Splus functions to perform regression calibration for logistic regression with multiple surrogates for one exposure

Edie Weller, Ruifeng Li, Donna Spiegelman

May 4, 2004

Abstract

The Splus functions described in this documentation perform regression calibration for multiple surrogates with one exposure as discussed in the paper by Weller et al (submitted to Biostatistics, 2004). This type of data is often encountered in occupational studies where the measurement of exposure can be quite complex and is characterized by numerous factors of the workplace; therefore, multiple surrogates often describe one exposure. In this paper, methodology is developed along the lines of the regression calibration method to adjust the estimates of exposure-response associations for the bias and additional uncertainty due to exposure measurement error. The health outcome is assumed to be binary and related to the quantitative measure of exposure by a logistic link function. The relationship between the conditional mean of quantitative exposure measurement and job characteristics is assumed to be linear. A simulation option is available in the function to evaluate the performance (% bias, MSE and coverage probability) of the estimator for the data at hand.

Keywords: regression calibration, measurement error, multiple surrogates, occupational study

Contents

1 Description

$Multisurr$ is a Splus function with two sub-functions: $adjfun$, for calculating the adjusted coefficients of exposure effect-response associations from regression calibration with multiple surrogates for one exposure; and $mesimfn$, for simulation study to compute % bias and MSE for assessing the performance of the estimator using the data at hand.

This function is appropriate for the following situation:

- One exposure is considered with multiple surrogates measured for this exposure.
- The measurement error model is linear.
- The logit of the probability of the binary outcome and the surrogates (and other covariates measured without error) is linear.
- The data can be divided into two parts:

  (1) Validation study data: the true exposure information, multiple surrogates and covariates measured without error are available on all n_2 subjects .

  (2) Main study data: the multiple surrogates, covariates measured without error, and the binary outcome are available on all n_1 subjects. There is no information about the true exposure on these subjects.

The inputs for the Splus function include the binary outcome and surrogates in main study data, the true exposure quantity and surrogates in validation study data. If some other covariates measured without error are needed to adjust the exposure-effect association, they should be available in both main study and validation study data. Also there are options to include confounders or not to do simulation study (and if so, what is the size). (For detail, see the next section)

The outputs from the Splus function include the estimates from logistic regression model in main study; the estimates from measurement error model in validation study; the adjusted coefficients and corresponding OR, 95% CI and p-values by combining the information from main and validation study;

and covariance between adjusted coefficients, surrogates, etc. (for detail, see the example output).

2 Invocation

The Splus code should include three parts: define Splus main function *multisurr* and its two sub-functions *adjfun* and *mesimfn*; read data into Splus for the input parameters of *multisurr*; and plug these values into *multisurr*. More specific, to use the Splus main function *multisurr*, the user must execute a three-step invocation:

Step 1

Make the Splus function *multisurr* and its two sub-functions ($adjfun$ and $mesimfn$) as Splus objects. In Splus mode:

source("adjfun.s")

source("mesimfn.s")

source("multisurr.s")

Step 2

Input values for the following Splus objects in Splus:

(1) Main study data with n_1 subjects:

- outcome: vector of disease outcome ($n_1 \times 1$)
- surrogatesInMain: matrix of surrogates ($n_1 \times r$)
- confoundersInMain: matrix of perfectly measured covariates ($n_1 \times s$), optional

(2) Validation study data with n_2 subjects:

- trueExposure: vector of correctly measured exposure ($n_2 \times 1$)
- surrogatesInValid: matrix of surrogates ($n_2 \times r$)

- confoundersInValid: matrix of perfectly measured covariates ($n_2 \times s$), optional

(3) includeConfounders:

- T: if the study includes perfectly measured covariates;
- F: otherwise (default is T).

(4) simulation:

- T: if needs the simulation to measure how good of the estimates. And *nsim*, size of simulation, is needed (default is 2000)
- F: otherwise (default is F)

Notice that if includeConfounders is F, skip the two options confoundersInMain, confoundersInValid.

(5) weight for uncorrected logistic model to give OR by the certain unit increase: vector to provide the increment of OR for each covariate in the uncorrected logistic model with interception as the first element . The length should be the total number of covariances in the model, including interception term, and the order should be exactly same as in the model. For example, if there are 3 surrogates and 2 confounders, and you would like to have 5 units increase for the first surrogate (continuous), then, wt1=c(1,5,1,1,1,1). By default, it is one unit increase.

(6) weight for corrected logistic model to give OR by the certain unit increase: vector to provide the increment of OR for each covariate in the corrected logistic model with interception as the first element. The length should be the total number of covariances in the model, including interception term, and the order should be exactly same as in the model. Using the same example as above, then wt2=c(1,5,1,1,1,1). By default, it is one unit increase.

(7) digits: to control printing, can change number of significant digits by setting digits (default is 4)

Step 3

Invoke *multisurr* with the specific values from step 2 by the following:

if includeConfounders is T and simulation study is not required, then in Splus mode, type

```
multisurr(outcome=maind, surrogatesInMain=mainw, trueExposure=validx,
surrogatesInValid=validw, includeConfounders=T, confoundersInMain=mainz,
confoundersInValid=validz)
```

if includeConfounders is F and simulation study is not required, then in Splus mode, type

```
multisurr(outcome=maind, surrogatesInMain=mainw, trueExposure=validx,
surrogatesInValid=validw, includeConfounders=F)
```

if includeConfounders is T and simulation study is required with 2000 simulations, then in Splus mode, type

```
multisurr(outcome=maind, surrogatesInMain=mainw, trueExposure=validx,
surrogatesInValid=validw, includeConfounders=T, confoundersInMain=mainz,
confoundersInValid=validz, simulation=T)
```

Of course, you can name Splus objects as different names and change correspondingly the right side of =. For example, instead of using *maind*, you may use the real outcome name.

3 Illustrative Example

In this section, we present an example of the use of the program *multisurr*, assuming it is an appropriate situation described in section *Description*.

The main dataset is a subset (with sample size n1=1040) of the workers in Greave et al.'s epidemiology study (1997) of auto workers; and the validation dataset is a subset (with sample size n2=83) of the workers in the exposure assessment study (Woskie et al., 1994). Suppose they are saved in the directory */udd/strui/edie/* in SAS datasets format with names **main.ssd01** and **valid.ssd01**. Notice that there is no missing value allowed for any relevant variable in the two datasets.

Suppose all the files are saved in the current directory, here is the example

Splus code in the file named **example.sn**

```
# define Splus function adjfun, for calculation adjusted coefficients
source(''adjfun.s'')

# define Splus function mesimfn, for simulation
source(''mesimfn.s'')

#define Splus main function multisurr, which invokes the above
#two sub-functions
source(''multisurr.s'')

# read SAS dataset valid.ssd01 into Splus.
# notice na.omit in the front of sas.get to avoid reading in missing values
valid _ na.omit(sas.get(".", mem="valid",
   var=c("truex","plant2","grinding","str","syn","agecat1","agecat2",
         "agecat3","racec","smokenow")))

# truex is the true exposure name in the validation dataset, which is
# the thoracic aerosol fraction in mg/m^3
validx_validm[,"truex"]

# The surrogates for this exposure are pant (1 or 2), machine type
# (grinding or not grinding), and metal working fluids type
# (no fluid, straight cutting oils or synthetic)
# use plant 1, not grinding, no fluid as references groups correspondingly
validw_validm[,c("plant2","grinding","str","syn")]

# Age, race and smoking status were the perfectly measured potential
# confounders of the (outcome, true exposure) association
validz_validm[,c("agecat1","agecat2","agecat3","racec","smokenow")]

# read SAS dataset main.ssd01 into Splus.
# notice na.omit in the front of sas.get to avoid reading in missing values
main _ na.omit(sas.get("/udd/strui/edie", mem="main",
   var=c("weezmost","plant2","grinding","str","syn","agecat1","agecat2",
         "agecat3","racec","smokenow")))
```

```
# binary outcome: prevalence of wheeze, named ''weezmost'' in the data
maind_main["weezmost"]

# surrogates, same as above
mainw_main[,c("plant2","grinding","str","syn")]

# confounders, same as above
mainz_main[,c("agecat1","agecat2","agecat3","racec","smokenow")]

# input the values into multisurr function
# notice the parameter orders.
# since just 1 unit increase, wt1, wt2 are default value, so you
# don't need to give values. Here is just show how to use them.
 multisurr(outcome=maind, surrogatesInMain=mainw, trueExposure=validx,
           surrogatesInValid=validw, includeConfounders=T,
           confoundersInMain= mainz,confoundersInValid=validz,
           wt1=c(1,1,1,1,1,1,1,1,1,1), wt2=c(1,1,1,1,1,1,1)),
```

Once you have done editing the above Splus file, save it as example.s, the run in any unix window with Splus software as the following:

```
Splus BATCH example.s example.output
```

Then the output is saved as a file called **example.output**, which looks like the following (in order to have shorter output, we suggest to source the three Splus functions in advance, so in the Splus file, you don't need to source them again to have a shorter and neater output):

```
(omit the code for the three functions)
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+      The related results from the real data are:                  +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

The results from fitting logistic regression model:
--------------------------------------------------
          Weights Estimate    S.E   Odds 95% LL  95%UL
Intercept       1  -2.5390 0.2665 0.0789 0.0468 0.1331
```

```
   plant2       1   0.7461 0.2125 2.1088 1.3905 3.1982
 grinding       1  -0.3487 0.3242 0.7056 0.3737 1.3320
      str       1   0.4955 0.1953 1.6414 1.1193 2.4070
      syn       1   0.6155 0.2211 1.8506 1.1997 2.8546
  agecat1       1  -0.1093 0.1926 0.8965 0.6147 1.3076
  agecat2       1  -0.1819 0.2490 0.8337 0.5118 1.3582
  agecat3       1  -0.0921 0.2633 0.9120 0.5443 1.5281
    racec       1   0.1595 0.1979 1.1729 0.7958 1.7286
 smokenow       1   1.1127 0.1631 3.0425 2.2100 4.1886
```

The adjusted coefficients and corresponding SE, OR, 95% CI:

```
          weights   Coeff   S.E. Wald Score   Odds 95% LL 95% UL pvalue
intercept       1 -2.7120 0.2911    86.8271 0.0664 0.0375 0.1175 0.0000
 beta exp       1  1.0560 0.3845     7.5415 2.8749 1.3530 6.1085 0.0060
  agecat1       1 -0.0355 0.2031     0.0305 0.9651 0.6481 1.4372 0.8613
  agecat2       1 -0.1590 0.2594     0.3754 0.8530 0.5130 1.4184 0.5401
  agecat3       1 -0.0902 0.2734     0.1088 0.9138 0.5347 1.5615 0.7415
    racec       1  0.1538 0.2043     0.5669 1.1663 0.7814 1.7408 0.4515
 smokenow       1  1.0914 0.1677    42.3337 2.9784 2.1439 4.1378 0.0000
```

The variance-covariance matrix of the estimated adjusted coefficients beta:

```
            (intercept) exposure agecat1 agecat2 agecat3   racec smokenow
(intercept)      0.0847  -0.0358 -0.0309 -0.0351 -0.0383 -0.0147  -0.0180
   exposure     -0.0358   0.1479  0.0051 -0.0087 -0.0165 -0.0116   0.0006
    agecat1     -0.0309   0.0051  0.0413  0.0276  0.0276  0.0017  -0.0015
    agecat2     -0.0351  -0.0087  0.0276  0.0673  0.0337  0.0062  -0.0044
    agecat3     -0.0383  -0.0165  0.0276  0.0337  0.0747  0.0041  -0.0007
      racec     -0.0147  -0.0116  0.0017  0.0062  0.0041  0.0418  -0.0009
   smokenow     -0.0180   0.0006 -0.0015 -0.0044 -0.0007 -0.0009   0.0281
```

The results from fitting linear ME model:

```
          Estimate    S.E  t value Pr(>|t|)
Intercept   0.1506 0.0711   2.1190   0.0375
   plant2  -0.0358 0.0751  -0.4767   0.6350
 grinding   0.0985 0.0669   1.4728   0.1451
```

```
      str   0.5010 0.0480  10.4347   0.0000
      syn   0.2982 0.0614   4.8548   0.0000
  agecat1  -0.0698 0.0614  -1.1382   0.2588
  agecat2  -0.0217 0.0718  -0.3019   0.7636
  agecat3  -0.0019 0.0699  -0.0265   0.9789
    racec   0.0053 0.0471   0.1131   0.9103
 smokenow   0.0202 0.0381   0.5295   0.5981

The results for the adjusted exposure parameters
------------------------------------------------
         Estimate     S.E   Odds 95% LL        95%UL Wald Chi-SQ p-value
  plant2 -20.8380 44.1143 0.0000 0.0000 3.164934e+28      0.2231  0.6367
grinding  -3.5418  4.0773 0.0290 0.0000 8.559320e+01      0.7546  0.3850
     str   0.9891  0.4012 2.6888 1.2247 5.903200e+00      6.0768  0.0137
     syn   2.0644  0.8549 7.8809 1.4752 4.210040e+01      5.8311  0.0157

The optimal weights for the surrogates of exposure:
---------------------------------------------------
     plant2 grinding    str    syn
[1,]      0   0.0154 0.8573 0.1273
```

4 Warnings

Make sure there are no missing values in validation dataset for true exposure, all surrogates, and adjusted covariances; in main dataset for outcome, all surrogates, and adjusted covariances.

When both validation and main datasets are in SAS dataset format, the way to avoids this missing value problem is to add Splus function *na.omit* in the front of *sas.get* when reading SAS datasets into Splus. If the dataset are in other format, also make sure to delete the whole observation if missing value happens in any of the variables.

In addition, make sure the order to input parameter values is correct.

Since the funcitons are written in Splus, make sure it obeys all the rules in

Splus.

5 Credits

The original Splus functions were written by Edie Weller, Ruifeng Li has extended them to the current version. Dr. Donna Spiegelman has given valuable suggestions. Questions can be directed to Ruifeng Li:

strui@channing.harvard.edu

6 See Also

There is a SAS version written by Ruifeng Li, please refer the corresponding manuscript for help.

There is another SAS macro called *%blinplus* to deal with multiple true exposure, each of them can only have one surrogate, please refer the corresponding manuscript for help.

7 References

Weller E., Spiegelman D., Milton D., Eisen E, *Regression Calibration for Logistic Regression with Multiple Surrogates for One Exposure*

Greaves IA, Eisen EA, Smith TJ, Pothier LJ, Kreibel D, Woskie SR, Kennedy SM, Shalat, S and Monson, RR (1997). Respiratory health of automobile workers exposed to metal-working fluid aerosols: respiratory symptoms. *American Journal of Industrial Medicine* **32**, 450-459.

Woskie SR, Smith TJ, Hallock MF, Hammond, SK, Rosenthal F, Eisen EA, Kreibel D, Greaves IA (1994). Size-selective pulmonary dose indices for metal-working fluid aerosols in machining and grinding operations in the automobile manufacturing industry. *American Industrial Hygiene Association* **55**,20-29.