DEPARTMENT OF TRANSPORT AND COMMUNICATIONS

FEDERAL OFFICE OF ROAD SAFETY

DOCUMENT RETRIEVAL INFORMATION

Report No.	Date	Pages	ISBN	ISSN
MR 12	August 1993	64	0 642 51372.4	0810-770X

Title and Subtitle

Estimation of Truncated Ordinal Regression Models

Author(s)

O'Neill T.J.
Barry S.

Performing Organisation

Department of Statistics
Australian national University

Sponsors

Federal Office of Road Safety
GPO Box 594
CANBERRA ACT 2601

Project Officer: Deanne Perry

Available from

Federal Office of Road Safety
GPO Box 594
CANBERRA ACT 2601

Abstract

This report discusses the implementation of a new procedure called Truncated Ordinal Regression (TOR). The technique is more general and efficient than the existing methods. It allows for an ordinal scale of injury such as uninjured, severe injury, dead.

Keywords

Accident Statistics, Statistics, Truncated Ordinal Regression, Databases, Data Analysis

Estimation of Truncated Ordinal Regression Models

Department of Transport & Communications Road Safety Seeding Research Grant

T.J. O'Neill
S. Barry
Department of Statistics
The Faculties, Australian National University

July 16, 1993

Abstract

A major obstacle to the regression analysis of road traffic fatality data is that data is typically only recorded for accidents where at least one fatality occurs. Examples are the FARS database in the USA and the Fatal File of the Federal Office of Road Safety. Data of this type is called group truncated data. A regression technique should allow for this truncation if it is to avoid serious biases. Two existing methods are Conditional Logistic Regression (CLR) (Lui, McGee, Rhodes, & Pollack (1988)) and Double Pair Comparisons (DPC) (Evans (1985)). This report discusses the implementation of a new procedure called Truncated Ordinal Regression (TOR). The technique is more general and efficient than the existing methods. It allows for an ordinal scale of injury such as uninjured, moderate injury, severe injury, dead. The software consists of an Splus interface to a suite of C routines. Help files, installation scripts and an example are provided with the software. As a more complicated example of its use, TOR is applied to the Fatal File in Appendix B.

0 Executive Summary

A common feature of mass databases on road traffic fatalities is that only accidents in which at least one fatality occurred are included. Define a binary response variable Y which is 0 if an individual survives and 1 if the individual dies. Then this road traffic fatality data is called group truncated data, since the data is only collected if at least one of the binary response variables is one. The aim of compiling the mass databases is to relate the fatalities observed to the variables that are thought to influence the chance of a fatality, for example age, sex and seatbelt use. Ordinary logistic regression will be subject to serious biases if it is applied to group truncated data. O'Neill & Barry (1993b) and O'Neill & Barry (1993a) recently proposed the method of Truncated Logistic Regression (TLR) and its extension TOR to analyze group truncated binary and ordinal data. The only other methods available to handle such data are Conditional Logistic Regression (CLR) which has been discussed in the road traffic context by Lui et al. (1988) and the Double Pair Comparisons (DPC) method of Evans (1985).

In general, since TLR uses the full information from the sample, TLR can be expected to lead to the most accurate estimates and the most powerful tests of the effects of variables on survival prospects. The following properties hold.

- TLR will give the best estimates of the effect on survival of various variables, for example seat belt usage or age of the occupant, followed by CLR and then DPC. TLR will also give the most powerful hypothesis tests. This is because TLR uses all the information that is available in the data. CLR can only use comparisons within a vehicle. For example the fact that a female died in one car while a male didn't die in exactly the same circumstances in another car does not contribute any information to the CLR estimate. Also cars in which all occupants die contribute no information to the CLR estimate. Both of these scenarios would add to the TLR information. DPC only uses the data which satisfies a condition on a single control variable and so will normally be less precise.

- More effects can be fitted using TLR. The conditional logistic regression likelihood equation 4 only includes terms which vary within a given accident. For example, for single vehicle accidents, since the speed of the car is constant for all the occupants, its effect on the survival prospects cannot be estimated using CLR. TLR on the other hand can be used to estimate its effect. CLR cannot estimate the effect of variables which do not vary within accidents.

- Only TOR can be used to estimate the relative seriousness of crashes for occupants. The TLR method allows us to estimate the probability that a given type of crash will kill a given type of occupant. The TOR method enables the estimation of the probabilities of the various categories of injury.

- Only TLR can be used to estimate the total number of potentially fatal crashes. The TLR method allows us to estimate the probability that a particular configuration of factors results in a fatality. By dividing the observed number of crashes of this type by this probability we obtain an estimate of the total number of potentially fatal crashes of this type. The estimates can then be summed over the categories of crashes to obtain an estimate of the total number of potentially fatal crashes.

- Unlike CLR, TOR can be generalized to different link functions. Various researchers have found that the logistic link given in equation 1 does not work well when dealing with very rare events. The TOR method allows us to choose the link function which best fits a given data set.

The aims of the seeding grant were:

- To extend the truncated logistic regression estimates of O'Neill & Barry (1993b) to ordinal response models.
- To develop software to fit the generalized model to the level where it is accessible to road traffic researchers.
- To apply the software to a suitable data set abstracted from the FORS 1988 Fatal File.

All of these aims have been met. The extension of the theory to Truncated Ordinal Regression is given in O'Neill & Barry (1993a) which is attached as Appendix A. The software which has been developed consists of an Splus interface to a suite of C routines. Help files, installation scripts and an example are provided with the software. As a more complicated example of its use, TOR is applied to the 1988 and 1990 Fatal Files in Appendix B. The resulting estimates of the effects of variables are consistent with expectations and are more precise than other methods.

Now that a computer package is available to perform TOR, it can be applied to a variety of road traffic accident databases by interested researchers. It will improve the accuracy of the estimates and conclusions and allow more general questions to be posed.

1 Introduction

A common feature of mass databases on road traffic fatalities is that only accidents in which at least one fatality occurred are included. Define a binary response variable Y which is 0 if an individual survives and 1 if the individual dies. Then this type of data is called group truncated data since the data is only collected if at least one of the binary response variables is one. The aim of compiling the mass databases is to relate the fatalities to the variables that are thought to influence the chance of a fatality, for example age, sex and seatbelt wearing. Ordinary logistic regression will be subject to serious biases if it is applied to group truncated data. O'Neill & Barry (1993b) recently proposed the method of Truncated Logistic Regression (TLR) to analyze group truncated binary data. The only other methods available to handle such data are Conditional Logistic Regression (CLR) which has been discussed in the road traffic context by Lui et al. (1988) and the Double Pair Comparisons (DPC) method of Evans (1985). The aims of the seeding grant were:

- To extend the truncated logistic regression estimates of O'Neill & Barry (1993b) to ordinal response models.
- To develop software to fit the generalized model to the level where it is accessible to road traffic researchers.
- To apply the software to a suitable data set abstracted from the FORS 1988 Fatal File.

All of these aims have been met. The theoretical extensions to group truncated ordinal responses such as uninjured, moderate injury, severe injury, dead have been made and are described in Section 2. The software has been developed and is described in Section 3. The methods and software are applied to some examples in Section 4 As a more complicated example of its use, TLR is applied to frontal collision data from the 1988 and 1990 FORS Fatal Files in Appendix B. The merits of the new software are discussed in section 5.

2 Methods

2.1 The estimators

The method of TLR is described in the paper of O'Neill & Barry (1993b) which is attached as Appendix B. Suppose that the binary variable Y is 0 if an individual survives and 1 if the individual dies. Also suppose that x is a vector of covariates thought to influence survival. Then the logistic model is that

$$Pr(Y=1) = \frac{\exp \beta' x}{1+\exp \beta' x} = p(\beta, x) = 1 - q(\beta, x), \tag{1}$$

where β is a vector of unknown covariates. The conventional logistic regression estimate of β is the maximizer of

$$\prod_{sample} p(\beta, x_i)^{Y_i} q(\beta, x_i)^{1-Y_i}. \tag{2}$$

This method will result in biased estimators of regression parameters if it is applied to truncated data.

The TLR approach conditions on the probability that an accident is observed which is the probability that it results in at least one fatality. This has the effect of introducing a divisor to logistic regression likelihood equation 2. The truncated logistic regression estimator of β is the maximizer of

$$\prod_{accidents} \frac{\prod_{i \in accident\, j} p(\beta, x_i)^{Y_i} q(\beta, x_i)^{1-Y_i}}{1 - \prod_{i \in accident\, j} q(\beta, x_i)} \tag{3}$$

This modification of the logistic regression likelihood equation 2 gives a well behaved estimator which has all the usual desirable properties of maximum likelihood estimators.

The conditional logistic regression estimator is obtained by conditioning on the exact number of deaths in an accident. In the example given by Lui et al. (1988) only accidents with two occupants where exactly one death occurred were used. The conditional logistic likelihood estimate of β is the maximizer of

$$\prod_{accidents\, j} \frac{\exp \beta' S_j}{\sum_{i \in accident\, j} \exp \beta' x_i} \tag{4}$$

where $S_j = \sum_{deaths\, i \in accident\, j} \beta' x_i$ is the sum of the covariates of the individuals who die in accident j.

The method of Evans (1985) is not a regression technique. For two levels of a factor of interest, it compares the relative frequency of deaths of each level to a specified control group in another seating position in the vehicle.

2.2 Ordinal Models

Of the three methods discussed above only the truncated logistic regression likelihood of equation 3 extends naturally to ordinal data. The arguments that can be advanced for the use of the conditional logistic regression method in the binary case fail to hold in the ordinal case. A full discussion of the Truncated Ordinal Regression is given in O'Neill & Barry (1993a) which is attached as Appendix A.

The ordinal response variable is assumed to have $k+1$ levels and the case $k = 1$ corresponds to binary data. An example where $k = 3$ would be a four point scale for injury:

1. No injury
2. Injury
3. Died after hospitalization
4. Died at scene

The data is said to be group truncated if the responses for a group are only known if at least one of the group attained a specified level, j say. In the above example the cutoff might be $j = 3$ in which case the injury levels are only recorded if at least one person in the accident dies. The truncated ordinal regression (TOR) likelihood is the natural generalization of equation 3. The method allows for different relationships between the covariates to the logistic link given in equation 1.

2.3 Theoretical Relative Merits of the Methods

Since the TLR uses the full information from the sample it can be expected to lead to the most accurate inference. The following general properties hold.

- TLR will give the best estimates of effects, such as for example seat belt usage or age of the occupant, followed by CLR and then DPC. TLR will also give the most powerful hypothesis tests. This is because TLR uses all the information that is available in the data. CLR can only use comparisons within a vehicle. For example the fact that a female died in one car while a male didn't die in exactly the same circumstances in another car does not contribute any information to the CLR estimate. Also cars in which all occupants die contribute no information to the CLR estimate. Both of these scenarios would add to the TLR information. DPC only uses the data which satisfies a condition on a single control variable and so will normally be less precise.

- More effects can be fitted using TLR. The conditional logistic regression likelihood equation 4 only includes terms which vary within a given accident. For example, for single vehicle accidents, since the speed of the car is constant for all the occupants, its effect on the survival prospects cannot be estimated using CLR. TLR on the other hand can be used to estimate its effect. CLR cannot estimate the effect of variables which do not vary within accidents.
- Only TOR can be used to estimate the relative seriousness of crashes for occupants. The TLR method allows us to estimate the probability that a given type of crash will kill a given type of occupant. The TOR method enables the estimation of the probabilities of the various categories of injury.
- Only TLR can be used to estimate the total number of potentially fatal crashes. The TLR method allows us to estimate the probability that a particular configuration of factors results in a fatality. By dividing the observed number of crashes of this type by this probability we obtain an estimate of the total number of potentially fatal crashes of this type. The estimates can then be summed over the categories of crashes to obtain an estimate of the total number of potentially fatal crashes.
- Unlike CLR, TOR can be generalized to different link functions. Various researchers have found that the logistic link given in equation 1 does not work well when dealing with very rare events. The TOR method allows us to choose the link function which best fits a given data set.

3 Software

3.1 Technical Issues

3.1.1 Introduction

This section describes installation and the technical operation of the software in greater detail. The technical detail is included for users who may wish to modify the software for a particular purpose, or are having installation problems. It is assumed that users wishing to modify the routines will have a sound knowledge of Splus and C, and will be familiar with the issues involved in fitting truncated ordinal regressions. The general reader may wish to skip to the installation section.

3.1.2 General description of routines

The programs are written to take advantage of the Splus enviroment for data analysis with the efficiency and control of C code. The roles are divided as follows:

The routines use the Splus language to allow the specification of the model and to ensure that the specification is consistent with the assumptions of the model. Thus it uses the built in functionality of Splus to construct factors/contrasts and hence the design matrix.

Splus is notoriously poor at performing iteration and in freeing memory after function calls. Hence the calls to C are used when these factors would come into play. These calls are characterised by the fact that they have been written at the lowest level possible, in the sense that Splus performs as much of the work as is sensible/possible before passing to the C code. The calls to C from Splus are

- formXblocks: This takes a (model) matrix and the number of levels of the response and returns a matrix correctly *blocked for the proportional odds model*. For example if the model matrix produced by Splus is:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

  with 3 (as 1 is redundant) levels of response to be parametrised the function would return (as 1 is redundant)

$$\begin{bmatrix} 1 & 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 1 & 2 & 3 & 4 \\ 1 & 0 & 5 & 6 & 7 & 8 \\ 0 & 1 & 5 & 6 & 7 & 8 \end{bmatrix}$$

- formZ: This takes a vector of the responses (assumes integer levels (ie 1,2,3 .. k)) and returns the response vector needed by the fitting func. For example if the possible response levels are 1-3 *then if* :

$$response = \begin{bmatrix} 1 & 3 & 2 \end{bmatrix}$$

  returns

$$zmat = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

- fitfunc: This performs the fisher scoring. It takes the design matrix, the response vec, the vector of groups, the truncation point and an *initial estimate of the parameters*, and iterates until convergence.

The result is that the C routines are effectively support routines to *Splus*, and *should not normally* be modified. As an example say there is a problem in the convergence of a model that is being fitted. This problem should not lie in the C code as the function it performs is fixed, in that all matrices have been specified. If they are passed correctly everything should work. The problem must lie with the data that is being passed to the C functions. For instance the fitting func assumes that the design matrix is full rank. If it is not, the problem *is arising on the Splus side* and the Splus function should be modified to ensure that the matrix is full rank.

3.1.3 Routines

The *Splus code* has been documented to some extent and should be faily self explanatory. Copies of the help files for *trunc.fit* and *trunclm.object* are included Appendix C. The Splus functions that are needed are:

- init.beta: This performs an untruncated logistic regression to find initial values for the fisher scoring algorithm.

- no.groups: returns the number of unique elements in a vector.

- setup.prop: construct model matrix and response vector for proportional odds model and calls init.beta.

- trunc.fit: Performs the actual fitting.

The following routines use some of the inheritance mechanisms in Splus. This implementation is fairly basic, but is set up to allow the use of the *summary()* function in Splus. The setup is as follows:

1. *trunc.fit()* produces an object of class "trunclm".

2. *print.summary.trunclm* and *summary.trunclm* are implemented along the lines of those for lm to produce similar output.

These functions can be found as source in *GTLRfuncs.S*.
The C code is in 5 files:

- matmult.c : defines all the matrix routines.

- matmult.h : header file for the C routines.

- fitfunc.c : defines all functions used in the fitting.

- xblock.c : defines functions to produce the design matrix and response vector.

- minv.f : invert matrix. (note this is a Fortran routine).

There is also a makefile listing the dependencies. The only file that should be modified is matmult.h. In this file the macro variable *maxgroups* can be modified to satisfy whatever space requirements you may have. For instance lowering maxgroups lowers the base (ie fixed) amount of space the routines take up. Note that the routines also use dynamic allocation and are not protected against out of memory signals from allocation requests.
The help functions are in the files:

- init.beta.d

- no.groups.d

- setup.prop.d

- trunc.fit.d

- trunclm.object.d

These files are copied to the *.Help* sub-directory that is being used, without the .d extension.

3.2 Installation

Installing the software involves three steps.

1. Compiling the source into the object file *fittrunclm.o* and placing it in the appropriate directory. This directory is either the directory that Splus is run from(which from here will be referred to as "Sdir") or an appropriate library.
2. Sourcing the Splus functions (using the Splus function *source()* to the *.Data* sub directory of *"SDir"/.Data* , or to a directory that is attached (using the Splus command *attach()*).
3. Copying the help files (without the .d postfix) to the .Help sub directory of whichever directory the Splus functions were placed in.

Installation should proceed as follows:

- Unix

  The file *truncpack.tar* is a tar file that has two components.

  1. *truncfit.tar* contains the code/functions.
  2. *truncfit.install* contains a simple shell script to install functions into a specified directory, that is, the directory that Splus will run from.

  Install as follows.

  1. Decide on the directory that Splus is to be run from, call it "Sdir". The directories *"Sdir/.Data"* and *"Sdir/.Data/.Help"* MUST exist.
  2. Unpack truncpack.tar by executing

     tar xf truncpack.tar

     this should extract the two files described above
  3. Run the C script by typing

     truncfit.install "Sdir"

     where Sdir is defined as in (1).

  This should compile the code and copy the help and Splus functions to the appropriate directories. If you wish a different setup simply modify *truncfit.install.*

  As an example, say you wish to install the software in the directory

  /home/stat1/barstat

  and you are presently in

  /home/stat1/barstat/treport/testarea

  all you need to do is move *truncpack.tar* to your present directory and execute

  tar xf truncpack.tar

  truncfit.install /home/stat1/barstat

This will install it provided */home/stat1/barstat/.Data* and */home/stat1/barstat/.Data/ Help* exist.

The other option is to install the code in a Splus library, "Slib", say. This is more complicated. To do this use

truncfit.install "Slib"

You will then have to modify *trunc.fit()* or the function *.First.lib()* to ensure that the object file *fittrunclm.o* is dynamically loaded. See Splus help for details on dynamically loading from libraries.

When you run Splus you may need to run the function *help.findsum(".Data")* to use the help facility.

- Other systems

  For other systems the files are packaged individually, and ALL functions included as text files

  You will need to:

  1. Produce *fittrunclm.o*. This will be compiler dependent, but will consist of compiling each of the source files to produce object files and then linking these together. See the documentation for your machine for details.
  2. Copy the help files listed above to the *"SDir"/.Data/.Help directory*, dropping the d postfix. See the Splus function *prompt()*.
  3. Start Splus and use the function *"source()"* to parse the Splus functions from GTLR-funcs.S into the working directory.

The makefile used for the compilation and the truncfit.install script are printed in the appendices. These should give a general idea of what goes on, and how to extract files independently.

3.3 Software features

During the testing of the software the major problem that was confronted was the situation where the data is sparse and as a result the model is not well specified. In this case the Fisher scoring algorithm for the parameters will not converge (although the fitted values for the probabilities will). This can be diagnosed by examining the path of the log likelihood as output by *trunc.fit()*. If these estimates fail to converge then certain parameters are tending to infinity. The particular parameters that are extreme provide information regarding the terms that are leading to the problem.

The problem occurs in this binomial regression context usually due to the sparseness of data for certain combinations of levels of factors. For example if there is only one observation at a particular level of a factor, then the parameter estimate for this level will go to + or - infinity, so that the data fits the model exactly. Note that the specification of the logistic model causes the problem to cease as the size of the data set grows

One way around this is to choose levels of factors such that it does not happen. The problem with this is that the recursive nature of this approach is incompatible with the assumptions needed for the asymptotic behaviour of the MLE estimators. Hence any inference that is made must be interpreted carefully.

Another problem that may arise is that Splus may produce design matrices that are not of full rank. In this case certain parameters are not identifiable, This problem can arise for various reasons, but the most usual is from using crossed/nested terms. In this case if some levels of a factor do not occur at levels of another factor Splus will still incorporate this nested parameter into the model matrix. If this problem arises it is recommended to rewrite the function to modify (ie remove non-identifiable columns) the design matrix before it is passed to the C routines. The use of *browser()* and *solve(t(design.mat) % * % design.mat)* to detect singularities is recommended.

If the routine continually crashes on large datasets it could be a problem with memory. The file *matmult.h* contains the macro variable *maxgroups* which should be modified and the the routines re-compiled.

4 Examples

4.1 Introduction

In this section the use of the software is demonstrated. Reference should be made to the help files and technical documentation included in the appendices. For pedagogical reasons, the examples are very detailed. There are of course alternative (and more preferable) ways to perform the data manipulations. All references to Splus functions and output is in *italics*. The data set used in the examples is the same as in the online help for the function *trunc.fit()*. For information on the fitting of statistical models in Splus see Chambers & Hastie (1992).

4.2 Example 1

Consider the following data:

```
>response [1] a c c a b a a b a b c a b c b a a b c b
> belt
[1] 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 1 0 0 0
> age
[1] 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2
> group
[1] 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
>
```

where all data is either simple numeric or simple character

We may consider this to be a sample of fatal accidents where *belt* is an indicator for whether the person was wearing a seatbelt (1=yes, 0=no) and *age* is their age. The vector *group* determines to which accident the individual belongs. The *response* variable is coded as:

1. a = uninjured
2. b = injured
3. c = killed

Assume that the data was observed given that at least one individual died in the accident. If we wish to fit the truncated model to this data we must manipulate it into the correct Splus formats for *trunc.fit()*. Suppose that for the above two accidents we only want to model the probability of being killed in the accident. Then the model is a truncated logistic regression model, and by definition the response is either killed, or not killed. Hence we must modify the vector response by assigning all observations with a response different to killed to another factor as follows:

```
> response2 <- response
> index <- response2!= "c"
> response2[index] <- "not killed"
> response2
[1] "not killed" "c" "c" "not killed" "not killed"
[6] "not killed" "not killed" "not killed" "not killed" "not killed"
[11] "c" "not killed" "not killed" "c" "not killed"
[16] "not killed" "not killed" "not killed" "c" "not killed"
```

To logically fit a truncated ordinal model the response must be ordered. We thus use

```
> response2 <- ordered(response2,levels=c("not killed", "c"))
> response2
[1] not killed c c not killed not killed not killed
[7] not killed not killed not killed not killed c not killed
[13] not killed c not killed not killed not killed not killed
[19] c not killed
not killed < c
```

To generate the ordered factor. If belt is an indicator variable for the use of a seat belt we obviously want to fit it as a factor. If we also wish to use treatment contrasts in the construction of the design matrix we can do this as follows:

```
> beltf <- factor(belt)
> class(beltf)
[1] "factor"
> beltf <- C(beltf,treatment)
```

For information on the use of factors and contrasts see the online help or Chambers & Hastie (1992).

We are now in a position to fit the model. Examining the documentation we see that the first argument to *trunc.fit()* is a formula. In this example we are interested in modelling response2 in terms of beltf and age. We express this as

```
response2 ~beltf+age
```

The second argument is a data frame. As we do not have one at this point we must construct it. The function *trunc.fit()* expects that all variables in the model formula will be on this frame, as well as the variable identifying which group it belongs to. We thus use

```
> exam1.frame <- data.frame(response2,beltf,age,group)
```

The next argument is the tolerance. We will, as an example, set the tolerance to .000001. By examining the convergence of the loglikelihood (output by *trunc.fit()*) we can asses if this is sufficient. The next argument is the number of iterations that we will run for before terminating. If the model is well specified convergence is rapid (in our experience in 4-8 iterations). We will set this to 20. The next argument is groupvar. In this case the name of the group variable is in fact "group", and so we use this as the argument. The next variable is trunc, the point at which the the distribution is truncated. In this case the data were only observed given that some one died in the accident. Being killed is coded as "c" in the response vector, or is the second level of the response. Thus the truncation point is level 1, or "not killed". Examining the help documentation we can use either trunc=1 or trunc="not killed".

The last parameter is ,the initial value of beta. This parameter is optional so we will allow the routine to choose its own starting value. We thus use

> exam.fit1 < - trunc.fit(formula = response2 ~ beltf + age , data = exam1.frame , tolerance = .000001 , iter=20 , groupvar = "group", trunc = "not killed")

-8.890799

-8.882969

-8.882967

-8.882967

Note the convergence of the log likelihood. Examining the fitted model:

> exam.fit1

$call:

trunc.fit(formula = response2 beltf + age, data = exam1.frame, tolerance = 1e-06, iter = 20, groupvar = "group", trunc = "not killed")

$fitted: [1] 0.4689223 0.5189034 0.7008293 0.7911075 0.8213461 0.907688 0.922616

[8] 0.9504274 0.9761095 0.3961386 0.5153058 0.6985921 0.789545 0.858243

[15] 0.9069980 0.9400892 0.9618748 0.9683336 0.3916230 0.515305

$variance: [,1] [,2] [,3]

[1,] 1.3165525 0.3729987 0.22590995

[2,] 0.3729987 1.4752376 -0.10944208

[3,] 0.2259099 -0.1094421 0.08691102

$x: [,1] [,2] [,3]

[1,] 1 -1 -1

[2,] 1 0 -2

[3,] 1 -1 -3

[4,] 1 -1 -4

[5,] 1 0 -5

[6,] 1 -1 -6

[7,] 1 0 -7

[8,] 1 0 -8

```
[9,] 1 -1 -9
[10,] 1 0 -1
[11,] 1 0 -2
[12,] 1 -1 -3
[13,] 1 -1 -4
[14,] 1 -1 -5
[15,] 1 -1 -6
[16,] 1 -1 -7
[17,] 1 -1 -8
[18,] 1 0 -9
[19,] 1 0 -1
[20,] 1 0 -2
$coefficients:
model.coeff beltf age
-0.7919592 -0.281752 -0.4737266
$z:
[1] 1 0 0 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 0 1
$linear.pred: [,1]
[1,] -0.03648064
[2,] 0.15549389
[3,] 0.91097248
[4,] 1.38469904
[5,] 1.57667357
[6,] 2.33215216
[7,] 2.52412669
[8,] 2.99785325
[9,] 3.75333184
[10,] -0.31823267
[11,] 0.15549389
[12,] 0.91097248
[13,] 1.38469904
[14,] 1.85842560
[15,] 2.33215216
[16,] 2.80587872
[17,] 3.27960528
[18,] 3.47157981
```

```
[19,] -0.31823267
[20,] 0.15549389
$log.lik:
[1] -8.882967
$iterations:
[1] 4
$tolerance:
[1] 3.10811e-07
$frame: response2 bellf age group
1 "not killed" "1" "1" "1"
2 "c" "0" "2" "1"
3 "c" "1" "3" "1"
4 "not killed" "1" "4" "1"
5 "not killed" "0" "5" "1"
6 "not killed" "1" "6" "1"
7 "not killed" "0" "7" "1"
8 "not killed" "0" "8" "1"
9 "not killed" "1" "9" "1"
10 "not killed" "0" "1" "1"
11 "c" "0" "2" "2"
12 "not killed" "1" "3" "2"
13 "not killed" "1" "4" "2"
14 "c" "1" "5" "2"
15 "not killed" "1" "6" "2"
16 "not killed" "1" "7" "2"
17 "not killed" "1" "8" "2"
18 "not killed" "0" "9" "2"
19 "c" "0" "1" "2"
20 "not killed" "0" "2" "2"
attr(, "class"):
[1] "trunclm"
```

A more convenient summary is found by

```
> summary(exam.fit1)
Call: trunc.fit(formula = response2 ~ bellf + age, data = exam1.frame, tolerance
= 1e-06, iter = 20, groupvar = "group", trunc = "not killed")
Residuals:
```

Min 1Q Median 3Q Max

-0.8582 -0.07999 0.06865 0.2093 0.6039

Coefficients:

	Value	Std.Error	zvalue	Pr(> |z|)
model.coeff	−0.7920	1.1474	−0.6902	0.4901
beltf	−0.2818	1.2146	−0.2320	0.8166
age	−0.4737	0.2948	−1.6069	0.1081

Correlation:

	model.coeff	beltf	age
model.coeff	1.0000	0.2676	0.6678
beltf	0.2676	1.0000	−0.3056
age	0.6678	−0.3056	1.0000

Log likelihood:

[1] -8.883

The parameters have there usual logistic regression interpretation as log odds ratios, keeping in mind the following.

- We are modelling the probability of the response being less than a certain level. In the logistic case, if p is the probability of death, we are in fact modelling *1-p*.
- The design matrix is constructed based on the negative of the covariate design matrix See Appendix E.
- The interpretation of parameters relating to factors will depend on the contrasts used.

For this example *beltf* was incorporated into the model using treatment contrasts. The odds of dying for an individual at the top level is then

$$odds\ for\ level\ 1\ of\ beltf = \frac{probability\ die\ given\ level\ 1}{probability\ don't\ die\ given\ level\ 1}$$

We are modelling the probability of not dying so

$$probability\ don't\ die\ at\ level\ 1 = \frac{exp(C + .2818)}{1 + exp(C + .2818)}$$

Where C depends on the other factors, and the negative value arises due to the negative value of the design matrix. Thus

$$odds\ for\ level\ 1\ of\ beltf = \frac{\frac{1}{1+exp(C+.2818)}}{\frac{exp(C+.2818)}{1+exp(C+.2818)}}$$

$$= exp(-C - .2818)$$

Similarly

$$odds\ for\ level\ 0\ of\ beltf = exp(-C)$$

So it follows that

$$odds\ ratio = \frac{exp(-C - .2818)}{exp(-C)} = exp(-0.218)$$

So

$$log(odds\ ratio) = -0.218$$

ie individuals with *beltf* at level 1 have a lower probability of dying.

4.3 Example 2

In this example we will fit the full ordinal model to the data from example 1. We thus need to construct a new data frame. Assuming *beltf* still exists and *response* holds the responses we use

```
> response < −ordered(response)
> exam2.frame < −data.frame(response,beltf, age, group)
```

We can now call the *trunc.fit()*. Assuming the data was only observed if someone died in the group the truncation point is "b", or 2. We thus use

```
> exam.fit2 < −trunc.fit(formula = response  beltf + age, data = exam2.frame,
tolerance = 1e-06, iter = 20, groupvar = "group", trunc = "b")
+ -18.357277
-18.002097
-17.992586
-17.992529
-17.992528
-17.992528
-17.992528
-17.992528
-17.992528
```

It has obviously converged with this tolerance. As the complete output would be long and messy we use *summary()*.

```
> summary(exam.fit2)
Call: trunc.fit(formula = response  beltf + age, data = exam2.frame, tolerance
= 1e-06, iter = 20, groupvar = "group", trunc = "b")
Residuals:
Min 1Q Median 3Q Max
-0.9146 -0.3775 0.05862 0.2283 0.6868
```

		Value	Std.Error	zvalue	Pr(¿ |z|)
	model.coeff	−2.6797	1.1854	−2.2607	0.0238
Coefficients.	model.coeff	−0.6199	1.0261	−0.6041	0.5458
	beltf	−1.6803	0.9429	−1.7821	0.0747
	age	−0.2737	0.1764	−1.5517	0.1207

Correlation:

	model.coeff	model.coeff	beltf	age
model.coeff	1.0000	0.7898	0.5132	0.7482
model.coeff	0.7898	1.0000	0.3758	0.6872
beltf	0.5132	0.3758	1 0000	0.0182
age	0.7482	0.6872	0.0182	1.0000

```
Log likelihood:
[1] -17.99
```

Note that there are now two intercepts, corresponding to the three levels of the factor.

5 Summary

The aim of the seeding grant was to provide a computer implementation of the new statistical procedure Truncated Ordinal Regression. This report has described its successful implementation in the Splus package using a suite of C routines. The software should make the technique generally available to road traffic researchers who have access to the package Splus, either on UNIX or in Windows. The advantages of combining Splus and C routines are:

- The Splus model language is used for the design aspects of the regression problem.
- The Splus generic functions can be used to extend the functions and make them user friendly.
- The computationally intensive aspects of the procedure can be relegated to C.
- Because of the division, the resulting software is both very efficient and very rich in the types of model structures that can be fitted.
- The speed of the software is such that it can be used as a primitive in such procedures as stepwise model selection, bootstrap and non-parametric modelling

Now that a package is available to perform TOR, it can be applied to a variety of road traffic accident databases by interested researchers. It will improve the accuracy of the estimates and conclusions and allow more general questions to be posed.

Reference

Chambers, M. C., & Hastie, T. J. (1992) *Statistical models in S.* Wadsworth & Brooks/Cole, California.

Evans, L. (1985). Double pair comparisons - a new method to determine how occupant characteristics affect fatality risk in traffic crashes. *Accident Analysis and Prevention*, *18*, 217–227.

Lui, K. J., McGee, D., Rhodes, P., & Pollack, D. (1988). An application of conditional logistic regression to study the effects the effects of safety belts, principal impact points and car weights on drivers fatalities. *Journal of Safety Research*, *19*, 197–203.

O'Neill, T. J., & Barry, S. (1993a). Group truncated ordinal regression. In preparation.

O'Neill, T. J., & Barry, S. (1993b). Truncated logistic regression. Submitted.

Appendices

A Truncated ordinal regression.

B Truncated logistic regression.

C Splus help files.

D Splus source listing.

E Install script.

F Makefile.

Appendix A

Group Truncated Ordinal Regression

by

Terence J O'Neill

and

Simon Barry

Department of Statistics, The Faculties,
Australian National University,
GPO Box 4, Canberra, ACT 2601, Australia

Abstract

O'Neill (1992) proposed truncated logistic regression as an alternative to conditional logistic regression (Breslow and Day, 1980) which has previously been used for the analysis of truncated binary data (Lui, Rhodes and Pollock, 1988). This paper extends truncated logistic regression to truncated ordinal regression. This is important since conditional logistic regression does not extend to ordinal data. *

* Keywords: Truncation, Conditional, Logistic, Ordinal.

1. Introduction

This paper considers the modelling of ordinal data from groups subject to truncation. As an example consider an ordinal scale for injuries sustained in a motor vehicle accident:

1 – no injury

2 – injury

3 – died subsequent to accident

4 – died immediately,

and suppose that data is only available on accidents involving a fatality. Then the ordinal responses for all individuals in a given accident are observed if and only if the maximum response over the group is at least 3. We call such data, group truncated ordinal data. For binary data only group truncation is meaningful since ordinary truncation would imply degenerate data. O'Neill & Barry(1992) recently proposed truncated logistic regression as an alternative to conditional logistic regression which has previously been used for truncated binary data (Lui *et al.*, 1988). The aim of this paper is to extend the methods of O'Neill (1992) to ordinal data. This is important since the conditional logistic model does not extend to either different link functions or true ordinal data.

In section 2 conditional logistic regression is briefly reviewed and it is shown that it depends crucially on the assumption of a logistic link and that it does not extend to ordinal data. In section 3 the estimates for group truncated ordinal regression are derived.

2. Conditional Logistic Regression

In this section we review conditional logistic regression and its possible extension to other link functions and ordinal regression. In conditional binary regression, the conditioning event is the number of responses greater than 1 where 1 indicates a null response and 2 a positive response. In general, for ordinal data with ordered responses $1,\ldots,k+1$ and a group of size n, the conditioning event would be the number of responses $C = c$ which are greater than ℓ where $\ell \leq k$. Now if $\Re$ denotes the set of individuals in the group, then

$$P(C=c)=\sum_{\Im\in\Re_{n-c}}\left\{\prod_{\Im}\gamma_{i\ell}/(1-\gamma_{i\ell})\right\}\prod_{\Re}(1-\gamma_{i\ell})$$

where $\Re_{n-c}$ is the set of all subsets of $\Re$ of size n - c and $\gamma_{i1},\ldots\gamma_{ik}$ is the set of cumulative probabilities of categories $1,\ldots,k$ for individual i. ie γ_{ij} is the probability that individual i's response is less than or equal to category j.

Following McCullagh and Nelder (1989) we consider link functions of the form

$$g(\gamma_{ij})=\eta_{ij}=\theta_j-\beta_1' x_i,$$

where x_i is the covariate measured on individual i. Then if g has inverse h,

$$P(C=c)=\sum_{\Im\in\Re_{n-c}}\left\{\prod_{\Im}h(\eta_{i\ell})/(1-h(\eta_{i\ell}))\right\}\prod_{\Re}(1-h(\eta_{i,\ell})),$$

and if y is the set of observed levels,

$$P(y/C=c)=\frac{\prod_{\Re}\left\{h(\eta_{i,yi})-h(\eta_{i,yi-1})\right\}}{\prod_{\Re}\left\{1-h(\eta_{i,\ell})\right\}\sum_{\Im\in\Re_{n-c}}\left\{\prod_{\Im}h(\eta_{i,\ell})/(1-h(\eta_{i,\ell}))\right\}}. \qquad (2.1)$$

Now if $k = 1$ and g is the logistic link, then we have conditional logistic regression and

$$P(y / C = c) = \frac{exp\left((n-c)\theta_1 - \beta_1 \sum_{\Im_{n-c}} x_i\right)}{\sum_{\Im \in \Re_{n-c}} exp\left((n-c)\theta_1 - \beta_1 \sum_{\Im} x_i\right)}$$

$$= exp\left(-\beta_1 \sum_{\Im_{n-c}} x_i\right) / \sum_{\Im \in \Re_{n-c}} exp\left(-\beta_1 \sum_{\Im} x_i\right). \tag{2.2}$$

Where $\Im_{n-c} = \{i : y_i \leq \ell\}$,the set of reponses less than or equal to ℓ.

The fact that the group level effect θ_1 does not appear in (2.2) is the primary reason for the popularity of conditional logistic regression. However the removal of the group level effect only occurs for the logistic link and for $k = 1$. It is a simple matter to check that the θ's do not cancel from (2.1) for $k > 1$, even for the logistic link. To check that cancellation implies the logistic link for $k = 1$, take $f = h/(1-h)$. Then cancellation must apply for groups of size 2 where the individual with response 1 has $x = 0$. So

$$f(\theta) / \{f(\theta) + f(\theta + \eta_1)\} = m(\eta_1)$$

where $m()$ is some function and $\eta_1 = \beta_1 x$ for the individual dead, or

$$f(\theta + y) = f(\theta) m_1(y).$$

where $m_1()$ is a function of $m()$. Taking $\theta = 0$, we obtain

$m_1(y) = f(y) / f(0)$ and $f(\theta) f(y) / f(0)$.

This has solution $f(\eta)=\exp(a+b\eta)$ for a and b which implies that the link function is logistic.

In generalized linear modelling, we are used to the inference being fairly robust to the link function, so it is a matter of some concern that the inclusion of group level effects in the conditional likelihood is determined by whether or not the logistic link function is assumed. Also, even if the logistic link is assumed, the property of no group level effects in the conditional likelihood is not preserved when we split categories.

Because of these problems with extending conditional logistic regression to ordinal responses and other link functions, it is necessary to consider the extension of truncated logistic regression to group truncated ordinal data.

3. Group Truncated Ordinal Regression

Since the likelihood equations for group truncated ordinal regression are very similar to those for conventional ordinal regression, we begin with a brief summary of the standard results presented in McCullagh and Nelder (1989). We suppose that for each of N distinct experimental situations, there are cumulative frequencies $z_{i1},\ldots,z_{ik}$ for the m_i individuals observed. So z_{ij} is the number of individuals with response at most j and $E(z_{ij}) = \gamma_{ij}$ Following McCullagh and Nelder (1989), we consider link functions of the form

$$g(\gamma_{ij})=\eta_{ij}=\theta_j-\beta_i' x_j.$$

where x_i is the p x 1 vector of covariates for observation i.

Let

$$D_i=\operatorname{diag}\left(\frac{\partial\gamma_{ij}}{\partial\eta_{ij}}\right)=\operatorname{diag}\left(1/g'(\gamma_{ij})\right)$$

where $\operatorname{diag}(a_{ij})$ refers to a $k\times k$ diagonal matrix with jth diagonal element a_{ij}.

$$=\operatorname{diag}\left(\gamma_{ij}(1-\gamma_{ij})\right).$$

for the logistic link and

$$\Gamma_i = m_i\left(\gamma_{ij}\left(1-\gamma_{ij'}\right)\right),\ j \le j'$$

where Γ_i is symmetric. Then Γ_i has a tri–diagonal inverse with entries

$$\Gamma_i^{jj} = \frac{\gamma_{i,j+1}-\gamma_{i,j-1}}{\left(\gamma_{i,j+1}-\gamma_{ij}\right)\left(\gamma_{ij}-\gamma_{i,j-1}\right)}, \qquad j=1,\ldots,k$$

$$\Gamma_i^{j,j+1} = -\left(\gamma_{i,j+1}-\gamma_{i,j}\right)^{-1}, \qquad j=1,\ldots,k-1.$$

where $\Gamma_i^{a,b}$ refers to the element in row a, and column b, of Γ_i.

Then for the sample of size N, let

$$D = B\text{diag}(D_i), \qquad \Gamma = B\text{diag}(\Gamma_i)$$

$$M = \text{diag}(m_i)\otimes I, \qquad z' = \left(z_1',\ldots,z_N'\right)$$

$$\gamma' = \left(\gamma_1',\ldots,\gamma_N'\right), \qquad X_i = \left(I_k, -e_k x_i'\right),$$

Where Bdiag(a_i) is a block diagonal matrix, with ith block a_i, I_k is the k $\times$ k identity matrix, and e_k is a k $\times$ 1 column vector of 1's,

and

$$X' = \left(X_1',\ldots,X_N'\right). \tag{3.1}$$

If

$$\beta' = \left(\theta_1,\ldots,\theta_k,\beta_1'\right)$$

then the score functions are

$$D_\gamma \ell = M\Gamma^{-1}(z - M\gamma)$$

and

$$D_\beta \ell = X'DM\Gamma^{-1}(z - M\gamma)$$

where D_β is the vector differential operator $D_\beta f = \left[\partial f / \partial \theta_1, \partial f / \partial \theta_2, \; \ldots\ldots \; , \partial f / \partial \beta_p\right]'$. The Fisher information is

$$\Im(\beta) = X'DM\Gamma^{-1}\, MDX.$$

So the Fisher scoring method is

$$(X'WX)\left(\hat{\beta}_{NEW} - \hat{\beta}_{OLD}\right) = X'W(MD)^{-1}(z - M\gamma)$$

where

$$W = DM\Gamma^{-1}MD$$

or if $\eta = X\beta$, it is a weighted linear regression of

$$\hat{\eta} + (MD)^{-1}(z - M\gamma) \text{ on } X \text{ with weights } W.$$

Subsequently in this discussion we will assume that $M = I$, that is we do not aggregate over individuals. Consider for the moment a single group subject to truncation and for clarity suppress the i subscript. Then the likelihood for the observed group is

$$L_T(\beta,X) = L(\beta,X) / \left(1 - \prod_{\Re} \gamma_\ell\right) = P(\beta,X)^{-1} L(\beta,X)$$

where $L(\beta,X)$ is the unconditional likelihood and

$$P(\beta,X) = 1 - \prod_{\Re} \gamma_\ell = 1 - Q(\beta,X)$$

is the probaility of observing the group. Now

$$D_\beta \, log \, P(\beta,X) = -\left(D_\beta \prod_{\Re} \gamma_\ell\right) / P(\beta,X)$$

$$= -\{Q(\beta,X)/P(\beta,X)\} \sum_{\Re} D_\beta \gamma_\ell / \gamma_\ell$$

$$= -\{Q(\beta,X)/P(\beta,X)\} \sum_{\Re} \frac{\partial \gamma_\ell}{\partial \eta_\ell} D_\beta \eta_\ell / \gamma_\ell$$

$$= -\{Q(\beta,X)/P(\beta,X)\} X' D E_T \gamma^{-1}$$

where $E_T = I \otimes E_{\ell\ell}$ where $E_{\ell\ell}$ is the $k \times k$ matrix with one in the ℓ,ℓ position and zeros elsewhere and I has dimension the number in the group. Also γ^{-1} is the vector of inverses of the elements of γ. So the score function for the group is

$$X'D\left\{\Gamma^{-1}(z-\gamma) + \frac{Q(\beta,X)}{P(\beta,X)} E_T \gamma^{-1}\right\}$$

$$= X'D\Gamma^{-1}\left\{z - \gamma + \frac{Q(\beta,X)}{P(\beta,X)} \Gamma E_T \gamma^{-1}\right\}$$

But the jth element of the vector

$$\left\{z-\gamma+\frac{Q(\beta,X)}{P(\beta,X)}\Gamma E_T\gamma^{-1}\right\}_j=\begin{cases} z_j-\dfrac{\gamma_j}{P(\beta,X)}+\dfrac{Q(\beta,X)}{P(\beta,X)}\dfrac{\gamma_j}{\gamma_\ell} & j\le\ell \\ z_j-\dfrac{\gamma_j}{P(\beta,X)}+\dfrac{Q(\beta,X)}{P(\beta,X)} & j>\ell \end{cases}$$

Now for $j \le \ell$

$$E(z_j \mid \text{observed}) = Pr(y\le j,\text{observed})\,/\,Pr(\text{observed})$$

$$= \gamma_j\left(1-Q(\beta,X)/\gamma_\ell\right)/\,P(\beta,X) = \gamma_j\,/\,P(\beta,X)-\frac{Q(\beta,X)}{P(\beta,X)}\frac{\gamma_j}{\gamma_\ell}$$

and for $j > \ell$

$$E(z_j \mid \text{observed}) = Pr(y> j,\text{observed})\,/\,Pr(\text{observed})$$

$$= 1-Pr(y>j)\,/\,Pr(\text{observed})$$

$$= 1-\left(1-\gamma_j\right)/\,P(\beta,X)$$

$$= \gamma_j\,/\,P(\beta,X)-Q(\beta,X)\,/\,P(\beta,X).$$

So the score statistic for β is

$$U_T(\beta,X)=U_T=X'D\Gamma^{-1}(z-\mu_T) \qquad (3.2)$$

where μ_T is the mean of an observed z given that it is subject to truncation. Now if we use Fisher scoring to estimate β, then we require

$$\Im(\beta)=-E\left(\frac{\partial^2\log L_T}{\partial\beta\partial\beta'}\right).$$

But

$$\Im(\beta) = E\left(u(\beta)u(\beta)'\right)$$
$$= X'D\Gamma^{-1}V_T\Gamma^{-1}DX \tag{3.3}$$

where $V_T(\beta, X) = V_T = \text{Var}(z \mid \text{observed})$. The diagonal blocks of this variance matrix contain the covariance of the response vector within an individuals response. These are analagous to the Γ matrix in the non-truncated case. Now for individual i ,since for $j \le j'$

$$E\left(z_j\, z_{j'} \mid \text{observed}\right) = E\left(z_{j'} \mid \text{observed}\right),$$

it follows that the $j\, j'$ entry of the diagonal blocks of V_T, for $j \le j'$ is

$$\mu_{T_j}\left(1 - \mu_{T_{j'}}\right)$$

In a departure from the non-truncated model, the truncation causes the responses between individuals in a group to be correlated. As defined previously let $z_{i1}, .., z_{ik}$ be the response for individual i in the group. Thus for two individuals, i and j , calculations analagous to those above lead to:

for $a \leq \ell, \quad b \leq \ell$

$$E\left(z_{i,a} z_{j,b} / observed\right) = \frac{\gamma_{i,a}\gamma_{i,b}}{P(\beta,X)} - \frac{Q(\beta,X)}{P(\beta,X)\gamma_{i,l}\gamma_{j,l}}$$

for $a > \ell, \quad b > \ell$

$$E\left(z_{i,a} z_{j,b} / observed\right) = \frac{\gamma_{i,a}\gamma_{i,b}}{P(\beta,X)} - \frac{Q(\beta,X)}{P(\beta,X)}$$

and for $a \leq \ell, \quad b > \ell$

$$E\left(z_{i,a} z_{j,b} / observed\right) = \frac{\gamma_{i,a}\gamma_{i,b}}{P(\beta,X)} - \frac{\gamma_{i,a}}{\gamma_{i,l}} \frac{Q(\beta,X)}{P(\beta,X)}$$

It is thus straight forward to complete the off diagonal blocks of V_T, given that the conditional expectations are known.

Hence for N truncated groups using the extended definitions of X, D, Γ and z in (3.1) and letting

$$\mu_T' = (\mu_{1T}', \ldots, \mu_{NT}')$$

where

$$\mu_{ijT} = \gamma_{ij} / P - (Q / P)\, \gamma_{i,j\wedge\ell} / \gamma_{i,\ell}$$

where $j \wedge \ell = \begin{cases} \ell & j > \ell \\ j & j \leq \ell \end{cases}$

and

$$V_T = B\text{diag}(V_{iT})$$

we have that the general score statistic and Fisher information are also given by (3.2) and (3.3). So the Fisher scoring algorithm for the estimation of β is defined by the equation

$$\left(X'D\Gamma^{-1}V_T\Gamma^{-1}DX\right)\left(\hat{\beta}_{\text{NEW}} - \hat{\beta}_{\text{OLD}}\right) = X'D\Gamma^{-1}(Z - \mu_T). \qquad (3.4)$$

The behaviour of the estimate will depend crucially on the condition number of $X'D\Gamma^{-1}V_T\Gamma^{-1}DX$. In the next section the impact of this on the efficiency is discussed.

References

Breslow, N.E. and Day, N.E. (1980). *Statistical Methods in Cancer Research:* Volume 1. *The Analysis of Case–Control Studies*. IARC, Lyon:

Lui, K., McGee, D., Rhodes, P. and Pollock, D. (1988).. An application of a conditional logistic regression to study the effects of safety belts, principal impact points, and car weights on driver's fatalities. *Journal of Safety Research*, **19**, 197–203.

McCullagh, P. and Nelder, J.A. (1989). *Generalized Linear Models*, Second Edition Chapman and Hall, London.

O'Neill, T.J. and Barry, S (1992). Truncated Logistic Regression. Submitted.

Appendix 5

Truncated Logistic Regression

by

Terence J. O'Neill

and

Simon C. Barry

Department of Statistics, The Faculties
Australian National University
Canberra, ACT 0200, Australia

Abstract

Truncated binary data occurs when a group of individuals, who each have a binary response, are observed only if at least one of the individuals has a positive response. This paper considers the regression modelling of such data when covariates are also observed and quantifies the loss of efficiency that can arise from the truncation. Although the efficiency loss compared to untruncated data can be substantial, viable estimation is still possible with truncated binary data. An alternative procedure called conditional logistic regression (Breslow and Day, 1980), which conditions on the actual number of deaths, has been previously used for this type of data. Truncated logistic regression is computationally simpler than conditional logistic for groups of size greater than two and is shown to be considerably more efficient generally.*

Acknowledgement

This research was supported by a seeding grant from the Federal Office of Road Safety, Australian Department of Transport and Communications.

---

* Keywords: Truncation, Logistic Regression, Conditional.

Introduction

This paper was motivated by a call for tenders for the analysis of the 1988 Fatal File. This is a file compiled by the Federal Department of Transport and Communications in Australia and consists of records of all road traffic accidents involving fatalities in Australia in 1988. The aim of the proposed analysis was to examine the effects of covariates such as for example seat position, seat belt usage, size of vehicle, on the probability of death in a road traffic accident. One obvious problem with this data set is that we only see the accidents involving fatalities. So we only observe the binary variables for death for individuals involved in an accident if at least one of the binary variables is 1. This truncation means that standard logistic regression is no longer appropriate and different methods must be used. The truncated logistic likelihood considered in this paper allows for the correlation induced between individuals involved in a given crash.

Conditional logistic regression has been previously used in this area by Lui *et . al* (1988) to analyse data from the Fatal Accident Reporting System. They restricted their analysis to two vehicle accidents where each vehicle had a single occupant and exactly one fatality resulted. It is obviously possible to generalise their approach to multi-occupant, multi-fatality crashes, but the simple nature of the conditional likelihood is not preserved and computational difficulties result. Some approximate methods of maximising the resulting conditional likelihood are discussed in Thompson (1991). The truncated logistic likelihood is no more complex for multi-occupant, multi-fatality crashes. Further, since it is only conditioning on at least one death, it will clearly be more efficient.

The aim of the present paper is to show that truncated binary regression is feasible computationally and to show that the truncated data contains a large component of the total information compared to untruncated data. This paper also evaluates the efficiency of conditional to truncated logistic regression for several examples and shows that the efficiency losses using conditional logistic regression can be substantial. Since the truncated likelihood is also computationally simpler, it should often be preferred to conditional logistic regression.

There is a large literature on truncation models where the response y_i for an individual is only observed if $y_i \geq c_i$ where c_i is some known constant. Some recent examples are Lagakos, Barraj and De Grutta (1988) who consider truncation models for AIDS survival data and Hodoshima (1988) who considered the effect of truncation on the identifiability of regression coefficients. The present example is believed to be non standard since rather than an individual being observed if and only if its response achieves a certain level, a group or cluster is observed if and only if the maximal response over the group achieves a specified level. Hence the truncation here is novel and standard truncation likelihood formulae do not apply. If the only covariate indicates a dichotomous treatment, then the responses of the group form a 2 x 2 contingency table where the total dead is conditioned to be at least one. Various examples of estimation for constrained contingency tables using quasi likelihood have been presented by McCullagh and Nelder (1989). The present situation however does not appear to have been considered.

Although the preceding discussion has been couched in terms of road traffic data, it is clear that there will be other areas of application. The likelihood proposed in this paper will often be an alternative to those proposed in proband studies (Thompson, 1986) if registered clusters are sampled on an equally likely basis. A possible economic application might be looking at employment in families where at least one member is registered unemployed.

In section 2, maximum likelihood estimation of truncated logistic regression is considered. Section 3 derives the efficiency losses that are caused by the truncation and conditioning. Section 4 evaluates the efficiency losses for some examples. Section 5 presents an analysis of some road safety data.

2. The Truncated Binary Regression Likelihood

In the following let

y_i = the response for the *i*th individual in the cluster.

x_i = the p x 1 vector of covariates specific to individual i.

X = the n x p matrix with x_i in the *i* th row.

$\mathfrak{R}$ = the set of individuals in the group.

β = the p x 1 vector of regression parameters.

Consider a group of size *n* subject to truncation. We suppose that a logistic model holds for the individual binary responses,

$$\begin{aligned}\Pr(y_i = 1) &= \exp(\beta' x_i) / \{1 + \exp(\beta' x_i)\} \\ &= p(\beta, x_i) \\ &= 1 - q(\beta, x_i).\end{aligned}$$

Note that the covariates X may include group level effects as well as individual specific covariates. For example in road traffic applications X would include information on the vehicle and the severity of the crash. Then the log likelihood for a truncated group becomes

$$\sum_{\mathcal{R}} \left\{ y_j \log p(\beta, x_j) + (1 - y_j) \log q(\beta, x_j) \right\} - \log P(\beta, X)$$

where

$$1 - P(\beta, X) = Q(\beta, X) = \prod_{\mathcal{R}} q(\beta, x_j) \, ,$$

and $\sum_{\mathcal{R}} = \sum_{j \in \mathcal{R}}$ denotes summation over the individuals in the group.

Now since

$$\partial \log P(\beta, X) / \partial\beta = Q(\beta, X) / P(\beta, X) \sum_{\mathcal{R}} p(b, x_j)\, x_j \, ,$$

the score function for a truncated group is

$$U(\beta) = \sum_{\mathcal{R}} \left\{ y_j - p(\beta, x_j) \right\} x_j - Q(\beta, X) / P(\beta, X) \sum_{\mathcal{R}} p(\beta, x_j)\, x_j$$

$$= \sum_{\mathcal{R}} x_j y_j - \mu(\beta, X) \qquad (2.1)$$

where for truncated binary regression

$$\mu(\beta, X) = E\left(\sum_{\mathcal{R}} x_j y_j \mid \mathcal{R} \, ; \, \sum_{\mathcal{R}} y_j \geq 1 \right)$$

$$= P(\beta, X)^{-1} \sum_{\mathcal{R}} p(\beta, x_j)\, x_j$$

The sample information matrix from the group is thus

$$I(\beta, X) = \partial\mu'(\beta, X) \ / \ \partial\beta$$

$$= \sum_{\mathcal{R}} p(\beta, x_j)\, q(\beta, x_j) / P(\beta, X) x_j x_j' - Q(\beta, X)\mu(\beta, X)\mu(\beta, X)' ,$$

$$= \sum_{\mathcal{R}} p(\beta, x_j) / P(\beta, X) x_j x_j' + \sum_{i \neq j} p(\beta, x_i)\, p(\beta, x_j) / P(\beta, X) x_i x_j'$$

$$- \mu(\beta, X)\mu(\beta, X)'$$

$$= \mathrm{Var}\left(\sum_{\mathcal{R}} x_j y_j \mid \mathcal{R} \; ; \; \sum_{\mathcal{R}} y_j \geq 1 \right)$$

$$= V(\beta, X).$$

So

$$I(\beta, X) = V(\beta, X). \qquad (2.2)$$

Hence letting

$$r = \sum_{\mathcal{R}} x_j y_j ,$$

then for a sample of N truncated groups, the score function is

$$\sum_{i=1}^{N} \left\{ r_i - \mu(\beta, X_i) \right\}$$

and the sample information matrix is

$$\sum_{i=1}^{N} V(\beta, X_i) .$$

Thus a simple Newton-Rhapson scheme can be used to find the maximum likelihood estimate $\hat{\beta}$ via

$$\hat{\beta}_t = \hat{\beta}_{t-1} + \left\{ \sum_{i=1}^{N} V(\hat{\beta}_{t-1}, X_i) \right\}^{-1} \sum_{i=1}^{N} \left\{ r_i - \mu(\hat{\beta}_{t-1}, X_i) \right\}$$

and the estimated covariance of $\hat{\beta}$ is given by

$$\left\{ \sum_{i=1}^{N} V(\hat{\beta}, X_i) \right\}^{-1}.$$

Inference can be performed using the asymptotic normality of the maximum likelihood estimators. In addition model testing can be done in the usual way using differences of deviances.

As a very simple case consider the following example:

Example 2.1: Psychiatric data

Cox and Snell (1989, p. 53) analyse the following paired comparison data of Maxwell (1961, p. 28) on the effect of a treatment on twenty three matched pairs of depressed patients.

Table 3.1 Recovery of psychiatric patients

Response:		
Depersonalised	Not Depersonalised	Number of Pairs
0	0	2
1	0	2
0	1	5
1	1	14

For illustration purposes only, we consider truncating this data by discarding pairs where neither patient responded. When a two group model is fitted to the full data we obtain estimated logits of the probabilities for the two groups of .827 and 1.558 with an estimated variance matrix of

$$\begin{pmatrix} .205 & 0 \\ 0 & .303 \end{pmatrix}.$$

When the two group model is fitted to the truncated data the estimated logits are 1.030 and 1.950 with an estimated variance matrix of

$$\begin{pmatrix} .271 & .071 \\ .071 & .571 \end{pmatrix}.$$

It is worth emphasising that for the truncated data, the number of truncated groups is not known. The estimated probability from the truncated fit of a group being observed is .967 which can be compared to an estimate of .947 from the full data set and an observed frequency of 21/23 = .913 of non-truncated pairs. In the next section, we consider the efficiency loss which arises from the truncation.

3. Efficiency of Truncated Logistic Regression

To compare the efficiency of the truncated model to the conditional model we will assume that the logistic model

$$\Pr(y_i = 1) = \exp(\beta' x_i) \,/\, \{1 + \exp(\beta' x_i)\}$$

holds and that the x_i's are known for each individual. This assumption is not restrictive. For example, in the case of road safety data, logistic regression would be a logical initial analysis technique. It is only the truncation of the responses that precludes its use. What this is effectively saying is that the probability structure induced by the logistic model is appropriate, but that the sampling scheme is causing complications.

In the truncated case, conditional logistic regression is used to avoid the difficulties introduced by the truncation. This is different to its use in, for example, matched case/control studies where its purpose is to remove the effect of group level covariates. Of course, the use of conditional logistic regression with truncated data will have the effect of removing the group level effects, but if these effects can be adequately modelled by the logistic model then the technique will be less efficient. In this section this efficiency loss is quantified.

In the following the efficiency will be compared on a prospective basis. This is done so that the methods are compared with respect to a common sample space. The underlying sample space is based on grouped, but not truncated data. The term prospective relates to the expected information provided by a group drawn from this sample space. For example, if the group contains no deaths it provides no information to the truncated likelihood.

If the group of individuals is not truncated, then the sample information is equal to the Fisher information and is

$$\mathcal{I}(\beta, X) = \sum_{\mathcal{R}} p(\beta, x_j) q(\beta, x_j)\, x_j x_j'$$

while the Fisher information from a (possibly unobserved) group subject to truncation is

$$\mathcal{I}_T(\beta, X) = P(\beta, X) \left\{ \sum_{\mathcal{R}} p(\beta, x_j) q(\beta, x_j) / P(\beta, X)\, x_j x_j' - Q(\beta, X)\, \mu(\beta, X)\, \mu(\beta, X)' \right\}$$

$$= \mathcal{I}(\beta, X) - P(\beta, X) Q(\beta, X)\, \mu(\beta, X)\, \mu(\beta, X)' \; .$$

So the information loss by truncation is

$$\mathcal{I}(\beta, X) - \mathcal{I}_T(\beta, X) = P(\beta, X) Q(\beta, X)\, \mu(\beta, X)\, \mu(\beta, X)' \; .$$

Next consider the information loss due to conditioning. Consider the group $\mathcal{R}$ of n individuals for which $x'_i = (u', v'_i)$ where u' is constant over the group and only v_i varies. If m deaths are observed, the conditional logistic likelihood is

$$\frac{\exp(\beta_2' S_2)}{\sum_{\mathcal{Z} \in \mathcal{R}_m} \exp(\beta_2' S_{\mathcal{Z},2})} \tag{3.1}$$

where

$$\beta' = (\beta_1', \beta_2') \, ,$$

such that

$$x'_i \beta = u' \beta_1 + v'_i \beta_2 \, ,$$

and $$S = \sum_{\text{observed deaths}} x_i \, , \qquad S_2 = \sum_{\text{observed deaths}} v_i$$

and

$$S_{\mathcal{Z}} = \sum_{\text{subset } \mathcal{Z} \text{ of } \mathcal{R}_m} x_i, \qquad S_{\mathcal{Z},2} = \sum_{\text{subset } \mathcal{Z} \text{ of } \mathcal{R}_m} v_i$$

and $\mathcal{R}_m$ is the set of subsets of $\mathcal{R}$ of size m. The likelihood (3.1) is identical to the Cox Proportional Hazards Likelihood (Cox 1972, 1975, Efron 1977). Then the sample information for estimating β_2 in the conditional logistic likelihood is $V_m(\beta_2, X)$, the variance of S_2 given the probability distribution (3.1) over $\mathcal{R}$. Hence *since* by *assumption* u *does not vary over* $\mathcal{R}$, the sample information for estimating β is $V_m(\beta, X)$,

$$V_m(\beta, X) = \begin{pmatrix} 0 & 0 \\ 0 & V_{m,22}(\beta_2, X) \end{pmatrix}$$

the variance of S given the probability distribution (3.1). So, letting $\mathcal{I}_c(X) = \mathcal{I}_c(\beta, X)$ denote the Fisher information matrix for β from the conditional logistic likelihood, we have

$$\mathcal{I}_C(X) = \begin{pmatrix} 0 & 0 \\ 0 & \mathcal{I}_{C,22}(X) \end{pmatrix}$$

where

$$\mathcal{I}_{C,22}(X) = \sum_{m=1}^{n-1} P_m(\beta, X) V_{m,22}(\beta_2, X) = \sum_{m=1}^{n} P_m(\beta, X) V_{m,22}(\beta_2, X),$$

where $P_m(\beta, X)$ is the probability that m deaths are observed in the group. $\mathcal{I}_{C,22}(X)$ is thus the expected, with respect to the distribution of M, *information from the group.*

Let M denote the actual number of deaths in the group and

$$\mathcal{I}_{M,m}(X) = \mathcal{I}_{M|M=m}(\beta, X),$$

the Fisher information in m deaths conditional on there being at least one death. This is the information in the conditional density of $M \mid M \geq 1$,

$$P_{M|M>1}(\beta, X) = \Pr(M = m \mid M \geq 1)$$

$$= \frac{\sum_{\chi \in \mathcal{R}_m} \exp(\beta' S_\chi)}{\prod_{\mathcal{R}} \left\{ 1 + \exp(\beta' x_i) \right\} - 1}.$$

Hence the expected information from the $\mathcal{I}_{M,m}(X)$ component is

$$\mathcal{I}_M(X) = \sum_{m=1}^{n} P_m(\beta, X) \mathcal{I}_{M,m}(X)$$

Hence if we let $F(X)$ describe the expected distribution of X where X can be chosen either stochastically or deterministically, then for each type of information $\mathcal{I}$, $\mathcal{I}_T$, $\mathcal{I}_C$ and $\mathcal{I}_M$ it follows that

$$\mathcal{I}. = \int \mathcal{I}.(X) dF(X)$$

and since

$$\Pr(y_1, y_2, \ldots, y_n | M \geq 1) = \Pr(y_1, y_2, \ldots, y_n | M) \Pr(M \mid M \geq 1)$$

It follows that

$$\mathcal{I}_T(X) = \mathcal{I}_C(X) + \mathcal{I}_M(X),$$

and we have that

$$\mathcal{I}_T = \mathcal{I}_C + \mathcal{I}_M.$$

We will now consider measures of relative efficiency. With a scalar parameter the choice of measure of efficiency is straight forward. With vector parameters the choice becomes less clear.

We will use as our measure of efficiency, when comparing method A to the less efficient method B,

$$\text{Efficiency loss} = p^{-1} \operatorname{tr} \mathcal{I}_A^{-1} \left\{ \mathcal{I}_A - \mathcal{I}_B \right\}.$$

where $\mathcal{I}_A$ and $\mathcal{I}_B$ are the information in the group using method A and B respectively and it is assumed that $\mathcal{I}_A^{-1}\mathcal{I}_B$ is positive semi-definite.

This has the obvious property that if $\mathcal{I}_A = \mathcal{I}_B$ the efficiency loss is zero, and if $\mathcal{I}_B$ is a matrix of zeroes the efficiency loss is one. It is also invariant under transformations of the parameters. It can be viewed as the approximate average efficiency loss for the orthogonal parameterisation $\beta^* = \mathcal{I}_A^{1/2}(\beta)\beta$. This parameterisation will be orthogonal for method A, but is not necessarily orthogonal for method B.

With this definition the efficiency loss by truncation is

$$ET = p^{-1} \operatorname{tr} \mathcal{I}^{-1} \left\{ \mathcal{I} - \mathcal{I}_T \right\} .$$

As an overall measure of the efficiency of conditional compared to truncated estimation of β we will use

$$EC = p^{-1} \operatorname{tr} \mathcal{I}_T^{-1} \left\{ \mathcal{I}_T - \mathcal{I}_C \right\} .$$

However we note that in the case when u never varies within $\mathcal{R}$, and thus the information for the corresponding parameters in the conditional likelihood is zero,

$$EC \le p_2 / p$$

where v_i has length p_2. The efficiency of conditional to truncated estimation of β_2 is

$$EC_2 = p_2^{-1} \operatorname{tr} \mathcal{I}_{T,22}^{-1} \left\{ \mathcal{I}_{T,22} - \mathcal{I}_{C,22} \right\},$$

i.e. the efficiency is only compared over the parameters that can be estimated by the two methods.

In section 4 we evaluate these efficiency measures for two examples.

4. Efficiency examples

Example 4.1 : Two group problem.

We consider the simplest possible regression where the covariate, x, is an indicator variable for the second of one of two treatments. We also suppose that each group subject to truncation has only two members who get different treatments. A practical example might be the driver and front passenger seat occupant of single vehicle accidents where the treatment is the seat location and the accident is only recorded in the fatal file if a fatality resulted. Without loss of generality we may assume that the design matrix for the group is

$$X = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Now let

$$p_i = \text{the probability that individual i dies.}$$

by straightforward algebra

$$ET = (p_1q_2 + p_2q_1) / 2(1 - q_1q_2) \geq 1/2$$

In Figure 1, we contour ET with respect to p_1 and p_2. Note that although the efficiency loss to truncation can be substantial, the retained efficiency is always at least 50% and viable estimation is possible. Also, it is well known that the discordant pairs contain all the relevant information concerning the difference between the two treatments or

$$EC = 1/2, \; EC_2 = 0$$

and so in this case, conditional logistic regression is fully efficient for the estimation of β_2. However this is the only case where conditional logistic regression is fully efficient for estimating a subset. If the group size is ever larger than 2 or X varies across groups, then $EC_2 > 0$.

[Insert Figure 1 about here]

Example 4.2 : Continuous covariates

We now consider the case where

$$X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \end{pmatrix} = \begin{pmatrix} x_{+1}' \\ x_{+2}' \end{pmatrix}$$

where x_1 and x_2 are chosen independently from the Uniform (0, 1) density. If $p_i = p(\beta, x_{+i})$ and $p_{2i} = p(\beta_2, x_i)$ where

$$p(\beta, x) = \exp(x'\beta) / (1 + \exp(x'\beta))$$

then

$$V(\beta, X) = (1 - q_1q_2)^{-1} \sum_i p_iq_i x_{+i} x_{+i}' - q_1q_2(1 - q_1q_2)^{-2}(p_1x_{+1} + p_2x_{+2})(p_1x_{+1} + p_2x_{+2})'$$

$$V_1(\beta_2, X) = \sum_i p_{2i}x_i^2 - \left(\sum_i p_{2i}x_i\right)^2 ,$$

$$P(\beta, X) = (1 - q_1q_2),$$

$$P_1(\beta_2, X) = p_1q_2 + q_1p_2$$

and

$$\mathcal{I}_c(X) = p_1q_1p_2q_2(x_1 - x_2)^2 / (p_1q_2 + p_2q_1).$$

Although $\mathcal{I}_T$ and $\mathcal{I}_c$ cannot be explicitly evaluated, they can be calculated by numerical integration for a range of β values. In Figures 2, 3 and 4 we contour ET, EC_2 and EC for a range of values of the intercept and slope parameters. It can be seen that truncated logistic is often very much more efficient than conditional logistic. It can also be seen that the loss caused by truncation is not usually extreme.

[Insert Figures 2, 3 and 4 about here]

5. Federal Office of road safety data.

This section examines the effects of various covariates on the probability of death for passengers involved in fatal car accidents. This analysis is based on the so called "fatal files" which are collected by the Australian Federal office of road safety on a biennial basis. These files consist of passenger and vehicle information for all fatal accidents that occur in the target year. The analysis is based on the records for the 1988 and 1990 calendar years.

The aim of the analysis was to estimate the effect of various group level and individual level covariates on probability of death. To simplify the analysis it was restricted to single vehicle, frontal impact collisions that involved passenger cars. This produced a data set with observations on 306 individuals involved in 111 accidents. Note that cars with only a driver are non informative for all methods.

From this data set the accidents involving a front seat passenger and driver were extracted for use in the conditional analysis. This data set consisted of information from 76 accidents. While this analysis could have been augmented by constructing the conditional likelihood for each car, the following points should be noted. Firstly, a large proportion of the accidents involved cars with driver and front seat passenger. Secondly the construction and maximisation of the conditional likelihood for varying size clusters, though analytically straight forward, would be complicated to numerically perform. Finally, in the other analysis published in this area (Lui *et al* 1988) the paired analysis considered here was performed.

Although a wide range of variables are available for each individual, a subset was chosen due to their previous association with fatalities in car accidents. See for example the work of Evans(1985) or Lui *et al* (1988). The variables selected are described in table 1.

The models were fitted using the method presented in section 2 and the results are presented in table 2. All covariates were treated as factors, and the design matrix was constructed using treatment (Chambers and Hastie 1992) contrasts. The effect of the lowest level of each factor was set to zero.

[Insert tables 1 and 2 about here]

The agreement between the two estimates is encouraging. Although both methods model the mean response equivalently, departures from the underlying models, such as missing covariates, could cause discrepancies to arise. Examining the standard errors of the estimates reveals the increase in efficiency that is achieved using the truncated model.

6. Summary

The results found in this paper are very encouraging for the analysis of truncated binary data. They show

- Viable regression estimation using the truncated likelihood is possible when the binary variables of groups of individuals are only observed if at least one is positive. The resulting likelihoods are well behaved and tractable. The truncated likelihood also avoids the difficulties of the conditional likelihood (Thompson, 1991) when more than one death in a group is observed.

- The efficiency loss due to truncation although substantial is not catastrophic. For larger groups the efficiency loss will be less.

- The efficiency of truncated versus conditional logistic regression has been evaluated in several realistic examples showing that the efficiency loss in using conditional logistic can often be high.

The choice between truncated and conditional logistic regression for the analysis of group truncated binary data will be governed by the group level effects. If they vary systematically across groups or clusters then truncated logistic should be used whereas for random or unstructured group level effects, conditional logistic should be used since it eliminates all group level effects. For example if the group level effects that affect the probability structure of the response are known (ie the individual probabilities can be adequately modelled) then truncated logistic regression is appropriate.

This appears the case with the road safety data used in the example. In this case conditional logistic regression has been used to remove the biases introduced by the sampling scheme. It may be argued that the nature of the data will ensure that slight unexplained variation between groups remains. While this may be true, these slight imperfections are also true for most stochastic models.

Truncated logistic regression provides a natural framework for the analysis of truncated binary data. It utilises the full likelihood, and is the logical extension of the non truncated case. Because of the superior efficiency and relative computational simplicity, truncated logistic regression should often be the preferred method for truncated binary data.

References

Breslow, N.E. and Day, N.E. (1980). *Statistical Methods in Cancer Research : Volume 1. The Analysis of Case-Control Studies.* IARC, Lyon.

Chambers, J.M. and Hastie, T.J. ed. (1992) *Statistical models in S*. Wadsworth and Brooks/Cole, Pacific Grove, California.

Cox, D.R. (1972). Regression models and life tables (with discussion). *Journal of the Royal Statistical Society B* **74**, 187-220.

Cox, D.R. (1975). Partial likelihood. *Biometrika* **62**, 269-76.

Cox, D.R. and Snell, E.J. (1989). *Analysis of Binary Data.* Chapman and Hall, London.

Efron, B. (1977). The efficiency of Cox's likelihood function for censored data. *Journal of the American Statistical Association* **72**, 557-65.

Evans, L. (1985) Double pair comparisons - a new method to determine how occupant characteristics affect fatality risk in traffic crashes. *Accident Analysis and Prevention* **18**, 217-227.

Hodeshima, J. (1988). The effect of truncation on the identifiability of regression coefficients. *Journal of the American Statistical Association* **83**, 1055-56.

Lagakos, S.W., Barraj, L.M. and De Grutta, V. (1988). Non-parametric analysis of truncated survival data, with application to AIDS. *Biometrika* **75**, 515-523.

Lui, K., McGee, D., Rhodes, P. and Pollock, D. (1988). An application of a conditional logistic regression to study the effects of safety belts, principal impact points, and car weights on driver's fatalities. *Journal of Safety Research*, **19**, 197-203.

Maxwell, A.E. (1961). *Analysing Qualitative Data.* Chapman and Hall, London.

McCullagh, P. and Nelder, J.A. (1989). *Generalized Linear Models,* Second Edition. Chapman and Hall, London.

Thompson, E.A. (1986). *Pedigree Analysis in Human Genetics.* The Johns Hopkins University Press, Baltimore, Maryland.

Thompson, J. (1991). Conditional Logistic regression in Genstat. *Genstat Newsletter*, **26**, 6-10.

Table 1: Description of variables for 1988/1989 FORS data

Variable	Level	Category
Damage	0	major damage
	1	extensive damage
Resavl	0	restraint worn/ restraint use unknown
	1	restraint definitely not worn
Sex	0	male
	1	female
Speedlim	0	speed limit 0-60 kmh
	1	speed limit 60+ kmh
Speedcat	0	not over speed limit
	1	orer/possibly over speed limit
Age	0	0-15 years old
	2	15-25 years old
	3	25-60 years old
	4	60 + years old
Perloc	0	front seat
	1	not front seat

Table 2: Log odds ratio estimates for single vehicle, frontal impact collisions, using the 1988/1990 FORS dat a.[2]

Variable[1]	Level	GTLR	Conditional Logistic
Intercept	1	3.46(.87)	NA
Damage	1	0.53(.61)	NA
Resavl	1	1.11(.36)	1.35(.65)
Speedlim	1	0.64(.51)	NA
Perloc	1	-0.34(.31)	Not estimated
Sex	1	0.36(.26)	0.61(.37)
Age	1	0.00(.50)	-0.45(.91)
	2	0.24(.53)	0.24(.91)
	3	1.21(.72)	1.77(1.4)
Speedcat	1	1.57(.48)	NA

1. Estimated standard errors are given in brackets.

2. $log\,\text{odds ratio}(\text{level i}) = log(\frac{\text{odds for level i}}{\text{odds for level 0}})$

Figure 1:Efficiency Loss for Truncated Binary Regression with Two Treatments

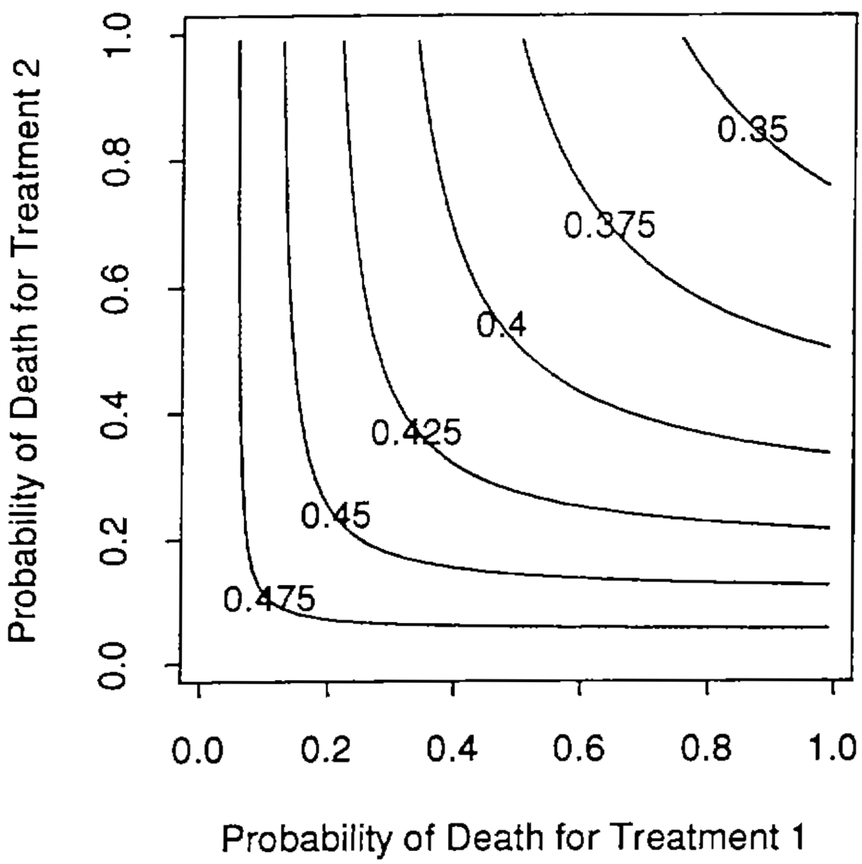


Figure 2:Efficiency Loss for Truncated Binary Regression with Uniform Covariates

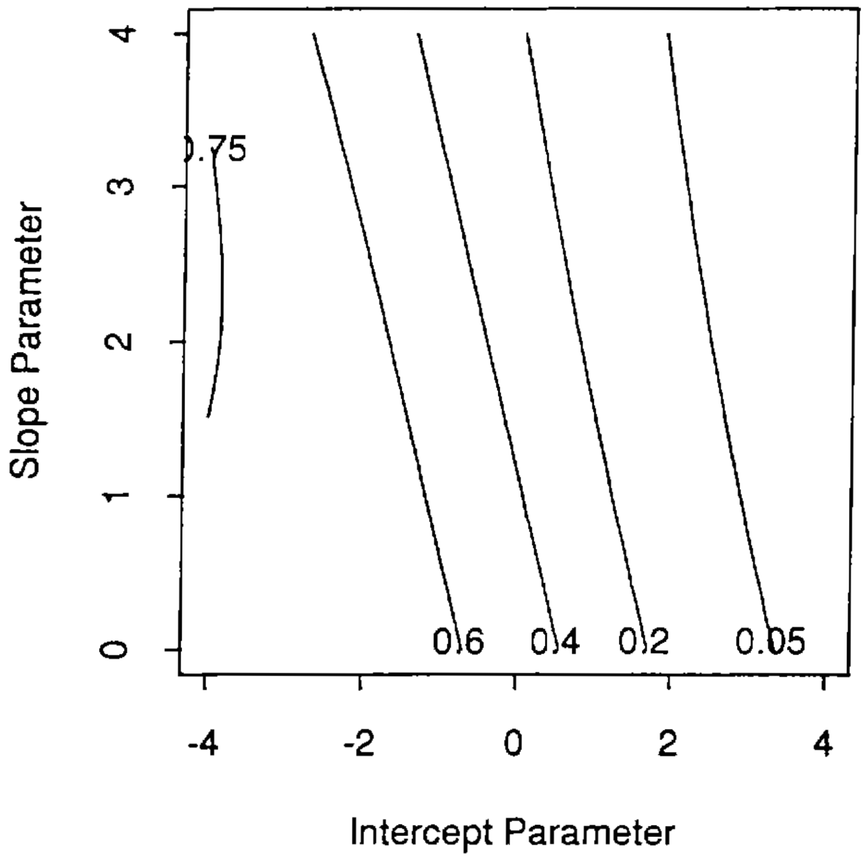


Figure 3:Slope Efficiency Loss for Conditional vs. Truncated Logistic with Uniform Covariates

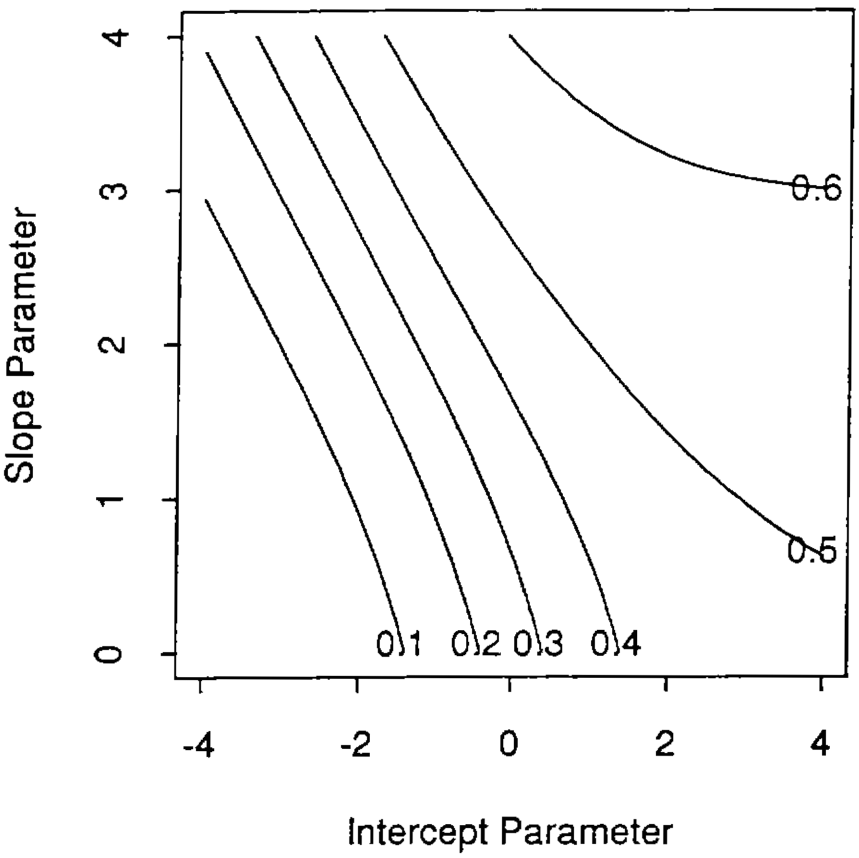
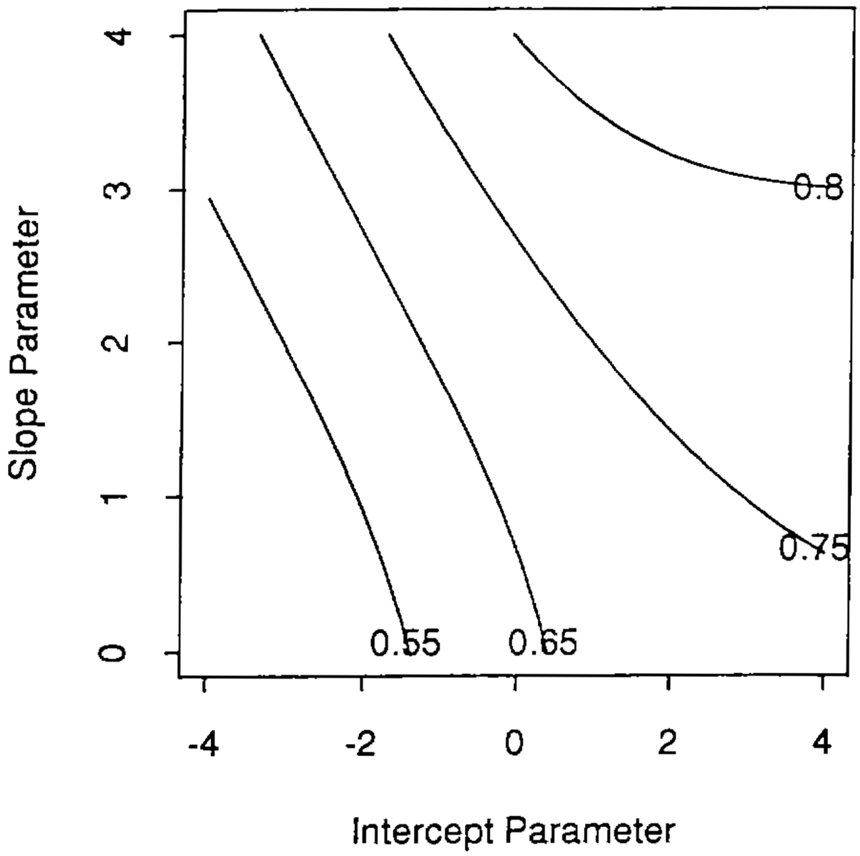


Figure 4:Efficiency Loss for Conditional vs Truncated Logistic with Uniform Covariates



Appendix C

trunc.fit()

Fit a group truncated ordinal regression.

DESCRIPTION

Produces an object of class trunclm which is the fit of a truncated logistic/ordinal grouped regression.

USAGE

trunc.fit(formula, data, tolerance, iter, groupvar, trunc, initbeta)

REQUIRED ARGUMENTS

formula: a formula expression as for other regression models, of the form response predictors. See the documentation of lm and formula for details.

data: a data frame in which to interpret the variables oc- curring in the formula. See data.frame for details.

tolerance: The required tolerance to be used in the fitting. It is the euclidean distance between the fitted expecta- tions from one iteration to the next.

iter: The maximum number of iterations used in the fisher scoring algorithm.

groupvar: A character string giving the name of the variable on the data frame that defines the group.

trunc: The level at which the truncation occurs (ie one response in the group must be above this level). This can either be the number of the level, or a character string for the name of the level.

OPTIONAL ARGUMENTS

initbeta: Optional initial value numeric vector for the parame- ter vector in the Fisher scoring algorithm. Otherwise logistic regression estimates are used.

VALUE

an object of class trunclm is returned. See trunclm.object for details.

DETAILS

The output can be examined using print and summary.

The models are fitted using the Fisher scoring algorithm, and the logistic link is assumed.

REFERENCES

O'Neill, T J and Barry, S C (1993). Estimation of Truncat- ed Ordinal Regression Models. Dept of transport an com- munications Road Safety Seeding Research Grant Report.

SEE ALSO

trunclm.object

EXAMPLES

Say we have a response vector "response" with three levels and a explanator vectors "belt" and "age". We must also have a vector, "group" say, defining which group each individual is in.

We would then use:

>response< –ordered(response)

>

to generate the ordered factor. If belt is a factor then we would use

>belt< –factor(belt)

>

Splus generates the design matrix by using the "contrasts" attribute of an factors in the model formula. To modify it, say to use treatment contrasts, use >belt< –C(belt,treatment)

>

So we have

> response

[1] a c c a b a a b a b c a b c b a a b c b

a < b < c

> belt

[1] 1 0 1 1 0 1 0 0 1 0 0 1 1 1 1 1 1 0 0 0

> age

[1] 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2

> group

[1] 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2

>

Last of all construct the model frame.

>exam.frame< –data.frame(response,belt,age,group)

>

> exam.fit2 < –trunc.fit(formula = response beltf + age, data = exam.frame, tolerance = 1e-06, iter = 20, groupvar = "group", trunc = "b")

+ -18.357277

-18.002097

-17.992586

-17.992529

-17.992528

-17.992528

-17.992528

-17.992528

-17.992528

It has obviously converged with this tolerance. As the complete output would be long and messy we use *summary()*.

> summary(exam.fit2)

Call: trunc.fit(formula = response beltf + age, data = exam2.frame, toleranc = 1e-06, iter = 20, groupvar = "group", trunc = "b")

Residuals:

Min 1Q Median 3Q Max

-0.9146 -0.3775 0.05862 0.2283 0.6868

		Value	*Std.Error*	*zvalue*	*Pr(¿ |z|)*
	model.coeff	−2.6797	1.1854	−2.2607	0.0238
Coefficients:	*model.coeff*	−0.6199	1.0261	−0.6041	0.5458
	beltf	−1.6803	0.9429	−1.7821	0.0747
	age	−0.2737	0.1764	−1.5517	0.1207

Correlation:

	model.coeff	*model.coeff*	*beltf*	*age*
model.coeff	1.0000	0.7898	0.5132	0.7482
model.coeff	0.7898	1.0000	0.3758	0.6872
beltf	0.5132	0.3758	1.0000	0.0182
age	0.7482	0.6872	0.0182	1.0000

Log likelihood:

[1] -17.99

trunclm.object

Group truncated ordinal regression object

DESCRIPTION:

These are objects of class "trunclm" which represent fits of group truncated ordinal regression models. This class of object is returned from the function trunc.fit(). The components can be extracted using the "$" operator.

The following components are included in the object.

COMPONENTS:

call: an image of the call that produced the object, but with the arguments all named and with the actual formula in- cluded as the formula argument.

fitted: the expected value of the response vector under the fitted model. Note that this is not equal to the fitted category probabilitys of the non truncated model. These should be found from $linear.pred.

variance: the estimated expected information for the parame- ters.

x: the matrix of predictors used in the fit. Note that the number of rows of this matrix is (number of levels of response - 1) * number of observations.

coefficients: the fitted regression coefficients. The names of the coefficients are the names of the single-degree- of-freedom effects (the columns of the model matrix) con- structed by Splus. The "model.coefficients" are the in- tercept terms. There are thus (number of levels of response -1) "model.coefficients". The coefficients have a one to one correspondence with the columns of the model matrix.

z: the vector of responses used in the fit of the model. This has the same number of rows a object$x. It is constructed of 1's and zeros. For observation i, z[i*(number of levels of response - 1) + j] = 0 if the response for individual i is greater that j, 1 otherwise.

linear.pred: the linear predictor, ie object$x % % object$coefficients.

log.lik: the value of the maximised log.likelihood.

iterations: The number of iterations used in the fit.

tolerance: The euclidean difference between the parameter vector in the last two iterations.

frame: The data frame used in the fit. This is included for cases where the group variable was not sorted, and trunc.fit performed the sort.

Appendix D

```
"no.groups"<-
function(group)
{
        num <- length(unique(group))
        num
}
"trunc.fit"<-
function(formula, data, tolerance, iter, groupvar, trunc, initbeta)
{
        if(!(is.loaded("_fitmodel"))) {
                dyn.load2("fittrunclm.o")
        }
        if(is.character(data[, groupvar])) {
                stop("group variable cannot be character")
        }
#is the data sorted?.  If not sort and warn.
        nums <- 1:nrow(data)
        permute <- order(data[, groupvar])
        sav.perm <- permute
        permute <- permute[ - nrow(data)]
        permute <- append(permute, 0, after = 0)
        if(sum((nums - permute) == 1) != nrow(data)) {
                warning("data.frame not sorted by group. sorted frame returned in $frame
                        )
        }
        data2 <- data[sav.perm,  ]
        call <- match.call()
        #setup.prop sets up starting values, design mats etc for the #proportional odds :
        derived <- setup.prop(formula, data2)
        new.X <- derived[[1]]
        new.Z <- derived[[2]]
        klength <- derived[[3]]
        new.beta <- derived[[4]]
        paramnames <- names(new.beta)
        if(!missing(initbeta)) {
                new.beta <- initbeta
        }
# get the trunc point
        if(is.character(trunc)) {
                temp.trunc <- trunc
                trunc <- match(trunc, levels(model.extract(model.frame(formula,
                        data2), "response")))
                if(is.na(trunc)) {
                        stop(paste("Level: ", temp.trunc,
                                "is not a valid level for the response"))
                }
        }
        if(trunc >= length(levels(model.extract(model.frame(formula, data2),
                "response")))) {
                stop(paste("Level: ", trunc,
                        "is the top level+ for the response and is thus invalid as a trur
                        ))
        }
        if(trunc < 1) {
                stop(paste("Level: ", trunc,
                        "is below the minimum level.  Level go 1,2,3..."))
        }
        num <- nrow(new.X)/klength
        change <- 9999999
        dev <- vector("numeric", 1)
        result <- .C("fitmodel",
                as.double(new.X),
                as.double(new.Z),
                as.double(new.beta),
                as.integer(num),
                as.integer(klength),
```

```
		as.integer(nrow(new.X)),
		as.integer(ncol(new.X)),
		as.integer(data2[, groupvar]),
		as.integer(trunc),
		as.integer(no.groups(data2[, groupvar])),
		as.double(tolerance),
		as.integer(iter),
		as.double(dev))
	indexvec <- 1:length(new.X)
	indexvec <- indexvec <= (ncol(new.X) * ncol(new.X))
	retlist <- list(call = call, fitted = result[[2]], variance = (matrix(
		as.vector(result[[1]])[indexvec], nrow = ncol(new.X))), x =
		new.X, coefficients = result[[3]], z = new.Z, linear.pred =
		NULL, log.lik = result[[13]], iterations = result[[12]],
		tolerance = result[[11]], frame = data2)
	attr(retlist$coefficients, "names") <- paramnames
	linpred <- retlist$x %*% retlist$coefficients
	retlist$linear.pred <- linpred
	fitted <- exp(linpred)/(1 + exp(linpred))
	if(sum(fitted > 0.999)) {
		warning("fitted value close to 1.  Could mean parameter estimates going
			)
	}
	if(sum(fitted < 0.0001)) {
		warning("fitted value close to 0.  Could mean parameter estimates going
			)
	}
	attr(retlist, "class") <- c("trunclm")
	retlist
}
"setup.prop"<-
function(formula, data)
{
#this function constructs the model matrix and response vec for the proportional #odds m
	model.terms <- terms(formula)
	mod.frame <- model.frame(model.terms, data)
	model.mat <- model.matrix(model.terms, data)
	model.mat <- as.matrix(model.mat[, -1])
	model.mat <- model.mat * -1
	response.vec <- model.extract(mod.frame, "response")
	if(!is.ordered(response.vec)) {
		stop("response is not an ordered factor")
	}
	klength <- length(levels(response.vec)) - 1
	evec <- vector("numeric", length = klength)
	evec[] <- 1
	new.X <- matrix(0, nrow = length(response.vec) * klength, ncol =
		klength + length(model.mat[1,  ]))
	xresult <- .C("formXblocks",
		as.double(model.mat),
		as.integer(nrow(model.mat)),
		as.integer(ncol(model.mat)),
		as.integer(klength),
		as.double(new.X))
	new.X <- matrix(xresult[[5]], nrow = length(response.vec) * klength,
		ncol = klength + length(model.mat[1,  ]))
	new.Z <- vector("numeric", length = klength * length(response.vec))
	zresult <- .C("formZ",
		as.double(as.numeric(response.vec)),
		as.integer(length(response.vec)),
		as.integer(klength),
		as.double(new.Z))
	new.Z <- zresult[[4]]
	new.beta <- init.beta(data, new.Z, klength, model.terms)
	result <- list(new.X, new.Z, klength, new.beta)
	result
```

```
}
"print.summary.trunclm"<-
function(x, digits = max(3, .Options$digits - 3), ...)
{
	cat("\nCall: ")
	dput(x$call)
	resid <- x$residuals
	df <- x$df
	rdf <- df
	if(rdf > 5) {
		cat("Residuals:\n")
		if(length(dim(resid)) == 2) {
			rq <- apply(t(resid), 1, quantile)
			dimnames(rq) <- list(c("Min", "1Q", "Median", "3Q",
				"Max"), dimnames(resid)[[2]])
		}
		else {
			rq <- quantile(resid)
			names(rq) <- c("Min", "1Q", "Median", "3Q", "Max")
		}
		print(rq, digits = digits, ...)
	}
	else if(rdf > 0) {
		cat("Residuals:\n")
		print(resid, digits = digits, ...)
	}
	cat("\nCoefficients:\n")
	print(format(round(x$coef, digits = digits)), quote = F, ...)
	correl <- x$cov.unscaled * (1/sqrt(diag(x$cov.unscaled)))
	correl <- t(correl) * (1/sqrt(diag(x$cov.unscaled)))
	dimnames(correl) <- list(dimnames(x$coef)[[1]], dimnames(x$coef)[[1]])
	cat("\nCorrelation:\n")
	print(format(round(correl, digits = digits)), quote = F, ...)
	cat("\n Log likelihood:\n")
	print(x$log.lik, digits = digits, ...)
	cat("\n")
	invisible(x)
}
"summary.trunclm"<-
function(object, correlation = T)
{
#this method is designed on the assumption that the coef method
# returns only the estimated coefficients.  It will (it's asserted)
# also work, however, with fitting methods that don't follow this
# style, but instead put NA's into the unestimated coefficients
	coef <- coefficients(object)
	log.lik <- object$log.lik
	cnames <- labels(coef)
	ctotal <- object$coef
	ptotal <- length(ctotal)
	resid <- object$z - fitted(object)
	fv <- fitted(object)
	n <- length(resid)
	p <- length(ctotal)
	var <- object$variance
	coef <- array(coef, c(p, 4))
	dimnames(coef) <- list(cnames, c("Value", "Std. Error", "z value",
		"Pr(>|z|)"))
	coef[, 2] <- sqrt(diag(var))
	coef[, 3] <- coef[, 1]/sqrt(diag(var))
	coef[, 4] <- pnorm( - abs(coef[, 3])) + 1 - pnorm(abs(coef[, 3]))
	object <- object[c("call", "terms")]
	object$residuals <- resid
	object$coefficients <- coef
	object$cov.unscaled <- var
	object$df <- length(resid) - length(ctotal)
```

```
        object$log.lik <- log.lik
        class(object) <- "summary.trunclm"
        object
}
"init.beta"<-
function(datafr, response.vec, klength, terms.obj)
{
#for each level of response a logistic regression is performed.
#The intercepts from these are returned as the model coeffs, while the
#average over the regressions are returned for the explanators.
        result <- vector()      #perform glm() for each level of response.
        for(i in 1:klength) {
                index.vec <- vector("numeric", length = length(response.vec))
                temp.vec <- vector("numeric", length = klength)
                temp.vec[i] <- 1
                index.vec[] <- temp.vec
                iter.response <- response.vec[as.logical(index.vec)]
                temp.data <- data.frame(datafr, iter.response)
                modeli <- glm(paste("iter.response~", paste(attr(terms.obj,
                        "term.labels"), collapse = "+")), data = temp.data,
                        family = binomial)
                model.coeff <- coef(modeli)
                result <- rbind(result, model.coeff)
        }
#extract model coeffs and average over others
        thetas <- result[, 1]
        params <- apply(result, 2, mean, na.rm = T)
        params <- params[-1]    #added line
        params <- params * -1
        result <- append(thetas, params)
        result
}
```

Appendix E

```
#!/bin/csh -f
tar xf truncfit.tar
make fittrunclm.o
cp fittrunclm.o $1
rm fittrunclm.o

cp trunc.fit.d $1/.Data/.Help/trunc.fit
cp trunclm.object.d $1/.Data/.Help/trunclm.object
cp init.beta.d $1/.Data/.Help/init.beta
cp no.groups.d $1/.Data/.Help/no.groups
cp setup.prop.d $1/.Data/.Help/setup.prop

rm trunc.fit.d init.beta.d no.groups.d setup.prop.d trunclm.object.d

rm minv.f matmult.c matmult.h matmult.o fitfunc.c fitfunc.o xblock.c xblock.o makefile m

setenv currdir $PWD
cd $1
Splus < $currdir/GTLRfuncs.S
cd $currdir
rm GTLRfuncs.S
unsetenv currdir
```

Appendix F

```
fittrunclm.o: minv.o matmult.o fitfunc.o xblock.o
        ld -r -d -o fittrunclm.o minv.o matmult.o fitfunc.o xblock.o

minv.o: minv.f
        f77 -c minv.f

matmult.o: matmult.c matmult.h
        cc -c matmult.c

fitfunc.o: fitfunc.c matmult.h
        cc -c fitfunc.c

xblock.o: xblock.c matmult.h
        cc -c xblock.c
```