1995

Monte Carlo Methods for Confidence Bands in Nonlinear Regression

Shantonu Mazumdar
University of North Florida

Monte Carlo Methods For Confidence Bands in Nonlinear Regression

by

Shantonu Mazumdar

A thesis submitted to the Department of Mathematics and Statistics
in partial fulfillment of the requirements for the degree of

Master of Science in Mathematical Sciences

UNIVERSITY OF NORTH FLORIDA

COLLEGE OF ARTS AND SCIENCES

May 1995

Certificate of Approval

The thesis of Shantonu Mazumdar is approved:

Signature deleted | (Date) 5/1/95

Committee Chairperson

Signature deleted | 5/1/95

Signature deleted | 5-1-95

Accepted for the Department:

Signature deleted | 5/1/95

Chairperson

Accepted for the College:

Signature deleted | 5-3-95

Dean

Accepted for the University:

Signature deleted | 5-3-95

Dean of Graduate Studies

Acknowledgment

My sincere gratitude goes to Dr. Donna Mohr for her constant encouragement throughout my graduate program, academic assistance and concern for my graduate success. The guidance, support and knowledge that I received from Dr. Mohr while working on this thesis is greatly appreciated.

My gratitude is extended to the members of my reviewing committee, Dr. Ping Sa and Dr. Peter Wludyka for their invaluable input into the writing of this thesis.

I would like to thank all the faculty and staff that I have been associated with at the Department of Mathematics and Statistics at the University of North Florida for all their help during the two very productive years that I spent here as a Master's candidate and Graduate Teaching Assistant. Special thanks to my teachers, Dr. Wahab, Dr. Wilson and Dr. Sen, the Graduate Directors, Dr. Panchal and Dr. Boules and the Chairperson, Dr. Caldwell.

I would like to thank my friends and colleagues, Peter Davis, Kurt Lohman, Cynthia Wallace, Felicia Kennedy, Linda Moore and the other teaching assistants in the department for their intellectual support and comradeship. I will always treasure their friendship.

I would like to thank my friends, Robin and Shampa Mukherjee for their support and guidance throughout my Master's program and for keeping me in focus during the past couple of years.

Finally, and by no means least, I would like to thank my parents, Drs. Subir and Sandhya Mazumdar for their love and support throughout my academic career and instilling in me a desire to learn and broaden my intellectual horizons.

Contents

Abstract

Confidence Bands for Nonlinear Regression Functions can be found analytically for a very limited range of functions with a restrictive parameter space. A computer intensive technique, the Monte Carlo Method will be used to develop an algorithm to find confidence bands for any given nonlinear regression functions with a broader parameter space.

The logistic regression function with one independent variable and two parameters will be used to test the validity and efficiency of the algorithm. The confidence bands for this particular function have been solved for analytically by Khorasani and Milliken (1982). Their derivations will be used to test the Monte Carlo algorithm.

Chapter 1 - Introduction

A substantial level of statistical research has been undertaken on finding point estimates for parameters of regression functions and confidence regions for these parameters. However, there does not seem to exist much literature on the development of confidence bands about the nonlinear regression function. We will consider the problem of constructing confidence bands for nonlinear regression models.

For a given non-linear regression function :

$$y = f\,(x; \underline{\theta}) + \epsilon \tag{1}$$

where $\underline{\theta}$ is a $r \times 1$ vector of parameters, the **Maximum Likelihood Estimator** (M.L.E.) of $\underline{\theta}$ is $\hat{\underline{\theta}}$ which is obtained by maximizing the likelihood function. The maximum likelihood estimator, $\hat{\underline{\theta}}$, may be found explicitly or for those functions where it is not possible to do so, a computer intensive technique may be employed. Thus the Maximum Likelihood function is the nonlinear regression function defined in (1) evaluated at $\hat{\theta}$. i.e.

$$E(y, \hat{\underline{\theta}}\,) = f\,(x; \hat{\theta}\,) \tag{2}$$

A **$(1-\alpha)\times 100$ % confidence band** for a nonlinear regression function is defined to be two functions, $y_L(x)$ and $y_U(x)$ between which we are $(1-\alpha)\times 100$ % confident that the true function will completely lie as illustrated in Fig 1.

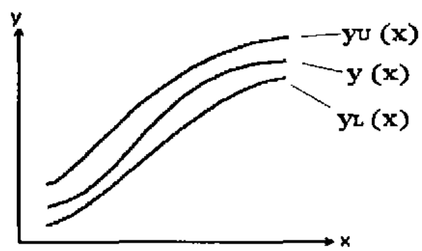


Fig.1 A non-linear regression function with upper and lower confidence bands

Our goal is thus to find the two functions $y_L(x)$ and $y_U(x)$ for any given non-linear regression function. When it is not possible to do so analytically, we will try to do so by utilizing computer intensive techniques.

Statistical Background

We will assume that the estimator $\hat{\underline{\theta}}$ is asymptotically distributed as a multivariate normal random variable with mean $\underline{\theta}^t$ and covariance matrix V, where $\underline{\theta}^t$ is the true value of $\underline{\theta}$. (**Myers and Milton**, 1991).

Let us define :

$$L(\underline{\theta}) = \text{Log likelihood function of } \underline{\theta} = \ln f(x_1, x_2, \ldots, x_n ; \underline{\theta})$$

and we will define

$$\lambda(\underline{\theta}_1, \underline{\theta}_2) = -2\,(L(\underline{\theta}_1) - L(\underline{\theta}_2)). \qquad (3)$$

An asymptotic result we may use is that $\lambda(\hat{\underline{\theta}}, \underline{\theta}^t)$ is distributed as a χ^2 random variable with r degrees of freedom (**Neter, Wasserman and Kutner**, 1990). i.e. $\lambda(\hat{\underline{\theta}}, \underline{\theta}^t) \sim \chi^2(r)$. ($\hat{\underline{\theta}}$ is the Maximum Likelihood Estimator for $\underline{\theta}$).

Confidence region for $\underline{\theta}$

If all the components of $\hat{\underline{\theta}}$ ($\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, \ldots\ldots, \hat{\theta}_r$) were statistically independent of each other, then the confidence region for $\underline{\theta}$ would assume the shape of a ellipsoid centered at $\hat{\underline{\theta}}$ with axes parallel to the coordinate axes. A frequently used geometric figure used to approximate this confidence region is a rectangular box, centered at $\hat{\underline{\theta}}$, with sides proportional to the standard errors of the respective $\hat{\theta}_i$'s (Fig 2) . If independence between the $\underline{\theta}$'s is assumed, a confidence region of probability $(1-\alpha)$ can be obtained by producing r confidence intervals, one for each parameter, with individual confidence level $\sqrt[r]{1-\alpha}$.

However, since independence between the $\underline{\theta}$'s cannot and should not be assumed, the true confidence region for $\underline{\theta}$ takes on an approximate ellipsoid form, encompassed almost within this parallelogram as shown in Fig 3. To find the confidence region for $\underline{\theta}^t$, we may use all the $\underline{\theta}$'s with ln likelihoods close to that of $\hat{\underline{\theta}}$. We can thus define the confidence region for $\underline{\theta}^t$ as :

$$\{ \text{All } \underline{\theta} \text{ such that } \lambda(\hat{\underline{\theta}} , \underline{\theta}) < \chi^2_{(1-\alpha)} (r) \} \qquad (4)$$

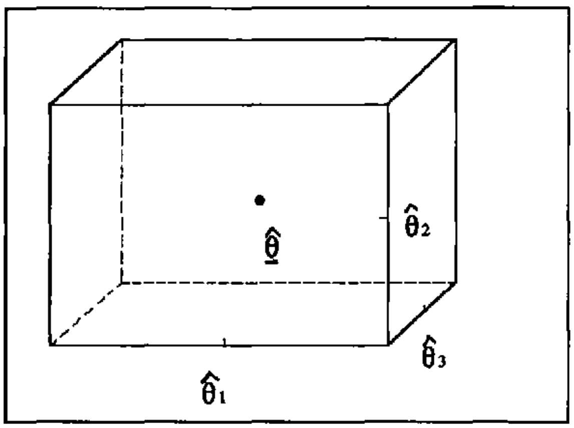


Figure 2
The rectangular box used to approximate confidence region for $\underline{\theta}$

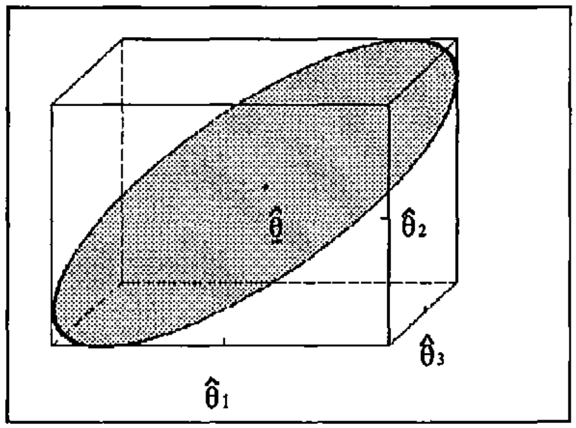


Figure 3
The true confidence region for $\underline{\theta}$, the ellipsoid

Another approach in defining the confidence region for $\underline{\theta}$ involves the covariance matrix V. This approach was taken by **Khorasani and Milliken** (1982) when they found confidence bands for a logistic regression function with two parameters. This method states that asymptotically $(\underline{\theta} - \hat{\underline{\theta}})^T V^{-1} (\underline{\theta} - \hat{\underline{\theta}})$ is distributed as an approximate χ^2 with r degrees of freedom. The corresponding confidence region for $\underline{\theta}^t$ would therefore be defined to be:

$$\{ \text{All } \underline{\theta} \text{ such that } (\underline{\theta} - \hat{\underline{\theta}})^T V^{-1} (\underline{\theta} - \hat{\underline{\theta}}) \leq \chi^2_{(1-\alpha)} (r) \} \qquad (5)$$

The shape of this confidence region takes on an exact ellipsoid form and can be used to find the confidence bands for a nonlinear function.

Confidence Bands for the Non-Linear Regression Function

Both the approaches introduced above may be used to translate confidence regions for $\underline{\theta}$ to confidence bands for the function which has $\underline{\theta}$ as its parameters.

The upper and lower functions that define the confidence bands for f $(x;\underline{\theta})$, ($y_U(x)$ and $y_L(x)$) respectively, can be found by using the definitions :

$$y_U(x) = \max \{ f(x;\underline{\theta}) ; \underline{\theta} \in CR \} \text{ and } y_L(x) = \min \{ f(x;\underline{\theta}) ; \underline{\theta} \in CR \} \qquad (6)$$

where CR is the confidence region for $\underline{\theta}^t$ and we choose various x's from the domain. We then evaluate the maximum and minimum values of the function at each x_i in this domain using the definitions given above. This would give us the upper and lower confidence bands for the function.

The analytic approach was employed by Khorasani and Milliken (1982) in finding confidence bands for two specific non-linear regression functions (the logistic regression function and the Michaelis-Menten Kinetic model, both with two parameters). This method employed the confidence region based on equation (5) and took advantage of the fact that the shape was that of an ellipsoid.

We can therefore use the confidence region for $\underline{\theta}$, as defined by the boundaries of the ellipsoid, to find a confidence band for the regression function. The bands would be defined by the value of the function evaluated at different $\underline{\theta}$'s chosen from the edges of the ellipsoid.

To generalize an analytical approach to solve for confidence bands of other non-linear regression function is not a trivial task. An increase in size of the parameter space or an introduction of complicated models may prove to exceed the limitations of a mathematical approach.

The **Monte-Carlo Method** we will employ will allow us to expand this range to many more non-linear functions with a much broader parameter space than what can be achieved theoretically.

Our method for finding the confidence band will rely on the asymptotic distribution of $\lambda(\hat{\underline{\theta}}, \underline{\theta})$. We will choose $\underline{\theta}$'s from a neighborhood around $\hat{\underline{\theta}}$ and use those $\underline{\theta}$'s that fall inside the confidence region for $\underline{\theta}^{t}$, i.e. pass the log-likelihood test.

An application - The logistic regression function.

The logistic regression function is used to model the probability of the occurrence of an event under conditions described by a variable.

The logistic regression function for two parameters is therefore defined to be:

$$f(x;\theta_1,\theta_2) = Prob\ (event\ \ occurs) = Prob\ (Y=1\,|\,X=x) = \frac{e^{\theta_1+\theta_2 x}}{1+e^{\theta_1+\theta_2 x}} \tag{7}$$

where Y is 1 if the event occurs and 0 if it does not (**Agresti**, 1990).

Khorasani and Milliken (1982) utilized analytical techniques to find confidence regions for θ_1 and θ_2 and hence confidence bands for the above function. Details of their calculations are attached in Appendix A. They employed the ellipsoid mentioned previously. They chose different points on the ellipsoid which gave them maximum and minimum values for f (x ; $\underline{\theta}$) using the following equations :

$$\theta_2 = \hat{\theta}_2 \pm \sqrt{\frac{\chi^2}{I_{11}K^2(x) - 2I_{12}K(x) + I_{22}}}$$

$$\theta_1 = \hat{\theta}_1 - K(x)(\theta_2 - \hat{\theta}_2) \tag{8}$$

$$where\ \ I = Information\ \ matrix = \begin{vmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{vmatrix}$$

$$and\ \ \ K(x) = \frac{I_{22} - x\, I_{12}}{I_{12} - x\, I_{11}}$$

Thus at each value of x in the domain, we evaluate two new values of θ_2 and corresponding values of θ_1. This gives us points on the boundary of the ellipsoid at which we evaluate the logistic regression function to obtain maximum and minimum values of the function at that x. When we plot these points for various x's, we will obtain two curves which will define the upper and lower confidence bands for the specified logistic function. We will attempt to use these calculations to verify our Monte Carlo results.

The limitations of analytical methods can be seen in the fact that for any logistic regression function with more than two parameters, the evaluation of equation (6) becomes quite complicated. The calculations for confidence bands for nonlinear regression functions other than the one specified above do not seem to have been explored extensively, presumably because of the broad nature of the forms that these functions can take. Our methods will attempt to generalize the results so that we may be able to find reasonable confidence bands for any specified function with a reasonable number of parameters.

The Monte-Carlo Method for finding confidence bands for non-linear regression functions will generate parameter estimates inside the confidence region of $\underline{\theta}'$ that can be used to approximate maximum and minimum values for the function at various values in the domain of X.

Chapter 2 - Monte Carlo Technique for Confidence Bands

The question of how to translate the confidence region for $\underline{\theta}^t$ to confidence bands for the nonlinear regression function (as defined in equation 6) is what we are attempting to answer. Analytical approaches have been suggested for very specific functions. We will employ a computer intensive method, the **Monte Carlo Technique**, to solve this problem for the general class of non-linear regression functions.

Let us summarize the steps in the Monte-Carlo Method for finding confidence bands for non-linear regression functions. The detailed steps will be discussed later in this chapter.

1. Find a Maximum Likelihood estimate for the parameters $\underline{\theta}$.
2. Choose $\underline{\theta}^*$ randomly from a region around $\hat{\theta}$.($\hat{\theta}$ is the MLE for $\underline{\theta}$.). We will employ a multivariate uniform distribution and a multinormal distribution to choose the $\underline{\theta}^*$'s.
3. Check to see if this particular $\underline{\theta}^*$ passes the log likelihood test. Proceed to next step if test is passed; return to step 2 if $\underline{\theta}^*$ is not in the confidence region of $\underline{\theta}^t$.

4. Evaluate the function at each value of x in the domain. If the function exceeds previous largest value at that particular x, designate this function value to be the new upper value. Do a similar test for the lower function value for each x.

5. Return to step 2 to choose another $\underline{\theta}^*$ using the same distribution employed above. Repeat the loop a large number of times to obtain enough $\underline{\theta}^*$'s from the confidence region.

6. Tabulate the maximum and minimum function values for the x's chosen in the domain. These will define the upper and lower confidence bands.

A flow chart describing the algorithm is shown in Fig. 4.

The logistic regression function

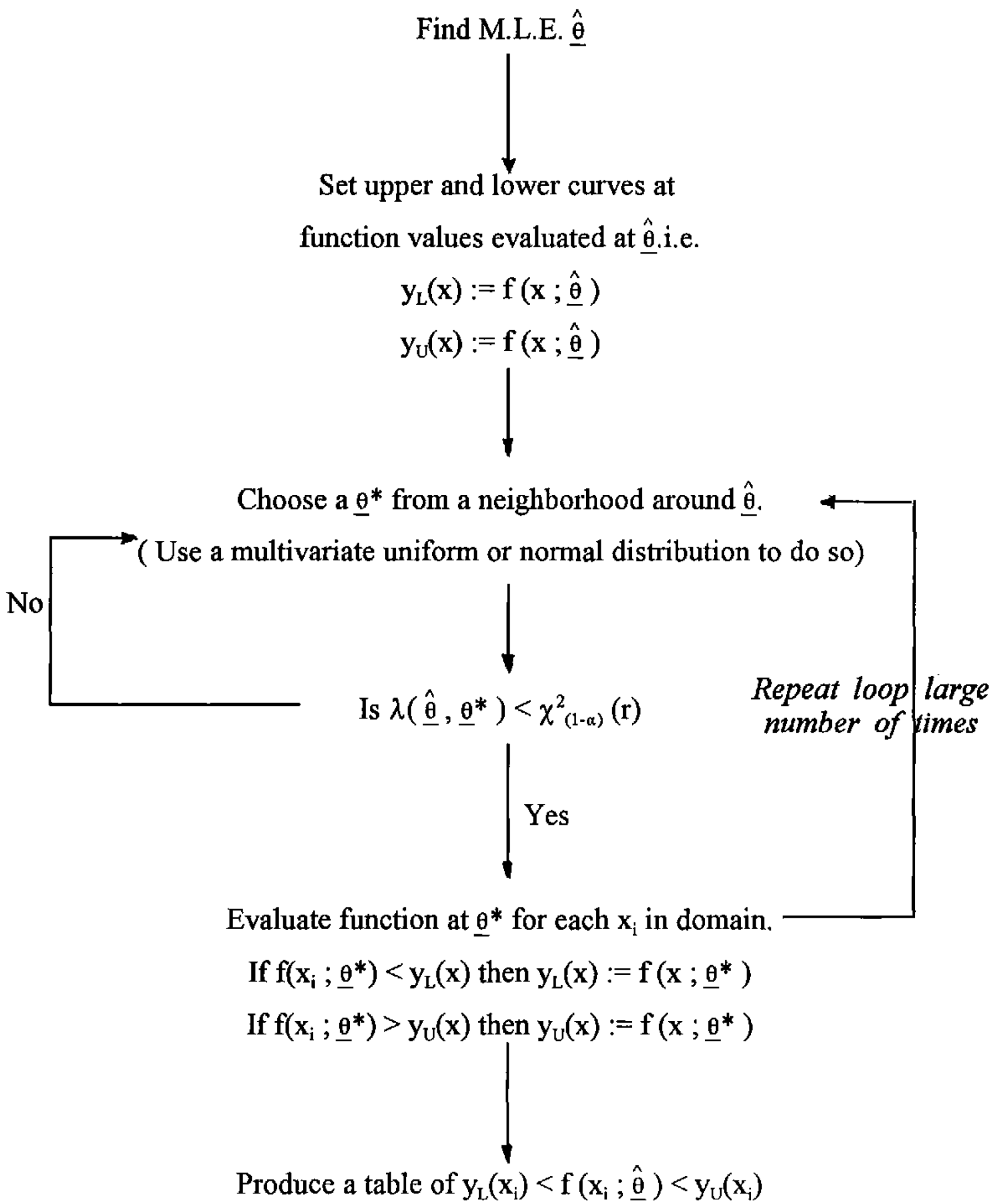
To develop confidence bands for non-linear regression functions, we will concentrate on the logistic regression function with two parameters as defined in Chapter 1, i.e.

$$P(Y=1\,|\,X=x) = f(x;\theta_1,\theta_2) = \frac{e^{\theta_1+\theta_2 x}}{1+e^{\theta_1+\theta_2 x}} \tag{9}$$

Y is a categorical variable which takes on a value of 1 if an event occurs and 0 otherwise, and X is a variable thought to affect Y.

Figure 4

Monte Carlo Technique for finding Confidence Bands



Since Khorasani and Milliken solved analytically to find confidence bands for the logistic regression function with two parameters, we will be able to compare our results to their theoretical conclusions. We should note, however, that our techniques should generalize for any given non-linear regression function with a given parameter space.

The first step in our procedure involves finding point estimates for our parameters. We will find the Maximum Likelihood Estimates (M.L.E.) for θ_1 and θ_2. This is done by maximizing the likelihood function :

$$f(y_1, y_2, \ldots, y_n; x_1, x_2, \ldots, x_n, \underline{\theta}) = \frac{\prod_{y_i = 1} e^{\theta_1 + \theta_2 x}}{\prod_{all\ i} (1 + e^{\theta_1 + \theta_2 x})} \tag{10}$$

To find the optimal values for θ_1 and θ_2 we need to differentiate the log likelihood function with respect to each parameter. It is not possible, however, to find closed form expressions for $\hat{\theta}_1$ and $\hat{\theta}_2$ when we attempt to solve for these maximum likelihood estimates. We thus have to find an alternative technique to find these estimated parameter values for a given data set and this is where a computer intensive method comes in useful.

We will find the values of θ_1 and θ_2 for which the log likelihood function attains the maximum in 3 dimensional space.

We are restricted by the routines available to us to finding the value of a set of parameters for which a function achieves its minimum. We will thus try to **minimize the negative log likelihood function**, which accomplishes the same goal. We will use the **gradient method** to find this minimum, that is, we will use the method which utilizes the gradient of the function (found by differentiating the function with respect to the two parameters) to search for the minimum value of the negative log likelihood function.

The Gradient Method for finding the M.L.E. for $\underline{\theta}$

The Gradient Method that we will employ to find the M.L.E for $\underline{\theta}$ is the **Method of Steepest Descent**. We use the fact that steepest slope to a function occurs when the gradient is perpendicular to the tangent of the function at any point. Thus to get to the minimum of a function, we may follow this slope to the least value of the function in the domain. Details of this method follow in the next section.

The Search for the M.L.E.

- We may note that rewriting our logistic regression function as:

$$\ln \left[\frac{f(x)}{1-f(x)} \right] = \theta_1 + \theta_2 x \tag{11}$$

we obtain a linear relationship between the left hand side and the independent variable x, and thus we may find a reasonable guess for the value of θ_1 and θ_2 by looking at the slope and intercept of this linear relationship given the data. This can be used as the starting point in our search for $\hat{\theta}_1$ and $\hat{\theta}_2$.

- With our starting values, we use the routine **Linmin** (Press et al) to search along the gradient of the function that we seek to minimize at these starting values. If we make note of the fact that the steepest slope at a certain point is defined by its gradient at this point, an efficient method of finding a path to the minimum for this function would be to follow the steepest slope. Thus the direction vector that we will use for the next step of the search is dictated to us by the gradient of the function at this point.
- When we reach the minimum value of the function along this gradient vector (which is found by Linmin), we may change directions by calculating the gradient of the function at that point and entering in a scaled value of the gradient as our new direction vector.

- We repeat the above step, taking turns switching directions according to our gradients until we find a value of $[\hat{\theta}_1, \hat{\theta}_2]$ which has not changed significantly from the last $[\hat{\theta}_1, \hat{\theta}_2]$ by finding the norm of the vectors and comparing it to a predefined value. This value is defined to be our tolerance and we end the search when this tolerance is reached or when we achieve a predetermined number of maximum iterations.
- The value of $[\hat{\theta}_1, \hat{\theta}_2]$ we obtain finally is our best estimate of the true parameter value which will maximize the likelihood function.

Standard error of the estimates

We will need to find the standard error of the estimates that we found above since we will require these to define a confidence region for $\underline{\theta}$.

The standard error can be obtained from the asymptotic covariance matrix V, which is defined to be $[E[-H]]^{-1}$ (**Press et al**, 1989). H is the **Hessian matrix** (i.e. the second partial derivative of the log-likelihood function with respect to each parameter in the main diagonal, and with respect to each other in the off diagonal). The Hessian matrix is presented in Appendix B.

The asymptotic covariance matrix can be found by inverting the expected value of the negative Hessian matrix. The variances of $\hat{\theta}_1$ and $\hat{\theta}_2$ will be found in the main diagonal and the covariance $[\hat{\theta}_1 , \hat{\theta}_2]$ will be in the off-diagonal.

Finding the Confidence Bands for a non-linear regression function

Once we have found the M.L.E for $\underline{\theta}$ and standard error for the estimates, a confidence band for the function can be constructed. We will use the asymptotic distribution of $\lambda(\hat{\underline{\theta}} , \underline{\theta}^{t})$ to find the confidence region for $\underline{\theta}^{t}$.

A detailed account of the steps we will follow to find the confidence bands for the function are presented below:

1) Find the Maximum Likelihood Estimator for $\underline{\theta}$.

As explained in the previous section, we will find the M.L.E. for $\underline{\theta}$ by the method of steepest descent. This will give us a point estimate $\hat{\underline{\theta}}$ upon which we can center our confidence region for $\underline{\theta}$.

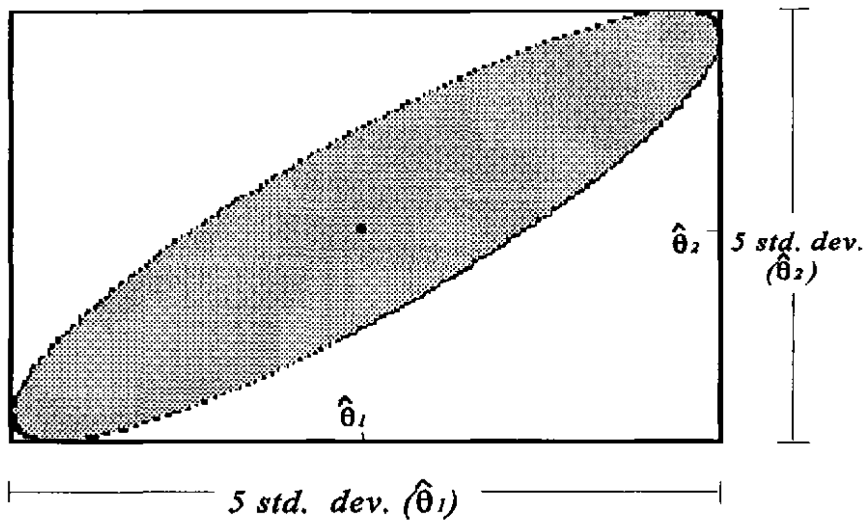


Figure 5
The rectangular box that will be used to select the $\underline{\theta}$*'s

2) Choose a $\underline{\theta}$* from the neighborhood around $\underline{\hat{\theta}}$.

We may randomly choose the $\underline{\theta}$*'s by using two well known distributions. The first that we shall explore is the bivariate uniform and the other will be the bivariate normal distribution.

The Uniform

The $\underline{\theta}$*'s will be chosen from a bivariate uniform distribution as shown in Fig.5 above. The ellipse which is shown will separate the $\underline{\theta}$*'s that will pass the log-likelihood test from those that will not. The $\underline{\theta}$*'s that will pass the test will be those that fall inside the ellipse. The length and width of the box will be about 5 standard deviations of $\hat{\theta}_1$ and $\hat{\theta}_2$ respectively. This will allow us to choose a very large percentage of all possible $\underline{\theta}$*'s.

The disadvantage of using a rectangular box is that possibly a relatively small percentage of all the $\underline{\theta}$*'s will actually pass the log-likelihood test since we will be getting many $\underline{\theta}$*'s far from the center, $[\hat{\theta}_1, \hat{\theta}_2]$. Thus we may have to choose a very large number of $\underline{\theta}$*'s to give us reasonable results when finding the confidence bands. On the other hand, we can intuitively say that the $\underline{\theta}$*'s that will play a leading role in defining the upper and lower confidence bands of the functions will probably be on the outer edges of the ellipse. We may therefore have more chances of catching these $\underline{\theta}$*'s when we use the rectangular box.

<u>The Multi-Normal</u>

The $\underline{\theta}$*'s will be chosen from a bivariate normal distribution centered at $\underline{\hat{\theta}}$ with variance and covariance defined by the asymptotic covariance matrix V that we obtained when we found the M.L.E. By the inherent shape of the bivariate normal distribution, we can see that many more $\underline{\theta}$*'s will be chosen closer to $\underline{\hat{\theta}}$ and thus will pass the log-likelihood test more often. The disadvantage would be that because of the proximity of the $\underline{\theta}$*'s to $\underline{\hat{\theta}}$, the outer edges of the ellipse may not be reached as often as it would be for the rectangular box.

Since both distributions have their positive and negative points, we will use both and see which will give us the better results.

3) Checking to see if the $\underline{\theta}$*'s pass the log-likelihood test.

We will calculate $\lambda(\hat{\underline{\theta}}, \underline{\theta}^*) = -2\,(\,L(\hat{\underline{\theta}}) - L(\underline{\theta}^*)\,)$. If this value is less than the χ^2 value for 2 d.f. at a prescribed α-level. (5.99 if $\alpha = 0.05$), we will proceed to the next step. If the test is not passed, we return to step 2 to choose another $\underline{\theta}^*$ by the method of choice.

4) Setting the confidence bands

For each x_i, we will evaluate the function with the $\underline{\theta}^*$ that passed the test. At each x_i, if the function value, $f(\,x_i, \underline{\theta}^*)$, exceeds the previous maximum upper function value, then this new value will become the new upper value $y_U\,(x_i)$. Similarly, if the function value evaluated at $\underline{\theta}^*$ is smaller than the previous minimum lower value, then we will assign this new value the label of lower function value, $y_L(x_i)$. Obviously, only one of the two assignments may be made on any one $\underline{\theta}^*$. On the first run through, the previous upper and lower function value is the function evaluated at $\hat{\underline{\theta}}$ as specified in step 1. Therefore, for each $\underline{\theta}^*$ that passes through this step, we can potentially broaden the confidence bands.

5) The next $\underline{\theta}$*.

We return to step 2 to choose another $\underline{\theta}^*$ and go through steps 3 and 4 with this $\underline{\theta}^*$. We will continue this loop for a large number of $\underline{\theta}^*$'s so that we may give the outer edges of the ellipse a chance of having a $\underline{\theta}^*$ picked from its locale. Obviously, the more $\underline{\theta}^*$'s we choose, the better our confidence bands will be. There is no danger of the band becoming "too wide" since the $\underline{\theta}^*$'s that will cause this to happen will not pass the log-likelihood test in the first place. The time required for the algorithm to run will increase with the number of iterations, thus the use of computer time is a restriction on how many $\underline{\theta}^*$'s we may choose.

6) The final confidence bands

We can now produce a table of the lower and upper function values for each x_i in the domain. If we plot the final lower and upper function values for each x_i in the domain, the curve that joins these points will be the $(1-\alpha)100$ % confidence band for the non-linear regression function given a certain data set.

Chapter 3 - Results

The algorithm that we have discussed in the previous chapter should work for any non-linear regression function with a reasonable number of parameters. We will use this algorithm to find upper and lower confidence bands (Equation 6) for the logistic regression function with two parameters . This particular function was chosen because of the work done by Khorasani and Milliken, who solved for the confidence band for $f(x;\underline{\theta})$ analytically. We will therefore be able to compare the results that we obtain from our Monte-Carlo method with their theoretical conclusions.

To test the algorithm for this application, we will generate a data set from a logistic distribution with a given set of two parameter values.

Generating the data

We will randomly generate a data set which follows a logistic regression distribution by the following process :

- We first choose the true values of θ_1 and θ_2 .

- We will then produce an array of the independent variable X over a predetermined domain. A random number generator which produces numbers from a uniform distribution over the interval [0,1] will be used and then the number will be scaled according to the specified domain.
- A corresponding array of the categorical variable Y will also be produced. This variable Y will assume values of 0 or 1 depending upon whether or not a random number generated from a uniform distribution is less than or greater than the logistic regression function evaluated at the corresponding value of X given the true values of θ_1 and θ_2 . That is, we evaluate the logistic regression function for the particular x_i. Recall that this number will take on a value in the interval [0,1]. We then generate another random number from the uniform distribution on [0,1]. If this generated number is less than the evaluated number then Y will take on the value 1, otherwise it will assume the value 0.

For our example, we will generate a data set of 100 pairs of (X,Y) values, where the X variable may take on any value between 0 and 10. The true values of θ_1 and θ_2 we shall enter will be $\theta_1 = -2.94$ and $\theta_2 = 0.51$.

The Maximum Likelihood Estimate

Once we generate the data, the next step is to calculate the Maximum Likelihood Estimate (M.L.E.) for the two parameters, θ_1 and θ_2.

For the data that we have generated, we find the M.L.E.'s of θ_1 and θ_2 to be -3.225 and 0.617 respectively. The asymptotic covariance matrix for $\hat{\theta}_1$, $\hat{\theta}_2$ was found to be :

$$V = \begin{bmatrix} 0.1987 & -0.0332 \\ -0.0332 & 0.0067 \end{bmatrix}$$

The tolerance level had been set at 10^{-7} and the maximum number of iterations allowed was 200.

Confidence Bands for the logistic regression function

As we noted previously, we may use one of two methods to sample from the confidence region for $\underline{\theta}$, i.e. sampling from the uniform and the multivariate normal distributions. We will discuss briefly the results of both these methods.

The Uniform

As we noted earlier, we chose the $\underline{\theta}^*$'s from a bivariate uniform distribution centered at $\hat{\underline{\theta}}$ with a range of 2.5 standard deviations on either side of $\hat{\theta}_1$ and $\hat{\theta}_2$. This gave us 5 standard deviations as the width of each side of the box as illustrated in Figure 5.

We decided to choose 100 $\underline{\theta}^*$'s from this distribution. The number that passed the log-likelihood test (at $\alpha = 0.05$) was 32 out of these 100. These 32 $\underline{\theta}^*$'s were used to create a confidence band for the function by the method described in Chapter 2.

We divided the domain of X [0,10], into 50 equal subintervals and found the maximum and minimum function values at each endpoint of these subintervals using the $\underline{\theta}^*$'s that passed the test. We thus had 51 x_i's on which we could calculate the maximum and minimum function values and thus plot the upper and lower confidence bands. The curve which joins these maximum and minimum values will define the upper and lower confidence bands respectively. (These bands, as well as the curves for the M.L.E. and the true parameters are illustrated in Fig. 6. The evaluations of the function values are given in the tables in Appendix C) .

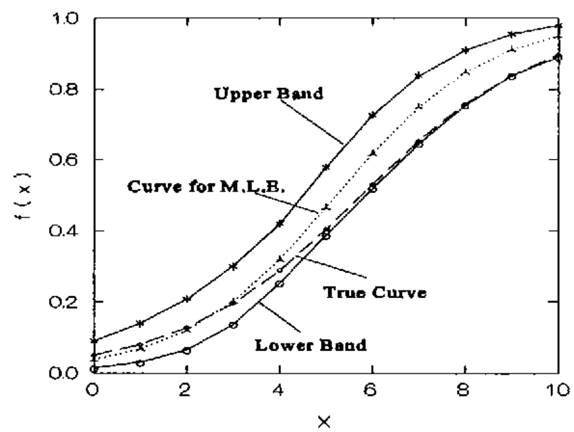


Figure 6
Confidence Bands produced using Uniform Method

We may note that the true curve is completely inside the confidence bands for the defined domain. This can be seen both in the graph and in the function values.

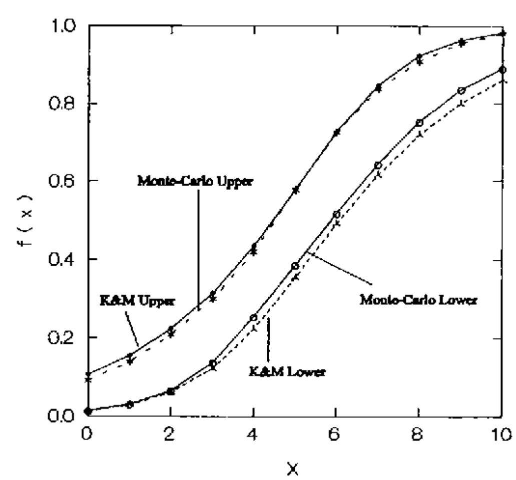


Figure 7
Khorasani/Milliken and Monte Carlo Confidence Bands
(Uniform Method)

To check the accuracy of our techniques, we may use the analytical results developed by Khorasani and Milliken. A plot of our confidence bands is shown along with theirs in Fig. 7. We may note the fact that our bands are narrower than theirs. We will address this issue in our discussion in next chapter. For each x_i, we calculated the difference between the Khorasani/ Milliken and the Monte Carlo upper band. A similar difference was found for the lower band. A plot of these differences is shown in Fig. 8.

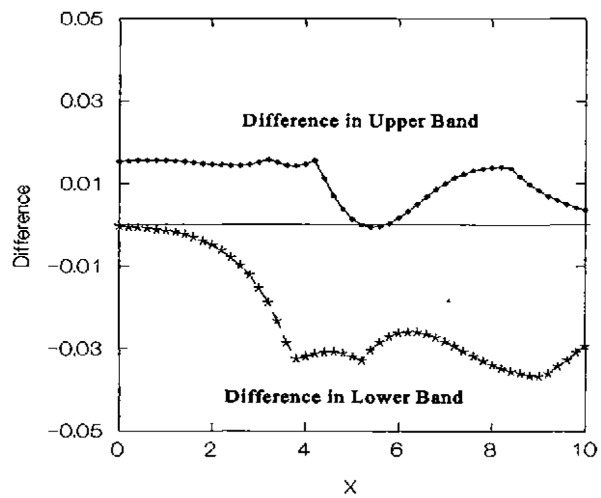


Figure 8
Plot of differences between Khorasani/ Milliken and the Monte Carlo bands for each x_i in the domain (Uniform Method)

The Bivariate Normal

We repeated the steps that we took above with the exception that we now choose the $\underline{\theta}^*$'s from a bivariate normal distribution centered at $\hat{\underline{\theta}}$ with the standard errors derived from the asymptotic covariance matrix V.

We again decided to choose 100 $\underline{\theta}^*$'s from this distribution but this time we noted a marked difference in the number of $\underline{\theta}^*$'s that passed the log-likelihood test. The number that passed the test when the $\underline{\theta}^*$'s were chosen from a bivariate normal distribution was 93 out of the 100 chosen.

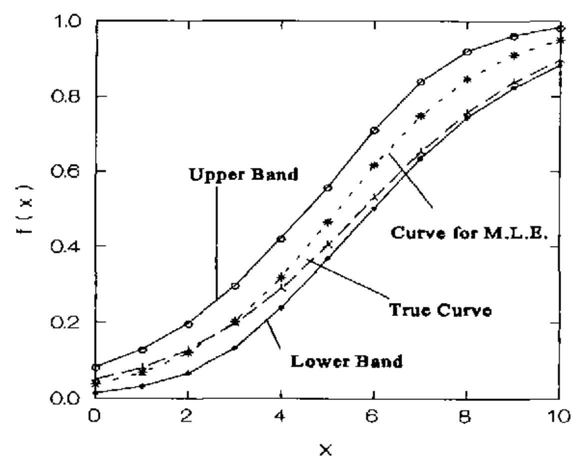


Figure 9
Confidence Bands produced using Normal Method

We found the confidence bands for the function using these $\underline{\theta}^*$'s by the same technique as before. The values of the upper and lower band as well as

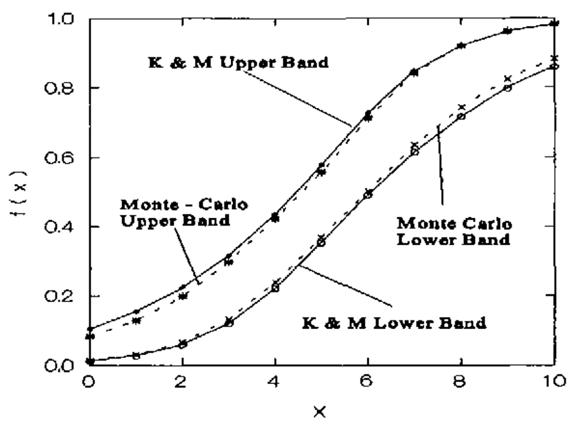


Figure 10
Khorasani/Milliken and Monte Carlo Bands
Normal Method

for the true parameters and the M.L.E. is given in Appendix C. A plot of our confidence bands is shown in Fig. 9. Our bands are shown in comparison to the Khorasani/Milliken bands in Fig. 10. A plot of the differences between our band and the Khorasani/Milliken is illustrated in Fig. 11.

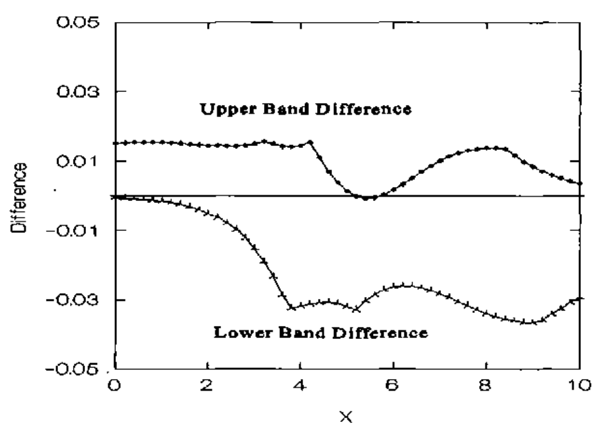


Figure 11
Plot of the differences between the Khorasani/Milliken and the Monte Carlo bands for each x_i in the domain
(Normal Method)

Examination of the accuracy of the Monte Carlo confidence bands

Another test of the validity of our results may be found by repeating our method a large number of times, using different simulated data sets and noting the frequency of the occurrence of the confidence bands completely enclosing the true curve. That is, for a 95 % confidence band, we would expect the true curve to be completely inside the confidence bands around 950 out of every 1000 repetitions.

The Uniform

We generated 1000 different data sets using the same true value of $\underline{\theta}$ on each occasion and created 1000 confidence bands for the logistic regression function, each time sampling 100 $\underline{\theta}$*'s. We found that when the $\underline{\theta}$*'s were chosen from a bivariate uniform distribution as defined above, our confidence bands enclosed the true curve 86.2 % of the time in the specified domain. When we increased the number of sampled $\underline{\theta}$*'s from 100 to 1000 for each iteration, the proportion of times the Monte-Carlo bands captured the true curve increased to 93.2%. A 99% confidence interval for the true proportion of times our bands would completely enclose the true curve is between 91.1% and 95.3%. In comparison, the proportion of times the true curve was completely inside the Khorasani / Milliken confidence band was 95.2%.

The mean percentage of $\underline{\theta}^*$'s that passed the log-likelihood test on each iteration was 34.16% with standard deviation of 2.8%.

The Bivariate Normal

As we did for the bivariate uniform case, we generated 1000 data sets given the true parameter values to see how well our confidence bands performed. The proportion of times that our confidence bands totally captured the true curve when we sampled 100 $\underline{\theta}^*$'s was 85.8%. When we increased the number of $\underline{\theta}^*$'s sampled to 1000, we saw an increase of this capture rate to 92.9%. A 99% confidence interval for the true proportion of times that the Monte Carlo bands would enclose the true curve is 90.8% to 95.0%. We will discuss the implication of these results in the following chapter.

The mean proportion of $\underline{\theta}^*$'s that passed the log-likelihood test on each iteration was 94.87% with standard deviation of 0.67%.

Chapter 4 - Discussion

As we can see from the results when we apply the Monte Carlo Technique to find confidence bands for non-linear regression functions, our bands are narrower than those found by analytical means.

An example is the logistic regression function with two parameters. The bands that were obtained by the analytical methods derived by Khorasani and Milliken were broader than those obtained by us, where we utilized computer intensive techniques. This fact can be attributed to the fact that the $\underline{\theta}^*$'s used by the theoretical methods were chosen from the edges of the ellipse which defined the confidence region for $\underline{\theta}^t$. The $\underline{\theta}^*$'s that we sampled, either from the uniform or the normal distributions, did not always come from the absolute edge of the ellipsoid. Hence, if for any particular x_i, the $\underline{\theta}^*$ used to maximize the upper band, or minimize the lower, did not fall on the edge of the confidence region, then at that particular x_i, our bands would be narrower than the Khorasani/Milliken bands. We did, however, achieve the maximum value for the upper band and the minimum value for the lower band at a couple of points in the domain of X, as can be seen in Figures 8 and 11, where the difference between our bands and those found by theoretical means was 0. It can be concluded, therefore, that at these points the $\underline{\theta}^*$'s we sampled were on the edge of the confidence region ellipsoid.

When we repeated our algorithm 1000 times for different randomly generated data sets, at an α-level of 0.05, we found that when the $\underline{\theta}^*$'s were sampled from a bivariate uniform distribution, our confidence bands totally encompassed the true function 93.2% of the time. When the $\underline{\theta}^*$'s were sampled from the bivariate normal distribution, the true curve was captured 92.9% of the time. The Khorasani/Milliken bands enclosed the true curve 95.2% of the time. This is further confirmation of our earlier conclusions that our bands are slightly narrower than the analytical ones and can be attributed to the reasons that we discussed earlier.

Let us briefly discuss the slight discrepancy between the results for the uniform and the normal methods for sampling the $\underline{\theta}^*$'s. As we noted earlier, the uniform distribution chooses $\underline{\theta}^*$'s indiscriminantly from a prescribed range and hence is just as likely to choose a $\underline{\theta}^*$ further away from $\hat{\underline{\theta}}$ as it is to choose one relatively closer. It is these distant $\underline{\theta}^*$'s that maximize the width of the confidence bands since the further away from $\hat{\underline{\theta}}$ a particular $\underline{\theta}^*$ is, the more likely it is to fall near the boundaries of the ellipse and therefore to maximize the width of the confidence bands.

If we sample from the normal distribution, however, we are more likely to get $\underline{\theta}^*$'s closer to $\hat{\underline{\theta}}$ and thus we do not find many $\underline{\theta}^*$'s near the edges of the ellipsoid which would maximize the width of the band. A $\underline{\theta}^*$ chosen from a bivariate normal distribution is much more likely to pass the log-likelihood test as can be seen from our results (an average of 94.87% passed compared to an average of 34.16% for the uniform distribution), but once this test is passed, it is likely to be too close to $\hat{\underline{\theta}}$ to maximize the width of the band. Hence the band for the normal distribution method is likely to be narrower than the one for the uniform. Therefore the normal distribution band will enclose the true curve on average less frequently than the uniform.

Though the successful capture rate of the true curve for our method may not seem to compare favorably with the theoretical results (especially for the normal case), it should be stressed that the power of our method comes from the generality of our algorithm. Thus when the non-linear function is such that analytical means cannot be employed to find confidence bands, the Monte Carlo method can be employed to a better than reasonable level of success.

Implications and Recommendations

How can we improve the efficiency of our methods ? The most obvious way would be to increase the number of samples of $\underline{\theta}^*$'s that we take. This would increase the number of $\underline{\theta}^*$'s that pass the log-likelihood test and also increase the probability of getting $\underline{\theta}^*$'s closer to the boundaries of the confidence region of $\underline{\theta}^t$. This would either get us closer to the maximum upper limit or the minimum lower limit for a particular $\underline{\theta}^*$. Hence, as a result a maximum width for the confidence band can be achieved.

Evidence of this improvement can be seen from the fact that when we increased the number of $\underline{\theta}^*$'s sampled from 100 to 1000, the capture rate of the true curve improved dramatically for both methods (86.2% to 93.2% for the uniform and 85.8% to 92.9% for the normal). Recall that only those $\underline{\theta}^*$'s that pass the log-likelihood test will play a part in the shaping of confidence bands. Therefore there is no chance of us broadening the bands more than they should be.

The disadvantage in increasing the volume of samples of $\underline{\theta}^*$'s is the increased time required for the algorithm to execute and thus more computer resources will be required for the method to run.

Another option in increasing the success rate for our method would be finding an improved method of sampling the $\underline{\theta}$*'s. If we could choose the $\underline{\theta}$*'s only from a band close to the edge of the ellipse (Fig. 12), then that would improve the frequency of those $\underline{\theta}$*'s that had a legitimate role to play in defining the confidence bands. As we have surmised, the closer the sampled $\underline{\theta}$* is to the edge of the confidence region, the better chance it has of broadening the confidence bands.

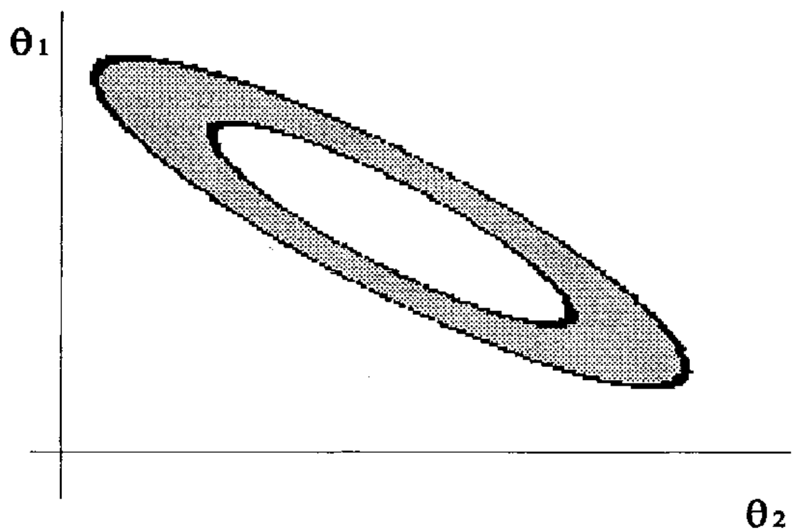


Figure 12
Sampling from an elliptical band

The next step in the development of the algorithm would be to try it on the logistic regression function with more than two parameters. We could also increase the number of independent variables as a way to generalizing the algorithm.

The Michaelis-Menten Kinetic model and the linear regression model are two functions upon which there are theoretical results for finding confidence bands. For the Michaelis-Menten model with two parameters :

$$f(x_i;\theta_1,\theta_2) = \frac{\theta_1 x_i}{\theta_2 + x_i} + \epsilon_i \tag{12}$$

Khorasani and Milliken (1982) developed an analytical approach to finding the confidence bands.

The linear regression model with two parameters :

$$f(x_i;\theta_1,\theta_2) = \theta_1 + \theta_2 x_i + \epsilon_i \tag{13}$$

has the Working-Hotelling approach for finding confidence bands (**Neter, Wasserman and Kutner, 1990**). These two models and respective analytical approaches can be used as further tests of our Monte-Carlo methods.

Conclusion

For any given nonlinear regression function, we may use the Monte Carlo method to find confidence bands for the true function curve. We can therefore find, given a set of values for the independent variables, a point estimate for the value of the dependant variable as well as an interval within which we have a certain degree of confidence that the true function value will lie for that set of independent variables.

Appendix A

<u>Khorasani and Milliken's Method for finding Confidence Bands for the logistic regression function with two parameters.</u>

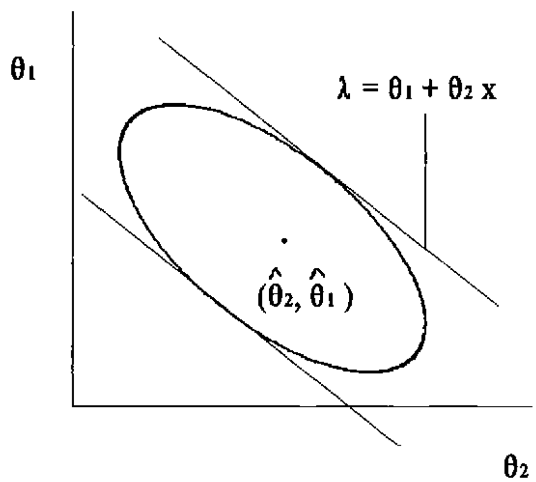


Figure 13
The ellipse which defines the confidence region for (θ_1 , θ_2) in the logistic regression model

The logistic response model is :

$$f(x;\theta_1,\theta_2) = \frac{e^{\theta_1+\theta_2 x}}{1+e^{\theta_1+\theta_2 x}}$$

where f (x ; θ_1 , θ_2) is the probability of response corresponding to dose x and θ_1 and θ_2 are the parameters of the model.

Let $\hat{\theta}_1$ and $\hat{\theta}_2$ be the maximum likelihood estimators of the parameters. Then, for a large sample, a 100(1- α) % confidence ellipse for the parameters is :

$$I_{11}(\hat{\theta}_1 - \theta_1)^2 + 2I_{12}(\hat{\theta}_1 - \theta_1)(\hat{\theta}_2 - \theta_2) + I_{22}(\hat{\theta}_2 - \theta_2)^2 < \chi^2_{(1-\alpha)}(2)$$

where $I_{i,j}$'s are the estimated components of the information matrix (Brand et al 1973) i.e.

$$I_{11} = \sum_{i=1}^{n} \frac{e^{\theta_1+\theta_2 x_i}}{[1+e^{\theta_1+\theta_2 x_i}]^2}$$

$$I_{12} = I_{21} = \sum_{i=1}^{n} x_i \frac{e^{\theta_1+\theta_2 x_i}}{[1+e^{\theta_1+\theta_2 x_i}]^2}$$

$$I_{22} = \sum_{i=1}^{n} x_i^2 \frac{e^{\theta_1+\theta_2 x_i}}{[1+e^{\theta_1+\theta_2 x_i}]^2}$$

To find the maximum and minimum of $f(x;\theta_1,\theta_2)$ over the confidence region (the ellipse) we may find the extremal values of $\lambda(x)$ over the ellipse where

$$\lambda(x) = \ln \frac{f(x;\theta_1,\theta_2)}{1+f(x;\theta_1,\theta_2)} = \theta_1 + \theta_2 x$$

For a fixed value of x, the extremal values of $\lambda(x)$ occur when the line with slope x is tangent to the boundary of the ellipse as shown in Fig. 13.

Therefore, we wish to find extremal values of $\lambda = \theta_1 + \theta_2 x$ where θ_1 and θ_2 are the parameters and x is the value of the independent variable.

Minimum and maximum values are attained when these lines are tangent to the ellipse. i.e. $\frac{\delta \alpha}{\delta \beta} = -x$ or equivalently when $\frac{\delta (\alpha - \hat{\alpha})}{\delta (\beta - \hat{\beta})} = -x$

Using implicit differentiation:

$$\frac{\delta (\theta_1 - \hat{\theta}_1)}{\delta (\theta_2 - \hat{\theta}_2)} = -\left[\frac{I_{12}(\theta_1 - \hat{\theta}_1) + I_{22}(\theta_2 - \hat{\theta}_2)}{I_{11}(\theta_1 - \hat{\theta}_1) + I_{12}(\theta_2 - \hat{\theta}_2)}\right] = -x$$

or

$$I_{12}(\theta_1 - \hat{\theta}_1) + I_{22}(\theta_2 - \hat{\theta}_2) = x I_{11}(\theta_1 - \hat{\theta}_1) + x I_{12}(\theta_2 - \hat{\theta}_2)$$

or

$$(\theta_1 - \hat{\theta}_1) = -K(x)(\theta_2 - \hat{\theta}_2) \quad \text{where } K(x) = \frac{I_{22} - I_{12}x}{I_{12} - I_{11}x}$$

To lie on the ellipse :

$$I_{11}K^2(x)(\theta_2 - \hat{\theta}_2)^2 - 2I_{12}K(x)(\theta_2 - \hat{\theta}_2)^2 + I_{22}(\theta_2 - \hat{\theta}_2)^2 = \chi^2$$

Therefore :

$$\theta_2 = \hat{\theta}_2 \pm \sqrt{\frac{\chi^2}{I_{11}K^2(x) - 2I_{12}K(x) + I_{22}}}$$

and $\theta_1 = \hat{\theta}_1 - K(x)(\theta_2 - \hat{\theta}_2)$

are the pair of values of θ_1 and θ_2 that will give the maximum and minimum function values for any x_i in the domain.

Appendix B

The Hessian Matrix

The Hessian Matrix is defined as :

$$H = \begin{bmatrix} \dfrac{\delta^2 \ln L}{\delta\, \theta_1^2} & \dfrac{\delta^2 \ln L}{\delta\, \theta_1\, \theta_2} \\ \dfrac{\delta^2 \ln L}{\delta\, \theta_2\, \theta_1} & \dfrac{\delta^2 \ln L}{\delta\, \theta_2^2} \end{bmatrix}$$

where L = Likelihood function = $f (x_1 , x_2 , ..., x_n ; \theta_1 , \theta_2)$

and θ_1 , θ_2 are the parameters of the function.

Appendix C

Results of Monte-Carlo Method and Khorasani-Milliken Confidence Bands for data generated for a logistic regression model with two parameters over the domain [0,10] of the independent variable x

Key:

M-C : Monte Carlo Method.

K-M : Analytical Method suggested by Khorasani and Milliken.

Indep. variable x	M-C lower value Uniform	M-C lower value Normal	K-M lower value	M.L.E $\hat{\underline{\theta}}$ value	True $\underline{\theta}$ value	K-M upper value	M-C upper value Normal	M-C upper value Uniform
0.0	0.0136	0.0136	0.0132	0.0382	0.0502	0.1058	0.0905	0.0905
0.2	0.0160	0.0160	0.0154	0.0430	0.0533	0.1143	0.0989	0.0989
0.4	0.0187	0.0187	0.0180	0.0484	0.0609	0.1235	0.1079	0.1079
0.6	0.0220	0.0220	0.0211	0.0544	0.0670	0.1333	0.1177	0.1177
0.8	0.0258	0.0258	0.0246	0.0611	0.0736	0.1438	0.1282	0.1282
1.0	0.0302	0.0302	0.0287	0.0686	0.0809	0.1550	0.1394	0.1394
1.2	0.0353	0.0353	0.0335	0.0769	0.0888	0.1670	0.1516	0.1516
1.4	0.0413	0.0413	0.0390	0.0862	0.0974	0.1798	0.1645	0.1645
1.6	0.0483	0.0483	0.0453	0.0964	0.1068	01.935	0.1784	0.1784
1.8	0.0564	0.0564	0.0525	0.1077	0.1169	0.2080	0.1931	0.1931
2.0	0.0657	0.0657	0.0608	0.1201	0.1279	0.2235	0.2088	0.2088
2.2	0.0764	0.0764	0.0703	0.1338	0.1397	0.2399	0.2253	0.2253
2.4	0.0888	0.0888	0.0810	0.1488	0.1524	0.2572	0.2428	0.2428
2.6	0.1029	0.1029	0.0932	0.1651	0.1660	0.2756	0.2612	0.2612
2.8	0.1189	0.1189	0.1068	0.1828	0.1806	0.2950	0.2804	0.2804
3.0	0.1371	0.1371	0.1219	0.2019	0.1962	0.3156	0.3005	0.3005
3.2	0.1576	0.1576	0.1387	0.2225	0.2128	0.3372	0.3214	0.3214

Indep. variable x	M-C lower value Uniform	M-C lower value Normal	K-M lower value	M.L.E $\hat{\theta}$ value	True θ value	K-M upper value	M-C upper value Normal	M-C upper value Uniform
3.4	0.1804	0.1804	0.1572	0.2446	0.2304	0.3599	0.3448	0.3448
3.6	0.2058	0.2058	0.1773	0.2681	0.2490	0.3837	0.3693	0.3693
3.8	0.2315	0.2315	0.1991	0.2930	0.2685	0.4087	0.3944	0.3944
4.0	0.2541	0.2541	0.2223	0.3192	0.2891	0.4348	0.4201	0.4201
4.2	0.2781	0.2781	0.2469	0.3466	0.3105	0.4619	0.4463	0.4463
4.4	0.3034	0.3034	0.2726	0.3751	0.3327	0.4900	0.4788	0.4788
4.6	0.3300	0.3300	0.2993	0.4044	0.3557	0.5190	0.5119	0.5119
4.8	0.3577	0.3577	0.3267	0.4344	0.3794	0.5488	0.5449	0.5449
5.0	0.3864	0.3864	0.3545	0.4650	0.4037	0.5790	0.5775	0.5775
5.2	0.4152	0.4152	0.3824	0.4957	0.4285	0.6095	0.6095	0.6095
5.4	0.4407	0.4407	0.4104	0.5266	0.4536	0.6399	0.6405	0.6405
5.6	0.4665	0.4665	0.4381	0.5572	0.4790	0.6700	0.6704	0.6704
5.8	0.4924	0.4924	0.4654	0.5874	0.5045	0.6994	0.6990	0.6990
6.0	0.5184	0.5184	0.4923	0.6169	0.5300	0.7279	0.7261	0.7261
6.2	0.5444	0.5444	0.5185	0.6456	0.5553	0.7550	0.7517	0.7517
6.4	0.5700	0.5700	0.5441	0.6733	0.5803	0.7807	0.7756	0.7756
6.6	0.5953	0.5953	0.5689	0.6999	0.6049	0.8047	0.7978	0.7978
6.8	0.6201	0.6201	0.5929	0.7251	0.6290	0.8269	0.8183	0.8183
7.0	0.6443	0.6443	0.6161	0.7490	0.6525	0.8473	0.8372	0.8372
7.2	0.6678	0.6678	0.6385	0.7715	0.6752	0.8659	0.8544	0.8544
7.4	0.6905	0.6905	0.6599	0.7925	0.6972	0.8826	0.8702	0.8702
7.6	0.7123	0.7123	0.6806	0.8121	0.7183	0.8976	0.8844	0.8844
7.8	0.7332	0.7332	0.7003	0.8302	0.7385	0.9109	0.8973	0.8973
8.0	0.7530	0.7530	0.7191	0.8469	0.7577	0.9228	0.9089	0.9089
8.2	0.7719	0.7719	0.7371	0.8622	0.7759	0.9332	0.9192	0.9192
8.4	0.7897	0.7897	0.7542	0.8762	0.7931	0.9423	0.9286	0.9286
8.6	0.8064	0.8064	0.7704	0.8890	0.8094	0.9503	0.9385	0.9385

Indep. variable x	M-C lower value Uniform	M-C lower value Normal	K-M lower value	M.L.E $\hat{\underline{\theta}}$ value	True $\underline{\theta}$ value	K-M upper value	M-C upper value Normal	M-C upper value Uniform
8.8	0.8222	0.8222	0.7857	0.9006	0.8246	0.9572	0.9473	0.9473
9.0	0.8369	0.8369	0.8003	0.9111	0.8389	0.9633	0.9458	0.9458
9.2	0.8500	0.8500	0.8140	0.9206	0.8522	0.9685	0.9614	0.9614
9.4	0.8611	0.8611	0.8269	0.9292	0.8646	0.9730	0.9670	0.9670
9.6	0.8716	0.8716	0.8390	0.9369	0.8761	0.9769	0.9718	0.9718
9.8	0.8813	0.8813	0.8505	0.9438	0.8868	0.9802	0.9760	0.9760
10.0	0.8905	0.8905	0.8612	0.9500	0.8966	0.9831	0.9795	0.9795

Appendix D

Programs in Pascal written to find the Confidence Band for a logistic regression function with two parameters. The two methods of sampling are from a bivariate uniform distribution and a bivariate normal distribution .

Method 1 Sampling from a bivariate uniform distribution

```
Program Monte-Carlo ;

const
  ndim = 2 ;
  n = 2   ;

type
  glnarray = array (.1..n.) of real ;
  glndim = array (.1..ndim.) of real ;
  realarray = array (.1..200.) of real ;
  plotxarray = array (.1..60.) of real;

var
  temp,diff1,diff2 : real ;
  xmax,xmin : real;
  plotx : plotxarray ;
  counter : integer;
  ncom,i,j,k,maxiter : integer;
  pcom, xicom : glnarray ;
  ftol,diff,norm1,norm2,fret : real ;
  b,lastb,dir,betastar : glnarray ;
  infile, outfile : text ;
  derv0,derv1,derv01 : real ;
  det,var1,var2,std1,std2 : real ;
  W,Z,lower,upper : realarray ;
  iseed : integer ;
  trueb : glnarray ;
  xl,xu : real ;
  glix1,glix2,glix3: integer;
  glr: ARRAY(.1..97.) OF real;
  ycnt,lambda : real;
  nrep : integer;
```

```
Function Ran1(VAR idum: integer): real;
{ This function generates a random number in the interval [0,1] (from Press et al) }

CONST
  m1=259200;
  ia1=7141;
  ic1=54773;
  rm1=3.8580247e-6;  (* 1.0/m1 *)
  m2=134456;
  ia2=8121;
  ic2=28411;
  rm2=7.4373773e-6;  (* 1.0/m2 *)
  m3=243000;
  ia3=4561;
  ic3=51349;

VAR  j: integer;

BEGIN
  IF (idum < 0) THEN BEGIN
    glix1 := (ic1-idum) MOD m1;
    glix1 := (ia1*glix1+ic1) MOD m1;
    glix2 := glix1 MOD m2;
    glix1 := (ia1*glix1+ic1) MOD m1;
    glix3 := glix1 MOD m3;
    FOR j := 1 to 97 DO BEGIN
      glix1 := (ia1*glix1+ic1) MOD m1;
      glix2 := (ia2*glix2+ic2) MOD m2;
      glr(.j.):= (glix1+glix2*rm2)*rm1
    END;
    idum := 1
  END;
  glix1 := (ia1*glix1+ic1) MOD m1;
  glix2 := (ia2*glix2+ic2) MOD m2;
  glix3 := (ia3*glix3+ic3) MOD m3;
  j := 1 + (97*glix3) DIV m3;
  IF ((j > 97) OR (j < 1)) THEN BEGIN
    writeln('pause in routine RAN1'); readln
  END;
  ran1 := glr(.j.);
  glr(.j.):= (glix1+glix2*rm2)*rm1
END;
```

```
Procedure GenLog ( TrueB : GLNArray; Var X,Y : Realarray; XL,XU : real; var ycnt:real);
{ Generates an independent variable X in the range [XL,XU] and a corresponding categorical
Y variable following a logistic distribution }

Var
  I : integer ;
  Func,FIRST,SECOND : real ;
  R : real ;

Begin
  ycnt := 0.0;
  for I:= 1 TO 200 DO
    BEGIN
      X(.I.) := XL + RAN1(ISEED)* (XU-XL) ;
      FIRST :=EXP(TRUEB(.1.)+TRUEB(.2.)*X(.I.));
      SECOND := 1 + EXP(TRUEB(.1.) + TRUEB(.2.) * X(.I.));
      FUNC := FIRST/SECOND;
      R := RAN1(ISEED);
      IF R < FUNC THEN Y(.I.) := 1 ELSE Y(.I.) := 0;
      ycnt := ycnt + y(.i.)
    END
END;

(****************************************************************)

Function FNC ( xt : glnarray;W,Z : realarray ) : real;
{ Calculates the value of the log likelihood function for a given value of X and the
parameters}

VAR
  I : INTEGER ;
  FIRST,SECOND,THIRD,TEMP : REAL ;

BEGIN
  TEMP := 0;
  FOR I:= 1 TO 200 DO
    BEGIN
      FIRST := Z(.I.) * (xt(.1.) + (xt(.2.)*W(.I.)));
      second := LN(1+EXP(xt(.1.)+xt(.2.)*W(.I.)));
      TEMP := TEMP - FIRST + SECOND
    END;
  FNC := TEMP
END;
```

```
Function DFunc0 (B:GLNARRAY ; W,Z : REALARRAY):REAL;
{ Calculates the first derivative of the ln likelihood function w.r.t. the first parameter }

 VAR
  I : INTEGER ;
  TEMP,FIRST,SECOND : REAL ;

BEGIN
  TEMP := 0;
  FOR I:= 1 TO 200 DO
    BEGIN
      FIRST := -(EXP(B(.1.)+B(.2.)*W(.I.))*(Z(.I.)-1) + Z(.I.));
      SECOND:= (1+EXP(B(.1.)+B(.2.)*W(.I.)));
      TEMP := TEMP + (FIRST/SECOND)
    END;
  DFUNC0 :=TEMP
END;

(************************************************************)

Function DFunc1 (B:GLNARRAY ; W,Z : REALARRAY):REAL;
{ Calculates the first derivative of the ln likelihood function w.r.t. the second parameter }

 VAR
  I : INTEGER ;
  TEMP,FIRST,SECOND : REAL ;

BEGIN
  TEMP := 0;
  FOR I:= 1 TO 200 DO
    BEGIN
      FIRST:=-W(.I.)*(EXP(B(.1.)+B(.2.)*W(.I.))*(Z(.I.)-1) + Z(.I.));
      SECOND:= (1+EXP(B(.1.)+B(.2.)*W(.I.)));
      TEMP := TEMP + (FIRST/SECOND)
    END;
  DFUNC1 :=TEMP
END;
```

```
Function D2Func0 (B:GLNARRAY ; W,Z : REALARRAY):REAL;
{ Calculates the second derivative of the ln likelihood function w.r.t. the first parameter }

 VAR
  I : INTEGER ;
  TEMP,FIRST,SECOND : REAL ;

BEGIN
  TEMP := 0;
  FOR I:= 1 TO 200 DO
    BEGIN
      FIRST := EXP(B(.1.)+B(.2.)*W(.I.));
      SECOND:= SQR((1+EXP(B(.1.)+B(.2.)*W(.I.))));
      TEMP := TEMP - (FIRST/SECOND)
    END;
  D2FUNC0 :=TEMP
END;

(*************************************************************)

Function D2Func1 (B:GLNARRAY ; W,Z : REALARRAY):REAL;
{ Calculates the second derivative of the ln likelihood function w.r.t. the second parameter}

 VAR
  I : INTEGER ;
  TEMP,FIRST,SECOND : REAL ;

BEGIN
  TEMP := 0;
  FOR I:= 1 TO 200 DO
    BEGIN
      FIRST := sqr(W(.I.)) * EXP(B(.1.)+B(.2.)*W(.I.));
      SECOND:= SQR((1+EXP(B(.1.)+B(.2.)*W(.I.))));
      TEMP := TEMP - (FIRST/SECOND)
    END;
  D2FUNC1 :=TEMP
END;
```

Function D2Func01 (B:GLNARRAY ; W,Z : REALARRAY):REAL;
{ Calculates the second derivative of the ln likelihood function w.r.t. both the parameters }

```
VAR
  I : INTEGER ;
  TEMP,FIRST,SECOND : REAL ;

BEGIN
  TEMP := 0;
  FOR I:= 1 TO 200 DO
    BEGIN
      FIRST := W(.I.) * EXP(B(.1.)+B(.2.)*W(.I.));
      SECOND:= SQR((1+EXP(B(.1.)+B(.2.)*W(.I.))));
      TEMP := TEMP - (FIRST/SECOND)
    END;
  D2FUNC01:=TEMP
END;

(*************************************************************)
```

Function F1dim(x: real;W,Z : REALARRAY): real;
{ Used by **Linmin** (Press et al) to find the minimum of a function along a vector }

```
VAR
  j: integer;
  xt: glnarray;
BEGIN
  FOR j := 1 to ncom DO BEGIN
    xt(.j.):= pcom(.j.)+x*xicom(.j.)
  END;
  f1dim := fnc(xt,W,Z)
END;

(*********************************************************)
```

Function Func (x:real) : real ;
{ Used by Linmin }

```
  begin
    func := f1dim (x,W,Z)
  end ;
```

```
Procedure Mnbrak (var AX,bx,cx,fa,fb,fc : real );
{ Used by Linmin }

LABEL 1;
CONST
  gold=1.618034;
  glimit=100.0;
  tiny=1.0e-20;
VAR
  ulim,u,r,q,fu,dum: real;
FUNCTION max(a,b: real): real;
  BEGIN
    IF (a > b) THEN max := a ELSE max := b
  END;
FUNCTION sign(a,b: real): real;
  BEGIN
    IF (b > 0.0) THEN sign := abs(a) ELSE sign := -abs(a)
  END;
BEGIN
  fa := func(ax);
  fb := func(bx);
  IF (fb > fa) THEN BEGIN
    dum := ax;
    ax := bx;
    bx := dum;
    dum := fb;
    fb := fa;
    fa := dum
  END;
  cx := bx+gold*(bx-ax);
  fc := func(cx);
1:  IF (fb >= fc) THEN BEGIN
    r := (bx-ax)*(fb-fc);
    q := (bx-cx)*(fb-fa);
    u := bx-((bx-cx)*q-(bx-ax)*r)/
      (2.0*sign(max(abs(q-r),tiny),q-r));
    ulim := bx+glimit*(cx-bx);
    IF ((bx-u)*(u-cx) > 0.0) THEN BEGIN
      fu := func(u);
      IF (fu < fc) THEN BEGIN
        ax := bx;
```

```
{ procedure MNBRAK continued }

        fa := fb;
        bx := u;
        fb := fu;
        GOTO 1 END
      ELSE IF (fu > fb) THEN BEGIN
        cx := u;
        fc := fu;
        GOTO 1
      END;
      u := cx+gold*(cx-bx);
      fu := func(u)
    END ELSE IF  ((cx-u)*(u-ulim) > 0.0) THEN BEGIN
      fu := func(u);
      IF (fu < fc) THEN BEGIN
        bx := cx;
        cx := u;
        u := cx+gold*(cx-bx);
        fb := fc;
        fc := fu;
        fu := func(u)
      END
    END ELSE IF  ((u-ulim)*(ulim-cx) >= 0.0) THEN BEGIN
      u := ulim;
      fu := func(u)
    END ELSE BEGIN
      u := cx+gold*(cx-bx);
      fu := func(u)
    END;
    ax := bx;
    bx := cx;
    cx := u;
    fa := fb;
    fb := fc;
    fc := fu;
    GOTO 1
  END
END;
```

Function Brent (ax,bx,cx,tol: real; VAR xmin: real): real;

```
{ Used by Linmin }

LABEL 1,2,3;
CONST
  itmax=100;
  cgold=0.3819660;
  zeps=1.0e-10;
VAR
  a,b,d,e,etemp: real;
  fu,fv,fw,fx: real;
  iter: integer;
  p,q,r,tol1,tol2: real;
  u,v,w,x,xm: real;
FUNCTION sign(a,b: real): real;
  BEGIN
    IF (b > 0.0) THEN sign := abs(a) ELSE sign := -abs(a)
  END;
BEGIN
  IF ax < cx THEN a := ax ELSE a := cx;
  IF ax > cx THEN b := ax ELSE b := cx;
  v := bx;
  w := v;
  x := v;
  e := 0.0;
  fx := func(x);
  fv := fx;
  fw := fx;
  FOR iter := 1 to itmax DO BEGIN
    xm := 0.5*(a+b);
    tol1 := tol*abs(x)+zeps;
    tol2 := 2.0*tol1;
    IF (abs(x-xm) <= (tol2-0.5*(b-a))) THEN GOTO 3;
    IF (abs(e) > tol1) THEN BEGIN
      r := (x-w)*(fx-fv);
      q := (x-v)*(fx-fw);
      p := (x-v)*q-(x-w)*r;
      q := 2.0*(q-r);
      IF (q > 0.0) THEN  p := -p;
      q := abs(q);
      etemp := e;
      e := d;
```

```
{ Function BRENT continued }

        IF((abs(p) >= abs(0.5*q*etemp)) OR (p <= q*(a-x))
          OR (p >= q*(b-x))) THEN GOTO 1;
        d := p/q;
        u := x+d;
        IF (((u-a)<tol2) OR ((b-u)<tol2)) THEN d := sign(tol1,xm-x);
        GOTO 2
      END;
1:      IF (x >= xm)  THEN e := a-x ELSE e := b-x;
      d := cgold*e;
2:      IF (abs(d) >= tol1)  THEN u := x+d ELSE u := x+sign(tol1,d);
      fu := func(u);
      IF (fu <= fx)  THEN BEGIN
        IF (u >= x)  THEN a := x ELSE b := x;
        v := w;
        fv := fw;
        w := x;
        fw := fx;
        x := u;
        fx := fu
      END ELSE BEGIN
        IF (u < x)  THEN a := u ELSE b := u;
        IF ((fu <= fw) OR (w = x))  THEN BEGIN
          v := w;
          fv := fw;
          w := u;
          fw := fu
        END ELSE IF ((fu <= fv) OR (v = x) OR (v = 2)) THEN BEGIN
          v := u;
          fv := fu
        END
      END
    END;
    writeln('pause in routine BRENT - too many iterations');
3:  xmin := x;
    brent := fx
END;
```

```
Procedure Linmin (VAR p,xi: glnarray; n: integer; VAR fret: real);

{ Finds the minimum value of a specified function along a given vector }

CONST
  tol=1.0e-4;
VAR
  j: integer;
  xx,xmin,fx,fb,fa,bx,ax: real;
BEGIN
  ncom := n;
  FOR j := 1 to n DO BEGIN
    pcom(.j.) := p(.j.);
    xicom(.j.) := xi(.j.)
  END;
  ax := 0.0;
  xx := 1.0;
  bx := 2.0;
  mnbrak(ax,xx,bx,fa,fx,fb);
  fret := brent(ax,xx,bx,tol,xmin);
  FOR j := 1 to n DO BEGIN
    xi(.j.) := xmin*xi(.j.);
    p(.j.) := p(.j.) + xi(.j.)
  END
END;

(**********************************************************)

Function Logistic (var beta:glnarray ; x:real ):real;
{ Calculates the logistic function for a given value x and the parameter vector beta }

var
  first,second : real ;

begin
  first := (exp(beta(.1.) + (beta(.2.)*x)));
  second := 1+(exp(beta(.1.) + (beta(.2.)*x)));
  logistic := first / second
end ;
```

```
Procedure Getbetastaruni ( var betastar,b :glnarray ; std1,std2 : real ) ;
{ Samples a betastar from a bivariate uniform distribution centered at b }

begin
   betastar(.1.) := ((ran1(iseed) - 0.5)*5*std1) + b(.1.);
   betastar(.2.) := ((ran1(iseed) - 0.5)*5*std2) + b(.2.)
end ;

(*****************************************************************)

BEGIN  {Main Program}

  TRUEB(.1.) :=-2.94; TRUEB(.2.) := 0.51;         {True value of the parameters}
  nrep := 100;
  ISEED := -346834287;
  XL:=0;XU:=10;                                   {Domain of the independent variable}
  GENLOG(TRUEB,W,Z,XL,XU,ycnt);
  i := 0;
  b(.1.) := ln(ycnt/(200-ycnt));                  {Starting values for the parameters }
  b(.2.) := 0.00 ;
  maxiter :=200 ;                                 {Maximum number of iteration to find min}
  ftol := 1.0e-07;                                {Tolerance level }
  lastb(.1.) :=0.35 ;
  lastb(.2.) := 0.5 ;
  norm1 := sqr (lastb(.1.) - b(.1.) );
  norm2 := sqr (lastb(.2.) - b(.2.) );
  diff := sqrt (norm1 + norm2 ) ;
  while ( diff > ftol ) and ( i < maxiter ) do    { Finds the M.L.E. for the parameters}
    begin
      lastb(.1.) := b(.1.) ;
      lastb(.2.) := b(.2.) ;
      I := I + 1 ;
      DIR(.1.):=DFUNC0(B,W,Z)/1000;
      DIR(.2.):=DFUNC1(B,W,Z)/1000;
      LINMIN(B,DIR,NDIM,FRET);
       norm1 := sqr (lastb(.1.) - b(.1.) );
      norm2 := sqr (lastb(.2.) - b(.2.) );
      diff := sqrt (norm1 + norm2 ) ;
    end ;
```

```
write(outfile,'Final value of log likelihood function is ');
writeln(outfile,fret:10:2);
write(outfile,'Final values of beta hat is ');
writeln(outfile,b(.1.):10:2,b(.2.):10:2);

DERV0 := D2FUNC0(B,W,Z);
DERV1 := D2FUNC1(B,W,Z);
DERV01:= D2FUNC01(B,W,Z);
DET := (-DERV0*(-DERV1))-SQR(DERV01);
VAR1 := (1/DET)*(-DERV1);                {Variances of parameter estimates}
VAR2 := (1/DET)*(-DERV0);
STD1 := SQRT(VAR1);                      {Std. errors of parameter estimates}
STD2 := SQRT(VAR2);
writeln(outfile,'std. dev. of betas is ',std1:10:2,std2:10:2);
writeln(outfile);

j:=1; xmax := xu ; xmin := xl; plotx(.1.) := xl;
while (j <= 51) do              {sets the upper and lower band to the M.L.E values}
 begin
   lower(.j.) := logistic(b,plotx(.j.));
   upper(.j.) := logistic(b,plotx(.j.));
   plotx(.j+1.) := plotx(.j.) + ((xu-xl)/50);
   j:=j+1;
 end;
counter := 0;

for k:= 1 to 100 do
  begin
    getbetastaruni (betastar,b,std1,std2);
     lambda := 2* (-fnc(b,w,z) + fnc(betastar,w,z));
    if Lambda < 5.99 then         {Checks to see if log-likelihood test is passed}
      begin
        counter := counter + 1;  {Counts the number of betastars that pass the test}
        for j:= 1 to 51  do
        begin
          temp := logistic(betastar,plotx(.j.));
          if temp < lower(.j.)
          then lower(.j.) := temp;        {Sets the lower band if necessary}
          if temp > upper(.j.)
          then upper(.j.) := temp         {Sets the upper band if necessary}
        end
      end;
  end;
```

```
      writeln(outfile,'Counter is ',counter);
      writeln(outfile);
      writeln(outfile,'  x(i)    lower    betahat  upper    true');
      writeln(outfile,'  ----    -----    -------  -----    ----');
      for j:= 1 to 51 do
         writeln(outfile,plotx(.j.):10:4,lower(.j.) :10:4,
                 logistic(b,plotx(.j.)):10:4  , upper(.j.):10:4,
                 logistic(trueb,plotx(.j.)):10:4);
      writeln(outfile);
      Write  (Outfile,'true value of beta1, beta2 : ');
      Writeln(Outfile, trueb(.1.) :7:5,trueb(.2.) :9:5);
END.  { Main Program }
```

Method 2 Sampling from a bivariate normal distribution.

The program that samples from a bivariate normal distribution will be identical to the one above except that Procedure Getbetastaruni will be replaced by the following procedure and a corresponding call to that procedure in the main program.

```
Procedure Getbetastarnorm (l:varmatrix;b:glnarray;var betastar :glnarray);

var
  i :integer;
  sum,z1,z2:real;
begin
  z1:=gasdev(iseed);z2:=gasdev(iseed);
  betastar(.1.) :=(l(.1,1.)*z1 + l(.1,2.)*z2) + b(.1.);
  betastar(.2.) :=(l(.2,1.)*z1 + l(.2,2.)*z2) + b(.2.);
end;
```

We would also insert the following routines :

```
 Function Gasdev(VAR idum: integer): real;
{ Generates a number from the normal distribution with mean 0 and variance 1 }

VAR
  fac,r,v1,v2: real;
BEGIN
  IF  (gliset = 0)  THEN BEGIN
    REPEAT
      v1 := 2.0*ran1(idum)-1.0;
```

```
      v2 := 2.0*ran1(idum)-1.0;
      r := sqr(v1)+sqr(v2);
    UNTIL (r < 1.0);
    fac := sqrt(-2.0*ln(r)/r);
    glgset := v1*fac;
    gasdev := v2*fac;
    gliset := 1
  END ELSE BEGIN
    gasdev := glgset;
    gliset := 0
  END
END;

(**************************************************************)

Procedure Choldc (var a:varmatrix ; n,np :integer;var p :varvector);

{ Uses Choleski Decomposition to decompose a matrix into the product of a lower triangular
matrix and its transpose }

var
  i,j,k : integer ;
  sum   : real ;

begin
  for i:= 1 to n do
    for j:= 1 to n do
      begin
        sum := a(.i,j.);
        for k:= (i-1) downto 1  do
          sum:= sum - a(.i,k.) * a(.j,k.);
        if (i=j) then
          begin
            If (sum <= 0) then writeln(outfile,'choldc failed');
            p(.i.) := sqrt(sum)
          end
        else
          a(.j,i.) := sum/p(.i.)
      end
end;
```

Bibliography

1. Khorasani, F. and Milliken, G.A. **Simultaneous Confidence Bands for Nonlinear Regression Models.** Communications in Statistics, Theory and Methods, 11(11), pg 1241-1253 (1982).

2. Brand, R.J., Pinnoch, D.E., and Jackson, K.L. **Large Sample Confidence Bands for the Logistic Response Curve and its Inverse**. American Statistician, 27, pg 157-160 (1973)

3. Press et al **Numerical Recipes in Pascal.** Cambridge University Press (1989).

4. Agresti, A. **Categorical Data Analysis**. Wiley and Sons (1990)

5. Neter, J., Wasserman, W. and Kutner, M.H. **Applied Linear Statistical Models, Third Edition**. Irwin (1990).

6. Myers, R.H. and Milton, J.S. A **First Course in the Theory of Linear Statistical Models**, PWS-Kent (1991).

VITA

Shantonu Mazumdar was born in on to Drs. Subir and Sandhya Mazumdar, both medical academicians. He spent the early years of his childhood in Siliguri, in the foothills of the Himalayas before moving to Kingston, Jamaica, where his parents took employment at the University of the West Indies.

He returned to India after three years and enrolled in St. Xaviers' High School in Calcutta. After two years there, he moved to Papua New Guinea. He finished his high school education at the Port Moresby International High School where he completed the N.S.W. (Australia) Higher School Certificate in 1985.

He completed his Bachelor of Science degree in Mathematics from the University of Papua New Guinea in 1991. He worked as a tutor in Mathematics and Computer Science at U.P.N.G. for a short period of time before proceeding to the University of North Florida in Jacksonville to complete his Master of Science degree, majoring in Statistics. After completion of this degree, he will proceed to the University of Georgia to pursue a Ph.D. degree in Statistics.