PharmaSUG 2012 - SP09

Automated forward selection for Generalized Linear Models with Categorical and Numerical Variables using PROC GENMOD

Manuel Sandoval, Pharmanet-i3, Mexico City, Mexico

ABSTRACT

Generalized linear models are a powerful tool to measure relationships between variables, as they can handle non-normal distributions without altering the properties of variables involved. When applied to risk factor analysis, they can help determine the most important factors contributing to the incidence, prevalence or acquisition of a particular medical condition.

This paper presents a particular case in which the aforementioned factors are unknown and a selection must be made from a pool containing both numerical and class variables. Since the model uses an option that is only present in the GENMOD procedure (and not in the LOGISTIC procedure, for example), an algorithm for selecting variables needed to be created from scratch.

The proposed macro was built in such a case. Several factors, both numerical and categorical, were tested using forward selection and defined criteria for entering the model and for keeping the variable in the model. The macro also selects the numeric and categorical variables to include only the later in the class statement of the PROC GENMOD.

INTRODUCTION

Risk factor analysis is a useful technique used in many areas and with different applications. For instance, it's the way the Credit Risk Bureaus estimate our probability of default and grade us as potential credit customers. Marketing studies use it to identify potential buyers for their products or potential factors that could inhibit customers from buying them.

In the health industry, we use it, among other things, to identify possible external influences that can help or worsen in the development of a medical condition or in the process of its cure. In this way, we can estimate the influence that different factors have in the acquisition of a completely new medical condition, as the infection rate or prevalence of an existing one within a given population. The example here detailed is about the acquisition of different serotypes of bacteria after the application of the tested vaccine.

The factors that are usually investigated are those which are not controllable but which can also have a huge impact in the response of a therapy or the propagation of the medical condition. Demographic data such as age, gender, the number of siblings (in case of child studies), physical characteristics as height or weight, and social factors (if there are smoking people in the patient's immediate area or the generalities of his way of life, such as cooking or housing conditions) are usually associated to the propagation of sickness. Knowing them and their specific contribution can be as important as developing the medicine to treat that condition.

GENERALIZED LINEAR MODELS

The usual approach for analyzing the relationship between variables is to create a linear regression model with them, and verify if the resulting estimation is statistically significant. If it is, there is a relationship between variables. Of course, we cannot conclude that there is a *causal* relationship between the factors and the study variable, but at least we can look for parameters that can guide the studies to come.

However, there are some times in which the study variable (and so the model's residual) cannot be modeled plausibly with a normal distribution, and the number of observations is small enough that also an expectation that the result will turn sufficiently normal by aggregation of data is not attainable. In clinical trials, as the number of patients have to be kept as a minimum to improve safety, this is usually the case, and so, a more powerful technique needs to be used: the Generalized Linear Models.

A Generalized Linear Model is a regression model where the residuals, rather than being related to the model by an identity with a linear function of the factors, it's done by a link function, and so, can be applied to any member of the exponential family.

STUDY MODEL

The general model used was a generalized linear model (created with PROC GENMOD) relating the flag for new acquisitions (new) with treatment, visit and the different factors to be studied. the distribution was binomial, as needed for the proportion, and, as there was no reason to think the link function needed to be different, the canonical link, i.e. a logit function. A generic code for this model is reproduced here:

```
ods output parameterestimates = parms type3 = type3;

proc genmod data = risk_all;
 class treatment visit patient &factor_c;
model new = treatment visit &factor/dist = binomial link = logit type3;
repeated subject = patient/ type = exch printmle;
lsmeans treatment/cl diff;
run;
```

where &factor is a list of the complete factors to be studied and &factor_c the list of the categorical ones. Treatment, visit and patient number were also considered to be class variables, even when they could be numeric in order to show that there was no increasing value with its number and was a label only.

The contrast between treatments was also evaluated and so the lsmeans statement was needed with the diff options. Notice that the same patient could have different visits and different serotype readings even in the same visit (reacting to two different serotypes, presenting new antibodies for both of them). Hence, the repeated statement with the patient number, with an exchangeable structure, was needed, as it was considered to be very important in the algorithm development.

THE ALGORITHM

As there were many different factors (about 39 of them), the need for a selection method arose quickly. There are two main methods used for selecting variables, forward and backward selection. Backward selection is the most straightforward method and intends to reduce the model from the complete one (i.e. with all the factors considered) to the best ones, that is, one where all factors in the model have p-value less than a previously set threshold. The other one, forward selection starts with all the single factor models and select the best one. To this new model, the rest of the factors will be tested, creating the best two factor model. This process is iterated until there is no new variable whose p-value is less that the significance level desired.

Another variant, called stepwise selection, utilizes one method (either forward or backward, but mostly forward), adding the factors (or deleting them) according to their single factor model, and keeping them in the model whenever the model that include them had a p-value (for that value) less than a secondary significance threshold, set by the investigator too.

UNSUCCESFUL SOLUTIONS

Before studying the solution that actually worked on the data, several solutions were tried. A couple of them shall be presented, those with the most important design flaws or who promised a best bet of success.

Backward selection model

Being the most straightforward method, it's not surprising that a backward selection model was tried first. The strategy was to create a couple of lists with the factors (as in the general model): &factor with the whole list and &factor_c with the class factors. After that, sort the factors by p-value and deleting the one with the greatest value, if it was greater than 0.2. Iterate this until there is no factor with p-value greater than 0.2.

The code for each model was the same as in the general model presented above. Unfortunately, there was a problem with this approach from the very beginning. The complete model, containing all the factors, did not converge. That would mean that there was no possible initial model and so no possible selection. Even when the algorithm worked and the design was sound, there was no way of implementing it with the data available.

PROC LOGISTIC automatic selection

A second approach involved using PROC LOGISTIC to run the data. Since a Generalized Linear Model with binomial distribution and logit link (that is, logistic regression) was being used, and since PROC LOGISTIC has an automatic selection option embedded within the procedure options, the following code was tried to run the process, which had the extra advantage that it didn't need an algorithm.

```
ods output Type3=pval(rename=effect=parm);
proc logistic data = risk_all;
    class patient visit treatment &factor_c;
    model new = treatment visit &factor/
           slentry =.2 slstay = .25 selection = s include =2;
      repeated subject=patient /type=exch;
run;
```

This is virtually the same PROC GENMOD used above, modified for PROC LOGISTIC use. The options SLENTRY, SLSTAY and SELECTION = s indicate that a stepwise automatic selection should be used, with an entry significance level of 0.2 and a staying significance level of 0.25.

Unfortunately, *this code did not run.* The problem was the REPEATED statement, which was not supported by the LOGISTIC procedure. Removing it, we had a valid result, but different from what was expected. Without the REPEATED statement, the possibility of a patient having different results for the same visit was not considered anymore and the proportion could be overestimated easily. This approach was consistent with the data, and ran, but the initial design was flawed.

FORWARD SELECTION ALGORITHM

The last remaining alternative was to build the forward selection algorithm from scratch. In order to minimize the number of models that needed to be run, a stepwise selection model was created, considering susceptible for entry all those variables with a p-value in a single model less than 0.2, and with a p-value in the aggregated model of less than 0.25.

In order to do this, the first part of the macro was the construction of all single factor models and saving their results in a dataset (aggregated), where those with p-values greater than 0.2 could be easily filtered out and would be useful, at the same time, as the source for the factors that could be considered in the final model.

Issues to be solved

There were several issues that needed to be solved to create an algorithm like this one. The first issue, the need to have two significance thresholds, was already solved by using the series of single factor models and the aggregated dataset. However, we had to keep track of the variables that were inside the model, and to create the list to add to the MODEL statement in the PROC GENMOD.

Also, not all the variables were numerical (nor categorical) and so the list of factors presented in the MODEL statement and the CLASS statement could not be the same. For instance, if numeric variables were added to the CLASS statement, the result would be completely different, as SAS considers the values of those variables as levels of a categorical variable instead of different possible values of a numeric one. Also, if some factors that would not be included in the MODEL statement were added, SAS would consider that a parameter was missing from the data, and could take a degree of freedom from the model for it, even if it did not end as part of the model.

The possibility that some of the intermediate models would not converge was also quite present, as it happened with the complete model. This had to be addressed, because, even when the program in the interactive SAS environment ran and could be completed while running in batch, the error would stop the program and the final output would never be created.

The Code

Once the aggregated dataset was created, it was filtered to consider only the variables that could be included in the study. Also, the necessary set of macro variables needed to be initialized as null, as the variables had to exist in order to be used in the rest of the code. Two different variables had to be considered, one to fill the MODEL statement (&invars) and one to fill the CLASS one (&catvar).

However, these two variables were destined for the complete list of variables to be included in the model, and the latest factor to be included (the one which is under consideration) needed to have a different macro variable. Those variables were &testvar for the MODEL statement and &catvar_t for the CLASS statement. Lastly, in order to fill the macro variables with the new factor in case it passed the second threshold, two dummy macro variables were also needed (&tempcar and &tempcat).

The next steps were creating a new dataset (elegibles) with the factors which had single model p-values less or equal to 0.2, and sorting the factors by their p-value, so that they were considered in increasing order of significance.

```
data elegibles;
    set aggregate;
    where probchisq le 0.2;
run;

proc sort data = elegibles;
    by probchisq;
run;
```

The following iterative process was ready to be run as many times as eligible factors were in the data. The first record in the dataset was the one tested (it would be deleted at the end of the iteration), and the macro variables &testvar and &catvar_t were filled with the name of the factor.

```
%let i = 1;

proc sql noprint;
    select count(source) into: totalfactor from elegibles;
quit;

%do %while (&i le &totalfactor);

data elegibles;
set elegibles;
if _n_ = 1 then do;
    call symput('testvar',source);
    if lowcase(source) in (/*#list of categoric variables by name*/) then do;
           call symput('catvar_t',source);
           end;
    else do;
           call symput('catvar_t','');
           end;
    delete;
end;
run;
```

The model was run with both the aggregated list of factors and the newly considered factor. After that, the p-values of the model were tested to check if the new factor had the potential to remain in the model, that is, if its p-value was less than 0.25.

```
ods output parameterestimates = parms type3 = type3;
proc genmod data = risk_all;
   class patient visit treatment &catvar &catvar_t;
   model new = rtrtn visit &invars &testvar /dist = binomial link = logit type3;
   repeated subject = patient/ type = exch printmle;
   lsmeans treatment/cl diff;
run;

data stay;
    set type3;
    If source = "&testvar" and probchisq le 0.25 then do;
           call symput ('invars',"&tempvar"||" "||strip("&testvar"));
           call symput ('catvar',"&tempcat"||" "||strip("&catvar_t"));
    end;
    else if source = "&testvar" then delete;
run;
```

The macro variables &invars and &catvars were filled the factors that populated the dummy macro variables &tempvar and &tempcat. If the variable was greater than 0.25 the variable was deleted from the dataset and was not introduced in the study factors. After that, the dummy variables were filled with the remaining factors in &invars and &catvars, and the next step was ready.

```
%let tempvar = &invars;
%let tempcat = &catvar;
```

```
%let i = %eval(&i+1);

%end;
```

After the process was finished, the resulting model would need to be run one more time, as two problems could arise. First, the last factor considered could be not significant, and so, the model would need to be rerun to have the p-values without that variable. Also, if the last model did not converge, the model would also need to be rerun to have the values of the next to last model.

```
ods listing close;
ods output parameterestimates = parms type3 = type3;
proc genmod data = risk_all;
    class patient visit treatment &catvar ;
    model new = treatment visit &invars/dist = binomial link = logit type3;
    repeated subject = patient/ type = exch printmle;
    lsmeans treatment/cl diff;
run;
```

Lastly, when the program was tested to run in interactive environment, the NOSYNTAXCHECK option was added, so that the batch would not stop after finding an error in the log. It cannot be stressed enough that this option should only be added if there is complete certainty that the program actually runs and all the errors are due to problems with convergence in the model, and not due to programming issues.

CONCLUSION

The algorithm here presented could be used with only minor variations in any other Generalized Linear Model or Regression Model that uses stepwise selection. A simple forward selection can also be built, simplifying the algorithm as no second threshold would be needed. Inserting the possibility of eliminating any factor that passes the second threshold would also be possible, but all the different combinations that could happen would then need to be considered then.

REFERENCES

Mc Cullagh, P; Nelder, J.A. 1989. Generalized Linear Models. 532 pp. Boca Raton, Florida: Chapman and Hall: CRC.

Myers, Raymond. 2010. Generalized Linear Models: with Applications in Engineering and the Sciences. 496 pp. Hoboken, New Jersey: Wiley.

ACKNOWLEDGMENTS

I would like to thank all the Pharmanet-i3 Pfizer Alliance Programming team for their continuous support, specially to Diane Voran (the study team leader), James Trammel (the study, and chief statistician), Isaac Vazquez and Mark Matthews (the managers).

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Manuel Sandoval
Enterprise: Pharmanet-i3
Address: Insurgentes Sur 716, 4th floor
City, State: Mexico City, Mexico DF
Work Phone: (+52) 55 5005 5526
E-mail: msandoval@pharmanet-i3.com