SAS MACRO QIF MANUAL: Version 0.2

Peter X.-K. Song[1] **and Zhichang Jiang**[2]
[1] Department of Statistics and Actuarial Science, University of Waterloo,
Waterloo, Ontario, N2L 3G1, Canada
[2] Alberta Cancer Board, Edmonton, Alberta, Canada, T6G 1Z2

May 15, 2007

TITLE

QIF: a SAS macro for fitting marginal generalized linear models (MGLMs) or population-average models using Proc GENMOD (to generate initial values), Proc IML, and the Output Delivery System (ODS). It was developed under SAS/STAT Version 9.1.3.

MERITS of QIF

Like generalized estimating equations (GEE available in PROC GENMOD), QIF is also a quasi-likelihood inference method. It has been showed that

1) QIF gives consistent estimators of the regression coefficients even if the correlation structure is misspecified. GEE has the same property.

2) QIF estimators are of the same efficiency as GEE estimators when the correlation structure is correctly specified, but more efficient when the correlation structure is misspecified.

3) QIF gives a goodness-of-fit test for the validity of the first moment assumption pertaining to the unbiasedness of inference function. This assumption is crucial to ensure the consistency in estimation. GEE cannot provide this test.

4) QIF is robust against a small portion of outliers/contaminated data; refer to Qu, A. and Song, P. (2004), "Assessing robustness of generalised estimating equations and quadratic inference functions", Biometrika, 91, 447-459.

5) QIF is analogous to -2*log-likelihood, so it enables naturally to define some model selection criteria, such as Akaike Information Criterion (AIC) and Bayes Information Criterion (BIC).

HISTORY

Initial coding was developed in MATLAB by Annie Qu and her research group at Department of Statistics, Oregon State University.

This QIF macro was developed by Peter Song and Zhichang Jiang at Department of Statistics and Actuarial Science, University of Waterloo in August, 2006. Part of the macro for fitting normal MGLM was translated from the original MATLAB coding. The other part of the macro was independently developed.

DESCRIPTION

The macro uses quadratic inference function (QIF) to estimate regression coefficients in MGLMs, where their estimates are found by Newton-Raphson algorithm; refer to Qu, A, Lindsay, B. and Li, B. (2000), "Improving generalized estimating equations using quadratic inference functions", Biometrika, 87, 823-836. Also see Song (2007) "Correlated Data Analysis: Modeling Analytics and Applications" (Chapter 5), New York: Springer.

This macro will work on the following types of models with the corresponding error distributions and canonical link functions listed below.

Model	Error Distribution	Link
Linear	Normal	identity
Logistic	Binomial	logit
Log-Linear	Poisson	log
Gamma-Reg	Gamma	reciprocal

Version 0.2 of this macro allows four types of working correlation structures: Independence, Exchangeable (or Compound Symmetry), AR-1 and Unstructured. In the AR-1 case, only two major basis matrices out of the total three were used. The QIF with 1-dependence correlation structure is still under development and will be released at a later time.

Version 0.1 only allowed three types of working correlation structures.

This macro uses PROC GENMOD to give initial values to begin Newton-Raphson algorithm for the search of the minimizers of the QIF, under the same working correlation structure.

Under the independence working correlation, this macro gives exactly the same results as those given by PROC GENMOD.

SYNTAX

Syntax for the macro is similar to that of PROC GENMOD. There are other options that are macro-specific, however.

```
%qif(data=,
     yvar=,
     xvar=,
     id=,
     weight=1,
     dist=,
     corr=,
     descend=Y,
     print=Y,
```

```
      outpar=,
      outqif=,
      outcov=,
      outres=
)
```

where

`data` specifies the data set you are using. The format is exactly the same as required in PROC GENMOD. Missing values are handled by the listwise deletion approach, so this macro does a complete data analysis under the assumption of missing completely at random (MCAR) mechanism.

`yvar` specifies the name of response variable.

`xvar` specifies a set of covariates, with only space in between.

`id` specifies the id variable (numeric) of clusters or subjects.

`weight` specifies the weight of each data point. The Default is set to be 1, but it can be assigned to any appropriate weights. This is useful in dealing with missing data using for example Robins' inverse probability weighting approach.

`dist` specifies the error distribution type, including NORMAL, BIN, POISSON, and GAMMA. The link functions are set to be their respective canonical link, as listed in the above table.

`corr` specifies the working correlation structure. The current version includes IND (Independence), AR (autoregressive AR-1), EXCH (exchangeable or compound symmetry) and UNSTR (unstructured). It is hoped to upgrade this version soon to allow 1-dependence. (**The previous version 0.1 had no UNSTR option.**)

`descend` specifies the probability of success in the BIN error distribution. The Default is Y (yes).

`print` specifies a flag of whether an on-screen print is needed. The Default is Y (yes), i.e. print the outputs.

`outpar` specifies the name for an output data set. This data set contains parameter estimates, standard errors and other basic summary statistics.

`outqif` specifies the name for an output data set. This data set stores the goodness-of-fit test statistic, AIC and BIC model selection statistics. For the case of IND, the QIF is the same as the GEE; therefore, Version 0.2 does not output AIC or BIC.

`outcov` specifies the name for an output data set. This data set saves the asymptotic covariance matrix.

`outres` specifies the same for an output data set. This data set includes fitted values and residuals of Pearson type and deviance type which are useful to conduct residual analysis.

OUTPUT

The output from this macro is an optional printout of selected tables. The Default is "print=Y", but the print can be muted by saying "print=N". All of these tables can be extracted by specifying their names of the data sets in the macro.

EXAMPLE SYNTAX

1) First Example: Marginal log-linear model for the epileptic seizures count data (Diggle et al., 2002, "Analysis of Longitudinal Data", 2nd Ed., Oxford Press).

```
%qif(data=epilepsy, yvar=y, xvar=baseline trt logage visit bstr,
     id=id, dist=poisson, corr=exch,
     print=y, outpar=par1, outqif=qif1, outcov=cov1, outres=resid);
run;
```

2) Second Example: Marginal logistic model for the Indonesian children's health data (Diggle et al., 2002, "Analysis of Longitudinal Data", 2nd Ed., Oxford Press). The response is binary, the presence of absence of respiratory or diarrheal infection.

```
%qif(data=xerop, yvar=respinf, xvar=sex height cos sin xerophthalmia age age2,
     id=id, dist=bin, corr=exch, descend=Y,
     print=Y, outpar=par2, outqif=qif2, outcov=cov2,outres=binres);
run;
```

Detailed data analyses with related SAS outputs can be found in the book by Song (2007).

DISCLAIMER

We have made every effort possible to ensure that this macro is free of any bugs or errors, however in no way is the macro to be considered error or bug free. This is a free software that is only for the purpose of research and still under testing. You assume all responsibility for any damages that may result from any errors in the macro. We appreciate very much if you can report any error or bug to us to improve this macro.

ACKNOWLEDGEMENTS

We are very grateful to Annie Qu and her former PhD student Dr. Tsai for generously sharing their MATLAB code with us.

SOURCE CODE

Download file "QIFv02.sas" for the source code of this macro from

`http://www.stats.uwaterloo.ca/stats_navigation/downloadRequest.shtml`