Difference in p values by PROC LOGISTIC and elrm in R

Wolfgang M. Hartmann

2015

The reason for writing this small report was to find out what could be behind the fact that I get rather different p values for β parameters in exact logistic regression by SAS PROC LOGISTIC and the `elrm` algorithm in CRAN which I have added as function `xctlog` into CMAT.

To keep things simple I think that the one parameter exact logistic regression is very similar to Fisher's exact test. The first table shows the setup of the exact logistic regression problem in CMAT (left column of table) and in SAS (right column of the table).

```
data = [ 17  36  0  ,
         3   30  1  ];
cnam = [" cancer n treat "];
data = cname(data,cnam);

model = "1/2 = 3";
exct = 1;
optn = [ "print"          2 ,
         "rmc"            2 ,
         "burnin"         0 ,
         "iter"       50000 ,
         "trial"          2 ];
< gof,parm,conf > = xctlog(data,
                    model,exct,optn);
```

```
data exctlog1;
  input cancer treat num;
  datalines;
             1   0   17
             0   0   19
             1   1    3
             0   1   27
             ;
run;

proc logistic data=exctlog1 desc;
  freq num;
  model cancer = treat;
  exact treat / estimate=both;
run;
```

The code in CMAT obtains $p = 0.0013$

```
************************************************
Exact Logistic Regression (D. Zamar, 2013, R_MC=2)
************************************************

 Effect    Estimate P-value P_AsyStdErr Nsample Nunique
```

```
 treat -2.43127761  0.0013  7.292e-005   50000      15

Confidence Intervals for Estimates (alpha=0.05)
          Effect        Lower        Upper
           treat -6.219380610 -0.610403011
```

and PROC LOGISTIC

```
                    Exact Parameter Estimates

                                        95% Confidence
Parameter    Estimate           Limits           p-Value

treat         -2.0543     -3.8576     -0.6314     0.0019
```

The following table shows the setup of Fisher's exact test in CMAT (left column) and with PROC FREQ (right column):

```
oddr = 1.;
optn = [ "print"       2 ,
         "alpha"    0.05 ,
         "alt"    "twos" ,
         "meth"   "minl" ];
x = [ 17 19 , 3 27 ];
< pval, ci > = xctfishr(x,oddr,optn);
```

```
data;
do a = 1 to 2;
do b = 1 to 2;
  input wt1@@;
  output;
end; end;
cards;
  17  19  3  27
;

proc freq;
  weight wt1;
  tables a * b / exact;
run;
```

The resulting $p = 0.0012$ by CMAT

```
Exact two-sided Fisher's test (MinLike Method)
          Testing True Odds Ratio

     Probability= 0.0012 Odds Ratio=7.80127
Min. Likelihood Method 95 Percent CI: [0.4398,35.3363]
```

is the same as that obtained by PROC FREQ:

```
                    Fisher's Exact Test
```

```
Cell (1,1) Frequency (F)        17
Left-sided Pr <= F          0.9999
Right-sided Pr >= F      9.621E-04

Table Probability (P)    8.585E-04
Two-sided Pr <= P           0.0012
```

The differences in the p values seem to be small considering the fact that the result of the elrm method in CMAT is based on an MCMC algorithm (however, with 50000 samples drawn). But it seems to be remarkable how close the `elrm` result is to Fisher's exact test.

The following table shows for a larger number of examples, that the p values obtained with

- the `elrm` method of the CMAT function `xctlog` are all **very close** to those obtained with Fisher's exact test method,
- but the p values obtained by SAS PROC LOGISTIC are rather different and always larger.

Name	**xctlog elrm**	**xctfishr Two-sided**	**SAS Logistic**	**SAS FREQ Two-sided**	**SAS FREQ 2*Right-sided**
exctlog1: 1	0.0013	0.0012	0.0019	0.0012	0.0019
exctlog1: 2	0.0639	0.0626	0.0977	0.0626	0.0978
exctlog1: 3	0.0272	0.0280	0.0429	0.0280	0.0428
exctlog2: 1	0.1762	0.1773	0.2307	0.1773	0.2308
exctlog2: 2	0.1018	0.1027	0.1338	0.1027	0.1338
exctlog3: 1	0.0952	0.0954	0.1558	0.0954	0.1558
exctlog3: 2	0.0918	0.1016	0.2031	0.1016	0.2032
exctlog3: 3	0.0115	0.0111	0.0183	0.0111	0.0184
exctlog3: 4	1.0000	1.0000	1.0000	1.0000	1.0000
exctlog3: 5	1.0000	1.0000	1.0000	1.0000	1.0000
exctlog3: 6	0.0918	0.1016	0.2031	0.1016	0.2032
exctlog4: 1	0.0841	0.0852	0.1234	0.0852	0.1234
exctlog4: 2	0.0793	0.0869	0.1738	0.0869	0.1738
exctlog4: 3	0.0097	0.0092	0.0123	0.0092	0.0122
exctlog4: 4	1.0000	1.0000	1.0000	1.0000	1.0000
exctlog4: 5	1.0000	1.0000	1.0000	1.0000	1.0000
exctlog4: 6	0.0793	0.0869	0.1738	0.0869	0.1738
exctlog5: 1.1	0.0829	0.0834	0.1165	0.0834	0.1166
exctlog5: 1.2	0.1639	0.1659	0.2363	0.1659	0.2364
exctlog5: 1.3	0.0100	0.0095	0.0174	0.0095	0.0174
exctlog5: 2.1	0.0841	0.0852	0.1234	0.0852	0.1234
exctlog5: 2.2	0.0346	0.0372	0.0745	0.0372	0.0744
exctlog5: 2.3	0.0043	0.0042	0.0053	0.0042	0.0054

That simple model which reduces the general exact logistic regression problem to Fisher's exact test model illustrates that the R program `elrm` (and therefore the CMAT function `xctlog()`) and the SAS PROC LOGISTIC compute the p value differently:

- `elrm` computes the p value as in a two-sided test
- SAS PROC LOGISTIC computes the p value as twice the `Right-sided Pr >= F`.

In all our examples `elrm` never resulted in a p value larger than that which SAS PROC LOGISTIC obtained. Thanks to the developer Bob Derr, of SAS Institute, for pointing that difference out.