quasi-likelihood of the bootstrap sample Y^* under the semi-parametric bootstrap setting can be written as

$$Q(\beta; X, Y^*, I) = \frac{1}{2}D'(Y^* - X\beta)'V^{-1}(Y^* - X\beta). \tag{5.41}$$

By taking the expectation with respect to the bootstrap distribution over the expression of (5.41)

and applying the equation (5.38), we have

$$\begin{aligned}
E_*Q(\beta; X, Y^*, I) &= E_*\{\frac{1}{2}D'(Y^* - X\beta)'V^{-1}(Y^* - X\beta)\} \\
&= \frac{1}{2}D'E_*\{(Y^* - X\beta)'V^{-1}(Y^* - X\beta)\} \\
&= \frac{1}{2}D'E_*\{(X\hat{\beta} + \hat{\xi}^* - X\beta)'V^{-1}(X\hat{\beta} + \hat{\xi}^* - X\beta)\} \\
&= \frac{1}{2}D'E_*\{(Y - X\beta + \hat{\xi}^* - (Y - X\hat{\beta}))'V^{-1}(Y - X\beta + \hat{\xi}^* - (Y - X\hat{\beta}))\} \\
&= \frac{1}{2}D'E_*\{(\xi - (\hat{\xi} - \hat{\xi}^*))'V^{-1}(\xi - (\hat{\xi} - \hat{\xi}^*))\} \\
&= \frac{1}{2}D'E_*\{\xi'V^{-1}\xi - 2\xi'V^{-1}(\hat{\xi} - \hat{\xi}^*) + (\hat{\xi} - \hat{\xi}^*)'V^{-1}(\hat{\xi} - \hat{\xi}^*)\} \\
&= \frac{1}{2}D'\{\xi'V^{-1}\xi - 2\xi'V^{-1}E_*(\hat{\xi} - \hat{\xi}^*) + E_*||V^{-1}(\hat{\xi} - \hat{\xi}^*)||^2\}. \qquad (5.42)
\end{aligned}$$

As the distribution of $\hat{\xi}^*$ is the same as the empirical distribution of $\hat{\xi}$ and E_* can also be viewed as the expectation taken under the empirical distribution, then

$$E_*(\hat{\xi} - \hat{\xi}^*) = E_*(\hat{\xi}) - E_*(\hat{\xi}^*) = \frac{1}{n}\sum_{i=1}^{n}\hat{\xi}_i - \frac{1}{n}\sum_{i=1}^{n}\hat{\xi}_i = 0.$$

Thus, the expectation (5.42) can be simplified to

$$\begin{aligned}
E_*Q(\beta; X, Y^*, I) &= \frac{1}{2}D'\{\xi'V^{-1}\xi\} \\
&= \frac{1}{2}D'\{(Y - X\beta)'V^{-1}(Y - X\beta)\} \\
&= Q(\beta; X, Y, I),
\end{aligned}$$

which has validated the proof of **Assumption 3** under semi-parametric bootstrap.

Non-parametric bootstrap

As discussed in Section 5.1.1, when implementing the non-parametric bootstrap approach, we should select data points with replacement at the individual level when there exists correlation in the data. For the subject indexes $i = 1, ..., n$, let the original pairs of sample to be (X_i, y_i) with y_i

denoting the response vector of the ith subject and X_i the corresponding covariate matrix. Then a bootstrap sample is represented as (X_i^*, y_i^*). According to the linear mixed model setting (3.1.1), let the vector $\xi_i = Z_i b_i + \epsilon_i$, $i = 1, ..., n$, then (3.1.1) becomes

$$y_i = X_i\beta + \xi_i, \quad i = 1, ..., n$$

and the associated bootstrap sample (X_i^*, y_i^*) can be expressed as

$$y_i^* = X_i^*\beta + \xi_i^*, \quad i = 1, ..., n.$$

With respect to the matrix notation (3.2), we have

$$\begin{aligned} Y &= X\beta + \xi \quad \text{and} \\ Y^* &= X^*\beta + \xi^* \end{aligned} \tag{5.43}$$

, where ξ and ξ^* are $N \times 1$ vectors with $\xi = (\xi_1', ..., \xi_n')'$ and $\xi^* = (\xi_1^{*\prime}, ..., \xi_n^{*\prime})'$, respectively. We can see from the expression (5.43) that, similar to the semi-parametric bootstrap, the bootstrap distribution of ξ^* is the same as the empirical distribution of the parameter ξ. let V^* be the variance covariance matrix of ξ^* and then V^* is a positive definite block diagonal matrix of n blocks of each block being $\frac{1}{n}\sum_{i=1}^{n}(\xi - \bar{\xi})(\xi - \bar{\xi})'$ with $\bar{\xi}$ to be the mean of ξ.As

$$\bar{\xi} = E(\xi) = E(Z_i b_i + \epsilon_i) = 0,$$

the corresponding block of the matrix V^* can be denoted by $\frac{1}{n}\sum_{i=1}^{n}\xi\xi'$.

Referring to the integrated log quasi-likelihood function (5.40), the corresponding log quasi-likelihood function of a bootstrap sample with respect to X^* and Y^* via non-parametric bootstrap

can be represented as

$$Q(\beta; X^*Y^*, I) = \frac{1}{2}D'(Y^* - X^*\beta)'V^{-1}(Y^* - X^*\beta).$$

Then, by plugging in the variance covariance V, we have

$$\begin{aligned}
E_*\{Q(\beta; X^*Y^*, I)\} &= \frac{1}{2}E_*\{D'(Y^* - X^*\beta)'V^{-1}(Y^* - X^*\beta)\} \\
&= \frac{1}{2}D'E_*||V^{-\frac{1}{2}}(Y^* - X^*\beta)||^2 \\
&= \frac{1}{2}D'E_*||V^{-\frac{1}{2}}(\xi^*)||^2 \\
&= \frac{1}{2}D'\{\text{trace}(V^{-1}V^*) + E_*(\xi^*)'V^{-1}E_*(\xi^*)\} \\
&= \frac{1}{2}D'\text{trace}(V^{-1}V^*) \\
&= \frac{1}{2}D'\xi'V^{-1}\xi \\
&= \frac{1}{2}D'||V^{-\frac{1}{2}}\xi||^2 \\
&= \frac{1}{2}D'||V^{-\frac{1}{2}}(Y - X\beta)||^2 \\
&= \frac{1}{2}D'(Y - X\beta)'V^{-1}(Y - X\beta) \\
&= Q(\beta; X, Y, I).
\end{aligned}$$

Thus, the **Assumption 3** holds true under non-parametric bootstrap framework.

To summary, we have showed that the **Assumption 3** is valid for any β of $\mathcal{B}$ under both the semi-parametric and non-parametric bootstrap approaches. Specially, when $\beta = \hat{\beta}$, which is an estimator of β by solving the corresponding quasi-score equations, we have the following assumption:

Assumption 3b. $E_*Q(\hat{\beta}; Y^*, I) = Q(\hat{\beta}; Y, I)$, where $\hat{\beta}$ is the estimator by solving the corresponding quasi-score equations.

The last assumption is provided to establish the asymptotic property of the second derivatives of the quasi-likelihood functions of both the original sample and bootstrap samples. The proof of

this assumption is also given.

Assumption 4. Let $\hat{\varphi}(Y,\beta) = -\frac{\partial^2}{\partial\beta\partial\beta'}\, Q(\beta; Y, I)$ and $\hat{\varphi}(Y^*,\beta) = -\frac{\partial^2}{\partial\beta\partial\beta'}\, Q(\beta; Y^*, I)$ be the observed fisher information over the original sample Y and bootstrap sample Y^*, respectively. Then as $n \to \infty$,

$$\begin{aligned} \hat{\varphi}(Y,\beta)/n \quad &\longrightarrow \quad \bar{\varphi}(\beta) \quad \text{a.s.} \quad \text{and} \\ \hat{\varphi}(Y^*,\beta)/n \quad &\longrightarrow \quad \bar{\varphi}_B(\beta) \quad \text{a.s.,} \end{aligned}$$

where $\bar{\varphi}(\beta)$ and $\bar{\varphi}_B(\beta)$ are two positive definite matrices. Moreover, $\bar{\varphi}(\bar{\beta}) = \bar{\varphi}_B(\bar{\beta})$.

Proof. From **Assumption 1**, we know that both the two quantities $\hat{\varphi}(Y,\beta)$ and $\hat{\varphi}(Y^*,\beta)$ exist and so do the corresponding limits $\hat{\varphi}(Y,\beta)/n$ and $\hat{\varphi}(Y^*,\beta)/n$. Let

$$\bar{\varphi}(\beta) = \lim_{n\to\infty} \frac{1}{n} E_0\{\hat{\varphi}(Y,\beta,I)\} \quad \text{and} \quad \bar{\varphi}_B(\beta) = \lim_{n\to\infty} \frac{1}{n} E_0 E_*\{\hat{\varphi}(Y^*,\beta,I)\}.$$

We claim that

$$\lim_{n\to\infty} \frac{1}{n}\hat{\varphi}(Y,\beta) = \bar{\varphi}(\beta) \quad \text{a.s.} \quad \text{and} \lim_{n\to\infty} \frac{1}{n}\hat{\varphi}_B(Y^*,\beta) = \bar{\varphi}(\beta) \quad \text{a.s..}$$

Next, we will show that $\bar{\varphi}(\beta) = \bar{\varphi}_B(\beta)$ under the semi-parametric and non-parametric boot-

strapping approaches. Based on **Assumptions 1** and **3**, we have

$$\begin{aligned}
\lim_{n\to\infty}\frac{1}{n}\hat{\varphi}(Y^*,\beta) &= \bar{\varphi}_B(\beta) \quad \text{a.s.} \\
&= \lim_{n\to\infty}\frac{1}{n}E_0E_*\{\hat{\varphi}(Y^*,\beta)\} \\
&= \lim_{n\to\infty}\frac{1}{n}E_0E_*\{-\frac{\partial^2}{\partial\beta\partial\beta'}Q(\beta;Y^*,I)\} \\
&= \lim_{n\to\infty}\frac{1}{n}E_0\{-\frac{\partial^2}{\partial\beta\partial\beta'}E_*\{Q(\beta;Y^*,I)\}\} \\
&= \lim_{n\to\infty}\frac{1}{n}E_0\{-\frac{\partial^2}{\partial\beta\partial\beta'}Q(\beta;Y,I)\} \\
&= \lim_{n\to\infty}\frac{1}{n}E_0\{\hat{\varphi}(Y,\beta)\} \\
&= \bar{\varphi}(\beta) \quad \text{a.s..}
\end{aligned}$$

As a special case, when $\beta = \bar{\beta}$ which is the global maximum of the log quasi-likelihood function $Q(\beta;Y)$, we have $\bar{\varphi}(\bar{\beta}) = \bar{\varphi}_B(\bar{\beta})$. Note that the $\bar{\beta}$ is not necessarily the global maximum under the independent model and can be the estimators over quasi-likelihood under any correlation structures. As the **Assumption 3** is validated based on the independent model as well as all the calculations involving quasi-likelihood functions for all the β in the parameter space $\mathcal{B}$, the global maximum of β under the independent model should also satisfy the **Assumption 4**. Therefore we abuse the notation $\bar{\beta}$ a little bit to simplify all the expressions.

5.3.1.1 Proof of Asymptotic Equivalence of QAICb1 and QAICb2.

After establishing the **Assumptions 1 - 4**, we can show the asymptotic equivalence of the two variants b1 and b2 defined by equations (5.30) and (5.35), respectively.

Consider a second-order expansion of the $-2Q(\hat{\beta};Y^*,I)$ about $\hat{\beta}^*$ as

$$-2Q(\hat{\beta};Y^*,I) = -2Q(\hat{\beta};Y^*,I) + (\hat{\beta}-\hat{\beta}^*)'\hat{\varphi}(Y^*,\beta^*)(\hat{\beta}-\hat{\beta}^*) \tag{5.44}$$

where β^* is a vector whose value is between $\hat{\beta}$ and $\hat{\beta}^*$.

By taking expectations with respect to the bootstrap distribution on both sides of the equation

(5.44), and referring to the consistency of $\hat{\beta}$ and $\hat{\beta}^*$ along with **Assumption 4**, we have

$$\begin{aligned}
&E_*\{-2Q(\hat{\beta}; Y^*, I)\} \\
&= E_*\{-2Q(\hat{\beta}^*; Y^*, I) + (\hat{\beta} - \hat{\beta}^*)'\hat{\varphi}(Y^*, \beta^*)(\hat{\beta} - \hat{\beta}^*)\} \\
&= E_*\{-2Q(\hat{\beta}^*; Y^*, I)\} + E_*\{(\hat{\beta} - \hat{\beta}^*)'\hat{\varphi}(Y^*, \beta^*)(\hat{\beta} - \hat{\beta}^*)\} \\
&= E_*\{-2Q(\hat{\beta}^*; Y^*, I)\} + E_*\{m(\hat{\beta} - \hat{\beta}^*)'\bar{\varphi}_B(\bar{\beta})(\hat{\beta} - \hat{\beta}^*)\}(1 + o(1)) \quad \text{a.s.}
\end{aligned}$$

and then

$$\begin{aligned}
&E_*\{-2Q(\hat{\beta}; Y^*, I)\} - E_*\{-2Q(\hat{\beta}^*; Y^*, I)\} \\
&= E_*\{m(\hat{\beta} - \hat{\beta}^*)'\bar{\varphi}_B(\bar{\beta})(\hat{\beta} - \hat{\beta}^*)\}(1 + o(1)) \quad \text{a.s..}
\end{aligned} \tag{5.45}$$

Similarly, consider a second-order expansion of the $-2Q(\hat{\beta}^*; Y, I)$ about $\hat{\beta}$ as

$$-2Q(\hat{\beta}^*; Y, I) = -2Q(\hat{\beta}; Y, I) + (\hat{\beta}^* - \hat{\beta})'\hat{\varphi}(Y, \beta^{**})(\hat{\beta}^* - \hat{\beta}) \tag{5.46}$$

where β^{**} is a vector lying between $\hat{\beta}^*$ and $\hat{\beta}$.

By taking expectations with respect to the bootstrap distribution on both sides of the equation (5.46), and referring to the consistency of $\hat{\beta}$ and $\hat{\beta}^*$ as well as the **Assumption 4**, we have

$$\begin{aligned}
&E_*\{-2Q(\hat{\beta}^*; Y, I)\} \\
&= E_*\{-2Q(\hat{\beta}; Y, I) + (\hat{\beta} - \hat{\beta}^*)'\hat{\varphi}(Y, \beta^{**})(\hat{\beta} - \hat{\beta}^*)\} \\
&= E_*\{-2Q(\hat{\beta}; Y, I)\} + E_*\{(\hat{\beta} - \hat{\beta}^*)'\hat{\varphi}(Y, \beta^{**})(\hat{\beta} - \hat{\beta}^*)\} \\
&= E_*\{-2Q(\hat{\beta}; Y, I)\} + E_*\{m(\hat{\beta} - \hat{\beta}^*)'\bar{\varphi}(\bar{\beta})(\hat{\beta} - \hat{\beta}^*)\}(1 + o(1)) \quad \text{a.s.}
\end{aligned}$$

and then

$$\begin{aligned} & E_*\{-2Q(\hat{\beta}^*;Y,I)\} - E_*\{-2Q(\hat{\beta};Y,I)\} \\ & = E_*\{m(\hat{\beta}-\hat{\beta}^*)'\bar{\varphi}(\bar{\beta})(\hat{\beta}-\hat{\beta}^*)\}(1+o(1)) \quad \text{a.s..} \end{aligned} \tag{5.47}$$

From **Assumption 4**, we know that $\bar{\varphi}_B(\bar{\beta}) = \bar{\varphi}(\bar{\beta})$, then we have

$$E_*\{m(\hat{\beta}-\hat{\beta}^*)'\bar{\varphi}_B(\bar{\beta})(\hat{\beta}-\hat{\beta}^*)\} = E_*\{m(\hat{\beta}-\hat{\beta}^*)'\bar{\varphi}(\bar{\beta})(\hat{\beta}-\hat{\beta}^*)\}.$$

Combining the two asymptotic results (5.45) and (5.47), we have

$$\begin{aligned} & E_*\{-2Q(\hat{\beta};Y^*,I)\} - E_*\{-2Q(\hat{\beta}^*;Y^*,I)\} \\ & = E_*\{-2Q(\hat{\beta}^*;Y,I)\} - E_*\{-2Q(\hat{\beta};Y,I)\}(1+o(1)) \quad a.s.. \end{aligned}$$

By the definitions of b1, b2 as well as **Assumption 3**, when $n \to \infty$, we have

$$\begin{aligned} \text{b1} &= E_*\{-2Q(\hat{\beta}^*;Y,I)\} - E_*\{-2Q(\hat{\beta}^*;Y^*,I)\} \\ &= E_*\{-2Q(\hat{\beta}^*;Y,I)\} - E_*\{-2Q(\hat{\beta};Y^*,I)\} \\ &\quad + E_*\{-2Q(\hat{\beta};Y^*,I)\} - E_*\{-2Q(\hat{\beta}^*;Y^*,I)\} \\ &= E_*\{-2Q(\hat{\beta}^*;Y,I)\} - \{-2Q(\hat{\beta};Y,I)\} \\ &\quad + E_*\{-2Q(\hat{\beta};Y^*,I)\} - E_*\{-2Q(\hat{\beta}^*;Y^*,I)\} \\ &= E_*\{-2Q(\hat{\beta}^*;Y,I) - \{-2Q(\hat{\beta};Y,I)\}\} \\ &\quad + E_*\{-2Q(\hat{\beta}^*;Y,I)\} - E_*\{-2Q(\hat{\beta};Y,I)\}(1+o(1)) \quad a.s. \\ &= 2E_*\{-2Q(\hat{\beta}^*;Y,I) - \{-2Q(\hat{\beta};Y,I)\}\}(1+o(1)) \quad a.s. \\ &= \text{b2}(1+o(1)) \quad a.s. \end{aligned}$$

Therefore, the two variants b1 and b2 are asymptotically equivalent, which leads to the asymptotic equivalence of QAICb1 and QAICb2.

5.3.2 Consistency of QAICb1 and QAICb2

In this section, we would like to show that b1 and b2 defined by equations (5.30) and (5.35) are consistent estimators of the bias correction term

$$E_0[E_0\{-2Q(\beta;Y,I)\}|_{\beta=\hat{\beta}}] - E_0[-2Q(\hat{\beta};Y,I)]. \tag{5.48}$$

To prove the consistence of QAICb1 and QAICb2, two lemmas **Lemma 1** and **Lemma 2** will be established. It should be pointed out that the term (5.48) is defined through the independent model assumption and ignoring the over-dispersion parameter. Thus, these two lemmas will also be validated using the independent assumption.

Let the quantity $\mathcal{H}_B$ be defined as the following

$$\mathcal{H}_B = E_*\{n(\hat{\beta}-\hat{\beta}^*)'\bar{\varphi}_B(\bar{\beta})(\hat{\beta}-\hat{\beta}^*)\} = E_*\{n(\hat{\beta}-\hat{\beta}^*)'\bar{\varphi}(\bar{\beta})(\hat{\beta}-\hat{\beta}^*)\}.$$

From the previous section 5.3.1, we know the two variants b1 and b2 are asymptotically equivalent to $2\mathcal{H}_B$. In other words, to prove the consistence of b1 and b2 is the same to show that $2\mathcal{H}_B$ is the consistent estimator of the bias (5.48). To start, we first define the term $\hat{\tau}(\beta,Y)$ using the first derivative of log quasi-likelihood functions by

$$\hat{\tau}(\beta,Y) = \{\frac{\partial}{\partial\beta}Q(\beta;Y,I)\frac{\partial}{\partial\beta'}Q(\beta;Y,I)\}.$$

By **Assumption 1**, similarly to the $\hat{\varphi}(\beta,Y)$ in the **Assumption 4**, the limit of $\hat{\tau}(\beta,Y)/n$ exists and if the limit is defined by $\bar{\tau}(\beta)$, we have

$$\begin{aligned}\bar{\tau}(\beta) &= \lim_{n\to\infty}\frac{1}{n}E_0\{\hat{\tau}(\beta,Y)\} \quad \text{and} \\ \lim_{n\to\infty}\frac{1}{n}\hat{\tau}(\beta,Y) &= \bar{\tau}(\beta) \quad \text{a.s..}\end{aligned} \tag{5.49}$$

Lemma 1.

$$\lim_{m\to\infty} \mathcal{H}_B = \text{tr}\{\bar{\tau}(\bar{\beta})\bar{\varphi}(\bar{\beta})^{-1}\} \quad \text{a.s..}$$

Proof of Lemma 1. Consider a first-order Taylor expansion of $\frac{\partial}{\partial\beta}Q(\hat{\beta};Y,I)$ around $\hat{\beta}^*$. With $\hat{\beta}$ being the estimator that maximizes $Q(\beta;Y,I)$, we have

$$\begin{aligned} 0 &= \frac{\partial}{\partial\beta}Q(\hat{\beta};Y,I) \\ &= \frac{\partial}{\partial\beta}Q(\hat{\beta}^*;Y,I) + \frac{\partial^2}{\partial\beta\partial\beta'}Q(\beta_\alpha;Y,I)(\hat{\beta}-\hat{\beta}^*), \end{aligned} \tag{5.50}$$

where β_α is a random vector between $\hat{\beta}$ and $\hat{\beta}^*$. By solving the equation (5.50), we have

$$\begin{aligned} \hat{\beta}-\hat{\beta}^* &= -\{\frac{\partial^2}{\partial\beta\partial\beta'}Q(\beta_\alpha;Y,I)\}^{-1}\frac{\partial}{\partial\beta}Q(\hat{\beta}^*;Y,I) \\ &= -\{\hat{\varphi}(Y,\beta_\alpha)\}^{-1}\frac{\partial}{\partial\beta}Q(\hat{\beta}^*;Y,I). \end{aligned} \tag{5.51}$$

Furthermore, by **Assumption 4**, we know

$$\begin{aligned} \lim_{n\to\infty} n(\hat{\beta}-\hat{\beta}^*) &= \lim_{n\to\infty} -\{\frac{1}{n}\hat{\varphi}(Y,\beta_\alpha)\}^{-1}\frac{\partial}{\partial\beta}Q(\hat{\beta}^*;Y,I) \\ &= \{-\bar{\varphi}(\bar{\beta})\}^{-1}\frac{\partial}{\partial\beta}Q(\hat{\beta}^*;Y,I) \quad \text{a.s..} \end{aligned} \tag{5.52}$$

By substituting the $\hat{\beta}-\hat{\beta}^*$ of $\mathcal{H}_B$ using (5.51) and applying the corresponding asymptotic property (5.52) as well as the consistency $\hat{\beta}$ and $\hat{\beta}^*$, we have

$$\begin{aligned} \lim_{n\to\infty} \mathcal{H}_B &= \lim_{n\to\infty} E_*\{n(\hat{\beta}-\hat{\beta}^*)'\bar{\varphi}(\bar{\beta})(\hat{\beta}-\hat{\beta}^*)\} \\ &= \lim_{n\to\infty} E_*\{\frac{1}{n}\{n(\hat{\beta}-\hat{\beta}^*)'\}\bar{\varphi}(\bar{\beta})\{n(\hat{\beta}-\hat{\beta}^*)\}\} \\ &= \lim_{n\to\infty} E_*\{\frac{1}{n}\frac{\partial}{\partial\beta'}Q(\hat{\beta}^*;Y,I)\{-\bar{\varphi}(\bar{\beta})\}^{-1}\bar{\varphi}(\bar{\beta})\{-\bar{\varphi}(\bar{\beta})\}^{-1}\frac{\partial}{\partial\beta}Q(\hat{\beta}^*;Y,I)\} \quad \text{a.s.} \\ &= \lim_{n\to\infty} E_*\text{tr}\{\{\frac{1}{n}\{\frac{\partial}{\partial\beta}Q(\hat{\beta}^*;Y,I)\frac{\partial}{\partial\beta'}Q(\hat{\beta}^*;Y,I)\}\}\bar{\varphi}(\bar{\beta})^{-1}\} \quad \text{a.s.} \\ &= \lim_{n\to\infty} E_*\text{tr}\{\{\frac{1}{n}\hat{\psi}(\hat{\beta}^*,Y)\}\bar{\varphi}(\bar{\beta})^{-1}\} \quad \text{a.s..} \end{aligned} \tag{5.53}$$

By using the (5.49) and the consistency of $\hat{\beta}^*$, we have

$$\lim_{n\to\infty} \frac{1}{n}\hat{\tau}(\hat{\beta}^*, Y) = \bar{\tau}(\bar{\beta}). \tag{5.54}$$

Combining (5.53) and (5.54), it can be shown that

$$\lim_{n\to\infty} \mathcal{H}_B = \text{tr}\{\bar{\tau}(\bar{\beta})\bar{\varphi}(\bar{\beta})^{-1}\} \quad \text{a.s.},$$

which validates the **Lemma 1**.

Next, we will provide **Lemma 2** with the corresponding proof of **Lemma 2**.

Lemma 2. $2\mathcal{H}_B$ is a consistent estimator of the bias adjustment (5.48).

Proof of Lemma 2. Similar to the proof of the asymptotic equivalence of the two variants b1 and b2, Taylor expansion is used to construct the equivalence of two expectations. First, we take a second order expansion of $E_0\{-2Q(\beta; Y, I)\}|_{\beta=\hat{\beta}}$ about $\bar{\beta}$ to obtain

$$\begin{aligned}
E_0\{-2Q(\beta; Y, I)\}|_{\beta=\hat{\beta}} &= E_0\{-2Q(\bar{\beta}; Y, I)\} + E_0\{-\frac{\partial}{\partial\beta}2Q(\bar{\beta}; Y, I)\}(\hat{\beta}-\bar{\beta}) \\
&\quad + (\hat{\beta}-\bar{\beta})' E_0\{-\frac{\partial^2}{\partial\beta\partial\beta'}2Q(\beta_\gamma; Y, I)\}(\hat{\beta}-\bar{\beta}) \\
&= E_0\{-2Q(\bar{\beta}; Y, I)\} \\
&\quad + (\hat{\beta}-\bar{\beta})' E_0\{-\frac{\partial^2}{\partial\beta\partial\beta'}2Q(\beta_\gamma; Y, I)\}(\hat{\beta}-\bar{\beta}) \\
&= E_0\{-2Q(\bar{\beta}; Y, I)\} + (\hat{\beta}-\bar{\beta})' E_0\{\hat{\varphi}(Y, \beta_\gamma)\}(\hat{\beta}-\bar{\beta}),
\end{aligned}$$

where β_γ is a random vector between $\hat{\beta}$ and $\bar{\beta}$. Then we have

$$E_0\{-2Q(\beta; Y, I)\}|_{\beta=\hat{\beta}} - E_0\{-2Q(\bar{\beta}; Y, I)\} = (\hat{\beta}-\bar{\beta})' E_0\{\hat{\varphi}(Y, \beta_\gamma)\}(\hat{\beta}-\bar{\beta}). \tag{5.55}$$

Next, consider a second order expansion of $-2Q(\bar{\beta}; Y, I)$ about $\hat{\beta}$ and $\hat{\beta}$ being obtained by

maximizing $Q(\beta; Y, I)$, we have

$$\begin{aligned}
-2Q(\bar{\beta}; Y, I) &= -2Q(\hat{\beta}; Y, I) - \frac{\partial}{\partial \beta} 2Q(\hat{\beta}; Y, I)(\bar{\beta} - \hat{\beta}) \\
&\quad + (\bar{\beta} - \hat{\beta})\{-\frac{\partial^2}{\partial \beta \partial \beta'} 2Q(\beta_\delta; Y, I)\}(\bar{\beta} - \hat{\beta}) \\
&= -2Q(\hat{\beta}; Y, I) + (\bar{\beta} - \hat{\beta})\{-\frac{\partial^2}{\partial \beta \partial \beta'} 2Q(\beta_\delta; Y, I)\}(\bar{\beta} - \hat{\beta}) \\
&= -2Q(\hat{\beta}; Y, I) + (\hat{\beta} - \bar{\beta})' \hat{\varphi}(Y, \beta_\delta)(\hat{\beta} - \bar{\beta}),
\end{aligned}$$

where β_δ is a random vector between $\hat{\beta}$ and $\bar{\beta}$. Similarly, we have

$$-2Q(\bar{\beta}; Y, I) - \{-2Q(\hat{\beta}; Y, I)\} = (\hat{\beta} - \bar{\beta})' \hat{\varphi}(Y, \beta_\delta)(\hat{\beta} - \bar{\beta}). \tag{5.56}$$

Take the expectation on both sides of the equation (5.56) with respect to the true model:

$$E_0\{-2Q(\bar{\beta}; Y, I)\} - E_0\{-2Q(\hat{\beta}; Y, I)\} = E_0\{(\hat{\beta} - \bar{\beta})' \hat{\varphi}(Y, \beta_\delta)(\hat{\beta} - \bar{\beta})\} \tag{5.57}$$

and get the summation of the two equations (5.55) and (5.57), which gives us

$$\begin{aligned}
&E_0\{-2Q(\beta; Y, I)\}|_{\beta=\hat{\beta}} - E_0\{-2Q(\hat{\beta}; Y, I)\} \\
&\quad = (\hat{\beta} - \bar{\beta})' E_0\{\hat{\varphi}(Y, \beta_\gamma)\}(\hat{\beta} - \bar{\beta}) + E_0\{(\hat{\beta} - \bar{\beta})' \hat{\varphi}(Y, \beta_\delta)(\hat{\beta} - \bar{\beta})\}.
\end{aligned} \tag{5.58}$$

By taking expectation on both sides of the equation (5.58) with respect to the true model, we can obtain

$$\begin{aligned}
&E_0\{E_0\{-2Q(\beta; Y, I)\}|_{\beta=\hat{\beta}}\} - E_0\{-2Q(\hat{\beta}; Y, I)\} \\
&\quad = E_0\{(\hat{\beta} - \bar{\beta})' E_0\{\hat{\varphi}(Y, \beta_\gamma)\}(\hat{\beta} - \bar{\beta}) + E_0\{(\hat{\beta} - \bar{\beta})' \hat{\varphi}(Y, \beta_\delta)(\hat{\beta} - \bar{\beta})\}\}
\end{aligned} \tag{5.59}$$

and the term (5.59) is is asymptotically equivalent to

$$\lim_{n\to\infty} (\hat{\beta}-\bar{\beta})' E_0\{\hat{\varphi}(Y,\beta_\gamma)\}(\hat{\beta}-\bar{\beta}) + E_0\{(\hat{\beta}-\bar{\beta})'\hat{\varphi}(Y,\beta_\delta)(\hat{\beta}-\bar{\beta})\}. \tag{5.60}$$

So if we need to show that $2\mathcal{H}_B$ is asymptotically equivalent to the bias (5.48), it is the same to establish the asymptotic equivalence of the $2\mathcal{H}_B$ and (5.60).

According to **Assumption 4** and the consistence of $\hat{\beta}$, we have

$$\lim_{n\to\infty}\{\frac{1}{n}\hat{\varphi}(Y,\beta_\delta)\} = \lim_{n\to\infty} E_0\{\frac{1}{n}\hat{\varphi}(Y,\beta_\gamma)\} = \bar{\varphi}(\bar{\beta}) \quad \text{a.s..} \tag{5.61}$$

Moreover, by taking the 1st order expansion of $\frac{\partial}{\partial\beta}Q(\hat{\beta};Y,I)$ about $\bar{\beta}$, we have

$$\begin{aligned} 0 = \frac{\partial}{\partial\beta}Q(\hat{\beta};Y,I) &= \frac{\partial}{\partial\beta}Q(\bar{\beta};Y,I) + \frac{\partial^2}{\partial\beta\partial\beta'}Q(\beta_\epsilon;Y,I)(\hat{\beta}-\bar{\beta}) \\ \Longrightarrow \hat{\beta}-\bar{\beta} &= -\{\frac{\partial^2}{\partial\beta\partial\beta'}Q(\beta_\epsilon;Y,I)\}^{-1}\frac{\partial}{\partial\beta}Q(\bar{\beta};Y,I) \\ &= -\hat{\varphi}(Y,\beta_\epsilon)^{-1}\frac{\partial}{\partial\beta}Q(\bar{\beta};Y,I), \end{aligned} \tag{5.62}$$

where β_ϵ is a random vector between $\hat{\beta}$ and $\bar{\beta}$.

Finally, by the previously established (5.49), (5.61) (5.62) and **Lemma 1** along with the con-

sistency of $\hat{\beta}$, the limit (5.60) can be reduced as:

$$
\begin{aligned}
&\lim_{n\to\infty} \ (\hat{\beta}-\bar{\beta})' E_0\{\hat{\varphi}(Y,\beta_\gamma)\}(\hat{\beta}-\bar{\beta}) + E_0\{(\hat{\beta}-\bar{\beta})'\hat{\varphi}(Y,\beta_\delta)(\hat{\beta}-\bar{\beta})\} \\
&= 2\lim_{n\to\infty}\{\frac{1}{n}\{n(\hat{\beta}-\bar{\beta})'\}\bar{\varphi}(\bar{\beta})\{n(\hat{\beta}-\bar{\beta})\}\} \\
&= 2\lim_{n\to\infty}\{\frac{1}{n}\{\frac{\partial}{\partial\beta'}Q(\bar{\beta};Y,I)\}\{\frac{1}{n}\hat{\varphi}(Y,\beta_\epsilon)\}^{-1}\bar{\varphi}(\bar{\beta})\{\frac{1}{n}\hat{\varphi}(Y,\beta_\epsilon)\}^{-1}\}\{\frac{\partial}{\partial\beta}Q(\bar{\beta};Y,I)\} \\
&= 2\lim_{n\to\infty}\{\frac{1}{n}\{\{\frac{\partial}{\partial\beta'}Q(\bar{\beta};Y,I)\}\bar{\varphi}(\bar{\beta})^{-1}\}\bar{\varphi}(\bar{\beta})\{\bar{\varphi}(\bar{\beta})^{-1}\{\frac{\partial}{\partial\beta}Q(\bar{\beta};Y,I)\}\}\} \quad \text{a.s.} \\
&= 2\lim_{n\to\infty}\text{tr}\{\{\frac{1}{n}\{\frac{\partial}{\partial\beta}Q(\bar{\beta};Y,I)\frac{\partial}{\partial\beta'}Q(\bar{\beta};Y,I)\}\}\bar{\varphi}(\bar{\beta})^{-1}\} \quad \text{a.s.} \\
&= 2\lim_{n\to\infty}\text{tr}\{\{\frac{1}{n}\hat{\tau}(Y,\bar{\beta})\}\bar{\varphi}(\bar{\beta})^{-1}\} \quad \text{a.s.} \\
&= 2\text{tr}\{\bar{\tau}(\bar{\beta})\bar{\varphi}(\bar{\beta})^{-1}\} \quad \text{a.s.} \\
&= 2\lim_{n\to\infty}\mathcal{H}_B \quad \text{a.s..}
\end{aligned}
$$

Therefore, we have established the asymptotic equivalence of $2\mathcal{H}_B$ and the limit (5.60), which completes the proof of **Lemma 2**.

Based on **Lemma 2**, we know that the two variants b1 and b2 are asymptotically equivalent to the bias (5.48) and more specifically, they are consistent estimators of the bias correction term (5.48).

5.3.3 Extension to Generalized Linear Models with Random Effects

As the over-dispersion parameter ϕ is scalar to the quasi-likelihood function, the proof of the equivalence and consistency of QAICb1 and QAICb2 in terms of generalized linear model with random effects is the same as in the linear mixed model except the addition of the over-dispersion parameter. We will show the proof of modified **Assumption 3** using the log and logit link functions to validate the extension of QAICb1 and QAICb2 by establishing the following assumption:

Assumption 3*. $E_* Q(\beta,\phi;Y^*,I) = E_0 Q(\beta,\phi;Y,I)$.

5.3.3.1 Proof via Logit Link Function

A generalized linear model defined by a logit link is the logistic regression model which, using matrix notation, is given by

$$\text{logit}(\mu) = E(Y) = X\beta \quad \text{and} \quad \mu = \frac{\exp(X\beta)}{1+\exp(X\beta)}.$$

Unlike the construction of quasi-likelihood function for the linear mixed model, it is convenient to consider observations at the individual level in this case. For $i = 1, ..., n$ and $j = 1, ..., m$, let y_{ij} be the response value for the ith individual and jth measurement with $E(y_{ij}) = \mu_{ij}$ and X_{ij} be the jth covariates for ith individual. Then, using the independence assumption, the quasi-likelihood function $Q(\beta; Y, I)$ is given by

$$\begin{aligned} Q(\beta, \phi; Y, I) &= \sum_{i=1}^{n}\sum_{j=1}^{m}\{y_{ij}\log(\frac{\mu_{ij}}{1-\mu_{ij}}) + \log(1-\mu_{ij})\}/\phi \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}\{y_{ij}\log(\frac{\mu_{ij}}{1-\mu_{ij}})/\phi + \log(1-\mu_{ij})/\phi\} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}\{y_{ij}\log(\frac{X_{ij}\beta}{1-X_{ij}\beta})/\phi + \log(1-X_{ij}\beta)/\phi\}. \end{aligned}$$

Moreover, the bootstrap distribution of the original sample provides us with the following two asymptotic results

$$\begin{aligned} &E_*\{y^*_{ij}\log(\frac{X^*_{ij}\beta}{1-X^*_{ij}\beta})/\phi\} = \mu_{ij}\log(\frac{X_{ij}\beta}{1-X_{ij}\beta})/\phi \quad \text{and} \\ &E_*\{\log(1-X^*_{ij}\beta)/\phi\} = \log(1-X_{ij}\beta)/\phi. \end{aligned} \tag{5.63}$$

Then, by utilizing (5.63), we have

$$\begin{aligned}
E_*Q(\beta,\phi;Y^*,I) &= E_*\{\sum_{i=1}^{n}\sum_{j=1}^{m}\{y_{ij}^*\log(\frac{X_{ij}^*\beta}{1-X_{ij}^*\beta})/\phi+\log(1-X_{ij}^*\beta)/\phi\}\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m}E_*\{y_{ij}^*\log(\frac{X_{ij}^*\beta}{1-X_{ij}^*\beta})/\phi+\log(1-X_{ij}^*\beta)/\phi\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m}E_*\{y_{ij}^*\log(\frac{X_{ij}^*\beta}{1-X_{ij}^*\beta})/\phi\}+\sum_{i=1}^{n}\sum_{j=1}^{m}E_*\{\log(1-X_{ij}^*\beta)/\phi\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m}\mu_{ij}\log(\frac{X_{ij}\beta}{1-X_{ij}\beta})/\phi+\sum_{i=1}^{n}\sum_{j=1}^{m}\log(1-X_{ij}\beta)/\phi.
\end{aligned}$$

Meanwhile,

$$\begin{aligned}
E_0Q(\beta,\phi;Y,I) &= E_0\{\sum_{i=1}^{n}\sum_{j=1}^{m}\{y_{ij}\log(\frac{X_{ij}\beta}{1-X_{ij}\beta})/\phi+\log(1-X_{ij}\beta)/\phi\}\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m}E_0\{y_{ij}\log(\frac{X_{ij}\beta}{1-X_{ij}\beta})/\phi\}+\sum_{i=1}^{n}\sum_{j=1}^{m}\log(1-X_{ij}^*\beta)/\phi \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m}\mu_{ij}\log(\frac{X_{ij}\beta}{1-X_{ij}\beta})/\phi+\sum_{i=1}^{n}\sum_{j=1}^{m}\log(1-X_{ij}^*\beta)/\phi \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m}\mu_{ij}\log(\frac{X_{ij}\beta}{1-X_{ij}\beta})/\phi+\sum_{i=1}^{n}\sum_{j=1}^{m}\log(1-X_{ij}\beta)/\phi \\
&= E_*Q(\beta,\phi;Y^*,I),
\end{aligned}$$

which completes the proof of **Assumption 3*** in the logistic regression setting.

5.3.3.2 Proof via Log Link Function

The Poisson regression model is defined via the log link function as

$$\log(\mu)=E(Y)=X\beta \quad \text{and} \quad \mu=\exp(X\beta).$$

By the use of the independence assumption as well, the quasi-likelihood function $Q(\beta; Y, I)$ for the Poisson regression model is defined as

$$\begin{aligned} Q(\beta, \phi; Y, I) &= \sum_{i=1}^{n}\sum_{j=1}^{m}\{y_{ij}\log(\mu_{ij}) - \mu_{ij}\}/\phi \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}\{y_{ij}\log(\mu_{ij})/\phi - \mu_{ij}/\phi\} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}\{y_{ij}\log(X_{ij}\beta)/\phi - X_{ij}\beta/\phi\}. \end{aligned}$$

Similar to the logistic regression case, two asymptotic results are established according to the bootstrap distribution

$$\begin{aligned} &E_*\{y^*_{ij}\log(X^*_{ij}\beta)/\phi\} = \mu_{ij}\log(X_{ij}\beta)/\phi \quad \text{and} \\ &E_*\{X^*_{ij}\beta/\phi\} = X_{ij}\beta/\phi. \end{aligned} \tag{5.64}$$

Then, through the results (5.64), we have

$$\begin{aligned} E_*Q(\beta, \phi; Y^*, I) &= E_*\{\sum_{i=1}^{n}\sum_{j=1}^{m}\{y^*_{ij}\log(X^*_{ij}\beta)/\phi - X^*_{ij}\beta/\phi\ \} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} E_*\{y^*_{ij}\log(X^*_{ij}\beta)/\phi - X^*_{ij}\beta/\phi\} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} E_*\{y^*_{ij}\log(X^*_{ij}\beta)/\phi\} - \sum_{i=1}^{n}\sum_{j=1}^{m} E_*\{X^*_{ij}\beta/\phi\} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} \mu_{ij}\log(X_{ij}\beta)/\phi - \sum_{i=1}^{n}\sum_{j=1}^{m} X_{ij}\beta/\phi \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}\{\mu_{ij}\log(X_{ij}\beta) - X_{ij}\beta\}/\phi. \end{aligned}$$

And then

$$
\begin{aligned}
E_0 Q(\beta, \phi; Y, I) &= E_0\{\sum_{i=1}^{n}\sum_{j=1}^{m}\{y_{ij}\log(X_{ij}\beta)/\phi - X_{ij}\beta/\phi\}\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} E_0\{y_{ij}\log(X_{ij}\beta)/\phi - X_{ij}\beta/\phi\} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} E_0\{y_{ij}\log(X_{ij}\beta)/\phi\} - \sum_{i=1}^{n}\sum_{j=1}^{m} X_{ij}\beta/\phi \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} \mu_{ij}\log(X_{ij}\beta)/\phi - \sum_{i=1}^{n}\sum_{j=1}^{m} X_{ij}\beta/\phi \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} \{\mu_{ij}\log(X_{ij}\beta) - X_{ij}\beta\}/\phi \\
&= E_* Q(\beta, \phi; Y^*, I),
\end{aligned}
$$

which completes the proof of **Assumption 3*** in the Poisson regression setting. The **Assumption 3** is proved to be valid for all the β in the parametric space under the non-parametric bootstrapping strategy, as a special case, we have the **Assumption 3b***:

$$E_* Q(\hat{\beta}, \hat{\phi}; Y^*, I) = E_0 Q(\hat{\beta}, \hat{\phi}; Y, I).$$

The rest of the proof following **Assumption 3*** and **Assumption 3b*** will be the same as been shown in Sections 5.3.1 and 5.3.2.

CHAPTER 6 SIMULATION STUDIES

6.1 Discussion on Simulations

6.1.1 Incorporating GEE

QAICb1 and QAICb2 are easy to implement when GEE is utilized to estimate the model parameters under the quasi-likelihood setting. It has been mentioned before but worth being stressed again that the difficulty of computing the quasi-likelihood lies in the complex correlation especially when the number of observations within the same individual is large. Besides, McCullagh (1989) pointed out that the quasi-likelihood function $Q(\beta; Y, R)$ may not exist until certain requirements are met. Most importantly, the asymptotic properties of QAICb1 and QAICb2 are established under the independent assumption. So when computing the quasi-likelihood functions in my simulation, the log quasi-likelihood $Q(\beta; Y, I)$ under the independent correlation structure is used.

The method of GEE solves the quasi-score equation iteratively based on a specified working correlation matrix. Let $\hat{\beta}_G(R)$ denote the GEE estimator of β under any working correlation R for a candidate model. Then $Q(\hat{\beta}_G(R); Y, I)$ is the log quasi-likelihood function corresponding to $\hat{\beta}(R)$ under the independent assumption. So $Q(\hat{\beta}_G(R); Y, I)$ is calculated by directly plugging $\hat{\beta}_G(R)$ into the log quasi-likelihood equation.

With regard to the two model selection criteria QAICb1 and QAICb2, for each candidate model, we can acquire $Q(\hat{\beta}_G(R); Y, I)$ from the original dataset given the working correlation matrix R. For each bootstrap sample indexed by i, we can obtain the quasi-likelihood functions $Q(\hat{\beta}^*_G(R)(i); Y, I)$ and $Q(\hat{\beta}^*_G(R)(i); Y^*(i), I)$. More specifically, to compute the log quasi-likelihood $Q(\hat{\beta}^*_G(R)(i); Y, I)$, we have to consider both the estimated parameter from the bootstrap sample $\hat{\beta}^*_G(R)(i)$ and the response values Y from the original sample. Therefore, when incorpo-

rating GEE, QAICb1 and QAICb2 for a linear mixed model can be expressed as

$$\begin{aligned} \text{QAICb1} &= -2Q(\hat{\beta}_G(R); Y, I) + \frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}^*_G(R)(i); Y, I)}{Q(\hat{\beta}^*_G(R)(i); Y^*(i), I)} \\ \text{QAICb2} &= -2Q(\hat{\beta}_G(R); Y, I) + 2\Big\{\frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}^*_G(R)(i); Y, I)}{Q(\hat{\beta}_G(R); Y, I)}\Big\}. \end{aligned} \tag{6.1}$$

The model with the smallest value of QAICb1 or QAICb2 is considered to be the best model.

6.1.2 Remarks on the Bias Correction Terms

During the simulation studies (not included in this dissertation), it has been found that QAICb1 has relatively consistent performance across all scenarios regardless of the correlation structure of the sample. In particular, when the correlation within the data becomes larger, the QAICb1 gains more power on correctly selecting the most appropriate model. On the contrary, QAICb2 performs surprisingly different from QAICb1 that it is much less efficient in large correlation cases. More specifically, QAICb2 tend to underestimate larger candidate models like QIC(R) and is less likely to choose the best model with small dimensions especially when the working correlation matrix used in the GEE procedure is the very close to the true working structure. As a result, when the GEE is applied to obtain the estimators and the format of the correlation matrix is correctly specified, the QAICb2 is not as efficient as the QAICb1, which indicates a modification when computing QAICb1 and QAICb2.

After deep investigation, the inefficiency of QAICb2 may be caused by the corresponding estimation procedure that the log quasi-likelihood $Q(\hat{\beta}_G(R); Y, I)$ is used twice in both the likelihood part and the bias correlation terms of the QAICb2. We know that QIC(R) performs similarly to AIC by underestimating the corresponding QIC(R) values of larger candidate models because the bias correction term is not powerful enough to address the relatively large quasi-likelihood values brought by the large candidate models. The same issue will happen to QAICb2 as the bias correlation terms utilizes the log quasi-likelihood $Q(\hat{\beta}_G(R); Y, I)$ of the original data as the denominator. Therefore, a candidate model with a larger $Q(\hat{\beta}_G(R); Y, I)$ is likely to bring down the estimation

of the bias correction term b2 of QAICb2.

In addition, the same working correlation matrix is used for both the construction of quasi-likelihood function and the bias correction term. For example, if we fit the candidate model by GEE under the exchangeable working correlation matrix over the original data, the same working correlation is used again over the bootstrap data. It is natural for the selection criterion to use the same pre-specified working correlation throughout the whole selection process. Unfortunately, during the simulation studies, doing so results in the underestimation of the bias correction terms for both QAICb1 and QAICb2.

Another issue brought by the use of the same working correlation is that the performance of QAICb1 and QAICb2 will be affected by the mis-specification of the true correlation structure. Even though the GEE estimators are consistent under different working correlations, a mis-specified correlation would, to some degree, lower the estimation efficiency (Liang and Zeger, 1986). When the log quasi-likelihood function is not efficiently estimated, so does the corresponding bias correction term. Therefore, the corresponding model selection results will be less efficient.

Based on our above discussion, the fitting process of the bias correlation term needs modification to improve the selection accuracy of both QAICb1 and QAICb2 when GEE is utilized. Before going deep into the revision of our proposed criterion QAICb1 and QAICb2, we first focus on another criterion QAIC_u discussed in Section (5.1.3). Given a GEE estimator $\hat{\beta}_G(R)$ of the β under any working correlation R, $\text{QAIC}_u(R)$ is expressed as

$$\text{QAIC}_u(R) = -2Q(\hat{\beta}_G(R); Y, I) + 2k.$$

It has been found in the simulation studies that $\text{QAIC}_u(R)$ is more efficient when the correlation of the data is relatively large, compared to the $\text{QIC}(R)$. A possible reason would be the use of the bias correction term $2k$ for $\text{QAIC}_u(R)$. Notice that the term $2k$ is only associated with the dimension of the candidate models and independent of the correlation structure, which stimulated us to modify the bias corrections of QAICb1 and QAICb2 to be free from the correlation of the

original data.

Moreover, when computing Pan's QIC(R), the bias correction term requires the update from the original model using the independent working correlation. Following the above discussion about QAIC$_u(R)$ and QAIC$_u(R)$, we are motivated to modify our proposed QAICb1 and QAICb2 by estimating the two corresponding terms b1 and b2 using the independent working correlation matrix. Unlike QIC(R), which tends to underestimate the corresponding discrepancy when the correlation is large, our bootstrap-adjusted criteria QAICb1 and QAICb2 are efficient and consistent across different correlation structures by taking advantage of the independence working correlation.

As a summary, to obtain the two bootstrap-based variants of QAICb1 and QAICb2, the parameters should be estimated using the independent working correlation to avoid the inconsistent performance caused by the different correlation structures. Therefore, our two proposed model selection criteria are modified to be:

$$\text{QAICb1} = -2Q(\hat{\beta}_G(R); Y, I) + \frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}^*_G(I)(i); Y, I)}{Q(\hat{\beta}^*_G(I)(i); Y^*(i), I)},$$

$$\text{QAICb2} = -2Q(\hat{\beta}_G(R); Y, I) + 2\Big\{\frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}^*_G(I)(i); Y, I)}{Q(\hat{\beta}_G(I); Y, I)}\Big\}$$

where $\hat{\beta}^*_G(I)(i)$ and $\hat{\beta}_G(I)$ are GEE estimators from the bootstrap sample i and the original sample based on the independent working correlation, respectively.

6.2 Simulations in Linear Mixed Models

In this section, the performance of the selection criteria QIC(R), QAIC$_u(R)$, QAICb1 and QAICb2 are compared using the longitudinal data. There are total 10 candidate models with increasing order with the largest model containing all the 10 covariates. The true parameters are $\beta_0 = (2, 2, 1, 1, 0.5, 0, 0, 0, 0, 0, 0)$. The covariates $X_1, ..., X_{10}$ are independently generated from the standard normal distribution. The error term is generated from a normal distribution with mean 0 and standard deviations $\sigma = 1, 1.3$ and 1.5. Four different number of individuals $n = 25$, $n = 50$,

$n = 100$ and $n = 200$ are considered and for each individual, there are $m = 3$ repeated measurements. The number of bootstrap samples B is set to be 250 across all the scenarios. The true correlation matrix is chosen as EX(ρ) or AR(ρ) or a mixture of different correlation matrices with $\rho = 0.2, 0.4, 0.6, 0.8$ and the working correlation matrix is exchangeable or auto-regressive defined in Section 2.3.3.

This section is divided into two parts. The first part contains simulation studies via the non-parametric approach and the part part is based on the semi-parametric bootstrap. Within both parts of simulations, we will investigate the model selection performance of QAICb1 and QAICb2 using a set of nested candidate models under different true correlation and working correlation structures. When non-parametric is utilized, QAICb1 and QAICb2 are also examined using candidate models constructed by different combinations of predictor variables.

6.2.1 Simulations via Non-parametric Bootstrap Approach

6.2.1.1 True Correlation is Correctly Specified

In this section, we are going to discuss the simulation results under the non-parametric bootstrap setting mentioned in Section 5.1.1. Moreover, the form of the working correlation matrix is set to be the same as the true correlation where the estimators from GEE are the most consistent. Due to the nature of the longitudinal data, all bootstrap samples are obtained with replacement from the original sample on the individual level. Based on 1000 iterations, the outcomes are summarized in the following tables.

For all of the four methods, regardless of the correlation coefficient ρ and the standard deviation σ, increasing the sample size will also increase the corresponding performance, which is reasonable as these criteria are derived based on the asymptotic results. When n is relatively small ($n = 25$ or 50), however, QIC(R) performs poorly with the selection percentages are down by at least 10% when compared to others. Besides, QIC(R) heavily relies on the independent assumption that as the correlation ρ increases, the corresponding selection efficiency decreases dramatically. When $\rho = 0.8$, the selection accuracy of QIC(R) can hardly reach 50% across all scenarios, which makes

Table 6.1: True model selection rates under $\sigma = 1$ (non-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Exchangeable				Auto-regressive			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
25	QIC(R)	51.2	47	47.8	40.1	53.8	51.3	46.9	40.0
	QAIC$_u$(R)	72.2	75.6	83.9	86.8	74.9	77.3	81.1	88.9
	QAICb1	62.5	63.8	71.4	79.3	64.9	66.3	70.4	79.6
	QAICb2	68.7	69.1	76.4	81.6	70.0	72.2	74.4	83.0
50	QIC(R)	61.5	57.9	55.4	49.4	58.2	60.8	55.2	45.5
	QAIC$_u$(R)	74.2	79.2	85.6	92.6	71.8	78.4	83.7	89.9
	QAICb1	68.2	71.3	78.4	87.8	64.1	71.7	77.4	85.5
	QAICb2	70.2	74.3	80.7	89.4	66.9	74.0	78.7	86.2
100	QIC(R)	66.8	61.5	59.4	49.4	68.3	59.6	57.3	50.5
	QAIC$_u$(R)	74.4	76.7	85.5	91.4	74.8	76.3	83.3	91.7
	QAICb1	70.7	73.2	80.2	89.8	71.4	72.0	78.7	89.4
	QAICb2	72.2	74.3	81.7	90.3	72.1	72.9	80.1	90.1
200	QIC(R)	69	65.3	59.4	52.1	68.0	66.9	58.9	52.3
	QAIC$_u$(R)	74.4	77.8	84.6	93.7	71.9	77.7	83.1	93.1
	QAICb1	72.2	76.0	84.0	91.3	70.3	75.6	80.5	92.1
	QAICb2	72.9	76.1	84.3	92.2	71.0	76.0	80.5	92.6

it unable to draw reliable conclusions.

On the other hand, QAIC$_u$(R) out performs QIC(R) and is slightly better than our proposed QAICb1 and QAICb2 under different values of ρ when $\sigma = 1$. In contrast to the pattern shown by QIC(R), the efficiency of QAIC$_u$(R), together with QAICb1 and QAICb2, increases with the increase of ρ. While QAIC$_u$(R) seems to be a better choice with a small σ, it becomes less efficient when σ is relatively large. In the case where σ is 1.3 or 1.5 with a relatively small correlation coefficient ($\rho = 0.2$ or 0.4), the selection rates are less than 55%, which is much smaller compared to those of QAICb1 and QAICb2. What's more, the QAIC$_u$(R) is not consistent for different values of ρ although it performs better when ρ is large. The inconsistency of QAIC$_u$(R) caused by ρ is not expected prior to the simulation studies. By using $2k$ as the bias correction term, the changing values of ρ will only affect the quasi-likelihood estimation because $2k$ has nothing to do with the correlation structure but is associated with dimension of candidate models. Under the circumstances that ρ is small and σ is large, $2k$ may not be able to penalize the quasi-likelihood enough to increase the corresponding selection efficiency.

Table 6.2: True model selection rates under $\sigma = 1.3$ (non-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Exchangeable				Auto-regressive			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
25	QIC(R)	47.9	47.4	45.6	40.1	48.0	52.1	45.8	38.0
	QAIC$_u$(R)	45.4	53.8	62.4	73.7	46.2	56.4	58.7	70.7
	QAICb1	60.0	64.3	68.5	75.2	59.7	65.9	68.2	72.3
	QAICb2	64.6	70.1	73.1	78.1	66.2	70.0	71.2	75.6
50	QIC(R)	61.1	58.8	53.3	42.2	60.2	60.6	57.4	47.4
	QAIC$_u$(R)	49.6	54.1	63.1	77.6	45.4	56.2	66.2	79.0
	QAICb1	68.5	71.4	77.4	85.6	66.2	73.7	78.9	87.1
	QAICb2	71.5	73.8	79.4	86.6	68.6	75.2	81.4	87.7
100	QIC(R)	64.5	62.5	58.7	47.7	65.7	65.5	56.8	50.7
	QAIC$_u$(R)	46.7	54.1	65.6	78.9	45.9	55.1	61.2	77.8
	QAICb1	69.2	74.3	81.8	89.2	69.2	76.0	77.2	88.6
	QAICb2	70.8	75.3	82.8	89.5	70.2	76.8	78.7	89.0
200	QIC(R)	70.5	64.4	59.1	48.8	67.3	64.9	57.6	49.2
	QAIC$_u$(R)	47.8	53.1	63.4	79.5	45.0	53.1	62.6	78.3
	QAICb1	73.3	76.4	82.0	90.9	70.9	75.5	80.4	91.0
	QAICb2	74.3	75.9	82.7	91.9	71.4	76.7	80.7	91.3

Table 6.3: True model selection rates under $\sigma = 1.5$ (non-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Exchangeable				Auto-regressive			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
25	QIC(R)	48.8	49.1	38.0	41.5	49.3	46.0	45.6	38.6
	QAIC$_u$(R)	31.4	38.2	47.8	63.6	33.8	39.3	45.9	61.8
	QAICb1	54.4	59.6	64.6	70.7	59.1	58.9	62.8	69.8
	QAICb2	59.4	64.0	68.1	72.9	63.3	63.3	65.7	72.7
50	QIC(R)	63.1	59.7	53.1	45.4	62.3	58.9	54.7	46.1
	QAIC$_u$(R)	33.6	41.5	50.0	65.5	33.0	38.7	51.1	67.4
	QAICb1	67.0	71.7	72.8	83.7	67.8	69.7	76.3	83.5
	QAICb2	68.9	73.7	75.7	85.6	70.5	73.2	77.7	84.9
100	QIC(R)	65.9	64.7	57.5	51.2	67.6	62.8	57.8	46.7
	QAIC$_u$(R)	32.2	40.9	51.8	69.1	33.2	41.6	50.3	65.8
	QAICb1	69.7	74.5	79.7	88.9	70.5	73.6	81.2	87.0
	QAICb2	72.2	75.9	80.7	89.2	71.5	74.5	82.0	87.1
200	QIC(R)	66.9	64.7	60.1	50.3	72.0	66.8	57.8	50.9
	QAIC$_u$(R)	31.1	41.3	53.0	67.3	34.7	40.1	48.9	60.8
	QAICb1	70.0	76.5	81.3	90.4	74.6	76.8	81.3	90.1
	QAICb2	70.7	77.0	81.5	90.8	75.9	77.5	81.9	90.3

The two bootstrap-adjusted QAICb1 and QAICb2 are more efficient than QIC(R) in almost all the scenarios, regardless of independent update applied, especially where the true correlation ρ is relatively large. Benefiting from the bootstrap, our two proposed selection criteria are much more consistent than the other two under different n, which is demonstrated by almost all the tables. For example, in Table 6.2 with a exchangeable working correlation, $\rho = 0.6$ and $n = 100$, both the QIC(R) and QAIC$_u(R)$ have rates for around 60% while the QAICb1 and QAICb2 have more than 80% percentages. Even though with a small sample size $n = 25$, their 70% correct rates are better than the other two criteria.

QAICb1 and QAICb2 not only inherit the advantage of QAIC$_u(R)$ that being relatively consistent throughout different correlation coefficients, but also remain efficient with respect to various sizes of the noise σ. It can be seen from Tables 6.2, 6.3 that when the sample size is large, the QAICb1 and QAICb2 are much more efficient than QAIC$_u(R)$ especially for small correlations. More importantly, changing the variability in the noise will merely affect their efficiency. By updating the corresponding two correction terms with independent working correlation matrix, they have been able to overcome the weakness of QAIC$_u(R)$ under situations with a smaller correlation and larger variance.

The overall performance of QAICb1 and QAICb2 are almost the same when it comes to different forms of the true correlation structures, which means the two criteria are fairly consistent under various situations. In fact, all of the four selection criteria have similar performance across different correlation structures as long as the correlation matrix is correctly specified, which is reasonable because the GEE estimators are consistent and gain largest efficiency with the correct specification of true correlation.

To summary, in the cases where the non-parametric bootstrap approach is utilized, when the correlation ρ is relatively small, the QIC(R) can barely match QAICb1 and QAICb2 in large sample situation but when ρ is relatively large, QAICb1 and QAICb2 are much more efficient than QIC(R). Meanwhile, the QAIC$_u(R)$ is only slightly better than QAICb1 and QAICb2 for the small variance scenario. When there exists large noise and high correlation within individuals, the QAICb1 and

QAICb2 are more powerful than the two existing ones. Furthermore, the QAICb1 and QAICb2 retain relatively stale performance across all the simulation settings in this section, which assures us to draw reliable conclusions based on the model chosen by these two criteria.

6.2.1.2 True Correlation is Not Correctly Specified

In this section, simulation results are presented for situations where the true correlation matrix is not correctly specified, which means the true correlation and the working correlation matrices are in different formats. The corresponding performance of three different settings are investigated. For the first scenario, EX(ρ) is the true correlation and auto-regressive (AR) is the working correlation. The second scenario is the opposite of the first one with the AR(ρ) being the true correlation and exchangeable (EX) being the working correlation. The third case is to use a mixed true correlation structure that 30% of the observations are from EX(0.5), another 30% come from AR(0.5) and the rest 40% from a self-defined matrix as following:

$$\begin{bmatrix} 1 & 0.9 & 0.4 \\ 0.9 & 1 & 0.8 \\ 0.4 & 0.8 & 1 \end{bmatrix}$$

where both the exchangeable and auto-regressive working correlation will be considered.

Tables 6.4 to 6.6 summarize the simulation results for the scenario that the true correlation is EX(ρ) and the working correlation is auto-regressive. We observe that all of the four selection criteria have similar performance as when the working correlation is correctly specified. When n increases, the selection accuracy tend to increase regardless of the correlation ρ and the standard deviation σ. As the correlation ρ becomes larger, QIC(R) undergoes a loss in efficiency while the other three are more and more powerful. The increase of σ would heavily affect the consistency of QAIC$_u(R)$. Meanwhile, QAICb1 and QAICb2 are more consistent. For example, referring to Tables 6.4 and 6.5, when σ increases from 1.3 to 1.5 under $\rho = 0.6$ and $n = 200$, the selection accuracy of QAIC$_u(R)$ decreases from 67.4% to 53.4% but those of the QAICb1 and QAICb2 are

Table 6.4: True model selection rates under $\sigma = 1$ (non-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Auto-regressive				Exchangeable			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
25	QIC(R)	50.4	50.1	49.7	44.1	54.0	48.8	47.1	43.3
	QAIC$_u$(R)	74.0	76.8	82.7	90.5	76.7	74.5	80.8	86.6
	QAICb1	61.9	66.1	72.0	80.3	65.0	63.7	70.4	76.2
	QAICb2	67.4	71.2	75.5	83.1	70.0	68.9	74.9	79.3
50	QIC(R)	60.3	60.0	56.3	49.8	63.4	62.1	57.4	49.8
	QAIC$_u$(R)	71.9	78.5	81.8	91.5	74.4	77.4	81.6	88.9
	QAICb1	66.0	72.5	74.4	86.5	69.6	72.0	76.0	83.3
	QAICb2	68.1	74.9	76.8	88.1	72.0	73.9	77.4	84.7
100	QIC(R)	66.1	63.5	60.8	55.3	66.1	64.8	56.9	49.9
	QAIC$_u$(R)	75.0	77.4	84.3	91.6	73.9	76.7	80.9	90.8
	QAICb1	70.6	72.4	80.5	88.9	69.1	73.8	77.2	88.7
	QAICb2	71.4	74.7	81.9	89.3	70.2	74.8	78.3	88.8
200	QIC(R)	70.2	66.6	67.8	58.0	70.3	65.6	62.7	55.0
	QAIC$_u$(R)	75.6	77.1	85.7	92.1	74.9	77.0	84.0	91.0
	QAICb1	72.8	75.1	83.6	90.4	72.8	74.8	82.1	88.7
	QAICb2	73.5	75.9	84.7	90.7	73.2	75.6	82.1	89.5

Table 6.5: True model selection rates under $\sigma = 1.3$ (non-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Auto-regressive				Exchangeable			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
25	QIC(R)	50.4	49.3	46.1	43.3	50.0	47.5	44.6	39.2
	QAIC$_u$(R)	46.7	52.5	59.3	75.5	45.9	48.6	55.8	69.5
	QAICb1	59.0	62.0	66.2	77.0	62.2	63.4	63.0	72.9
	QAICb2	64.6	62.7	70.0	79.3	66.8	67.8	67.3	75.8
50	QIC(R)	59.3	59.3	53.0	49.2	62.5	60.8	56.9	48.0
	QAIC$_u$(R)	42.5	53.7	59.0	77.9	48.3	54.3	60.2	72.9
	QAICb1	64.9	70.8	75.0	85.3	68.1	70.6	75.9	82.9
	QAICb2	67.4	73.1	78.5	85.8	70.8	72.8	78.0	83.7
100	QIC(R)	70.1	64.0	62.0	54.8	67.3	64.9	58.2	54.5
	QAIC$_u$(R)	47.9	51.5	65.0	78.4	46.0	51.9	58.9	76.6
	QAICb1	72.5	72.8	82.0	89.4	69.5	72.5	76.7	88.3
	QAICb2	73.6	74.5	83.5	90.0	71.5	73.1	78.7	88.9
200	QIC(R)	70.1	65.3	67.7	57.7	70.4	68.0	62.1	54.6
	QAIC$_u$(R)	46.5	51.5	67.4	80.0	45.5	52.7	60.8	76.1
	QAICb1	72.9	74.1	83.2	91.5	72.4	76.2	78.5	89.0
	QAICb2	73.9	74.3	83.7	91.6	72.5	76.7	79.2	89.5

Table 6.6: True model selection rates under $\sigma = 1.5$ (non-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Auto-regressive				Exchangeable			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
25	QIC(R)	48.4	47.0	48.5	41.2	48.8	48.7	44.9	40.8
	QAIC$_u$(R)	34.7	38.9	51.3	61.5	32.8	37.7	45.5	59.7
	QAICb1	59.5	60.6	64.5	69.7	59.2	60.4	61.5	70.6
	QAICb2	63.1	63.8	67.8	71.5	62.4	64.5	65.8	73.8
50	QIC(R)	59.0	61.8	57.0	51.5	60.6	59.3	55.2	46.4
	QAIC$_u$(R)	33.0	41.8	52.6	68.4	31.7	36.8	46.7	62.2
	QAICb1	64.8	71.3	75.8	83.4	66.9	69.6	74.2	81.4
	QAICb2	67.3	73.8	77.0	85.9	69.7	72.1	76.3	82.7
100	QIC(R)	66.0	64.3	63.3	54.7	67.0	63.6	60.4	53.3
	QAIC$_u$(R)	30.9	40.2	54.0	68.6	31.9	38.1	48.1	63.1
	QAICb1	70.0	75.7	80.5	86.7	69.2	71.5	79.9	86.4
	QAICb2	70.3	76.8	81.8	87.2	70.9	73.9	79.9	87.3
200	QIC(R)	67.4	66.8	65.7	55.4	68.9	64.1	65.1	52.0
	QAIC$_u$(R)	31.7	38.2	53.4	70.1	30.8	34.7	48.5	62.9
	QAICb1	70.2	76.9	83.2	91.4	71.5	71.1	82.1	87.7
	QAICb2	71.2	76.8	84.3	91.2	72.5	71.1	82.3	88.2

almost the same. Although the QIC(R) is relatively consistent under different values of σ as well, the selection efficiency is much less than the two proposed criteria.

We have also noticed that the selection accuracy of QIC(R) is higher when the true correlation is mis-specified. For instance, from Tables 6.1 and 6.4 where the working correlation is auto-regressive, QIC(R) has an average of 3% increase while the other three selection criteria have an average of 1% decrease in the rates. If the true correlation is mis-classified, the GEE estimators will be less efficient even though they are overall consistent, which would explain why QAIC$_u$(R), QAICb1 and QAICb2 have sightly worse performance. However, we did not expect QIC(R) to be slightly more efficient even though the corresponding selection accuracy is much lower than those of QAICb1 and QAICb2.

6.2.1.3 Mixture Model Involving Different Correlation Structures

Tables 6.7 to 6.9 provide us with the simulation results when the true correlation is a mixture of several correlation matrices. The resulted true correlation is relatively large from the matrices

Table 6.7: True model selection rates under $\sigma = 1$ (non-parametric).

True Correlation	Mixture			Mixture		
Working Correlation	Exchangeable			Auto-Regressive		
n	50	100	200	50	100	200
QIC(R)	54.0	55.4	60.4	54.8	59.8	60.4
QAIC$_u(R)$	84.3	80.8	83.0	84.6	83.4	84.0
QAICb1	76.1	76.3	80.8	77.7	78.5	82.1
QAICb2	78.1	77.6	81.9	80.0	79.9	82.6

Table 6.8: True model selection rates under $\sigma = 1.3$ (non-parametric).

True Correlation	Mixture			Mixture		
Working Correlation	Exchangeable			Auto-Regressive		
n	50	100	200	50	100	200
QIC(R)	56.5	59.9	58.3	54.6	58.1	63.5
QAIC$_u(R)$	65.1	63.3	60.2	63.1	64.0	66.9
QAICb1	76.2	79.5	77.9	77.4	80.6	85.2
QAICb2	77.6	80.4	77.5	80.0	81.6	86.1

we choose. In the mixture correlation case, compared to the previous two scenarios when ρ is relatively large, there is a certain degree of decrease in the selection rates for all the four criteria, which is expected because the GEE estimators should undergo certain loss in estimating efficiency when the working correlation matrix is quite different from the true correlation structure. Based on Table 6.7, for $\sigma = 1$, QAIC$_u(R)$ is better than the QAICb1 and QAICb2 with QIC(R) having the worst performance, which follows the same pattern for a small σ. However, from the same table, when the number of observations n is 200, QAIC$_u(R)$, QAICb1 and QAICb2 have almost the same true model selection rates.

When σ increases to 1.3 or 1.5, the efficiency of QAICb1 and QAICb2 remains unchanged while the other two suffer a huge decrease in correctly selecting the true model. Referring to Table 6.9, no matter which working correlation matrix is utilized, QAICb1 and QAICb2 are about 25% more than the other two in the true model selection rates. Therefore in the mixed true correlation scenario, the simulation results show that QAICb1 and QAICb1 are the most consistent selection criteria with the ability to maintain a high selection efficiency.

Table 6.9: True model selection rates under $\sigma = 1.5$ (non-parametric).

True Correlation	Mixture			Mixture		
Working Correlation	Exchangeable			Auto-Regressive		
n	50	100	200	50	100	200
QIC(R)	55.8	57.5	60.4	55.8	56.6	61.2
QAIC$_u$(R)	50.6	50.2	50.2	50.6	50.1	52.2
QAICb1	76.5	79.2	80.8	76.5	81.2	83.1
QAICb2	78.4	79.8	81.5	78.4	82.3	82.9

Table 6.10: Selection results under $n = 100$ and $\sigma = 1.3$ (non-parametric).

True Correlation: EX(0.6)				Working Correlation: Exchangeable						
Criteria	M1	M2	M3	TRUE	M5	M6	M7	M8	M9	M10
QIC(R)	0	0	0	587	119	77	69	43	53	52
QAIC$_u$(R)	0	0	0	656	114	75	51	37	43	24
QAICb1	0	0	0	818	74	48	27	9	17	7
QAICb2	0	0	0	828	70	52	22	7	15	6

6.2.1.4 Discussion on Selecting Over-fitted Candidate Models

In the previous section, we investigate the true model selection rates of different model selection criteria. The simulation results demonstrate that QAICb1 and QAICb2 have more efficient and consistent overall performance than some existing criteria. In this section, we will go deeper into the simulation results to show that QAICb1 and QAICb2 are less likely to select over-fitted candidate models. Three table are provided using different model settings.

We can see from Tables 6.10 to 6.12 that all of four criteria have similar selection pattern that they are more likely to choose over-fitted candidate models than under-fitted candidate models. However, QAICb1 and QAICb2 will significantly reduce the impact from over-fitted candidate models. In other words, the true model selection rates of QAICb1 and QAICb2 are much higher than those of QIC(R) and QAIC$_u$(R) by not selecting overfitted models. Focusing on Table 6.10, the percentages of choosing "M5" as the final model are 11.9% and 11.4% for QIC(R) and QAIC$_u$(R) while the percentages for QAICb1 and QAICb2 are 7.4% and 7%, which is a noticeable decrease by around 4% on only one overfitted model. For the largest candidate model "M10", QIC(R) still has a selection rate of 5.2% while the corresponding rates for QAICb1 and QAICb2

Table 6.11: Selection results under $n = 100$ and $\sigma = 1.3$ (non-parametric).

True Correlation: AR(0.6)				Working Correlation: Exchangeable						
Criteria	M1	M2	M3	TRUE	M5	M6	M7	M8	M9	M10
QIC(R)	0	0	0	582	130	83	65	55	48	37
QAIC$_u$(R)	0	0	0	589	133	86	64	49	43	36
QAICb1	0	0	0	767	106	54	30	16	19	8
QAICb2	0	0	0	787	100	45	31	15	17	5

Table 6.12: Selection results under $n = 100$ and $\sigma = 1.3$ (non-parametric).

True Correlation: Mixture				Working Correlation: Auto-regressive						
Criteria	M1	M2	M3	TRUE	M5	M6	M7	M8	M9	M10
QIC(R)	0	0	0	581	133	81	46	55	46	58
QAIC$_u$(R)	0	0	0	640	130	78	39	38	35	40
QAICb1	0	0	0	806	116	42	12	8	6	10
QAICb2	0	0	0	816	114	38	10	7	7	8

are 0.7% and 0.6%.

When the true correlation is mis-specified, the true model selection rates for QAICb1 and QAICb2 are lowered. By comparing the simulation results from Tables 6.10 and 6.11, we can see that the major difference is caused by candidate model "M5". There is an increase in selecting "M5" as the final model by about 3% for QAICb2 and QAICb2 while QIC(R) has only 1% increase, whose difference is small though. This may be the reason that QIC(R) is more consistent with different working correlations than QAICb1 and QAICb2.

The simulation results in Table 6.12 demonstrate that when the true correlation is a mixture of different correlation matrices, QAICb1 and QAICb2 are less likely to select highly over-fitted candidate models compared to the results in the previous two tables. The selection rates for candidate models "M7" to "M10" are all around or less than 1% which are lower than the corresponding rates in the other two tables.

To summary, the simulation results demonstrate that QAICb1 and QAICb2 are more likely to avoid over-fitted candidate models than QIC(R) and QAIC$_u$(R) in addition to the more efficient and consistent performance. As a result, the highly over-fitted candidate models will be rarely chosen as the final model, which responds to the philosophy of model that has been discussed in

Table 6.13: Description of candidate models.

Model	Covariates	Model	Covariates
M1	X_1, X_2	M5	X_1, X_2, X_3, X_4
M2	X_1, X_2, X_3	M6	X_1, X_2, X_3, X_5
M3	X_1, X_2, X_4	M7	X_1, X_2, X_4, X_5
M4 (True)	X_1, X_2, X_5	M8	X_1, X_2, X_3, X_4, X_5

Section 1.1.

6.2.1.5 Simulations for Combinations of Predictor Variables

In the previous section, we investigate the selection performance of the proposed QAICb1 and QAICb2 through nested candidate models. In this section, we will use smaller models with different combinations of predictor variables. In other words, we will evaluate all the possible candidate models to select the most appropriate model. We independently generate 5 different covariates X_1 to X_5 from the standard normal distribution and set the true model parameter to be $\beta_0 = (1, -1, 0.5, 0, -0.5)$. Always keeping the intercept, X_1 and X_2 so with different combinations of the three covariates X_3, X_4 and X_5, we can form 8 candidate models summarized in Table 6.13.

Similar to the nested candidate model selection, we evaluate the performance of the selection criterion under different number of individuals with $n = 25$, $n = 50$, $n = 100$ and $n = 200$ as well as a fixed variance of the noise $\sigma = 1.3$. The true correlation matrix is set as EX(ρ) or AR(ρ) with $\rho = 0.2, 0.4, 0.6, 0.8$ and the working correlation matrix is either exchangeable or auto-regressive. The number of bootstrap samples B is chosen to be 250 across all the scenarios.For the mixed true correlation structure, 30% of the observations are from EX(0.5), another 30% come from AR(0.5) and the rest 40% from a self-defined matrix as following:

$$\begin{bmatrix} 1 & 0.8 & 0.3 \\ 0.8 & 1 & 0.6 \\ 0.3 & 0.6 & 1 \end{bmatrix}$$

where both the exchangeable and auto-regressive working correlation will be considered.

Table 6.14: True model selection rates via variable combinations (non-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Exchangeable				Auto-regressive			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
25	QIC(R)	57.3	52.4	53.6	48.6	56.6	55.9	52.2	44.0
	QAIC$_u$(R)	53.8	53.2	64.4	72.5	53.3	56.5	61.3	70.1
	QAICb1	63.4	60.6	68.2	72.6	62.1	64.5	66.3	71.1
	QAICb2	66.3	62.8	69.9	74.0	64.4	67.6	68.7	72.4
50	QIC(R)	66.1	60.3	59.8	53.4	64.3	61.2	60.1	51.9
	QAIC$_u$(R)	54.5	56.9	65.7	78.3	52.3	55.0	61.6	74.0
	QAICb1	68.7	70.0	75.0	84.6	67.1	66.8	74.2	81.5
	QAICb2	71.0	70.6	75.5	85.6	68.2	67.7	75.8	82.2
100	QIC(R)	68.5	64.3	60.7	53.6	67.8	67.2	59.5	56.8
	QAIC$_u$(R)	56.5	58.5	65.7	77.2	54.6	59.8	62.8	77.3
	QAICb1	70.4	72.3	77.0	86.2	69.9	73.5	75.2	85.0
	QAICb2	71.0	71.9	77.7	87.3	71.1	74.5	75.9	85.5
200	QIC(R)	66.5	62.9	59.9	55.4	69.5	68.2	63.4	56.1
	QAIC$_u$(R)	51.9	56.4	63.8	76.2	54.5	58.3	62.4	70.8
	QAICb1	70.5	71.1	77.6	87.2	70.3	73.4	76.0	83.8
	QAICb2	70.0	71.3	78.2	87.5	71.6	73.2	75.9	83.7

Table 6.15: True model selection rates via variable combinations (non-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Auto-regressive				Exchangeable			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
25	QIC(R)	57.8	55.8	55.8	48.0	58.0	56.0	50.9	50.5
	QAIC$_u$(R)	54.0	56.7	56.7	69.4	51.6	55.8	55.1	68.4
	QAICb1	63.0	64.8	64.8	70.4	63.3	63.4	61.2	70.8
	QAICb2	65.3	67.3	67.3	72.8	65.7	65.5	63.5	72.9
50	QIC(R)	64.7	63.4	59.3	53.2	65.1	63.6	61.2	55.8
	QAIC$_u$(R)	54.9	60.1	66.2	76.6	53.5	56.6	63.6	71.8
	QAICb1	68.2	71.2	74.1	83.2	66.9	69.3	73.0	78.4
	QAICb2	68.8	71.8	75.6	84.3	68.5	71.1	74.4	80.0
100	QIC(R)	66.7	68.2	61.6	57.7	67.0	64.8	62.3	57.1
	QAIC$_u$(R)	50.8	60.2	63.3	76.5	54.3	56.4	62.1	73.6
	QAICb1	68.0	74.3	75.9	84.5	68.2	70.7	75.4	83.0
	QAICb2	69.2	74.9	76.3	85.0	69.2	71.9	76.1	83.7
200	QIC(R)	67.4	66.4	63.3	60.5	69.3	65.5	66.5	56.4
	QAIC$_u$(R)	51.5	58.1	63.9	74.2	52.5	55.0	65.0	74.2
	QAICb1	68.6	71.5	76.8	85.6	71.8	71.2	78.5	86.2
	QAICb2	69.1	71.7	77.2	86.4	71.8	71.5	78.2	86.6

Table 6.16: True model selection rates via variable combinations (non-parametric).

True Correlation	Mixture			Mixture		
Working Correlation	Exchangeable			Auto-Regressive		
n	50	100	200	50	100	200
QIC(R)	63.1	63.6	64.5	61.1	61.5	62.2
QAIC$_u$(R)	62.7	62.6	62.2	64.6	63.6	62.7
QAICb1	74.7	74.7	75.9	73.5	78.8	76.3
QAICb2	75.5	75.9	76.6	74.5	78.8	76.7

The overall pattern of performance is similar to that in Section 6.2.1. The selection rates from QIC(R) decreases with the increase of the correlation coefficient ρ while QAIC$_u$(R), QAICb1 and QAICb2 have higher selection rates with relatively large ρ. Of all the simulation settings in this section, the performance of QAICb1 and QAICb2 is almost the same and they are in general much more efficient than QAIC$_u$(R). From Table 6.14 we can see that when $n = 25$ and $\rho = 0.8$, this is the only situation that QAIC$_u$(R) is comparable to QAICb1 and QAICb2. Of the rest cases, the selection rates of QAIC$_u$(R) are about 5% to 18% less than those from QAICb1 and QAICb2.

However, with the candidate models not being nested, there exists some difference with respect to different values of n and ρ. In previous situations, the selection efficiency of QAICb1 and QAICb2 improves noticeably with the increase of n, which is not seen from Tables 6.14 to 6.16 especially in the situation where the true correlation is mixture. Furthermore, starting from $n = 50$, the corresponding true model selection rates are seldom impacted by the sample size. As a result, QAICb1 and QAICb2 are more consistent with smaller n during this set of simulations and will be more capable than QIC(R) and QAIC$_u$(R) in small sample model selection.

In addiction to the consistency of QAICb1 and QAICb2 over different n, the impact from the true correlation coefficient ρ is not large as well. Take Table 6.15 as an example. When $n = 100$ and the working correlation is auto-regressive, the selection rates of QAIC$_u$(R) under $\rho = 0.2$ and $\rho = 0.8$ are 50.8% and 76.5% with a difference 25.7% while the corresponding rates difference for QAICb1 and QAICb2 is around 16%.

To summary, during this simulation where the candidate models are formed by different combinations of predictor variables, QAICb1 and QAICb2 have more consistent selection performance

with respect to both the sample size and the true correlation coefficient. More importantly, our proposed selection criteria could be relatively more efficient in a small sample scenario.

6.2.2 Simulations via Semi-parametric Bootstrap Approach

In this section, the semi-parametric bootstrap is utilized to construct the two variants of QAICb1 and QAICb2. As been stated in Section 5.1.1, the semi-parametric bootstrap involves sampling with replacement over the residuals of all the candidate models. It is preferred in certain scenarios where the candidate models are of high accuracy. When implementing the semi-parametric bootstrap method, the bootstrap samples should be drawn model by model as well as the computation of our model selection criteria QAICb1 and QAICb2. For each candidate model, the first step is to obtain the corresponding fitted values and residuals. Then the resampling is done over the residuals at the individual level and the bootstrapped residuals are added to the original fitted values to form the bootstrap sample, which will be fitted again by the candidate model to calculate the values of QAICb1 and QAICb2.

6.2.2.1 True Correlation Correctly Specified

We will use the same model setting as that for non-parametric bootstrap in Section 6.2.1 except for the number of individuals n. In this section, n is chosen to be 15, 25, 35 and 50. As mentioned above, semi-parametric bootstrap requires resampling model by model so the corresponding time cost will be 10 times higher than that of the non-parametric bootstrap with 10 candidate models. Therefore, we will utilize relatively small samples to investigate the selection performance of QAICb1 and QAICb2. We first present simulation results when the true correlation structure is correctly specified in Tables 6.17 to 6.19.

The overall pattern of selection rates using semi-parametric bootstrap is similar to that based on non-parametric bootstrap. We have close true model selection rates for both the exchangeable and auto-regressive working correlation matrices, which is reasonable because when the true correlation is correctly specified in both cases, the GEE estimators are the most efficient. From the perspective of the number of individuals, when n increases, all of the four selection criteria be-

Table 6.17: True model selection rates under $\sigma = 1$ (semi-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Exchangeable				Auto-regressive			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
15	QIC(R)	42.2	41.7	38.9	35.6	41.3	38.0	36.8	36.0
	QAIC$_u$(R)	73.2	74.6	79.0	84.6	71.3	70.5	76.2	81.7
	QAICb1	42.5	44.7	49.4	57.4	40.3	42.8	47.6	54.6
	QAICb2	46.8	51.5	57.4	68.1	45.5	47.8	53.8	66.0
25	QIC(R)	52.9	50.6	46.7	41.7	51.7	49.1	46.1	41.4
	QAIC$_u$(R)	73.8	76.3	81.1	88.6	73.1	75.6	82.1	87.0
	QAICb1	50.2	52.2	55.3	62.1	49.5	52.9	57.8	62.4
	QAICb2	55.5	61.0	67.5	80.3	54.7	61.1	71.0	79.7
35	QIC(R)	59.2	55.9	52.1	47.1	59.8	53.5	50.1	43.0
	QAIC$_u$(R)	76.1	79.9	84.4	91.6	76.2	75.9	85.7	90.1
	QAICb1	53.3	56.4	59.6	64.5	53.9	54.1	58.7	61.3
	QAICb2	62.1	67.7	78.7	88.1	61.5	66.7	77.6	83.8
50	QIC(R)	61.8	57.2	54.1	45.4	61.0	56.8	51.9	42.0
	QAIC$_u$(R)	73.8	77.0	83.5	92.7	74.4	75.7	82.2	91.5
	QAICb1	52.6	52.0	59.3	63.8	51.5	54.3	58.2	63.1
	QAICb2	64.5	71.1	82.0	91.4	63.8	69.0	79.1	90.9

Table 6.18: True model selection rates under $\sigma = 1.3$ (semi-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Exchangeable				Auto-regressive			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
15	QIC(R)	33.2	38.0	36.4	30.9	35.8	38.3	35.1	32.5
	QAIC$_u$(R)	43.3	51.0	57.6	64.5	42.9	48.8	52.3	65.6
	QAICb1	36.0	41.3	46.5	48.0	36.6	39.9	40.6	50.5
	QAICb2	38.1	46.5	52.9	56.7	38.8	46.0	49.6	60.0
25	QIC(R)	52.0	49.8	48.1	39.5	50.3	49.3	45.6	38.6
	QAIC$_u$(R)	48.8	54.5	62.1	71.4	46.9	52.8	61.2	70.3
	QAICb1	49.1	50.8	54.6	60.9	46.6	52.0	56.4	58.2
	QAICb2	53.6	58.0	66.8	72.3	53.0	60.1	64.8	73.9
35	QIC(R)	58.5	55.3	48.6	45.3	59.0	56.9	52.4	47.0
	QAIC$_u$(R)	47.2	53.7	63.0	79.4	46.9	52.3	63.7	78.8
	QAICb1	52.5	55.6	57.1	63.3	53.6	55.8	58.9	63.3
	QAICb2	61.6	66.4	75.3	84.4	61.1	66.2	74.2	82.5
50	QIC(R)	61.4	58.8	55.5	48.5	62.7	59.3	56.0	45.3
	QAIC$_u$(R)	47.6	53.5	64.5	80.3	47.7	53.1	65.1	77.5
	QAICb1	52.9	57.8	57.6	63.6	53.6	55.9	60.7	63.2
	QAICb2	63.3	72.0	80.9	90.4	65.1	71.0	80.2	87.7

Table 6.19: True model selection rates under $\sigma = 1.5$ (semi-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Exchangeable				Auto-regressive			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
15	QIC(R)	31.3	32.1	32.8	28.5	32.2	35.0	31.0	30.3
	QAIC$_u$(R)	28.8	33.7	43.0	53.2	30.7	34.2	39.1	51.1
	QAICb1	32.2	36.5	41.7	45.4	34.2	36.0	38.6	41.5
	QAICb2	34.2	39.6	46.5	55.0	36.4	39.3	45.4	51.4
25	QIC(R)	48.8	47.0	43.3	37.0	49.6	48.0	46.8	37.6
	QAIC$_u$(R)	34.1	39.1	47.5	61.7	32.2	37.2	50.1	56.7
	QAICb1	44.7	48.8	52.0	58.3	47.7	48.6	50.0	54.5
	QAICb2	52.8	56.6	62.4	73.0	53.2	58.5	63.0	68.7
35	QIC(R)	55.7	53.0	51.8	39.4	57.2	56.8	46.2	43.2
	QAIC$_u$(R)	33.1	41.2	53.0	62.6	33.6	42.8	46.9	64.1
	QAICb1	51.4	53.2	58.0	60.8	51.7	55.5	56.4	59.7
	QAICb2	59.4	65.5	73.8	77.7	61.5	63.9	70.0	77.7
50	QIC(R)	60.8	58.8	53.6	44.6	61.7	59.6	55.1	44.6
	QAIC$_u$(R)	32.5	40.0	50.7	66.8	32.5	38.9	49.8	65.3
	QAICb1	52.2	53.8	58.6	61.9	53.0	56.0	57.8	58.9
	QAICb2	64.4	70.4	78.2	84.9	63.9	70.8	77.7	82.9

come more and more efficient regardless of the correlation coefficient ρ and the size of the noise σ, which matches the corresponding asymptotic properties. In most of the cases, QIC(R) has the lowest selection rates except when n and σ are relatively large and ρ is relatively small. In Table 6.19, for example, when $n = 50$, $\sigma = 1.5$ and $\rho = 0.2$, the selection rate of QIC(R) is 60.8% which is much higher than that of QAIC$_u$(R) but slightly lower than QAICb2's 64.4%. On the other hand, QAIC$_u$(R) only has higher selection rates with a relatively small σ and a relatively large ρ, which is shown from Table 6.17.

It is also worth pointing out that when the number of observations n is relatively small ($n = 15$ or 25) and σ is not large ($\sigma = 1$), QAIC$_u$(R) has higher true model selection rates than QAICb1 and QAICb2. For example, in Table 6.17 where $n = 15$, $\rho = 0.2$ and the working correlation is exchangeable, the selection rate of QAIC$_u$(R) is 74.6% which is much higher than the 44.7% and 51.5% for QAICb1 and QAICb2. The simulation results indicate that the estimation of the bias correction terms b1 and b2 may not be of high efficiency when the sample size is small even though bootstrap is utilized. However, when n increases to 50, QAICb2 will have comparable true

model selection rates to QAIC$_u(R)$, which can also be seen in the same Table 6.17.

During the simulations using the non-parametric bootstrap approach in Section 6.2.1, QAICb1 and QAICb2 have very close selection efficiency where the selection rates of QAICb1 is only slightly lower than those achieved by QAICb2. However, the performance of QAICb1 is not as optimal as that of QAICb2 in the simulations using the semi-parametric bootstrap. One possible reason is that when semi-parametric bootstrap is utilized, QAICb1 tends to underestimate the expected K-L distance between the true model and candidate models. As a result, QAICb1 is more likely to select relatively large candidate models as the final model. Such issue will receive more in-depth discussion during the next part of this section.

Even though QAICb2 does not always have the highest true model selection rates, it is the most consistent model selection criterion during semi-parametric simulations. In particular, when n is relatively large ($n = 50$), QAICb2 would achieve selection rates ranging from 64% to 90% which are much higher than those from the others. Referring to Table 6.19 again where QIC(R) has similar performance to QAICb2 for smaller values of ρ. Unlike QIC(R) that the corresponding selection rates decrease with the increase of ρ, QAICb2 becomes more and more powerful that it reaches around 84.9% when working correlation is exchangeable, $n = 50$ and $\rho = 0.8$.

Furthermore, when comparing the results based on non-parametric and semi-parametric bootstrap approaches, it is worth noting that the semi-parametric simulation results are slightly better than those from the non-parametric ones for relatively larger values of ρ. Taking Table 6.17 as an example, when $n = 50$ and the working correlation is auto-regressive, the selection rates of QAICb2 are $63.8\%, 69.0\%, 79.1\%$ and 90.9% while the corresponding rates using non-parametric approach are $66.9\%, 74.0\%, 78.7\%$ and 86.2% from Table 6.1. However, the semi-parametric bootstrap approach has a higher computation cost than the non-parametric bootstrap. Therefore, if the time allows and the within individual correlation is assumed to be relatively large, the semi-parametric bootstrap approach and QAICb2 are preferred for normal linear mixed models to obtain a more accurate selection rate.

To sum up, when the true correlation is correctly specified under the semi-parametric bootstrap

Table 6.20: True model selection rates under $\sigma = 1$ (semi-parametric).

True Correlation	Mixture				Mixture			
Working Correlation	Exchangeable				Auto-Regressive			
n	15	25	35	50	15	25	35	50
QIC(R)	37.5	49.4	54.3	58.0	42.4	50.6	54.3	59.8
QAIC$_u(R)$	73.8	77.2	80.8	79.0	75.2	76.7	77.7	78.8
QAICb1	43.8	53.6	57.5	58.8	47.2	53.0	55.5	55.9
QAICb2	49.4	62.3	72.6	74.8	52.5	61.1	70.3	76.2

approach, QAICb2 is the most consistent and has the highest true model selection rates with a relatively larger noise ($\sigma = 1.3$ or 1.5). Even with a smaller σ, if the number of individuals is large enough ($n = 35$ or 50), QAICb2 will have the most efficiency.

6.2.2.2 Mixture Model Involving Different Correlation Structures

In this section, we will present simulation results when the sample is constructed by observations from three different groups. That is, 40% of the observations are from EX(0.5), another 40% come from AR(0.5) and the rest 20% from a self-defined matrix as following:

$$\begin{bmatrix} 1 & 0.4 & 0.6 \\ 0.4 & 1 & 0.3 \\ 0.6 & 0.3 & 1 \end{bmatrix}.$$

Both the exchangeable and auto-regressive working correlations are used to fit candidate models and three values of σ are considered as well.

The simulation results are summarized in Tables 6.20 to 6.22. We observe the similar overall pattern demonstrated by the simulations where the true correlation is correctly specified. Firstly, it can be seen that QIC(R), QAICb1 and QAICb2 are more consistent than QAIC$_u(R)$ with respect to the change of σ although QAIC$_u(R)$ is more efficient than others if both n and σ are small. In Table 6.20 when $\sigma = 1$, $n = 25$ and the working correlation is exchangeable, the true model selection rate of QAIC$_u(R)$ is 77.2% which is higher than the 62.3% of QAICb2, respectively. However, when n increases to 50, the selection rate of QAICb2 is 74.8% which is slightly higher

Table 6.21: True model selection rates under $\sigma = 1.3$ (semi-parametric).

True Correlation	Mixture				Mixture			
Working Correlation	Exchangeable				Auto-Regressive			
n	15	25	35	50	15	25	35	50
QIC(R)	35.6	48.3	55.2	58.0	35.5	47.4	57.3	56.9
QAIC$_u(R)$	49.8	56.7	59.8	55.3	49.0	54.9	56.9	55.3
QAICb1	39.9	52.0	56.5	57.8	39.4	50.0	56.0	58.1
QAICb2	45.1	59.1	69.7	74.2	44.0	60.6	69.6	72.7

Table 6.22: True model selection rates under $\sigma = 1.5$ (semi-parametric).

True Correlation	Mixture				Mixture			
Working Correlation	Exchangeable				Auto-Regressive			
n	15	25	35	50	15	25	35	50
QIC(R)	31.9	45.4	53.6	58.5	35.6	44.8	56.0	59.1
QAIC$_u(R)$	37.4	40.6	42.4	43.8	39.5	40.9	43.6	44.1
QAICb1	37.5	46.8	54.3	57.2	40.7	49.4	54.5	54.7
QAICb2	40.6	56.2	66.8	75.2	42.9	58.0	66.8	72.3

than the 79.0% of QAIC$_u(R)$. In addition, when $\sigma = 1.5$ and $n = 50$ with the exchangeable working correlation, Table 6.22 shows that QAIC$_u(R)$ has a 43.8% true model selection rate while QAICb2 achieves a much higher 75.2% rate. With the increase of n, QAICb2 becomes more and more efficient. Especially in the cases where σ is relatively large ($\sigma = 1.3$ or 1.5), QAICb2 has noticeably higher true model selection rates than QIC(R) and QAIC$_u(R)$.

Secondly, in general, QAICb2 has the best overall selection performance while QAICb1 is not as efficient as QAICb2 across all model settings. The detailed explanation will be provided in the following part. As a summary, when the true correlation is a mixture of several correlation structures and semi-parametric bootstrap is utilized, QAICb2 has the most consistent and efficient overall selection performance than the others. Possibly affected by the bias introduced from over-fitted candidate models, the performance of QAICb1 is not optimal when compared to the results from non-parametric bootstrap.

6.2.2.3 Discussion on Selecting Overfitted Candidate Models

In this section, we will discuss the simulation results of QAICb1 and QAICb2 using semi-parametric bootstrap based on three specific model settings. We intend to show that QAICb2 is less

Table 6.23: Selection results under $n = 100$ and $\sigma = 1.3$ (semi-parametric).

True Correlation: EX(0.6)				Working Correlation: Exchangeable						
Criteria	M1	M2	M3	TRUE	M5	M6	M7	M8	M9	M10
QIC(R)	0	0	0	576	132	81	70	42	56	43
QAIC$_u$(R)	0	0	0	648	129	68	58	33	39	25
QAICb1	0	0	0	535	247	111	52	27	14	14
QAICb2	0	0	0	849	103	19	17	4	6	2

Table 6.24: Selection results under $n = 100$ and $\sigma = 1.3$ (semi-parametric).

True Correlation: AR(0.6)				Working Correlation: Auto-regressive						
Criteria	M1	M2	M3	TRUE	M5	M6	M7	M8	M9	M10
QIC(R)	0	0	0	599	113	91	68	51	40	38
QAIC$_u$(R)	0	0	0	654	112	82	57	42	30	23
QAICb1	0	0	0	550	237	104	60	25	13	11
QAICb2	0	0	0	875	68	30	20	5	2	0

likely to select over-fitted candidate models while QAICb1 is more likely to do so, which results in the non-optimal performance of QAICb1. The corresponding simulation results are presented in Tables 6.23 to 6.25.

From the three tables, we can see that all of the four selection criteria will not select candidate models smaller than the true model and QAICb2 attains the highest true model selection rates by not selecting relatively large candidate models. In Table 6.23, the selection rates of "M6" for QIC(R) and QAIC$_u$(R) are 8.1% and 3.8% while the corresponding rate for QAICb2 is 1.9%. Not to mention QAICb2 will rarely choose highly overfitted candidate models "M8" to "M10". Similar pattern can be found in Tables 6.24 and 6.25 where the selection rate of candidate model "M5" is greatly reduced by QAICb2.

Tables 6.23 to 6.25 also demonstrate that QAICb1 is not as efficient as QAICb2 because QAICb1 tends to choose more relatively larger models than QAICb2. The reason may lie in the estimation of the bias correction term b1 (5.30) in QAICb1 defined as

$$\frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}^*(i); Y, I)}{Q(\hat{\beta}^*(i); Y^*(i), I)}. \tag{6.2}$$

In a semi-parametric bootstrap process, the resampling is done over the residuals obtained from

Table 6.25: Selection results under $n = 100$ and $\sigma = 1.3$ (semi-parametric).

True Correlation: Mixture				Working Correlation: Exchangeable						
Criteria	M1	M2	M3	TRUE	M5	M6	M7	M8	M9	M10
QIC(R)	0	0	0	627	110	66	53	57	48	39
QAIC$_u$(R)	0	0	0	584	118	75	56	66	50	51
QAICb1	0	0	0	529	236	94	56	43	23	19
QAICb2	0	0	0	811	97	40	20	19	8	5

the corresponding candidate models. In other words, the bootstrap estimation of b1 relies on the candidate model generating the residuals. From the term (6.2), we know that, for the QAICb1 value of each candidate model, both the numerator and denominator are calculated using GEE estimators from the bootstrap sample corresponding to this model. Moreover, the denominator in the fraction (6.2) utilizes the $Y^*(i)$ from the bootstrap sample. As a result, if this candidate model is biased, the estimation of b1 tends to be biased as well.

On the other hand, the estimation of the bias correction term b2 (5.35) in QAICb2 is defined as

$$2\Big\{\frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}^*(i);Y,I)}{Q(\hat{\beta};Y,I)}\Big\}. \tag{6.3}$$

The use of the quasi-likelihood functions based on both the bootstrap and the original samples in term (6.3) reduces the bias in the estimation brought by the semi-parametric bootstrap samples.

6.3 Simulations in Generalized Linear Models with Random Effects

For the second part of the simulation study, the model selection efficiency of QIC(R), QAIC$_u$(R), QAICb1 and QAICb2 are investigated under the generalized linear models with random effects. Specifically, we will use the logit and log link functions to construct logistic and Poisson regression models, respectively.

6.3.1 Logistic Regression Models via Non-parametric Approach

In this section, we will evaluate the performance of our proposed selection criteria QAICb1 and QAICb2 based on the logistic regression models. The candidate models are 7 nested models with increasing order and the largest model contains all the 7 covariates. Unlike the normal case

with 10 candidate models, we reduce the model number because a larger logistic regression model is much more easier to overfit the data where the corresponding parameters are very difficult to estimate. For example, the candidate models with 5 or 6 redundant variables are sometimes not correctly fitted and their quasi-likelihood values would be much larger than the true model, which makes them more likely to be chosen as the final model. By providing fewer valid candidate models, we would be able to achieve more credible simulation results. On the other hand, the sample size needed to fit a logistic regression model should be increased to achieve more reliable estimators when bootstrap is used. As the response variable only takes two values 0 and 1, the model fitting process requires a larger sample to ensure that both categories have enough data values for parameters estimation.

Moreover, there is no over-dispersion parameter included in the logistic regression model. Although the variance is connected to the mean in the binary distribution, when it comes to fitted results, the estimated over-dispersion parameter is either 1 or very close to 1. Thus, to maintain the consistence of the estimated correction terms of QAICb1 and QAICb2, we ignore the estimation of the over-dispersion parameter like we do for linear mixed models. A generalized linear model with the logit link function is defined by:

$$\mathrm{E}(Y_{ij}|X_{1,ij}, X_{2,ij}, X_{3,ij}, X_{4,ij}, X_{5,ij}, X_{6,ij}, X_{7,ij}) = \mu_{ij} \quad \text{and}$$

$$\text{logit}(\mu_{ij}) = \beta_0 + \beta_1 X_{1,ij} + \beta_2 X_{2,ij} + \beta_3 X_{3,ij} + \beta_4 X_{4,ij} + \beta_5 X_{5,ij} + \beta_6 X_{6,ij} + \beta_7 X_{7,ij},$$

where $i \in \{1, ..., n\}$ and $j \in \{1, ..., m\}$.

The true model parameters are $\beta_0 = (0.5, -0.5, 0.25, -0.25, 0, 0, 0)$. The covariates X_1 to X_7 are independently generated from the standard normal distribution. Three different number of individuals $n = 150$, $n = 200$, $n = 300$ are considered with $m = 3$ repeated measurements. The number of bootstrap samples is set to be 250 across all the scenarios. The true correlation matrix is chosen as EX(ρ) or AR(ρ) with $\rho = 0.2, 0.4, 0.6, 0.8$ and the working correlation matrix is exchangeable or auto-regressive. There is no error term in the logistic regression model. For the case where the true correlation is mixture, 30% of the observations are from EX(0.5), another 30%

Table 6.26: True model selection rates using logit link (non-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Exchangeable				Auto-regressive			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
150	QIC(R)	65.9	60.5	57.8	57.1	62.5	61.9	62.0	58.6
	QAIC$_u$(R)	66.6	62.7	62.5	66.7	62.8	63.7	64.7	64.7
	QAICb1	66.4	62.5	62.5	67.2	63.8	63.3	63.8	64.8
	QAICb2	67.4	63.8	63.0	68.2	63.6	64.9	64.8	64.4
200	QIC(R)	69.6	67.1	65.0	62.5	70.3	67.3	67.4	61.8
	QAIC$_u$(R)	70.5	68.6	72.2	74.4	72.1	70.1	73.0	71.2
	QAICb1	71.2	69.1	72.5	75.3	71.3	71.0	74.0	71.7
	QAICb2	71.8	70.2	73.2	74.4	72.2	71.9	74.7	73.1
300	QIC(R)	73.6	73.6	68.8	66.6	73.0	73.6	66.9	67.4
	QAIC$_u$(R)	74.6	77.4	78.3	80.1	73.4	76.4	74.9	79.8
	QAICb1	74.9	77.1	77.7	80.2	73.7	76.8	75.7	79.6
	QAICb2	76.0	78.0	78.2	80.9	75.0	77.7	76.6	80.2

come from AR(0.5) and the rest 40% from a self-defined matrix as following:

$$\begin{bmatrix} 1 & 0.9 & 0.6 \\ 0.9 & 1 & 0.3 \\ 0.6 & 0.3 & 1 \end{bmatrix}$$

where both the exchangeable and auto-regressive working correlation will be considered. The simulation results are summarized in Tables 6.26 to 6.27.

It can be seen from these three tables that as the number of subjects n increases, all of the four selection criteria become more and more efficient. Similar to the simulations using normal models, with the same n, the increment in the correlation coefficient ρ will, in general, also bring up the true model selection rates of QAIC$_u$(R), QAICb1 and QAICb2, which could be seen from Tables 6.26 and 6.27. However, the influence of different values of ρ for the logistic model is much smaller than the normal model. Consider the case in Table 6.26 where $n = 200$ and working correlation is EX, when ρ changes from 0.2 to 0.8, the selection rates of QAICb1 and QAICb2 go from 71.2% to 75.3% and 71.2% to 75.3% with an increase of around 4%, respectively. In other words, the two proposed selection criteria are relatively consistent with respect to various ρ.

Table 6.27: True model selection rates using logit link (non-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Auto-regressive				Exchangeable			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
150	QIC(R)	61.4	61.1	60.6	59.9	61.7	58.6	61.0	57.2
	QAIC$_u$(R)	63.2	63.1	63.4	66.4	64.1	59.1	65.0	62.4
	QAICb1	63.8	62.2	63.3	65.7	63.3	59.9	64.2	61.9
	QAICb2	64.6	62.3	64.4	65.9	64.5	61.0	65.7	61.9
200	QIC(R)	70.4	65.2	67.4	63.5	67.6	69.9	65.0	65.6
	QAIC$_u$(R)	70.8	68.5	73.6	73.4	68.4	71.8	70.7	75.5
	QAICb1	71.8	68.2	74.7	73.4	68.6	71.3	70.3	74.7
	QAICb2	72.9	69.7	75.5	74.3	69.1	72.6	71.6	75.7
300	QIC(R)	73.8	72.7	70.1	70.6	75.1	73.9	68.8	68.4
	QAIC$_u$(R)	75.1	75.6	77.7	82.3	75.8	76.1	73.5	80.2
	QAICb1	74.2	75.0	77.8	81.7	76.4	75.1	73.4	79.5
	QAICb2	75.5	76.2	78.3	82.3	76.6	76.1	74.7	80.4

Based on the simulation results of the logistic regression, QAICb2 has the best overall performance across all the different settings with QAICb1 being slightly less efficient than QAICb2. However, we also notice that QAIC$_u(R)$ has achieved comparable selection rates with respect to QAICb1 and QAICb2. In the situations where n is relatively small ($n = 150$ or 200), QAIC$_u(R)$ could be the most efficient selection criterion. For example, in the Table 6.28, when $n = 150$ and the working correlation is auto-regressive, QAIC$_u(R)$ obtains 62.8% correction rate while QAICb1 and QAICb2 have rates 61.2% and 62.7%, respectively. When n is large ($n = 300$), the overall performance of QAIC$_u(R)$ is slightly better than that of QAICb1 but slightly worse than that of QAICb2. This may be resulted from the relatively small size of noise in the logistic regression model. From the simulation of linear mixed models in Section 6.2, we know that QAIC$_u(R)$ is relatively more efficient when the σ is small which is exactly the case with our model setting in this section.

6.3.2 Poisson Regression Models via Non-parametric Approach

In this section, we will investigate the selection performance of the proposed QAICb1 and QAICb2 in the Poisson regression model setting. Different from the data used in logistic regression that only two categories are involved, the response variable of the Poisson regression takes the form

Table 6.28: True model selection rates using logit link (non-parametric).

True Correlation	Mixture			Mixture		
Working Correlation	Exchangeable			Auto-regressive		
n	150	200	300	150	200	300
QIC(R)	57.5	67.0	70.7	65.1	68.0	71.1
QAIC$_u$(R)	62.8	72.2	75.6	68.1	72.9	75.7
QAICb1	61.2	71.6	75.9	67.4	72.5	75.5
QAICb2	62.7	73.2	77.1	67.8	72.9	75.9

of discrete count values. Due to the fact that the mean and variance of a Poisson distribution are the same, which means the estimation of the variance is totally dependent on the corresponding mean, the over-dispersion parameter is no longer fixed at 1 in this case. In fact, the Poisson regression model is known for the over-dispersion issue, from which some traditional estimating approaches will be more likely to produce unreliable conclusions.

Therefore, besides estimating the parameters associated with the predictor variables, the over-dispersion parameter should be considered as a modification of the proposed model selection criteria QAICb1 and QAICb2 as well. By including the estimation of the over-dispersion parameter $\hat{\phi}$, we modify QAICb1 and QAICb2 as:

$$\text{QAICb1} = -2Q(\hat{\beta}_G(R), \hat{\phi}_G(R); Y, I) + \frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}_G(I)^*(i), \hat{\phi}_G(I)^*(i); Y, I)}{Q(\hat{\beta}_G(I)^*(i), \hat{\phi}_G(I)^*(i); Y^*(i), I)},$$

$$\text{QAICb2} = -2Q(\hat{\beta}_G(R), \hat{\phi}_G(R); Y, I) + 2\Big\{\frac{1}{B}\sum_{i=1}^{B} -2\frac{Q(\hat{\beta}_G(I)^*(i), \hat{\phi}_G(I)^*(i); Y, I)}{Q(\hat{\beta}_G(I), \hat{\phi}_G(I); Y, I)}\Big\}.$$

It is worth pointing out that, when updating the estimation of bias correlation terms of both QAICb1 and QAICb2 with the independence assumption, all the corresponding values of $\hat{\phi}_G$ are also updated from models using independent working correlation. Moreover, the quasi-likelihood functions in the selection criteria QIC(R) and QAIC$_u$(R) are also computed with the $\hat{\phi}_G$ included.

The Poisson regression model is a generalized linear mixed model using *log* as the link function. Similar to the model setting in Section 6.2, there are 10 predictor variables $X_1, ..., X_{10}$, from which we could obtain 10 nested candidate models with an increasing order. The largest candidate

model could be described by

$$\mathrm{E}(Y_{ij}|X_{1,ij}, X_{2,ij}, ..., X_{7,ij}) = \mu_{ij} \quad \text{and}$$
$$\text{logit}(\mu_{ij}) = \beta_0 + \beta_1 X_{1,ij} + \beta_2 X_{2,ij} + ... + \beta_{10} X_{10,ij},$$

where $i \in \{1, ..., n\}$, $j \in \{1, ..., m\}$ and β_0 is a parameter representing the random intercept.

The true model is the fourth model having parameters, including the intercept, $(0.5, -0.5, 0.25, -0.25, 0.25)$ for β_0 to β_5. Three numbers of $n = 50, 100$ and 200 and four values of $\rho : 0.2, 0.4, 0.6$ and 0.8 are considered. There are two different scenarios with the first having correctly specified correlations and the second having mis-specified correlations. For the mis-specified case, a mixture of correlations will also be included. When it comes to the implementation of QAICb1 and QAICb2, the bootstrap sample size is set to be $B = 250$. There is no noise term for Poisson model although the variance can be estimated as a combination of the mean and the over-dispersion parameter. When the true correlation structure is mixture, 30% of the observations are from EX(0.5), another 30% come from AR(0.5) and the rest 40% from a self-defined matrix as following:

$$\begin{bmatrix} 1 & 0.2 & 0.5 \\ 0.2 & 1 & 0.8 \\ 0.5 & 0.8 & 1 \end{bmatrix}$$

where both the exchangeable and auto-regressive working correlation will be considered. The simulation results are summarized in Tables 6.29 to 6.31.

We can observe that as the n increases, all of the four model selection criteria have higher selection rates with QIC(R) being the most influenced by n. In other words, QIC(R) is inconsistent with the change of n and much less efficient than others when the sample size is relatively small. Referring to Table 6.29 for $\rho = 0.4$ and working correlation is exchangeable, the selection rate for QIC(R) is 49.8% and QAICb1 has 65.2% accuracy for $n = 50$ while the corresponding selection rates when $n = 200$ are 69.3% and 80.4% for QIC(R) and QAICb1, respectively. Even though the QIC(R) has a larger bump of the rate increase, it is not comparable with QAICb1.

Table 6.29: True model selection rates under Poisson model (non-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Exchangeable				Auto-regressive			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
50	QIC(R)	55.3	49.8	43.2	33.2	53.9	52.9	47.8	35.1
	QAIC$_u$(R)	67.6	66.9	73.6	74.6	66.8	70.6	72.0	78.0
	QAICb1	65.5	65.2	69.1	68.1	65.0	68.5	70.4	73.9
	QAICb2	64.9	62.5	62.5	63.7	63.7	66.7	64.9	68.4
100	QIC(R)	70.5	65.3	53.9	36.6	68.1	64.2	55.9	38.0
	QAIC$_u$(R)	78.4	79.4	83.1	87.0	76.7	79.4	83.6	86.4
	QAICb1	76.3	78.2	80.7	83.3	75.3	77.8	81.1	85.5
	QAICb2	80.3	80.8	82.1	84.6	79.9	81.0	82.0	86.1
200	QIC(R)	71.4	69.3	63.5	43.2	70.9	72.3	62.7	46.2
	QAIC$_u$(R)	77.5	80.8	88.1	92.8	77.9	83.1	85.3	93.2
	QAICb1	76.8	80.4	87.4	92.6	76.5	82.1	84.4	90.8
	QAICb2	82.0	86.0	90.6	94.4	82.3	86.3	88.6	94.2

Table 6.30: True model selection rates under Poisson model (non-parametric).

True Correlation		EX(ρ)				AR(ρ)			
Working Correlation		Auto-regressive				Exchangeable			
n	ρ	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
50	QIC(R)	52.3	53.2	46.7	37.3	54.8	53.0	45.4	37.3
	QAIC$_u$(R)	68.2	70.8	70.8	79.3	68.8	70.0	69.4	77.4
	QAICb1	66.8	68.1	66.4	73.6	68.0	66.4	67.5	73.1
	QAICb2	64.7	65.3	63.8	67.7	67.1	65.4	64.5	66.7
100	QIC(R)	68.7	64.4	59.5	48.8	68.3	64.4	61.8	43.5
	QAIC$_u$(R)	76.6	77.5	82.5	88.8	76.1	77.6	83.3	88.3
	QAICb1	75.1	76.1	81.5	87.1	74.2	75.3	80.8	85.2
	QAICb2	79.6	79.0	82.1	86.2	78.3	79.5	82.6	84.9
200	QIC(R)	75.6	72.1	64.5	50.2	75.2	70.1	62.9	47.9
	QAIC$_u$(R)	81.1	83.7	85.4	90.9	79.6	79.9	84.2	89.3
	QAICb1	79.8	82.6	84.3	90.3	78.5	78.7	81.9	86.8
	QAICb2	85.5	86.6	87.8	92.6	83.7	84.6	86.9	90.3

Table 6.31: True model selection rates under Poisson model (non-parametric).

True Correlation	Mixture			Mixture		
Working Correlation	Exchangeable			Auto-Regressive		
n	50	100	200	50	100	200
QIC(R)	50.3	61.5	64.3	49.3	63.2	68.3
QAIC$_u$(R)	74.2	80.6	83.0	69.7	80.6	81.9
QAICb1	70.7	79.3	81.9	67.6	80.7	81.8
QAICb2	66.3	80.4	87.0	64.0	81.7	85.4

With a relatively small sample size ($n = 50$ or $n = 100$, QAICb1 is more optimal than QAICb2, which is not common during our simulation studies especially the semi-parametric bootstrap approach is utilized. From Table 6.30, when $\rho = 0.8$ and the working correlation is auto-regressive, the selection rates of QAICb1 and QAICb2 are 73.6% and 67.7% for $n = 50$, which is a noticeable difference. However, when $n = 200$, the two corresponding rates of QAICb1 and QAICb2 are 90.3% and 92.6%. It could also be seen that the overall performance of QAICb2 is slightly better than QAICb1 if n is relatively large. One possible reason may be that when the sample size is relatively small, the estimator $\hat{\phi}_G(I)$ from the original sample may not be efficiently estimated. As a result, the estimation of the bias correction term in QAICb2 may not be efficient as well because it is based on both the bootstrap samples and the original sample. On the other side, with the bias correction term of QAICb1 is fully estimated using the bootstrap samples, it is more consistent than that from QAICb2 when the sample size is relatively small.

Furthermore, in Section 5.3, we have proved the equivalence of QAICb1 and QAICb2 under a Poisson model when the over-dispersion parameter is included. This would explain the similar performance of QAICb1 and QAICb2 with a relatively large sample size. During our previous simulations, there is no over-dispersion parameter considered and QAICb1 is equivalent to or slightly less efficient than QAICb2, which is consistent with the asymptotic results. Due to the fact that the over-dispersion parameter ϕ is a scalar added to the quasi-likelihood function, QAICb1 and QAICb2 should be asymptotically equivalent as well.

From Tables 6.29 and 6.30, we notice that QAIC$_u$(R) is more efficient than QAICb1 and QAICb2 across some simulation settings especially when n is relatively small and ρ is relatively

large. From previous Sections 6.2 and 6.3.1, the performance of QAIC$_u(R)$ is better with the increase of the true correlation coefficient ρ, which is also demonstrated in this section. However, QAIC$_u(R)$ is not as consistent as QAICb2 with respect to different values of ρ when n is large enough. For instance, based on Table 6.30, when $n = 200$, QAICb2 out performs QAIC$_u(R)$ under all different values of ρ with QAICb2 being noticeably for smaller values of ρ.

The simulation results for the mixture correlation model setting is presented in Table 6.31. The selection rates are similar for EX and AR working correlation matrices. When the n is relatively large, QAICb1 has almost the same rate as QAIC$_u(R)$ while QAICb2 is slightly better than QAICb1, which is consistent with simulation results in Tables 6.29 and 6.30.

To summary, in this log linear (Poisson) mixed model simulation study, QAICb1 and QAICb2 are more consistent than QIC(R) with respect to different sample sizes. Moreover, they provide more consistent selection results than QAIC$_u(R)$ with different true correlation coefficients. For a relatively smaller sample, QAICb1 is more efficient than QAICb2 when the over-dispersion parameter included in the model.

CHAPTER 7 APPLICATIONS

The simulation studies in Chapter 6 have demonstrated that both the proposed model selection criteria QAICb1 and QAICb2 are efficient and consistent under various model settings. To further investigate the performance of the two criteria in mixed model selection, two applications with real data sets are provided in this chapter. Within each application, we construct the candidate models first and then apply QAICb1 and QAICb2 to obtain the most appropriate model. Finally, we compare the results with those from some other existing model selection criteria.

7.1 The Respiratory Illness Clinical Trial Data

7.1.1 Data Description

The respiratory illness clinical trial data (Davis, 1991) was from a randomized clinical trial aiming to examine the effect of two treatments for respiratory illness. The primary response variable is the respiratory status of each patient and it was recorded 5 times. The first record is at the start of the study, which serves as a patient's baseline status while he rest four were assessed during the four visits of the study. These repeated measurements within the same individual over a period of time form a typical longitudinal data.

There are 111 patients randomized to receive either active or placebo treatments across two different medical centers. The total number of observations is 444 with each individual having 4 repeated assessments. In addition to the respiratory status and treatment conditions, the study location, sex and age of a patient are also recorded. Most of the predictor variables in the dataset are categorical and a description of the coded variables is included in Table 7.1.

7.1.2 Candidate Models

As there are total 6 predictor variables, we are able to form 32 possible candidate models based on whether or not a predictor variable is included or not. However, this extensive way of searching the most appropriate model is not only time consuming but also ignoring the context of the practical problem. As for this clinical study, the two predictors "baseline" and "treatment" should always

Table 7.1: A description of predictor variables.

Name	Description
status	respiratory status measured during the study, “poor” = 0, “good” = 1.
baseline	respiratory status measured before the study, “poor” = 0, “good” = 1.
treatment	treatment received of a patient, “placebo” = 0, “active” = 1.
center	index of study locations, “center 1” = 0, “center 2” = 1.
sex	index of a patient’s sex, “female” = 0, “male” = 1.
age	numerical value of a patient’s age.
id	index of a patient’s id.

Table 7.2: P-values of predictor variables based on the full model.

Coefficients	Estimate	Standard Error	Wald Statistic	P-Value
Intercept	-0.9002	0.4603	3.82	0.05052
baseline	1.8820	0.3501	28.91	7.6×10^{-8}
treatment	1.2992	0.3508	13.72	0.00021
center	0.6716	0.3568	3.54	0.05981
sex	0.1192	0.4432	0.07	0.78789
age	-0.0182	0.0130	1.95	0.16244

be included in a candidate model. The factor “baseline” is important as whether or not a patient has a respiratory illness before the study directly determines the status of this patient at the end of the study. Moreover, the factor “treatment” should be considered as well because the goal of this clinical study is to investigate the effectiveness of the treatment for the respiratory illness. Thus, the candidate models without these two predictor variables would be meaningless.

Based on the p-values constructed using Wald statistic from Table 7.2, the “baseline” and “treatment” are identified as two highly significant variables with regard to any significance levels, which also indicates the inclusion of these two predictor variables into the base model. As a result, we define the base model using the “baseline” and “treatment” and then construct 8 candidate models by different combinations of the rest three variables “center”, “sex” and “age”. A summary of all the candidate models is provided by Table 7.3.

7.1.3 Presentation of Selection Results

The response variable represents two different kinds of status labeled as “0” and “1”, which indicates us to fit the date using the setting of logistic regression. We apply the GEE approach

Table 7.3: A list of candidate models.

Model	Covariates	Model	Covariates
M1	baseline, treatment	M5	baseline, treatment, center, sex
M2	baseline, treatment, center	M6	baseline, treatment, center, age
M3	baseline, treatment, sex	M7	baseline, treatment, sex, age
M4	baseline, treatment, age	M8	baseline, treatment, center, sex, age

Table 7.4: Selection results using different selection criteria.

Criterion	Final Model	Criterion	Final Model
QIC(R)	M6	QAICb1	M2
QAIC$_u(R)$	M6	QAICb2	M2

to obtain estimated parameters and the corresponding quasi-likelihood functions for all the candidate models.Three different working correlation matrix EX, AR and UN are considered. When calculating the estimation of bias correction terms of QAICb1 and QAICb2, bootstrap samples with size 500 are used. The existing model selection criteria QIC(R) and QAIC$_u(R)$ are utilized to implement the selection procedure as well. By choosing the model with smallest value of the selection criteria, we have the selection results presented in Table 7.4. Note that even though different working correlations are used, they lead to the same final model.

From Table 7.4 we can see that both QIC(R) and QAIC$_u(R)$ have selected the model "M6" with "center" and "age" as additional predictor variables while the final model chosen by QAICb1 and QAICb2 is "M2" with only "center" included.

With the responses and all of the predictor variables nested within each patient, a typical generalized linear mixed model with a random intercept could also be applied (Zhang *et al.*, 2011). They fit the same data set using the logistic mixed model and identified "baseline", "treatment" and "center" as significant predictor variables based on the corresponding p-values with the 0.05 significance level. Moreover, using the same set of candidate models fitted by the logistic mixed model, we take advantage of the AIC selection criteria and find that the most appropriate model selected is "M2", which is consistent with our selection result from QAICb1 and QAICb2. From this point of view, we conclude our proposed model selection criteria could identify the most appropriate model and reduce the possibility of selecting overfitted models.

Table 7.5: A description of predictor variables.

Name	Description
PD	status of Parkinson's Disease, "yes" = 1, "no" = 0.
chr12 rs34637584 GT	genetic piece
chr17 rs11868035 GT	genetic piece
chr17 rs11012 GT	genetic piece
chr17 rs393152 GT	genetic piece
chr17 rs12185268 GT	genetic piece
chr17 rs199533 GT	genetic piece
UPDRS part I	part I of the Unified Parkinson's Disease Rating Scale
UPDRS part II	part II of the Unified Parkinson's Disease Rating Scale
UPDRS part III	part II of the Unified Parkinson's Disease Rating Scale
sex	index of an individual's sex, "female" = 0, "male" = 1.
weight	numerical value of an individual's age.
age	numerical value of an individual's age.
FID	index of an individual's id.

7.2 Parkinson's Progression Markers Initiative (PPMI) Data

7.2.1 Data Description

The Parkinson's Progression Markers Initiative (PPMI) (Marek *et al.*, 2011) is a longitudinal clinical study trying to identify the significant factors contributing to the progression of Parkinson's disease. Taking place at various sites from four different countries, the PPMI comprehensively evaluated the characteristics of participated subjects based on different perspectives including the brain imaging, genetics, behavioral assessments. The data of the PPMI study was provided by *The Michael J. Fox Foundation for Parkinson's Research*.

There are 441 subjects participated in this clinical study with each measured by the baseline and 3 follow-ups so there are total 1764 data points in the data. To simplify the process of this application, we chose 12 predictor variables featuring genetics and individual assessments out of the original variables. A detained description of all the variables is displayed in Table 7.5. The goal is to select the most appropriate model that can measure the progress of Parkinson's Disease based on the data.

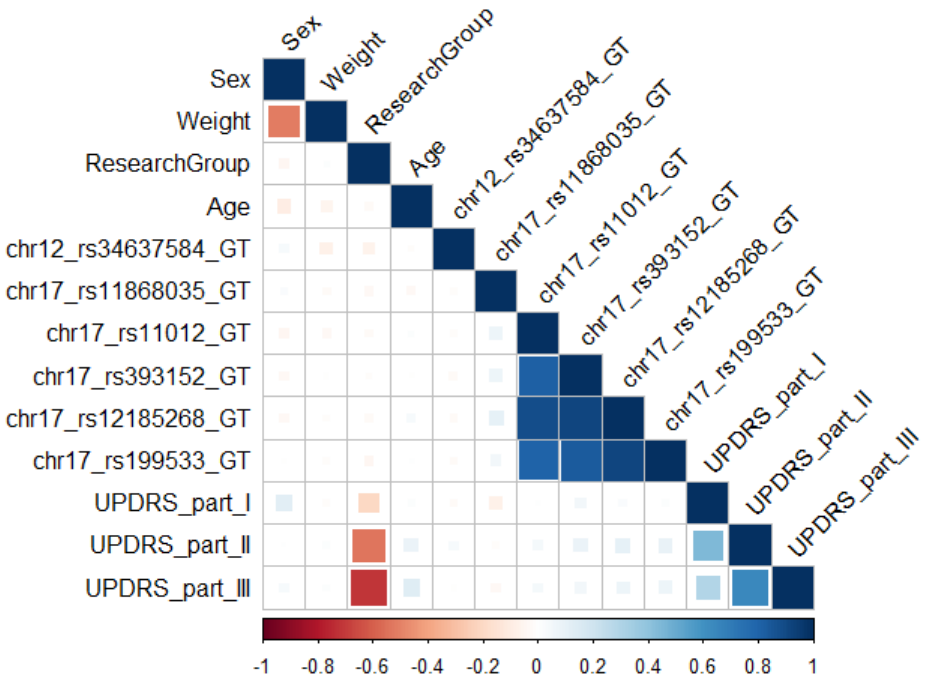

Figure 7.1: Visualization of PPMI correlation matrix.

7.2.2 Data Processing and Candidate Models

Unlike the respiratory data in the first application, the PPMI data encounters missing values. One way to deal with this issue is to delete the data points with missing inputs if the percentage of missing points is low and they are missing at random. However, after the deletion we have 1155 out of 1764 observations left. Such a high missing rate indicates us to consider impute the missing values using multiple imputation. Therefore, we obtain the complete data by the *weighted predictive mean matching* imputation approach and all of our model selection analysis is based on the imputed data set.

There are total 12 predictor variables except the response variable "PD" and the indexing variable "FID". If we consider all the possible candidate models, there are 2^{12} possibilities, which is not realistic. Based on the correlation Figure 7.1, we notice that there are two sets of variables that are highly correlated with each other. The first one contains the four correlated variables "chr17 rs11012 GT", "chr17 rs393152 GT", "chr17 rs12185268 GT" and "chr17 rs199533 GT". These

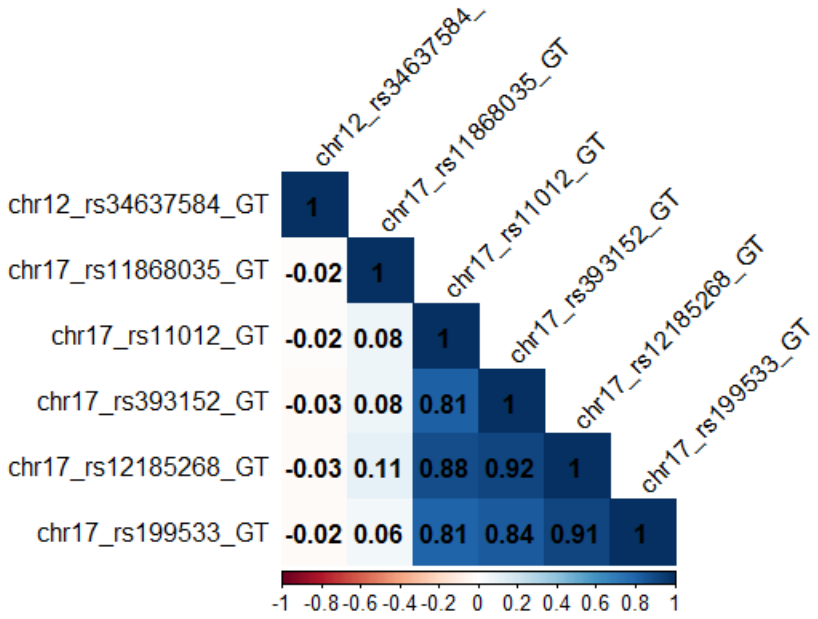

Figure 7.2: Correlation of gene expression.

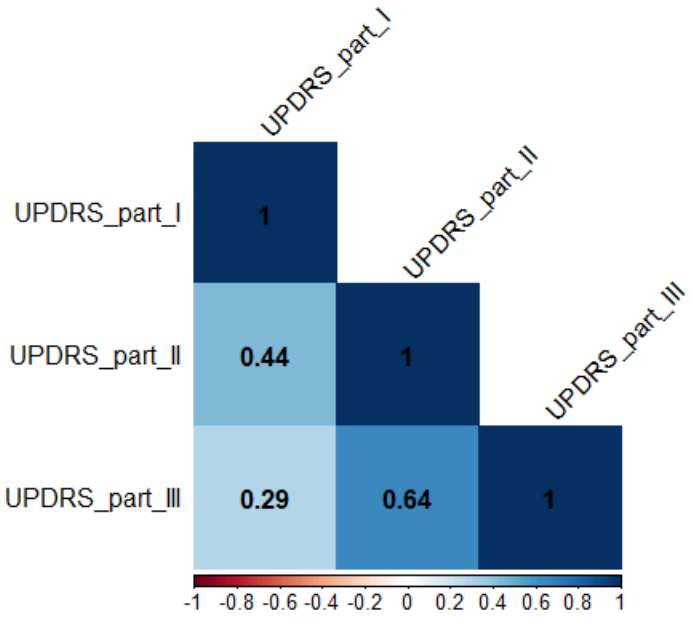

Figure 7.3: Correlation of UPDRS.

Table 7.6: P-values of predictor variables based on the full model.

Coefficients	Estimate	Standard Error	Wald Statistic	P-Value
Intercept	3.9668	0.7816	25.76	3.9×10^{-7}
UPDRS part I	0.6765	0.1703	15.79	7.1×10^{-5}
UPDRS part II	-0.4769	0.0863	30.55	3.3×10^{-8}
UPDRS part III	-0.5038	0.0575	76.89	$< 2 \times 10^{-16}$
age	0.6978	0.1750	15.90	6.7×10^{-5}
sex	-0.4597	0.4534	1.03	0.311
weight	0.0834	0.2133	0.15	0.696
chr12 rs34637584 GT	-39.8910	1.4052	805.83	$< 2 \times 10^{-16}$
chr17 rs11868035 GT	-0.3437	0.2376	2.09	0.148
chr17 rs11012 GT	-1.1220	0.6341	3.13	0.077
chr17 rs393152 GT	0.3716	0.5708	0.42	0.515
chr17 rs12185268 GT	2.2947	1.3773	2.78	0.096
chr17 rs199533 GT	-0.8195	1.1056	0.55	0.459

predictor variables are of great interest to us because our goal is to detect whether these genetic pieces are connected with Parkinson's Disease and then to find the most appropriate model. Moreover, by considering the p-values in Table 7.6, we may exclude the two variables "chr17 rs393152 GT" and "chr17 rs199533 GT" from the base model as the corresponding p-values are 0.515 and 0.459, respectively.

The second set consists of the three variables: "UPDRS part I", "UPDRS part II" and "UPDRS part III", which is reasonable as these are measurements of the same research subject from different aspects. As a person's status of Parkinson's Disease is evaluated based on these assessments, these three deterministic predictor variables should be included in the base model otherwise the model fitting procedure will become very unstable.

$$\begin{aligned}\text{Base Model} = \text{UPDRS part I} + \text{UPDRS part II} + \text{UPDRS part III} \\ + \text{sex} + \text{age} + \text{weight} + \text{chr12 rs34637584 GT}\end{aligned} \tag{7.1}$$

We will also include the predictor variables "sex", "weight" and into the base model to better characterize a subject even though they are not statistically significant based on the p-values from the Table 7.6. Considering these three variables will greatly simplify the process of model selec-

Table 7.7: Candidate models for PPMI data analysis.

Name	M1	M2	M3	M4	M5	M6	M7	M8
UPDRS part I	×	×	×	×	×	×	×	×
UPDRS part II	×	×	×	×	×	×	×	×
UPDRS part III	×	×	×	×	×	×	×	×
age	×	×	×	×	×	×	×	×
sex	×	×	×	×	×	×	×	×
weight	×	×	×	×	×	×	×	×
chr12 rs34637584 GT	×	×	×	×	×	×	×	×
chr17 rs11868035 GT		×			×	×		×
chr17 rs11012 GT			×		×		×	×
chr17 rs12185268 GT				×		×	×	×

tion. Therefore, the base model (7.1) is constructed using 7 predictor variables. To conclude, we have a summary of all the candidate models in Table 7.7.

7.2.3 Presentation of Selection Results

We have presented the results of model selection in Table 7.8. The bootstrap sample size is set to be 500. Three different working correlation matrix EX, AR and UN are considered in the GEE procedure to estimate parameters for all the candidate models and they all produce the same selection result.

It can be seen that except for QAICb2, the other three model selection criteria QIC(R), QAIC$_u(R)$ and QAICb1 choose the largest model "M8" as the most appropriate model based on the given data. QAICb2 chooses "M7" whose difference from "M8" lies in the inclusion of the predictor variable "chr17 rs11868035 GT". Notice that all of the four selection criteria agree to include the two questionable predictor variables "chr17 rs11012 GT" and "chr17 rs12185268 GT" whose p-values are less than 0.1 but large are 0.05 into the final model. Without further investigation, we are not able to decide the better model between "M7" and "M8". However, by selecting a simpler model, QAICb2 may reduce the possibility of select overfitted candidate models.

Table 7.8: Selection results using different selection criteria.

Criterion	Final Model	Criterion	Final Model
QIC(R)	M8	QAICb1	M8
QAIC$_u$(R)	M8	QAICb2	M7

CHAPTER 8 CONCLUDING REMARKS AND FUTURE RESEARCH

In this dissertation, we mainly propose two criteria QAICb1 and QAICb2 for mixed model selection by adopting the quasi-likelihood function and bootstrap approach. In this chapter, we will provide summary and concluding remarks of our proposed criteria as well as discuss possible research in the future.

Our proposed QAICb1 and QAICb2 originate from the construction of the model selection criterion AIC. The AIC is derived based on the Kullback-Leibler discrepancy, which involves the expectation of log likelihood functions. To obtain the log likelihood of the data, a parametric distribution must be fully specified and corresponding computation process will be complicated if there exists correlation within data points. Moreover, fully specifying the distribution of the data may not be realistic when it comes to some practical problems. To overcome the limitations of the traditional model selection criteria, we intend to propose a selection criterion that depends less on the parametric distribution and is easy to implement.

We replace the log likelihood function in the Kullback-Leibler discrepancy by the log quasi-likelihood, in which our approaches will depend less on the parametric distribution and much easier to implement with the existence of correlation in the data. QAICb1 and QAICb2 are developed serving as unbiased estimators of the quasi-likelihood based Kullback-Leibler discrepancy between the true model and candidate models.

Both QAICb1 and QAICb2 consist of two components: the log quasi-likelihood and an estimation of the bias correction term. To compute the log quasi-likelihood function, we utilize GEE to obtain the model parameters from the original data with a pre-specified working correlation matrix and assume all the data points are independently distributed. We employ independent assumption to ensure the existence and validation of the corresponding quasi-likelihood function. The estimation of the bias correlation term is based on the bootstrap approach, which provides us with more consistent selection performance of QAICb1 and QAICb2 for mixed models.

Furthermore, we deeply investigate the theoretical properties of QAICb1 and QAICb2. We

prove that both of our proposed criteria are consistent estimators of the corresponding Kullback-Leibler discrepancy by showing the bias correction term is consistently estimated using variants b1 and b2 from QAICb1 and QAICb2, respectively. Moreover, we prove that QAICb1 and QAICb2 are asymptotically equivalent even though the corresponding bias correction terms are differently estimated.

Besides the linear mixed model, we extend QAICb1 and QAICb2 to generalized linear models with random effects by including the estimation of over-dispersion parameter. We also provide proofs to establish corresponding properties under the logit and log link function, which serves as theoretical foundations for the application of QAICb1 and QAICb2 to logistic and Poisson regression models.

For the computation of QAICb1 and QAICb2, the two variants b1 and b2 are estimated using GEE models with the independent working correlation matrix. Doing so reduces the possibility of underestimating overfitted candidate models and the impact brought by using the same working correlation matrix for the estimation of both the log quasi-likelihood term and the bias correction term.

To fully study the model selection performance of QAICb1 and QAICb2, simulation studies with different kinds of model settings are conducted. The simulation results demonstrate that our proposed criteria have more efficient and consistent overall selection performance when compared to some existing selection criteria. The results also show that the performance of QAICb1 and QAICb2 are almost the same in large sample scenarios except the situation where the semi-parametric bootstrap approach is used to generate bootstrap samples in Section 6.2.2. In this case, the selection performance of QAICb1 is not as optimal as that of QAICb2. Based on the simulation results, QAICb2 is preferred when semi-parametric bootstrap is utilized.

We apply our proposed model selection criteria QAICb1 and QAICb2 to two real data sets to further evaluate their model selection performance. In both applications, the selection results show that our proposed two selection criteria are comparable to existing selection criteria and have the ability to select the most appropriate candidate model.

The asymptotic results show that b1 and b2 are consistent estimators of the bias correction terms corresponding to QAICb1 and QAICb2, respectively. Thus, a suitable bootstrap sample size B should be considered to ensure the selection efficiency of QAICb1 and QAICb2. During the simulation studies, B is chosen to be 250 which serves as the minimal number of bootstrap sample size needed to efficiently two terms b1 and b2 of QAICb1 and QAICb2. In other words, increasing B from 250 will not significantly increase the performance of our proposed selection criteria. As been discussed above, QAICb1 is less efficient than QAICb2 when parametric bootstrap is used. We may be able to improve the selection performance of QAICb1 by increasing the number of bootstrap sample B to 1000 or 2000.

Furthermore, when establishing theoretical properties and conducting simulation studies, we initially propose QAICb1 and QAICb2 for mixed models suitable for longitudinal data with repeated measurements. The use of the quasi-likelihood and GEE could also be extended to a typical generalized linear mixed effects models with both fixed and random effects included (Zeger *at al.*, 1988), which indicates a possible extension of QAICb1 and QAICb2.

BIBLIOGRAPHY

[1] Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. *Second International Symposium on Information Theory*, 267–281.

[2] Anderson, D. and K. Burnham (2001). Commentary on models in ecology. *Bulletin of the Ecological Society of America 82*, 160–161.

[3] Bancroft, T. and C.-P. Han (1977). Inference based on conditional specification: a note and a bibliography. *International Statistical Review/Revue Internationale de Statistique*, 117–127.

[4] Box, G. E. (1976). Science and statistics. *Journal of the American Statistical Association 71*(356), 791–799.

[5] Buckland, S. T., K. P. Burnham, and N. H. Augustin (1997). Model selection: an integral part of inference. *Biometrics*, 603–618.

[6] Burnham, K. P. and D. R. Anderson (2004). Multimodel inference: understanding aic and bic in model selection. *Sociological methods & research 33*(2), 261–304.

[7] Copas, J. B. (1983). Regression, prediction and shrinkage. *Journal of the Royal Statistical Society: Series B (Methodological) 45*(3), 311–335.

[8] Davis, C. S. (1991). Semi-parametric and non-parametric methods for the analysis of repeated measurements with applications to clinical trials. *Statistics in medicine 10*(12), 1959–1980.

[9] Dvison, A., D. V. Hinkley, and E. Schechtman (1986). Efficient bootstrap simulation. *Biometrika 73*(3), 555–566.

[10] Efron, B. (1982). *The jackknife, the bootstrap, and other resampling plans*, Volume 38. Siam.

[11] Efron, B. (1983). Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of the American statistical association 78*(382), 316–331.

[12] Efron, B. (1986). How biased is the apparent error rate of a prediction rule? *Journal of the American statistical Association 81*(394), 461–470.

[13] Efron, B. (1992). Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pp. 569–593. Springer.

[14] Freedman, D. A. et al. (1981). Bootstrapping regression models. *The Annals of Statistics 9*(6), 1218–1228.

[15] Hurvich, C. M. and C.-L. Tsai (1989). Regression and time series model selection in small samples. *Biometrika 76*(2), 297–307.

[16] Ishiguro, M. and K. Morita (1991). Application of an estimator-free information criterion (wic) to aperture synthesis imaging. In *International Astronomical Union Colloquium*, Volume 131, pp. 243–248. Cambridge University Press.

[17] Ishiguro, M., Y. Sakamoto, and G. Kitagawa (1997). Bootstrapping log likelihood and eic, an extension of aic. *Annals of the Institute of Statistical Mathematics 49*(3), 411–434.

[18] Ishiguro, M., S. Y. (1991). *WIC: an estimation-free information criterion*. Research Memorandum, Institute of Statistical Mathematics, Tokyo.

[19] Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, Volume 14, pp. 1137–1145. Montreal, Canada.

[20] Kullback, S. and R. A. Leibler (1951). On information and sufficiency. *The annals of mathematical statistics 22*(1), 79–86.

[21] Laird, N. M., J. H. Ware, et al. (1982). Random-effects models for longitudinal data. *Biometrics 38*(4), 963–974.

[22] Lehmann, E. L. and G. Casella (2006). *Theory of point estimation*. Springer Science & Business Media.

[23] Liang, K.-Y. and S. L. Zeger (1986). Longitudinal data analysis using generalized linear models. *Biometrika 73*(1), 13–22.

[24] Lindley, D. V. (1981). The relationship between the number of factors and size of an experiment. Technical report, WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER.

[25] Linhart, H. and W. Zucchini (1986). *Model selection.* John Wiley & Sons.

[26] Marek, K., D. Jennings, S. Lasch, A. Siderowf, C. Tanner, T. Simuni, C. Coffey, K. Kieburtz, E. Flagg, S. Chowdhury, et al. (2011). The parkinson progression marker initiative (ppmi). *Progress in neurobiology 95*(4), 629–635.

[27] McCullagh, P. et al. (1983). Quasi-likelihood functions. *The Annals of Statistics 11*(1), 59–67.

[28] McCullagh, P. and J. A. Nelder (1989). *Generalized Linear Models, Second Edition*. New York: Chapman and Hall.

[29] McQuarrie, A. D. and C.-L. Tsai (1998). *Regression and time series model selection*. World Scientific.

[30] Müller, S., J. L. Scealy, A. H. Welsh, et al. (2013). Model selection in linear mixed models. *Statistical Science 28*(2), 135–167.

[31] Pan, W. (2001a). Akaike's information criterion in generalized estimating equations. *Biometrics 57*(1), 120–125.

[32] Pan, W. (2001b). Model selection in estimating equations. *Biometrics 57*(2), 529–534.

[33] Rao, J. S. and R. Tibshirani (1997). The out-of-bootstrap method for model averaging and selection. *University of Toronto*.

[34] Roecker, E. B. (1991). Prediction error and its estimation for subset-selected models. *Technometrics 33*(4), 459–468.

[35] Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics 6*(2), 461–464.

[36] Shang, J. and J. E. Cavanaugh (2008). Bootstrap variants of the akaike information criterion for mixed model selection. *Computational Statistics & Data Analysis 52*(4), 2004–2021.

[37] Shibata, R. (1997). Bootstrap estimate of kullback-leibler information for model selection. *Statistica Sinica*, 375–394.

[38] Simar, L. (1992). Estimating efficiencies from frontier models with panel data: a comparison of parametric, non-parametric and semi-parametric methods with bootstrapping. In *International Applications of Productivity and Efficiency Analysis*, pp. 167–199. Springer.

[39] Sugiura, N. (1978). Further analysts of the data by akaike's information criterion and the finite corrections: Further analysts of the data by akaike's. *Communications in Statistics-Theory and Methods 7*(1), 13–26.

[40] Thompson, W. A. et al. (1962). The problem of negative estimates of variance components. *The Annals of Mathematical Statistics 33*(1), 273–289.

[41] Wedderburn, R. W. (1974). Quasi-likelihood functions, generalized linear models, and the gauss—newton method. *Biometrika 61*(3), 439–447.

[42] White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the Econometric Society*, 1–25.

[43] White, H. et al. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *econometrica 48*(4), 817–838.

[44] Zeger, S. L., K.-Y. Liang, and P. S. Albert (1988). Models for longitudinal data: a generalized estimating equation approach. *Biometrics*, 1049–1060.

[45] Zhang, H., N. Lu, C. Feng, S. W. Thurston, Y. Xia, L. Zhu, and X. M. Tu (2011). On fitting generalized linear mixed-effects models for binary responses using different statistical packages. *Statistics in Medicine 30*(20), 2562–2572.

APPENDIX A. SELECTED R CODES FOR SIMULATION STUDIES

Simulation Example in CHAPTER 1

```
library(gee)
library(MuMIn)
library(copula)
library(MASS)
###### The parameters
beta_0 <- 1
beta_1 <- 1
beta_2 <- 1
beta_3 <- 1
beta_4 <- 1
###### The longitudinal data structure has %n subjects, each with %m replications
n <- 50
m <- 3  ###### Treated as the number of random effects
k <- c(2:11)
N <- n*m
###### Group index
Group <- rep(1,3)
for (i in 2:n){
  Group <- c(Group, rep(i, 3))
}
##### Function of computing Quasi-Likelihood
GetQuasi <- function(fitted_model){
  y <- fitted_model$y
  sum((y - fitted_model$fitted.values)^2/-2)
}
##### Function of computing the penalty term of QIC
GetQIC.P<- function(fit){
  #update the model with independent working correlation
  model.R <- fit
  model.independence <- suppressMessages(update(fit, corstr = "independence"))
  #obtain trace term of QAIC
  AIinverse <- solve(model.independence$naive.variance)
  V.msR <- model.R$robust.variance
  2 * sum(diag(AIinverse%*%V.msR))
}
###### Number of iterations
```

```
itr <- 1000
#### Start of the iteration
Freq.QIC.all <- matrix(rep(0,32), nrow = 4)
for(tao in 1:4){
  ###### The true correlation structure
  p <- 0.2 * tao
  myCop <- normalCopula(p, dim = 3, dispstr = "ex")

  Freq.QIC.New <- rep(0,8)
  QIC.New <- matrix(rep(0,itr * 8), nrow = itr)
  Penality.QIC <- matrix(rep(0,itr * 8), nrow = itr)
  Quasi.liklihood <- matrix(rep(0,itr * 8), nrow = itr)

  for(pitao in 1:itr)
  {
    ###### Generation of Data
    x_1 <- rnorm(n * m, 0, 1)
    x_2 <- rnorm(n * m, 0, 1)
    x_3 <- rnorm(n * m, 0, 1)
    x_4 <- rnorm(n * m, 0, 1)
    x_5 <- rnorm(n * m, 0, 1)
    x_6 <- rnorm(n * m, 0, 1)
    x_7 <- rnorm(n * m, 0, 1)
    x_8 <- rnorm(n * m, 0, 1)

    mu <- beta_0 + beta_1 * x_1 + beta_2 * x_2 + beta_3 * x_3 + beta_4 * x_4

    y <- matrix(rep(0, n * m), ncol = 3)
    for (i in 1 : n){
      mymvd <- mvdc(copula=myCop, margins=c("norm", "norm", "norm"),
                    paramMargins=list(list(mean = mu[3*i-2], sd = 1),
                                      list(mean = mu[3*i-1], sd = 1),
                                      list(mean = mu[3*i], sd = 1)))
      y[i,] <- rMvdc(1, mymvd)
    }
    y <- as.vector(t(y))

    Data <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, y))

    #### The original QIC values of all the candidate models
```

```
    fit_1_ori <- suppressMessages(gee(y ~ x_1 + 0, id = Group,data = Data, corstr = "exchangeable
        ", scale.fix = TRUE))
    fit_2_ori <- suppressMessages(gee(y ~ x_1 + x_2 + 0, id = Group,data = Data, corstr = "
        exchangeable", scale.fix = TRUE))
    fit_3_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + 0, id = Group,data = Data, corstr = "
        exchangeable", scale.fix = TRUE))
    fit_4_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + 0, id = Group,data = Data,
        corstr = "exchangeable", scale.fix = TRUE))
    fit_5_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + 0, id = Group,data = Data
        , corstr = "exchangeable", scale.fix = TRUE))
    fit_6_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + 0, id = Group,data
        = Data, corstr = "exchangeable", scale.fix = TRUE))
    fit_7_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + 0, id = Group
        ,data = Data, corstr = "exchangeable", scale.fix = TRUE))
    fit_8_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + 0, id =
         Group,data = Data, corstr = "exchangeable", scale.fix = TRUE))

    Quasi.liklihood[pitao,] <- c(GetQuasi(fit_1_ori), GetQuasi(fit_2_ori), GetQuasi(fit_3_ori),
        GetQuasi(fit_4_ori), GetQuasi(fit_5_ori), GetQuasi(fit_6_ori), GetQuasi(fit_7_ori),
        GetQuasi(fit_8_ori))

    Penality.QIC[pitao,] <- c(GetQIC.P(fit_1_ori), GetQIC.P(fit_2_ori), GetQIC.P(fit_3_ori),
        GetQIC.P(fit_4_ori),GetQIC.P(fit_5_ori), GetQIC.P(fit_6_ori), GetQIC.P(fit_7_ori), GetQIC
        .P(fit_8_ori))

    QIC.New[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.QIC[pitao,]
    Freq.QIC.New[which.min(QIC.New[pitao,])] <- Freq.QIC.New[which.min(QIC.New[pitao,])] + 1
  }
  Freq.QIC.all[tao,] <- Freq.QIC.New
}
Freq.QIC.all
```

Simulation Studies in Chapter 6

```
### Functions for computing first variants of QAICb1 and QAICb2 (Continuous Variable) (Overfitted
      Version)

Quasi.b1.1 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,2]))
```

```
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum((y.boot - mu.fit.2)^2/-2)
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b1.12 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:3)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum((y.boot - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b1.13 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:4)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum((y.boot - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
```

```
Quasi.b1.14 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:5)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum((y.boot - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.15 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:6)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum((y.boot - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.16 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:7)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum((y.boot - mu.fit.2)^2/-2)
```

```
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.17 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:8)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum((y.boot - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.18 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:9)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum((y.boot - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.19 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:10)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
```

```
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum((y.boot - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.110 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:11)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum((y.boot - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}

### Functions for computing second variants of QAICb1 and QAICb2 (Continuous Variable) (
    Overfitted Version)

Quasi.b2.1 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,2]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum((y.ori - mu.fit.2)^2/-2)
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b2.12 <- function(fit.ori, fit.boot, data){
  ##### First Part
```

```
  X <- as.matrix(cbind(int, data[,c(2:3)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum((y.ori - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b2.13 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:4)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum((y.ori - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b2.14 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:5)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum((y.ori - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.15 <- function(fit.ori, fit.boot, data){
```

```
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:6)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum((y.ori - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.16 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:7)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum((y.ori - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.17 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:8)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum((y.ori - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
```

```
Quasi.b2.18 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:9)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum((y.ori - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.19 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:10)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum((y.ori - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.110 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:11)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- X %*% beta.boot
  quasi.fit.1 <- sum((y.ori - mu.fit.1)^2/-2)
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum((y.ori - mu.fit.2)^2/-2)
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
```

```
}



###### The true correlation structure
###### With different true correlation, "dispstr" can be changed
myCop <- normalCopula(0.2, dim = 3, dispstr = "ex")
##### Sigma = 1
###### The parameters
beta_0 <- 2
beta_1 <- 2
beta_2 <- 1
beta_3 <- 1
beta_4 <- 0.5

###### The longitudinal data structure has %n subjects, each with %m replications
n <- 25
m <- 3  ###### Treated as the number of random effects
k <- c(2:11)
N <- n*m

###### Group index
Group <- rep(1,3)
for (i in 2:n){
  Group <- c(Group, rep(i, 3))
}

##### Function of computing Quasi-Likelihood
GetQuasi <- function(fitted_model){
  y <- fitted_model$y
  sum((y - fitted_model$fitted.values)^2/-2)
}

##### Function of computing the penalty term of QIC
GetQIC.P<- function(fit){
  #
  model.R <- fit
  model.independence <- suppressMessages(update(fit, corstr = "independence"))
  #obtain trace term of QAIC
  AIinverse <- solve(model.independence$naive.variance)
  V.msR <- model.R$robust.variance
  2 * sum(diag(AIinverse%*%V.msR))
```

```
}

###### Number of iterations
itr <- 1000

#### Start of the iteration
Freq.QIC.b1 <- rep(0,10)
Freq.QIC.b2 <- rep(0,10)
Freq.QAIC <- rep(0,10)
Freq.QIC.New <- rep(0,10)

QIC.New <- matrix(rep(0,itr * 10), nrow = itr)
QIC.ori <- matrix(rep(0,itr * 10), nrow = itr)
QIC.b1 <- matrix(rep(0,itr * 10), nrow = itr)
QIC.b2 <- matrix(rep(0,itr * 10), nrow = itr)
QAIC <- matrix(rep(0,itr * 10), nrow = itr)

Penality.QIC <- matrix(rep(0,itr * 10), nrow = itr)
Penality.b1 <- matrix(rep(0,itr * 10), nrow = itr)
Penality.b2 <- matrix(rep(0,itr * 10), nrow = itr)
Quasi.liklihood <- matrix(rep(0,itr * 10), nrow = itr)

################################################
for(pitao in 1:itr)
{
  set.seed(as.numeric(Sys.time()))
  ########
  sink("file")

  ###### Generation of Data
  x_1 <- rnorm(n * m, 0, 1)
  x_2 <- rnorm(n * m, 0, 1)
  x_3 <- rnorm(n * m, 0, 1)
  x_4 <- rnorm(n * m, 0, 1)
  x_5 <- rnorm(n * m, 0, 1)
  x_6 <- rnorm(n * m, 0, 1)
  x_7 <- rnorm(n * m, 0, 1)
  x_8 <- rnorm(n * m, 0, 1)
  x_9 <- rnorm(n * m, 0, 1)
  x_10 <- rnorm(n * m, 0, 1)
```

```
mu <- beta_0 + beta_1 * x_1 + beta_2 * x_2 + beta_3 * x_3 + beta_4 * x_4

y <- matrix(rep(0, n * m), ncol = 3)
for (i in 1 : n){
  mymvd <- mvdc(copula=myCop, margins=c("norm", "norm", "norm"),
                paramMargins=list(list(mean = mu[3*i-2], sd = 1),
                                  list(mean = mu[3*i-1], sd = 1),
                                  list(mean = mu[3*i], sd = 1)))
  y[i,] <- rMvdc(1, mymvd)
}
y <- as.vector(t(y))

Data <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_10, y))

#### The original QIC values of all the candidate models
fit_1_ori <- suppressMessages(gee(y ~ x_1, id = Group,data = Data, corstr = "exchangeable",
    scale.fix = TRUE))
fit_2_ori <- suppressMessages(gee(y ~ x_1 + x_2, id = Group,data = Data, corstr = "exchangeable
    ", scale.fix = TRUE))
fit_3_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3, id = Group,data = Data, corstr = "
    exchangeable", scale.fix = TRUE))
fit_4_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4, id = Group,data = Data, corstr = "
    exchangeable", scale.fix = TRUE))
fit_5_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5, id = Group,data = Data,
    corstr = "exchangeable", scale.fix = TRUE))
fit_6_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6, id = Group,data = Data
    , corstr = "exchangeable", scale.fix = TRUE))
fit_7_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7, id = Group,data
    = Data, corstr = "exchangeable", scale.fix = TRUE))
fit_8_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8, id = Group
    ,data = Data, corstr = "exchangeable", scale.fix = TRUE))
fit_9_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9, id =
     Group,data = Data, corstr = "exchangeable", scale.fix = TRUE))
fit_10_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_
    10, id = Group,data = Data, corstr = "exchangeable", scale.fix = TRUE))

#### Independent Update of the original model
fit_1_ind <- suppressMessages(update(fit_1_ori, corstr = "independence"))
fit_2_ind <- suppressMessages(update(fit_2_ori, corstr = "independence"))
fit_3_ind <- suppressMessages(update(fit_3_ori, corstr = "independence"))
fit_4_ind <- suppressMessages(update(fit_4_ori, corstr = "independence"))
```

```
fit_5_ind <- suppressMessages(update(fit_5_ori, corstr = "independence"))
fit_6_ind <- suppressMessages(update(fit_6_ori, corstr = "independence"))
fit_7_ind <- suppressMessages(update(fit_7_ori, corstr = "independence"))
fit_8_ind <- suppressMessages(update(fit_8_ori, corstr = "independence"))
fit_9_ind <- suppressMessages(update(fit_9_ori, corstr = "independence"))
fit_10_ind <- suppressMessages(update(fit_10_ori, corstr = "independence"))

#### Number of bootstrap samples
B <- 250

#### Bootstrap iteration to find weighted QIC
variants.b1 <- matrix(rep(0, 10*B), nrow = B)
variants.b2 <- matrix(rep(0, 10*B), nrow = B)

for(p in 1:B){
  index <- sample(c(1:n), replace = TRUE)
  Data.boot <- Data[which(Data$Group == index[1]),]
  for(boot in 2:n){
    Data.temp <- Data[which(Data$Group == index[boot]),]
    Data.boot <- rbind(Data.boot, Data.temp)
  }
  Data.boot <- Data.boot[order(Data.boot$Group),]
  Data.boot$Group <- factor(Data.boot$Group)

  ######## Fit the candidate models with bootstrap data

  fit_1 <- suppressMessages(gee(y ~ x_1, id = Group,data = Data.boot, corstr = "independence",
      scale.fix = TRUE))
  fit_2 <- suppressMessages(gee(y ~ x_1 + x_2, id = Group,data = Data.boot, corstr = "
      independence", scale.fix = TRUE))
  fit_3 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3, id = Group,data = Data.boot, corstr = "
      independence", scale.fix = TRUE))
  fit_4 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4, id = Group,data = Data.boot, corstr
      = "independence", scale.fix = TRUE))
  fit_5 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5, id = Group,data = Data.boot,
      corstr = "independence", scale.fix = TRUE))
  fit_6 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6, id = Group,data = Data.
      boot, corstr = "independence", scale.fix = TRUE))
  fit_7 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7, id = Group,data =
      Data.boot, corstr = "independence", scale.fix = TRUE))
  fit_8 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8, id = Group,
```

```
    data = Data.boot, corstr = "independence", scale.fix = TRUE))
  fit_9 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9, id =
    Group,data = Data.boot, corstr = "independence", scale.fix = TRUE))
  fit_10 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_
    10, id = Group,data = Data.boot, corstr = "independence", scale.fix = TRUE))

  ########
  int <- as.vector(rep(1, n * m))

  ######## First bootstrap variants
  variants.b1[p, ] <- c(Quasi.b1.1(fit_1_ind, fit_1, Data), Quasi.b1.12(fit_2_ind, fit_2, Data)
    , Quasi.b1.13(fit_3_ind, fit_3, Data), Quasi.b1.14(fit_4_ind, fit_4, Data), Quasi.b1.15(
    fit_5_ind, fit_5, Data), Quasi.b1.16(fit_6_ind, fit_6, Data), Quasi.b1.17(fit_7_ind, fit_
    7, Data), Quasi.b1.18(fit_8_ind, fit_8, Data), Quasi.b1.19(fit_9_ind, fit_9, Data), Quasi
    .b1.110(fit_10_ind, fit_10, Data))



  ######## Second bootstrap variants
  variants.b2[p, ] <- c(Quasi.b2.1(fit_1_ind, fit_1, Data), Quasi.b2.12(fit_2_ind, fit_2, Data)
    , Quasi.b2.13(fit_3_ind, fit_3, Data), Quasi.b2.14(fit_4_ind, fit_4, Data), Quasi.b2.15(
    fit_5_ind, fit_5, Data), Quasi.b2.16(fit_6_ind, fit_6, Data), Quasi.b2.17(fit_7_ind, fit_
    7, Data), Quasi.b2.18(fit_8_ind, fit_8, Data), Quasi.b2.19(fit_9_ind, fit_9, Data), Quasi
    .b2.110(fit_10_ind, fit_10, Data))
}

########## The Quasi-likelihood of the original data
y <- Data$y

Quasi.liklihood[pitao,] <- c(GetQuasi(fit_1_ori), GetQuasi(fit_2_ori), GetQuasi(fit_3_ori),
    GetQuasi(fit_4_ori),
                             GetQuasi(fit_5_ori), GetQuasi(fit_6_ori), GetQuasi(fit_7_ori),
                                 GetQuasi(fit_8_ori),
                             GetQuasi(fit_9_ori), GetQuasi(fit_10_ori))

########## QAIC
QAIC[pitao,] <- -2 * Quasi.liklihood[pitao,] + 2*k
Freq.QAIC[which.min(QAIC[pitao,])] <- Freq.QAIC[which.min(QAIC[pitao,])] + 1

########## Test the results of self-defined QIC function

Penality.QIC[pitao,] <- c(GetQIC.P(fit_1_ori), GetQIC.P(fit_2_ori), GetQIC.P(fit_3_ori), GetQIC
```

```
    .P(fit_4_ori), GetQIC.P(fit_5_ori), GetQIC.P(fit_6_ori), GetQIC.P(fit_7_ori), GetQIC.P(fit_
    8_ori), GetQIC.P(fit_9_ori), GetQIC.P(fit_10_ori))
  QIC.New[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.QIC[pitao,]
  Freq.QIC.New[which.min(QIC.New[pitao,])] <- Freq.QIC.New[which.min(QIC.New[pitao,])] + 1

  ########## First bootstrap-based QIC
  Penality.b1[pitao,] <- apply(variants.b1, 2, mean)
  QIC.b1[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.b1[pitao,]
  Freq.QIC.b1[which.min(QIC.b1[pitao,])] <- Freq.QIC.b1[which.min(QIC.b1[pitao,])] + 1

  ########## Second bootstrap-based QIC
  Penality.b2[pitao,] <- 2 * apply(variants.b2, 2, mean)
  QIC.b2[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.b2[pitao,]
  Freq.QIC.b2[which.min(QIC.b2[pitao,])] <- Freq.QIC.b2[which.min(QIC.b2[pitao,])] + 1

  ######
  sink()

  print(pitao)
  print(Quasi.liklihood[pitao,])
  print(rbind(2*k, Penality.QIC[pitao,], Penality.b1[pitao,], Penality.b2[pitao,]))
  print(rbind(QIC.New[pitao,], QAIC[pitao,], QIC.b1[pitao,], QIC.b2[pitao,]))
  print(rbind(Freq.QIC.New, Freq.QAIC, Freq.QIC.b1, Freq.QIC.b2))
}



###### The true correlation structure
myCop_ex <- normalCopula(0.5, dim = 3, dispstr = "ex")  # 30%
myCop_ar <- normalCopula(0.5, dim = 3, dispstr = "ar1") # 30%
myCop_un <- normalCopula(c(0.9, 0.6, 0.7), dim = 3, dispstr = "un") # 40%

###### The parameters
beta_0 <- 2
beta_1 <- 2
beta_2 <- 1
beta_3 <- 1
beta_4 <- 0.5

###### The longitudinal data structure has %n subjects, each with %m replications
n <- 200
n_ex <- 0.3*n
```

```
n_ar <- 0.3*n
n_un <- 0.4*n

############
m <- 3  ###### Treated as the number of random effects
k <- c(2:11)
N <- n*m

###### Group index
Group <- rep(1,3)
for (i in 2:n){
  Group <- c(Group, rep(i, 3))
}

###### Number of iterations
itr <- 1000

#### Start of the iteration
for(pitao in 1:itr)
{
  set.seed(as.numeric(Sys.time()))
  ########
  sink("file")

  ###### Generation of Data
  x_1 <- rnorm(n * m, 0, 1)
  x_2 <- rnorm(n * m, 0, 1)
  x_3 <- rnorm(n * m, 0, 1)
  x_4 <- rnorm(n * m, 0, 1)
  x_5 <- rnorm(n * m, 0, 1)
  x_6 <- rnorm(n * m, 0, 1)
  x_7 <- rnorm(n * m, 0, 1)
  x_8 <- rnorm(n * m, 0, 1)
  x_9 <- rnorm(n * m, 0, 1)
  x_10 <- rnorm(n * m, 0, 1)

  mu <- beta_0 + beta_1 * x_1 + beta_2 * x_2 + beta_3 * x_3 + beta_4 * x_4
  y <- matrix(rep(0, n * m), ncol = 3)

  ###############################################################
  ### EX
```

```
  for (i in 1 : n_ex){
    mymvd <- mvdc(copula=myCop_ex, margins=c("norm", "norm", "norm"),
                  paramMargins=list(list(mean = mu[3*i-2], sd = 1),
                                    list(mean = mu[3*i-1], sd = 1),
                                    list(mean = mu[3*i], sd = 1)))
    y[i,] <- rMvdc(1, mymvd)
  }

  ### AR
  for (i in (n_ex + 1) : (n_ex + n_ar)){
    mymvd <- mvdc(copula=myCop_ar, margins=c("norm", "norm", "norm"),
                  paramMargins=list(list(mean = mu[3*i-2], sd = 1),
                                    list(mean = mu[3*i-1], sd = 1),
                                    list(mean = mu[3*i], sd = 1)))
    y[i,] <- rMvdc(1, mymvd)
  }

  ### UN
  y_un <- matrix(rep(0, n_un * m), ncol = 3)
  for (i in (n_ex + n_ar + 1) : n){
    mymvd <- mvdc(copula=myCop_un, margins=c("norm", "norm", "norm"),
                  paramMargins=list(list(mean = mu[3*i-2], sd = 1),
                                    list(mean = mu[3*i-1], sd = 1),
                                    list(mean = mu[3*i], sd = 1)))
    y[i,] <- rMvdc(1, mymvd)
  }
  y <- as.vector(t(y))

  Data <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_10, y))

  ####### All the rest follows the same estimating procedure as above
}



###### The true correlation structure
myCop <- normalCopula(0.2, dim = 3, dispstr = "ex")
##### Sigma = 1
sig = 1
n <- 25

###### The parameters
```

```
beta_0 <- 1
beta_1 <- -1
beta_2 <- 1
beta_4 <- -1

###### The longitudinal data structure has %n subjects, each with %m replications
m <- 3  ###### Treated as the number of random effects
k <- c(3,4,4,4,5,5,5,6)
N <- n*m

###### Group index
Group <- rep(1,3)
for (i in 2:n){
  Group <- c(Group, rep(i, 3))
}

###### Number of iterations
itr <- 1000
###### Number of models
M <- 8

#### Start of the iteration
for(pitao in 1:itr)
{
  set.seed(as.numeric(Sys.time()))
  ########
  sink("file")

  ###### Generation of Data
  x_1 <- rnorm(n * m, 0, 1)
  x_2 <- rnorm(n * m, 0, 1)
  x_3 <- rnorm(n * m, 0, 1)
  x_4 <- rnorm(n * m, 0, 1)
  x_5 <- rnorm(n * m, 0, 1)

  mu <- beta_0 + beta_1 * x_1 + beta_2 * x_2 + beta_4 * x_4

  y <- matrix(rep(0, n * m), ncol = 3)
  for (i in 1 : n){
    mymvd <- mvdc(copula=myCop, margins=c("norm", "norm", "norm"),
                  paramMargins=list(list(mean = mu[3*i-2], sd = sig),
```

```
                                    list(mean = mu[3*i-1], sd = sig),
                                    list(mean = mu[3*i], sd = sig)))
  y[i,] <- rMvdc(1, mymvd)
}
y <- as.vector(t(y))

Data <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, y))

#### The original QIC values of all the candidate models
fit_1_ori <- suppressMessages(gee(y ~ x_1 + x_2, id = Group,data = Data, corstr = "exchangeable
    ", scale.fix = TRUE))
fit_2_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3, id = Group,data = Data, corstr = "
    exchangeable", scale.fix = TRUE))
fit_3_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_4, id = Group,data = Data, corstr = "
    exchangeable", scale.fix = TRUE))
fit_4_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_5, id = Group,data = Data, corstr = "
    exchangeable", scale.fix = TRUE))
fit_5_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4, id = Group,data = Data, corstr = "
    exchangeable", scale.fix = TRUE))
fit_6_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_5, id = Group,data = Data, corstr = "
    exchangeable", scale.fix = TRUE))
fit_7_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_4 + x_5 , id = Group,data = Data, corstr =
    "exchangeable", scale.fix = TRUE))
fit_8_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5, id = Group,data = Data,
    corstr = "exchangeable", scale.fix = TRUE))

#### Independent Update of the original model

fit_1_ind <- suppressMessages(update(fit_1_ori, corstr = "independence"))
fit_2_ind <- suppressMessages(update(fit_2_ori, corstr = "independence"))
fit_3_ind <- suppressMessages(update(fit_3_ori, corstr = "independence"))
fit_4_ind <- suppressMessages(update(fit_4_ori, corstr = "independence"))
fit_5_ind <- suppressMessages(update(fit_5_ori, corstr = "independence"))
fit_6_ind <- suppressMessages(update(fit_6_ori, corstr = "independence"))
fit_7_ind <- suppressMessages(update(fit_7_ori, corstr = "independence"))
fit_8_ind <- suppressMessages(update(fit_8_ori, corstr = "independence"))

#### Number of bootstrap samples
B <- 250

#### Bootstrap iteration to find weighted QIC
```

```
variants.b1 <- matrix(rep(0, M*B), nrow = B)
variants.b2 <- matrix(rep(0, M*B), nrow = B)

for(p in 1:B){
  index <- sample(c(1:n), replace = TRUE)
  Data.boot <- Data[which(Data$Group == index[1]),]
  for(boot in 2:n){
    Data.temp <- Data[which(Data$Group == index[boot]),]
    Data.boot <- rbind(Data.boot, Data.temp)
  }
  Data.boot <- Data.boot[order(Data.boot$Group),]
  Data.boot$Group <- factor(Data.boot$Group)

  ######## Fit the candidate models with bootstrap data

  fit_1 <- suppressMessages(gee(y ~ x_1 + x_2, id = Group,data = Data.boot, corstr = "
      independence", scale.fix = TRUE))
  fit_2 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3, id = Group,data = Data.boot, corstr = "
      independence", scale.fix = TRUE))
  fit_3 <- suppressMessages(gee(y ~ x_1 + x_2 + x_4, id = Group,data = Data.boot, corstr = "
      independence", scale.fix = TRUE))
  fit_4 <- suppressMessages(gee(y ~ x_1 + x_2 + x_5, id = Group,data = Data.boot, corstr = "
      independence", scale.fix = TRUE))
  fit_5 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4, id = Group,data = Data.boot, corstr
      = "independence", scale.fix = TRUE))
  fit_6 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_5, id = Group,data = Data.boot, corstr
      = "independence", scale.fix = TRUE))
  fit_7 <- suppressMessages(gee(y ~ x_1 + x_2 + x_4 + x_5, id = Group,data = Data.boot, corstr
      = "independence", scale.fix = TRUE))
  fit_8 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5, id = Group,data = Data.boot,
      corstr = "independence", scale.fix = TRUE))

  ########
  int <- as.vector(rep(1, n * m))

  ######## First bootstrap variants
  variants.b1[p, ] <- c(Quasi.b1.1(fit_1_ind, fit_1, Data), Quasi.b1.12(fit_2_ind, fit_2, Data)
      , Quasi.b1.13(fit_3_ind, fit_3, Data), Quasi.b1.14(fit_4_ind, fit_4, Data), Quasi.b1.15(
      fit_5_ind, fit_5, Data), Quasi.b1.16(fit_6_ind, fit_6, Data), Quasi.b1.17(fit_7_ind, fit_
      7, Data), Quasi.b1.18(fit_8_ind, fit_8, Data))
```

```
  ######## Second bootstrap variants
  variants.b2[p, ] <- c(Quasi.b2.1(fit_1_ind, fit_1, Data), Quasi.b2.12(fit_2_ind, fit_2, Data)
      , Quasi.b2.13(fit_3_ind, fit_3, Data), Quasi.b2.14(fit_4_ind, fit_4, Data), Quasi.b2.15(
      fit_5_ind, fit_5, Data), Quasi.b2.16(fit_6_ind, fit_6, Data), Quasi.b2.17(fit_7_ind, fit_
      7, Data), Quasi.b2.18(fit_8_ind, fit_8, Data))
}

########## The Quasi-likelihood of the original data
y <- Data$y

Quasi.liklihood[pitao,] <- c(GetQuasi(fit_1_ori), GetQuasi(fit_2_ori), GetQuasi(fit_3_ori),
    GetQuasi(fit_4_ori), GetQuasi(fit_5_ori), GetQuasi(fit_6_ori), GetQuasi(fit_7_ori),
    GetQuasi(fit_8_ori))

########## QAIC
QAIC[pitao,] <- -2 * Quasi.liklihood[pitao,] + 2*k
Freq.QAIC[which.min(QAIC[pitao,])] <- Freq.QAIC[which.min(QAIC[pitao,])] + 1

########## Test the results of self-defined QIC function
Penality.QIC[pitao,] <- c(GetQIC.P(fit_1_ori), GetQIC.P(fit_2_ori), GetQIC.P(fit_3_ori), GetQIC
    .P(fit_4_ori), GetQIC.P(fit_5_ori), GetQIC.P(fit_6_ori), GetQIC.P(fit_7_ori), GetQIC.P(fit_
    8_ori))

QIC.New[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.QIC[pitao,]
Freq.QIC.New[which.min(QIC.New[pitao,])] <- Freq.QIC.New[which.min(QIC.New[pitao,])] + 1

########## First bootstrap-based QIC
Penality.b1[pitao,] <- apply(variants.b1, 2, mean)
QIC.b1[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.b1[pitao,]
Freq.QIC.b1[which.min(QIC.b1[pitao,])] <- Freq.QIC.b1[which.min(QIC.b1[pitao,])] + 1

########## Second bootstrap-based QIC
Penality.b2[pitao,] <- 2 * apply(variants.b2, 2, mean)
QIC.b2[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.b2[pitao,]
Freq.QIC.b2[which.min(QIC.b2[pitao,])] <- Freq.QIC.b2[which.min(QIC.b2[pitao,])] + 1

######
sink()

print(pitao)
print(Quasi.liklihood[pitao,])
```

```
  print(rbind(2*k, Penality.QIC[pitao,], Penality.b1[pitao,], Penality.b2[pitao,]))
  print(rbind(QIC.New[pitao,], QAIC[pitao,], QIC.b1[pitao,], QIC.b2[pitao,]))
  print(rbind(Freq.QIC, Freq.QIC.New, Freq.QAIC, Freq.QIC.b1, Freq.QIC.b2))
}



###### The true correlation structure
myCop_ex <- normalCopula(0.5, dim = 3, dispstr = "ex")  # 30%
myCop_ar <- normalCopula(0.5, dim = 3, dispstr = "ar1") # 30%
myCop_un <- normalCopula(c(0.9, 0.6, 0.7), dim = 3, dispstr = "un") # 40%

###### The parameters
beta_0 <- 2
beta_1 <- 2
beta_2 <- 1
beta_3 <- 1
beta_4 <- 0.5

###### The longitudinal data structure has %n subjects, each with %m replications
n <- 200
n_ex <- 0.3*n
n_ar <- 0.3*n
n_un <- 0.4*n

#############
m <- 3  ###### Treated as the number of random effects
k <- c(2:11)
N <- n*m

###### Group index
Group <- rep(1,3)
for (i in 2:n){
  Group <- c(Group, rep(i, 3))
}

###### Number of iterations
itr <- 1000
#### Start of the iteration
for(pitao in 1:itr)
{
  set.seed(as.numeric(Sys.time()))
```

```
########
sink("file")

###### Generation of Data
x_1 <- rnorm(n * m, 0, 1)
x_2 <- rnorm(n * m, 0, 1)
x_3 <- rnorm(n * m, 0, 1)
x_4 <- rnorm(n * m, 0, 1)
x_5 <- rnorm(n * m, 0, 1)
x_6 <- rnorm(n * m, 0, 1)
x_7 <- rnorm(n * m, 0, 1)
x_8 <- rnorm(n * m, 0, 1)
x_9 <- rnorm(n * m, 0, 1)
x_10 <- rnorm(n * m, 0, 1)

mu <- beta_0 + beta_1 * x_1 + beta_2 * x_2 + beta_3 * x_3 + beta_4 * x_4
y <- matrix(rep(0, n * m), ncol = 3)

#################################################################
### EX
for (i in 1 : n_ex){
  mymvd <- mvdc(copula=myCop_ex, margins=c("norm", "norm", "norm"),
                paramMargins=list(list(mean = mu[3*i-2], sd = 1),
                                  list(mean = mu[3*i-1], sd = 1),
                                  list(mean = mu[3*i], sd = 1)))
  y[i,] <- rMvdc(1, mymvd)
}

### AR
for (i in (n_ex + 1) : (n_ex + n_ar)){
  mymvd <- mvdc(copula=myCop_ar, margins=c("norm", "norm", "norm"),
                paramMargins=list(list(mean = mu[3*i-2], sd = 1),
                                  list(mean = mu[3*i-1], sd = 1),
                                  list(mean = mu[3*i], sd = 1)))
  y[i,] <- rMvdc(1, mymvd)
}

### UN
y_un <- matrix(rep(0, n_un * m), ncol = 3)
for (i in (n_ex + n_ar + 1) : n){
  mymvd <- mvdc(copula=myCop_un, margins=c("norm", "norm", "norm"),
```

```
                paramMargins=list(list(mean = mu[3*i-2], sd = 1),
                                  list(mean = mu[3*i-1], sd = 1),
                                  list(mean = mu[3*i], sd = 1)))
  y[i,] <- rMvdc(1, mymvd)
}
y <- as.vector(t(y))

Data <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_10, y))

#### The original QIC values of all the candidate models
fit_1_ori <- suppressMessages(gee(y ~ x_1, id = Group,data = Data, corstr = "exchangeable",
    scale.fix = TRUE))
fit_2_ori <- suppressMessages(gee(y ~ x_1 + x_2, id = Group,data = Data, corstr = "exchangeable
    ", scale.fix = TRUE))
fit_3_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3, id = Group,data = Data, corstr = "
    exchangeable", scale.fix = TRUE))
fit_4_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4, id = Group,data = Data, corstr = "
    exchangeable", scale.fix = TRUE))
fit_5_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5, id = Group,data = Data,
    corstr = "exchangeable", scale.fix = TRUE))
fit_6_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6, id = Group,data = Data
    , corstr = "exchangeable", scale.fix = TRUE))
fit_7_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7, id = Group,data
    = Data, corstr = "exchangeable", scale.fix = TRUE))
fit_8_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8, id = Group
    ,data = Data, corstr = "exchangeable", scale.fix = TRUE))
fit_9_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9, id =
     Group,data = Data, corstr = "exchangeable", scale.fix = TRUE))
fit_10_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_
    10, id = Group,data = Data, corstr = "exchangeable", scale.fix = TRUE))

#### Independent Update of the original model
fit_1_ind <- suppressMessages(update(fit_1_ori, corstr = "independence"))
fit_2_ind <- suppressMessages(update(fit_2_ori, corstr = "independence"))
fit_3_ind <- suppressMessages(update(fit_3_ori, corstr = "independence"))
fit_4_ind <- suppressMessages(update(fit_4_ori, corstr = "independence"))
fit_5_ind <- suppressMessages(update(fit_5_ori, corstr = "independence"))
fit_6_ind <- suppressMessages(update(fit_6_ori, corstr = "independence"))
fit_7_ind <- suppressMessages(update(fit_7_ori, corstr = "independence"))
fit_8_ind <- suppressMessages(update(fit_8_ori, corstr = "independence"))
fit_9_ind <- suppressMessages(update(fit_9_ori, corstr = "independence"))
```

```
fit_10_ind <- suppressMessages(update(fit_10_ori, corstr = "independence"))

#### Number of bootstrap samples
B <- 250

#### Bootstrap iteration to find weighted QIC
variants.b1 <- matrix(rep(0, 10*B), nrow = B)
variants.b2 <- matrix(rep(0, 10*B), nrow = B)

#residuals <- matrix(rep(0, 10*m*n), ncol = 10)
residuals_1 <- data.frame(cbind(Group, fit_1_ori$residuals))
residuals_2 <- data.frame(cbind(Group, fit_2_ori$residuals))
residuals_3 <- data.frame(cbind(Group, fit_3_ori$residuals))
residuals_4 <- data.frame(cbind(Group, fit_4_ori$residuals))
residuals_5 <- data.frame(cbind(Group, fit_5_ori$residuals))
residuals_6 <- data.frame(cbind(Group, fit_6_ori$residuals))
residuals_7 <- data.frame(cbind(Group, fit_7_ori$residuals))
residuals_8 <- data.frame(cbind(Group, fit_8_ori$residuals))
residuals_9 <- data.frame(cbind(Group, fit_9_ori$residuals))
residuals_10 <- data.frame(cbind(Group, fit_10_ori$residuals))

int <- as.vector(rep(1, n * m))

####Model 1####
for(p in 1:B){
  index <- sample(c(1:n), replace = TRUE)
  residuals.boot <- residuals_1[which(residuals_1$Group == index[1]),]
  for(boot in 2:n){
    residuals.temp <- residuals_1[which(residuals_1$Group == index[boot]),]
    residuals.boot <- rbind(residuals.boot, residuals.temp)
  }
  residuals.boot <- residuals.boot[order(residuals.boot$Group),]
  ########
  y.boot <- as.matrix(cbind(int, Data[,2])) %*% as.vector(fit_1_ori$coefficients) + residuals.
      boot[,2]
  Data.boot <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_10, "y.
      boot" = as.numeric(y.boot)))
  ######## Fit the candidate models with bootstrap data
  fit_1_ind <- suppressMessages(update(fit_1_ori, corstr = "independence"))
  fit_1 <- suppressMessages(gee(y.boot ~ x_1, id = Group,data = Data.boot, corstr = "
      independence", scale.fix = TRUE))
```

```
  variants.b1[p, 1] <- Quasi.b1.1(fit_1_ind, fit_1, Data)
  variants.b2[p, 1] <- Quasi.b2.1(fit_1_ind, fit_1, Data)
}
####Model 2####
for(p in 1:B){
  index <- sample(c(1:n), replace = TRUE)
  residuals.boot <- residuals_2[which(residuals_2$Group == index[1]),]
  for(boot in 2:n){
    residuals.temp <- residuals_2[which(residuals_2$Group == index[boot]),]
    residuals.boot <- rbind(residuals.boot, residuals.temp)
  }
  residuals.boot <- residuals.boot[order(residuals.boot$Group),]
  ########
  y.boot <- as.matrix(cbind(int, Data[,c(2:3)])) %*% as.vector(fit_2_ori$coefficients) +
      residuals.boot[,2]
  Data.boot <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_10, "y.
      boot" = as.numeric(y.boot)))
  ######## Fit the candidate models with bootstrap data
  fit_2_ind <- suppressMessages(update(fit_2_ori, corstr = "independence"))
  fit_2 <- suppressMessages(gee(y.boot ~ x_1 + x_2, id = Group,data = Data.boot, corstr = "
      independence", scale.fix = TRUE))

  variants.b1[p, 2] <- Quasi.b1.12(fit_2_ind, fit_2, Data)
  variants.b2[p, 2] <- Quasi.b2.12(fit_2_ind, fit_2, Data)
}
####Model 3####
for(p in 1:B){
  index <- sample(c(1:n), replace = TRUE)
  residuals.boot <- residuals_3[which(residuals_3$Group == index[1]),]
  for(boot in 2:n){
    residuals.temp <- residuals_3[which(residuals_3$Group == index[boot]),]
    residuals.boot <- rbind(residuals.boot, residuals.temp)
  }
  residuals.boot <- residuals.boot[order(residuals.boot$Group),]
  ########
  y.boot <- as.matrix(cbind(int, Data[,c(2:4)])) %*% as.vector(fit_3_ori$coefficients) +
      residuals.boot[,2]
  Data.boot <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_10, "y.
      boot" = as.numeric(y.boot)))
  ######## Fit the candidate models with bootstrap data
```

```
  fit_3_ind <- suppressMessages(update(fit_3_ori, corstr = "independence"))
  fit_3 <- suppressMessages(gee(y.boot ~ x_1 + x_2 + x_3, id = Group,data = Data.boot, corstr =
      "independence", scale.fix = TRUE))

  variants.b1[p, 3] <- Quasi.b1.13(fit_3_ind, fit_3, Data)
  variants.b2[p, 3] <- Quasi.b2.13(fit_3_ind, fit_3, Data)
}
####Model 4####
for(p in 1:B){
  index <- sample(c(1:n), replace = TRUE)
  residuals.boot <- residuals_4[which(residuals_4$Group == index[1]),]
  for(boot in 2:n){
    residuals.temp <- residuals_4[which(residuals_4$Group == index[boot]),]
    residuals.boot <- rbind(residuals.boot, residuals.temp)
  }
  residuals.boot <- residuals.boot[order(residuals.boot$Group),]
  ########
  y.boot <- as.matrix(cbind(int, Data[,c(2:5)])) %*% as.vector(fit_4_ori$coefficients) +
      residuals.boot[,2]
  Data.boot <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_10, "y.
      boot" = as.numeric(y.boot)))
  ######## Fit the candidate models with bootstrap data
  fit_4_ind <- suppressMessages(update(fit_4_ori, corstr = "independence"))
  fit_4 <- suppressMessages(gee(y.boot ~ x_1 + x_2 + x_3 + x_4, id = Group,data = Data.boot,
      corstr = "independence", scale.fix = TRUE))

  variants.b1[p, 4] <- Quasi.b1.14(fit_4_ind, fit_4, Data)
  variants.b2[p, 4] <- Quasi.b2.14(fit_4_ind, fit_4, Data)
}
####Model 5####
for(p in 1:B){
  index <- sample(c(1:n), replace = TRUE)
  residuals.boot <- residuals_5[which(residuals_5$Group == index[1]),]
  for(boot in 2:n){
    residuals.temp <- residuals_5[which(residuals_5$Group == index[boot]),]
    residuals.boot <- rbind(residuals.boot, residuals.temp)
  }
  residuals.boot <- residuals.boot[order(residuals.boot$Group),]
  ########
  y.boot <- as.matrix(cbind(int, Data[,c(2:6)])) %*% as.vector(fit_5_ori$coefficients) +
      residuals.boot[,2]
```

```
  Data.boot <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_10, "y.
      boot" = as.numeric(y.boot)))
  ######## Fit the candidate models with bootstrap data
  fit_5_ind <- suppressMessages(update(fit_5_ori, corstr = "independence"))
  fit_5 <- suppressMessages(gee(y.boot ~ x_1 + x_2 + x_3 + x_4 + x_5, id = Group,data = Data.
      boot, corstr = "independence", scale.fix = TRUE))

  variants.b1[p, 5] <- Quasi.b1.15(fit_5_ind, fit_5, Data)
  variants.b2[p, 5] <- Quasi.b2.15(fit_5_ind, fit_5, Data)
}
####Model 6####
for(p in 1:B){
  index <- sample(c(1:n), replace = TRUE)
  residuals.boot <- residuals_6[which(residuals_6$Group == index[1]),]
  for(boot in 2:n){
    residuals.temp <- residuals_6[which(residuals_6$Group == index[boot]),]
    residuals.boot <- rbind(residuals.boot, residuals.temp)
  }
  residuals.boot <- residuals.boot[order(residuals.boot$Group),]
  ########
  y.boot <- as.matrix(cbind(int, Data[,c(2:7)])) %*% as.vector(fit_6_ori$coefficients) +
      residuals.boot[,2]
  Data.boot <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_10, "y.
      boot" = as.numeric(y.boot)))
  ######## Fit the candidate models with bootstrap data
  fit_6_ind <- suppressMessages(update(fit_6_ori, corstr = "independence"))
  fit_6 <- suppressMessages(gee(y.boot ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6, id = Group,data =
      Data.boot, corstr = "independence", scale.fix = TRUE))

  variants.b1[p, 6] <- Quasi.b1.16(fit_6_ind, fit_6, Data)
  variants.b2[p, 6] <- Quasi.b2.16(fit_6_ind, fit_6, Data)
}
####Model 7####
for(p in 1:B){
  index <- sample(c(1:n), replace = TRUE)
  residuals.boot <- residuals_7[which(residuals_7$Group == index[1]),]
  for(boot in 2:n){
    residuals.temp <- residuals_7[which(residuals_7$Group == index[boot]),]
    residuals.boot <- rbind(residuals.boot, residuals.temp)
  }
  residuals.boot <- residuals.boot[order(residuals.boot$Group),]
```

```
  ########
  y.boot <- as.matrix(cbind(int, Data[,c(2:8)])) %*% as.vector(fit_7_ori$coefficients) +
      residuals.boot[,2]
  Group.boot <- residuals.boot[,1]
  Data.boot <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_10, "y.
      boot" = as.numeric(y.boot)))
  ######## Fit the candidate models with bootstrap data
  fit_7_ind <- suppressMessages(update(fit_7_ori, corstr = "independence"))
  fit_7 <- suppressMessages(gee(y.boot ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7, id = Group,
      data = Data.boot, corstr = "independence", scale.fix = TRUE))

  variants.b1[p, 7] <- Quasi.b1.17(fit_7_ind, fit_7, Data)
  variants.b2[p, 7] <- Quasi.b2.17(fit_7_ind, fit_7, Data)
}
####Model 8####
for(p in 1:B){
  index <- sample(c(1:n), replace = TRUE)
  residuals.boot <- residuals_8[which(residuals_8$Group == index[1]),]
  for(boot in 2:n){
    residuals.temp <- residuals_8[which(residuals_8$Group == index[boot]),]
    residuals.boot <- rbind(residuals.boot, residuals.temp)
  }
  residuals.boot <- residuals.boot[order(residuals.boot$Group),]
  ########
  y.boot <- as.matrix(cbind(int, Data[,c(2:9)])) %*% as.vector(fit_8_ori$coefficients) +
      residuals.boot[,2]
  Data.boot <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_10, "y.
      boot" = as.numeric(y.boot)))
  ######## Fit the candidate models with bootstrap data
  fit_8_ind <- suppressMessages(update(fit_8_ori, corstr = "independence"))
  fit_8 <- suppressMessages(gee(y.boot ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8, id =
      Group,data = Data.boot, corstr = "independence", scale.fix = TRUE))

  variants.b1[p, 8] <- Quasi.b1.18(fit_8_ind, fit_8, Data)
  variants.b2[p, 8] <- Quasi.b2.18(fit_8_ind, fit_8, Data)
}
####Model 9####
for(p in 1:B){
  index <- sample(c(1:n), replace = TRUE)
  residuals.boot <- residuals_9[which(residuals_9$Group == index[1]),]
  for(boot in 2:n){
```

```
    residuals.temp <- residuals_9[which(residuals_9$Group == index[boot]),]
    residuals.boot <- rbind(residuals.boot, residuals.temp)
  }
  residuals.boot <- residuals.boot[order(residuals.boot$Group),]
  ########
  y.boot <- as.matrix(cbind(int, Data[,c(2:10)])) %*% as.vector(fit_9_ori$coefficients) +
      residuals.boot[,2]
  Data.boot <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_10, "y.
      boot" = as.numeric(y.boot)))
  ######## Fit the candidate models with bootstrap data
  fit_9_ind <- suppressMessages(update(fit_9_ori, corstr = "independence"))
  fit_9 <- suppressMessages(gee(y.boot ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9,
      id = Group,data = Data.boot, corstr = "independence", scale.fix = TRUE))

  variants.b1[p, 9] <- Quasi.b1.19(fit_9_ind, fit_9, Data)
  variants.b2[p, 9] <- Quasi.b2.19(fit_9_ind, fit_9, Data)
}
####Model 10####
for(p in 1:B){
  index <- sample(c(1:n), replace = TRUE)
  residuals.boot <- residuals_10[which(residuals_10$Group == index[1]),]
  for(boot in 2:n){
    residuals.temp <- residuals_10[which(residuals_10$Group == index[boot]),]
    residuals.boot <- rbind(residuals.boot, residuals.temp)
  }
  residuals.boot <- residuals.boot[order(residuals.boot$Group),]
  ########
  y.boot <- as.matrix(cbind(int, Data[,c(2:11)])) %*% as.vector(fit_10_ori$coefficients) +
      residuals.boot[,2]
  Data.boot <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_10, "y.
      boot" = as.numeric(y.boot)))
  ######## Fit the candidate models with bootstrap data
  fit_10_ind <- suppressMessages(update(fit_10_ori, corstr = "independence"))
  fit_10 <- suppressMessages(gee(y.boot ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 +
       x_10, id = Group,data = Data.boot, corstr = "independence", scale.fix = TRUE))

  variants.b1[p, 10] <- Quasi.b1.110(fit_10_ind, fit_10, Data)
  variants.b2[p, 10] <- Quasi.b2.110(fit_10_ind, fit_10, Data)
}
########## The Quasi-likelihood of the original data
y <- Data$y
```

```
  Quasi.liklihood[pitao,] <- c(GetQuasi(fit_1_ori), GetQuasi(fit_2_ori), GetQuasi(fit_3_ori),
      GetQuasi(fit_4_ori), GetQuasi(fit_5_ori), GetQuasi(fit_6_ori), GetQuasi(fit_7_ori),
      GetQuasi(fit_8_ori), GetQuasi(fit_9_ori), GetQuasi(fit_10_ori))

  ########## QAIC
  QAIC[pitao,] <- -2 * Quasi.liklihood[pitao,] + 2*k
  Freq.QAIC[which.min(QAIC[pitao,])] <- Freq.QAIC[which.min(QAIC[pitao,])] + 1

  ########## Self-defined QIC function
  Penality.QIC[pitao,] <- c(GetQIC.P(fit_1_ori), GetQIC.P(fit_2_ori), GetQIC.P(fit_3_ori), GetQIC
      .P(fit_4_ori), GetQIC.P(fit_5_ori), GetQIC.P(fit_6_ori), GetQIC.P(fit_7_ori), GetQIC.P(fit_
      8_ori), GetQIC.P(fit_9_ori), GetQIC.P(fit_10_ori))

  QIC.New[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.QIC[pitao,]
  Freq.QIC.New[which.min(QIC.New[pitao,])] <- Freq.QIC.New[which.min(QIC.New[pitao,])] + 1

  ########## First bootstrap-based QIC
  Penality.b1[pitao,] <- apply(variants.b1, 2, mean)
  QIC.b1[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.b1[pitao,]
  Freq.QIC.b1[which.min(QIC.b1[pitao,])] <- Freq.QIC.b1[which.min(QIC.b1[pitao,])] + 1

  ########## Second bootstrap-based QIC
  Penality.b2[pitao,] <- 2 * apply(variants.b2, 2, mean)
  QIC.b2[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.b2[pitao,]
  Freq.QIC.b2[which.min(QIC.b2[pitao,])] <- Freq.QIC.b2[which.min(QIC.b2[pitao,])] + 1

  ######
  sink()

  print(pitao)
  print(Quasi.liklihood[pitao,])
  print(rbind(2*k, Penality.QIC[pitao,], Penality.b1[pitao,], Penality.b2[pitao,]))
  print(rbind(QIC.New[pitao,], QAIC[pitao,], QIC.b1[pitao,], QIC.b2[pitao,]))
  print(rbind(Freq.QIC, Freq.QIC.New, Freq.QAIC, Freq.QIC.b1, Freq.QIC.b2))
}


### Functions for computing first variants of QAICb1 and QAICb2 (Logistic Model) (Overfitted
    Version)
```

```
Quasi.b1.1 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,2]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b1.12 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:3)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b1.13 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:4)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
```

```
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b1.14 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:5)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.15 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:6)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.16 <- function(fit.ori, fit.boot, data){
  ##### First Part
```

```
  X <- as.matrix(cbind(int, data[,c(2:7)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.17 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:8)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))

  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.18 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:9)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  y.boot <- fit.boot$y
```

```
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))

  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.19 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:10)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.110 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:11)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}

### Functions for computing second variants of QAICb1 and QAICb2 (Logistic Model) (Overfitted
    Version)
```

```
Quasi.b2.1 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,2]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b2.12 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:3)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b2.13 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:4)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
```

```
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b2.14 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:5)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.15 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:6)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.16 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:7)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
```

```
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.17 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:8)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.18 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:9)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.19 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:10)]))
  beta.boot <- as.vector(fit.boot$coefficients)
```

```
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.110 <- function(fit.ori, fit.boot, data){
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:11)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.logit.1 <- X %*% beta.boot
  mu.fit.1 <- exp(mu.logit.1)/(1 + exp(mu.logit.1))
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1/(1 - mu.fit.1)) + log(1 - mu.fit.1))
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2/(1 - mu.fit.2)) + log(1 - mu.fit.2))
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}



###### The true correlation structure
myCop <- normalCopula(0.2, dim = 3, dispstr = "ar1")

###### The parameters
beta_0 <- 0.5
beta_1 <- -0.5
beta_2 <- 0.25
beta_3 <- -0.25
beta_4 <- 0.25

##### Function of computing Quasi-Likelihood
GetQuasi <- function(fitted_model){
  y <- fitted_model$y
  mu <- fitted_model$fitted.values
```

```
  sum(y * log(mu/(1 - mu)) + log(1 - mu))
}

###### Number of iterations
itr <- 1000

###### Number of models
M <- 7
k <- c(2:(M + 1))

#### Start of the iteration
for(pitao in 1:itr)
{
  set.seed(as.numeric(Sys.time()))
  ########
  sink("file")

  ###### Generation of Data
  x_1 <- rnorm(n * m, 0, 1)
  x_2 <- rnorm(n * m, 0, 1)
  x_3 <- rnorm(n * m, 0, 1)
  x_4 <- rnorm(n * m, 0, 1)
  x_5 <- rnorm(n * m, 0, 1)
  x_6 <- rnorm(n * m, 0, 1)
  x_7 <- rnorm(n * m, 0, 1)

  mu_logit <- beta_0 + beta_1 * x_1 + beta_2 * x_2 + beta_3 * x_3  + beta_4 * x_4
  mu <- exp(mu_logit)/(1 + exp(mu_logit))

  y <- matrix(rep(0, n * m), ncol = 3)
  for (i in 1 : n){
    mymvd <- mvdc(copula=myCop, margins=c("binom", "binom", "binom"),
                  paramMargins=list(list(size = 1, prob = mu[3*i-2]),
                                    list(size = 1, prob = mu[3*i-1]),
                                    list(size = 1, prob = mu[3*i])))
    y[i,] <- rMvdc(1, mymvd)
  }
  y <- as.vector(t(y))

  Data <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, y))
```

```
#### The original QIC values of all the candidate models
tryCatch(
  {
    fit_1_ori <- suppressMessages(gee(y ~ x_1, id = Group,data = Data, family = "binomial",
        corstr = "exchangeable", scale.fix = TRUE))
    fit_2_ori <- suppressMessages(gee(y ~ x_1 + x_2, id = Group,data = Data, family = "binomial
        ", corstr = "exchangeable", scale.fix = TRUE))
    fit_3_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3, id = Group,data = Data, family = "
        binomial", corstr = "exchangeable", scale.fix = TRUE))
    fit_4_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4, id = Group,data = Data, family
         = "binomial", corstr = "exchangeable", scale.fix = TRUE))
    fit_5_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5, id = Group,data = Data,
        family = "binomial", corstr = "exchangeable", scale.fix = TRUE))
    fit_6_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6, id = Group,data =
        Data, family = "binomial", corstr = "exchangeable", scale.fix = TRUE))
    fit_7_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7, id = Group,
        data = Data, family = "binomial", corstr = "exchangeable", scale.fix = TRUE))
  }, error=function(e){cat("ERROR :",conditionMessage(e), "\n")}
)

#### Independent Update of the original model
tryCatch(
  {
fit_1_ind <- suppressMessages(update(fit_1_ori, corstr = "independence"))
fit_2_ind <- suppressMessages(update(fit_2_ori, corstr = "independence"))
fit_3_ind <- suppressMessages(update(fit_3_ori, corstr = "independence"))
fit_4_ind <- suppressMessages(update(fit_4_ori, corstr = "independence"))
fit_5_ind <- suppressMessages(update(fit_5_ori, corstr = "independence"))
fit_6_ind <- suppressMessages(update(fit_6_ori, corstr = "independence"))
fit_7_ind <- suppressMessages(update(fit_7_ori, corstr = "independence"))
  }, error=function(e){cat("ERROR :",conditionMessage(e), "\n")}
)

#### Number of bootstrap samples
B <- 200

#### Bootstrap iteration to find weighted QIC
variants.b1 <- matrix(rep(0, M*B), nrow = B)
variants.b2 <- matrix(rep(0, M*B), nrow = B)

for(p in 1:B){
```

```
index <- sample(c(1:n), replace = TRUE)
Data.boot <- Data[which(Data$Group == index[1]),]
for(boot in 2:n){
  Data.temp <- Data[which(Data$Group == index[boot]),]
  Data.boot <- rbind(Data.boot, Data.temp)
}
Data.boot <- Data.boot[order(Data.boot$Group),]
Data.boot$Group <- factor(Data.boot$Group)

######## Fit the candidate models with bootstrap data
tryCatch(
  {
  fit_1 <- suppressMessages(gee(y ~ x_1, id = Group,data = Data.boot, family = "binomial",
      corstr = "independence", scale.fix = TRUE))
  fit_2 <- suppressMessages(gee(y ~ x_1 + x_2, id = Group,data = Data.boot, family = "
      binomial", corstr = "independence", scale.fix = TRUE))
  fit_3 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3, id = Group,data = Data.boot, family = "
      binomial", corstr = "independence", scale.fix = TRUE))
  fit_4 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4, id = Group,data = Data.boot,
      family = "binomial", corstr = "independence", scale.fix = TRUE))
  fit_5 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5, id = Group,data = Data.boot,
       family = "binomial", corstr = "independence", scale.fix = TRUE))
  fit_6 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6, id = Group,data = Data
      .boot, family = "binomial", corstr = "independence", scale.fix = TRUE))
  fit_7 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7, id = Group,data
      = Data.boot, family = "binomial", corstr = "independence", scale.fix = TRUE))
  }, error=function(e){cat("ERROR :",conditionMessage(e), "\n")}
  )

########
int <- as.vector(rep(1, n * m))

######## First bootstrap variants
variants.b1[p, ] <- c(Quasi.b1.1(fit_1_ind, fit_1, Data), Quasi.b1.12(fit_2_ind, fit_2, Data)
    , Quasi.b1.13(fit_3_ind, fit_3, Data), Quasi.b1.14(fit_4_ind, fit_4, Data), Quasi.b1.15(
    fit_5_ind, fit_5, Data), Quasi.b1.16(fit_6_ind, fit_6, Data), Quasi.b1.17(fit_7_ind, fit_
    7, Data))

######## Second bootstrap variants
variants.b2[p, ] <- c(Quasi.b2.1(fit_1_ind, fit_1, Data), Quasi.b2.12(fit_2_ind, fit_2, Data)
    , Quasi.b2.13(fit_3_ind, fit_3, Data), Quasi.b2.14(fit_4_ind, fit_4, Data), Quasi.b2.15(
```

```
    fit_5_ind, fit_5, Data), Quasi.b2.16(fit_6_ind, fit_6, Data), Quasi.b2.17(fit_7_ind, fit_
    7, Data))
}

########## The Quasi-likelihood of the original data
y <- Data$y

Quasi.liklihood[pitao,] <- c(GetQuasi(fit_1_ori), GetQuasi(fit_2_ori), GetQuasi(fit_3_ori),
    GetQuasi(fit_4_ori), GetQuasi(fit_5_ori), GetQuasi(fit_6_ori), GetQuasi(fit_7_ori))

########## QAIC
QAIC[pitao,] <- -2 * Quasi.liklihood[pitao,] + 2*k
Freq.QAIC[which.min(QAIC[pitao,])] <- Freq.QAIC[which.min(QAIC[pitao,])] + 1

########## Self-defined QIC function

Penality.QIC[pitao,] <- c(GetQIC.P(fit_1_ori), GetQIC.P(fit_2_ori), GetQIC.P(fit_3_ori), GetQIC
    .P(fit_4_ori),
                          GetQIC.P(fit_5_ori), GetQIC.P(fit_6_ori), GetQIC.P(fit_7_ori))
QIC.New[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.QIC[pitao,]
Freq.QIC.New[which.min(QIC.New[pitao,])] <- Freq.QIC.New[which.min(QIC.New[pitao,])] + 1

########## First bootstrap-based QIC
Penality.b1[pitao,] <- apply(variants.b1, 2, mean)
QIC.b1[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.b1[pitao,]
Freq.QIC.b1[which.min(QIC.b1[pitao,])] <- Freq.QIC.b1[which.min(QIC.b1[pitao,])] + 1

########## Second bootstrap-based QIC
Penality.b2[pitao,] <- 2 * apply(variants.b2, 2, mean)
QIC.b2[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.b2[pitao,]
Freq.QIC.b2[which.min(QIC.b2[pitao,])] <- Freq.QIC.b2[which.min(QIC.b2[pitao,])] + 1

######
sink()

print(pitao)
print(Quasi.liklihood[pitao,])
print(rbind(2*k, Penality.QIC[pitao,], Penality.b1[pitao,], Penality.b2[pitao,]))
print(rbind(QIC.New[pitao,], QAIC[pitao,], QIC.b1[pitao,], QIC.b2[pitao,]))
print(rbind(Freq.QIC.New, Freq.QAIC, Freq.QIC.b1, Freq.QIC.b2))
```

```
}


### Functions for computing first variants of QAICb1 and QAICb2 (Poisson Model) (Overfitted
    Version)

Quasi.b1.1 <- function(fit.ori, fit.boot, data){
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,2]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2) - mu.fit.2) / phi.boot
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b1.12 <- function(fit.ori, fit.boot, data){
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:3)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2) - mu.fit.2) / phi.boot
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b1.13 <- function(fit.ori, fit.boot, data){
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:4)]))
```

```
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2) - mu.fit.2) / phi.boot
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b1.14 <- function(fit.ori, fit.boot, data){
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:5)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2) - mu.fit.2) / phi.boot
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.15 <- function(fit.ori, fit.boot, data){
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:6)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2) - mu.fit.2) / phi.boot
  ##### Output
```

```
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.16 <- function(fit.ori, fit.boot, data){
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:7)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2) - mu.fit.2) / phi.boot
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.17 <- function(fit.ori, fit.boot, data){
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:8)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2) - mu.fit.2) / phi.boot
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.18 <- function(fit.ori, fit.boot, data){
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:9)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
```

```
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2) - mu.fit.2) / phi.boot
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.19 <- function(fit.ori, fit.boot, data){
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:10)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2) - mu.fit.2) / phi.boot
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b1.110 <- function(fit.ori, fit.boot, data){
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:11)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  y.boot <- fit.boot$y
  mu.fit.2 <- fit.boot$fitted.values
  quasi.fit.2 <- sum(y.boot * log(mu.fit.2) - mu.fit.2) / phi.boot
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
```

```
### Functions for computing second variants of QAICb1 and QAICb2 (Poisson Model) (Overfitted
    Version)

Quasi.b2.1 <- function(fit.ori, fit.boot, data){
  phi.ori <- fit.ori$scale
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,2]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2) - mu.fit.2) / phi.ori
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b2.12 <- function(fit.ori, fit.boot, data){
  phi.ori <- fit.ori$scale
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:3)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2) - mu.fit.2) / phi.ori
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b2.13 <- function(fit.ori, fit.boot, data){
  phi.ori <- fit.ori$scale
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:4)]))
  beta.boot <- as.vector(fit.boot$coefficients)
```

```
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2) - mu.fit.2) / phi.ori
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
#####
Quasi.b2.14 <- function(fit.ori, fit.boot, data){
  phi.ori <- fit.ori$scale
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:5)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2) - mu.fit.2) / phi.ori
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.15 <- function(fit.ori, fit.boot, data){
  phi.ori <- fit.ori$scale
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:6)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2) - mu.fit.2) / phi.ori
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
```

```
####
Quasi.b2.16 <- function(fit.ori, fit.boot, data){
  phi.ori <- fit.ori$scale
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:7)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2) - mu.fit.2) / phi.ori
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.17 <- function(fit.ori, fit.boot, data){
  phi.ori <- fit.ori$scale
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:8)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2) - mu.fit.2) / phi.ori
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.18 <- function(fit.ori, fit.boot, data){
  phi.ori <- fit.ori$scale
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:9)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
```

```
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2) - mu.fit.2) / phi.ori
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.19 <- function(fit.ori, fit.boot, data){
  phi.ori <- fit.ori$scale
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:10)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2) - mu.fit.2) / phi.ori
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
####
Quasi.b2.110 <- function(fit.ori, fit.boot, data){
  phi.ori <- fit.ori$scale
  phi.boot <- fit.boot$scale
  ##### First Part
  X <- as.matrix(cbind(int, data[,c(2:11)]))
  beta.boot <- as.vector(fit.boot$coefficients)
  y.ori <- fit.ori$y
  mu.fit.1 <- exp(X %*% beta.boot)
  quasi.fit.1 <- sum(y.ori * log(mu.fit.1) - mu.fit.1) / phi.boot
  ##### Second Part
  mu.fit.2 <- fit.ori$fitted.values
  quasi.fit.2 <- sum(y.ori * log(mu.fit.2) - mu.fit.2) / phi.ori
  ##### Output
  -2 * (quasi.fit.1 - quasi.fit.2)
}
```

```
###### The true correlation structure
myCop <- normalCopula(0.2, dim = 3, dispstr = "ex")

###### The parameters
beta_0 <- 0.5
beta_1 <- -0.5
beta_2 <- 0.25
beta_3 <- -0.25
beta_4 <- 0.25

##### Function of computing Quasi-Likelihood
GetQuasi <- function(fitted_model){
  phi <- fitted_model$scale
  y <- fitted_model$y
  mu <- fitted_model$fitted.values
  sum(y * log(mu) - mu) / phi
}

###### Number of iterations
itr <- 1000

#### Start of the iteration
for(pitao in 1:itr)
{
  ########
  sink("file")

  ###### Generation of Data
  x_1 <- rnorm(n * m, 0, 1)
  x_2 <- rnorm(n * m, 0, 1)
  x_3 <- rnorm(n * m, 0, 1)
  x_4 <- rnorm(n * m, 0, 1)
  x_5 <- rnorm(n * m, 0, 1)
  x_6 <- rnorm(n * m, 0, 1)
  x_7 <- rnorm(n * m, 0, 1)
  x_8 <- rnorm(n * m, 0, 1)
  x_9 <- rnorm(n * m, 0, 1)
  x_10 <- rnorm(n * m, 0, 1)

  mu <- exp(beta_0 + beta_1 * x_1 + beta_2 * x_2 + beta_3 * x_3 + beta_4 * x_4)
```

```
y <- matrix(rep(0, n * m), ncol = 3)
for (i in 1 : n){
  mymvd <- mvdc(copula=myCop, margins=c("pois", "pois", "pois"),
                paramMargins=list(list(lambda = mu[3*i-2]),
                                  list(lambda = mu[3*i-1]),
                                  list(lambda = mu[3*i])))
  y[i,] <- rMvdc(1, mymvd)
}
y <- as.vector(t(y))

Data <- data.frame(cbind(Group, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_10, y))

#### The original QIC values of all the candidate models
fit_1_ori <- suppressMessages(gee(y ~ x_1, id = Group,data = Data, family = "poisson", corstr =
    "exchangeable"))
fit_2_ori <- suppressMessages(gee(y ~ x_1 + x_2, id = Group,data = Data, family = "poisson",
    corstr = "exchangeable"))
fit_3_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3, id = Group,data = Data, family = "
    poisson", corstr = "exchangeable"))
fit_4_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4, id = Group,data = Data, family = "
    poisson", corstr = "exchangeable"))
fit_5_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5, id = Group,data = Data,
    family = "poisson", corstr = "exchangeable"))
fit_6_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6, id = Group,data = Data
    , family = "poisson", corstr = "exchangeable"))
fit_7_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7, id = Group,data
    = Data, family = "poisson", corstr = "exchangeable"))
fit_8_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8, id = Group
    ,data = Data, family = "poisson", corstr = "exchangeable"))
fit_9_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9, id =
    Group,data = Data, family = "poisson", corstr = "exchangeable"))
fit_10_ori <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_
    10, id = Group,data = Data, family = "poisson", corstr = "exchangeable"))


QIC.ori[pitao,] <- c(QIC(fit_1_ori), QIC(fit_2_ori), QIC(fit_3_ori), QIC(fit_4_ori),
                     QIC(fit_5_ori), QIC(fit_6_ori), QIC(fit_7_ori), QIC(fit_8_ori),
                     QIC(fit_9_ori), QIC(fit_10_ori))

Freq.QIC[which.min(QIC.ori[pitao,])] <- Freq.QIC[which.min(QIC.ori[pitao,])] + 1
```

```
#### Independent Update of the original model

fit_1_ind <- suppressMessages(update(fit_1_ori, corstr = "independence"))
fit_2_ind <- suppressMessages(update(fit_2_ori, corstr = "independence"))
fit_3_ind <- suppressMessages(update(fit_3_ori, corstr = "independence"))
fit_4_ind <- suppressMessages(update(fit_4_ori, corstr = "independence"))
fit_5_ind <- suppressMessages(update(fit_5_ori, corstr = "independence"))
fit_6_ind <- suppressMessages(update(fit_6_ori, corstr = "independence"))
fit_7_ind <- suppressMessages(update(fit_7_ori, corstr = "independence"))
fit_8_ind <- suppressMessages(update(fit_8_ori, corstr = "independence"))
fit_9_ind <- suppressMessages(update(fit_9_ori, corstr = "independence"))
fit_10_ind <- suppressMessages(update(fit_10_ori, corstr = "independence"))

#### Number of bootstrap samples
B <- 250

#### Bootstrap iteration to find weighted QIC
variants.b1 <- matrix(rep(0, 10*B), nrow = B)
variants.b2 <- matrix(rep(0, 10*B), nrow = B)

for(p in 1:B){
  index <- sample(c(1:n), replace = TRUE)
  Data.boot <- Data[which(Data$Group == index[1]),]
  for(boot in 2:n){
    Data.temp <- Data[which(Data$Group == index[boot]),]
    Data.boot <- rbind(Data.boot, Data.temp)
  }
  Data.boot <- Data.boot[order(Data.boot$Group),]
  Data.boot$Group <- factor(Data.boot$Group)

  ######## Fit the candidate models with bootstrap data

  fit_1 <- suppressMessages(gee(y ~ x_1, id = Group,data = Data.boot, family = "poisson",
      corstr = "independence"))
  fit_2 <- suppressMessages(gee(y ~ x_1 + x_2, id = Group,data = Data.boot, family = "poisson",
       corstr = "independence"))
  fit_3 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3, id = Group,data = Data.boot, family = "
      poisson", corstr = "independence"))
  fit_4 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4, id = Group,data = Data.boot, family
      = "poisson", corstr = "independence"))
  fit_5 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5, id = Group,data = Data.boot,
```

```
    family = "poisson", corstr = "independence"))
  fit_6 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6, id = Group,data = Data.
    boot, family = "poisson", corstr = "independence"))
  fit_7 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7, id = Group,data =
    Data.boot, family = "poisson", corstr = "independence"))
  fit_8 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8, id = Group,
    data = Data.boot, family = "poisson", corstr = "independence"))
  fit_9 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9, id =
    Group,data = Data.boot, family = "poisson", corstr = "independence"))
  fit_10 <- suppressMessages(gee(y ~ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_
    10, id = Group,data = Data.boot, family = "poisson", corstr = "independence"))

  int <- as.vector(rep(1, n * m))

  ######## First bootstrap variants
  variants.b1[p, ] <- c(Quasi.b1.1(fit_1_ind, fit_1, Data), Quasi.b1.12(fit_2_ind, fit_2, Data)
    , Quasi.b1.13(fit_3_ind, fit_3, Data), Quasi.b1.14(fit_4_ind, fit_4, Data), Quasi.b1.15(
    fit_5_ind, fit_5, Data), Quasi.b1.16(fit_6_ind, fit_6, Data), Quasi.b1.17(fit_7_ind, fit_
    7, Data), Quasi.b1.18(fit_8_ind, fit_8, Data), Quasi.b1.19(fit_9_ind, fit_9, Data), Quasi
    .b1.110(fit_10_ind, fit_10, Data))



  ######## Second bootstrap variants
  variants.b2[p, ] <- c(Quasi.b2.1(fit_1_ind, fit_1, Data), Quasi.b2.12(fit_2_ind, fit_2, Data)
    , Quasi.b2.13(fit_3_ind, fit_3, Data), Quasi.b2.14(fit_4_ind, fit_4, Data), Quasi.b2.15(
    fit_5_ind, fit_5, Data), Quasi.b2.16(fit_6_ind, fit_6, Data), Quasi.b2.17(fit_7_ind, fit_
    7, Data), Quasi.b2.18(fit_8_ind, fit_8, Data),
  Quasi.b2.19(fit_9_ind, fit_9, Data), Quasi.b2.110(fit_10_ind, fit_10, Data))
}

########## The Quasi-likelihood of the original data
y <- Data$y
Quasi.liklihood[pitao,] <- c(GetQuasi(fit_1_ori), GetQuasi(fit_2_ori), GetQuasi(fit_3_ori),
    GetQuasi(fit_4_ori), GetQuasi(fit_5_ori), GetQuasi(fit_6_ori), GetQuasi(fit_7_ori),
    GetQuasi(fit_8_ori), GetQuasi(fit_9_ori), GetQuasi(fit_10_ori))

########## QAIC
QAIC[pitao,] <- -2 * Quasi.liklihood[pitao,] + 2*k
Freq.QAIC[which.min(QAIC[pitao,])] <- Freq.QAIC[which.min(QAIC[pitao,])] + 1

########## Self-defined QIC function
```

```
  Penality.QIC[pitao,] <- c(GetQIC.P(fit_1_ori), GetQIC.P(fit_2_ori), GetQIC.P(fit_3_ori), GetQIC
      .P(fit_4_ori), GetQIC.P(fit_5_ori), GetQIC.P(fit_6_ori), GetQIC.P(fit_7_ori), GetQIC.P(fit_
      8_ori), GetQIC.P(fit_9_ori), GetQIC.P(fit_10_ori))

  QIC.New[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.QIC[pitao,]
  Freq.QIC.New[which.min(QIC.New[pitao,])] <- Freq.QIC.New[which.min(QIC.New[pitao,])] + 1

  ########## First bootstrap-based QIC
  Penality.b1[pitao,] <- apply(variants.b1, 2, mean)
  QIC.b1[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.b1[pitao,]
  Freq.QIC.b1[which.min(QIC.b1[pitao,])] <- Freq.QIC.b1[which.min(QIC.b1[pitao,])] + 1

  ########## Second bootstrap-based QIC
  Penality.b2[pitao,] <- 2 * apply(variants.b2, 2, mean)
  QIC.b2[pitao,] <- -2 * Quasi.liklihood[pitao,] + Penality.b2[pitao,]
  Freq.QIC.b2[which.min(QIC.b2[pitao,])] <- Freq.QIC.b2[which.min(QIC.b2[pitao,])] + 1

  ######
  sink()

  print(pitao)
  print(Quasi.liklihood[pitao,])
  print(rbind(2*k, Penality.QIC[pitao,], Penality.b1[pitao,], Penality.b2[pitao,]))
  print(rbind(QIC.New[pitao,], QAIC[pitao,], QIC.b1[pitao,], QIC.b2[pitao,]))
  print(rbind(Freq.QIC, Freq.QIC.New, Freq.QAIC, Freq.QIC.b1, Freq.QIC.b2))
}
```