

A short Introduction to the SIMEX and MCSIMEX

by Wolfgang Lederer and Helmut Küchenhoff

In statistical practice variables are often contaminated with measurement error. This may be the case due to bad measurement tools or just because the true variable can not be measured directly. In the case of discrete variables, measurement error is referred to as misclassification. In the framework of general regression measurement error or misclassification can lead to serious bias in the estimated parameters. In most cases the estimated effect of the contaminated variable is attenuated

Among many other methods the simulation and extrapolation method (SIMEX) by Cook and Stefanski (1994) has become a useful tool for correcting effect estimates in the presences of additive measurement error. The method is especially helpful for complex models with a simple measurement error structure. The same basic idea of simulation and extrapolation is transferred to the case of misclassification (MC-SIMEX) by Küchenhoff et al. (2005).

The R package `simex` provides functions to use the SIMEX or MC-SIMEX methods for various kinds of regression objects and to produce graphics and summary statistics for corrected objects. There are also functions to provide help in constructing misclassification matrices.

Theory

The SIMEX-method uses the relationship between the size of the measurement error, described by the measurement error variance σ_u^2 and the bias of the effect estimator when ignoring the measurement error. So we can define the function

$$\sigma_u^2 \longrightarrow \beta^*(\sigma_u^2) := \mathcal{G}(\sigma_u^2)$$

where β^* is the limit to which the naive estimator converges as the sample size $n \rightarrow \infty$. A consistent estimator of β when there is no measurement error is usually called the naive estimator if it is used although there is measurement in the data. It is easily seen, that $\mathcal{G}(0) = \beta$ is the true parameter, and $\mathcal{G}(\sigma_u^2) = \beta_{na}$ the result of the naive estimator. The idea of the SIMEX method is to approximate the function $\mathcal{G}(\sigma_u^2)$ by a parametric approach $\mathcal{G}(\sigma_u^2; \Gamma)$, for example with a quadratic approximation $\mathcal{G}_{quad}(\sigma_u^2; \Gamma) = \gamma_0 + \gamma_1 \sigma_u^2 + \gamma_2 (\sigma_u^2)^2$.

To estimate Γ the method adds in the simulation step to a given data set additional measurement error with variance $\lambda \sigma_u^2$ to the contaminated variable. The resulting measurement error variance is then $(1 +$

$\lambda) \sigma_u^2$. The naive estimator for this increased measurement error is calculated and repeated B times. The average over B converges to $\mathcal{G}((1 + \lambda) \sigma_u^2)$. Repeating this simulation for a fixed grid of λ , leads to an estimator for $\hat{\Gamma}$ of the parameters $\mathcal{G}(\sigma_u^2, \Gamma)$, for example by least squares.

In the extrapolation step the approximated function $\mathcal{G}(\sigma_u^2, \hat{\Gamma})$ is extrapolated back to the case of no measurement error and so the SIMEX estimator is defined by $\beta_{simex} := \mathcal{G}(0, \hat{\Gamma})$, which corresponds to $\lambda = -1$.

The misclassification error can be described by the misclassification matrix Π which is defined via its components

$$\pi_{ij} = P(X^* = i | X = j)$$

where X^* is the misclassified version of X . Π is a $k \times k$ matrix where k is the number of possible outcomes of X . The estimator β^* is depending on the amount of misclassification and therefore defined by

$$\lambda \longrightarrow \beta^*(\Pi^\lambda)$$

where Π^λ is defined via its spectral decomposition $\Pi^\lambda := E \Lambda^\lambda E^{-1}$, with Λ being the diagonal matrix of eigenvalues and E the corresponding matrix of eigenvectors. This allows the SIMEX method to be applied to misclassification problems. The MC-SIMEX estimator is then defined by the parametric approximation of the function

$$\lambda \rightarrow \beta^*(\Pi^\lambda) \approx \mathcal{G}_\Pi(1 + \lambda, \Gamma).$$

In the simulation step we simulate B new pseudo data sets for a fixed grid of λ by the misclassification operation defined by

$$X_i^* := MC[\Pi^\lambda](X_i).$$

The misclassification operation $MC[M](X_i)$ generates by simulation a misclassified version of the true, but unknown, variable X_i denoted by X_i^* which is related to X_i by the misclassification matrix M . For each of this B pseudo data sets the naive estimators are calculated and averaged for each λ . These averaged naive estimators converge to $\mathcal{G}_\Pi(1 + \lambda, \Gamma)$ and the estimation of Γ via e.g. least squares and so an approximation of $\mathcal{G}_\Pi(1 + \lambda, \Gamma)$ is feasible.

The MCSIMEX estimator is then defined by

$$\beta_{MCSIMEX} := \mathcal{G}(0, \hat{\Gamma}).$$

which corresponds again to $\lambda = -1$.

Variance estimation

The ease of getting corrected parameter estimates is somewhat offset by the complexity of the calculation of the parameter's standard error. With its simulation character it is a natural candidate for the bootstrap, although it is a valid method for obtaining standard errors it is rather time consuming and for complex models not feasible. There were two methods developed, for the estimation of standard errors which are a smaller computational burden. The jackknife method was developed for the SIMEX method by Stefanski and Cook (1995). For the MCSIMEX method, it lacks theoretical foundation but simulation results indicate valid estimates for the standard errors.

An asymptotic approach based on estimation equations was developed by Carroll et al. (1996) for the SIMEX method and extended to the MCSIMEX method by Küchenhoff et al. (2006). It is possible to take the uncertainty of an estimated misclassification matrix or an estimated measurement error variance into account.

Example

To illustrate the application of the `simex` package, we use a data set of a study about chronic bronchitis and dust concentration of the Deutsche Forschungsgemeinschaft (German research foundation). The data were recorded during the years 1960 and 1977 in a Munich plant (1246 workers). The data can be downloaded from http://www.stat.uni-muenchen.de/service/datenarchiv/dust/dust_e.html.

The data set contains 4 variables described in table 1, and is read into the data.frame `dat` via the `read.table` command.

<code>cbr</code>	Chronic Bronchitis Reaction	1: Yes 0: No
<code>dust</code>	Dust concentration at work	(in mg / m ³)
<code>smoking</code>	Does worker smoke?	1: Yes 0: No
<code>expo</code>	Duration of exposure	in years

Figure 1: Description of variables

The naive model is then given by

```
> naive <- glm(cbr ~ dust + smoking + expo
+             , family= binomial
+             , data =dat, x=T, y=T)
```

mind that `cbr` and `smoking` must be factors. The options 'x','y' must be enabled for asymptotic variance estimation.

Continuous data

It is possible, that the the variabel `dust` is subject to measurement error. Because it is a continuous variable the SIMEX-method is to be used here. Let us

assume that there is a measurement error with standard deviation $\sigma = 2$ the correction via the `simex` function would be made as follows

```
> mod.sim <- simex(naive
+                 , measurement.error = 2
+                 , SIMEXvariable = "dust")
> mod.sim
```

Naive model:

```
glm(formula = cbr ~ dust + smoking + expo
     , family = binomial
     , data = dat, x = T, y = T)
```

SIMEX-Variables: dust

Number of Simulations: 100

Coefficients:

```
(Intercept)      dust      smoking1      expo
-3.16698      0.13549      0.67842      0.03969
```

The default extrapolation method for the function `simex` is quadratic, although the most appropriate extrapolation would be a nonlinear. Unfortunately The nonlinear extrapolation is numerically not stable and it is therefore advised to use it via the `refit` function. The `refit` function fits a new extrapolation function to the data obtained by the simulation step and yields therefore different estimators. It can be applied to objects of class MCSIMEX as well.

```
> refit(mod.sim, "nonl")
```

Naive model:

```
glm(formula = cbr ~ dust + smoking + expo
     , family = binomial
     , data = dat, x = T, y = T)
```

SIMEX-Variables: dust

Number of Simulations: 100

Coefficients:

```
(Intercept)      dust      smoking1      expo
-3.33167      0.18904      0.67854      0.03956
```

A comparison of both extrapolation functions is shown in figure 2

Discreet Data

It is known, that some participants lie when they are asked about their smoking behavior. Research indicates, that about 8% of smokers self-report them as non-smokers, so the misclassification matrix for smoking is defined by

```
> mc.s <- matrix(c(1,0,0.08,0.92),nrow=2)
> dimnames(mc.s) <- list(levels(dat$smoking)
+                        ,levels(dat$smoking))
> mc.s
```

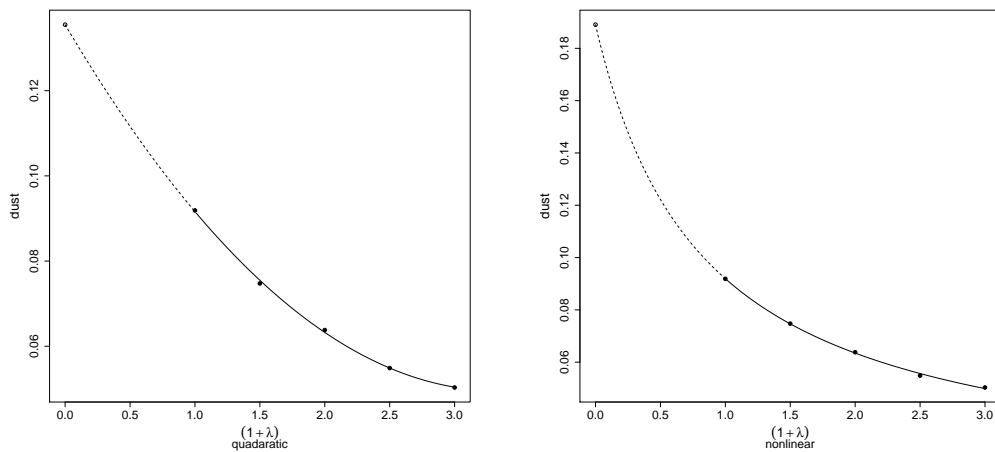


Figure 2: The effects of measurement error in variable *dust* produced with `plot(mod.sim), ask=c(F,T,F,F)` and `plot(refit(mod.sim,"nonl"), ask=c(F,T,F,F))`

```
0 1
0 1 0.08
1 0 0.92
```

and so the MCSIMEX-Algorithm can be used by calling the function `mcsimex()` an a quick overview can be obtained using the `print` method.

```
>mod.smoking <- mcsimex(naive
+                       , mc.matrix = mc.s
+                       , SIMEXvariable = "smoking")
> mod.smoking
```

```
Naive model:
glm(formula = cbr ~ dust + smoking + expo,
    family = binomial, data = dat
    , x = T, y = T)
```

```
SIMEX-Variables: smoking
Number of Simulations: 100
```

```
Coefficients:
(Intercept)      dust      smoking1      expo
-3.25827      0.09269      0.88086      0.04026
```

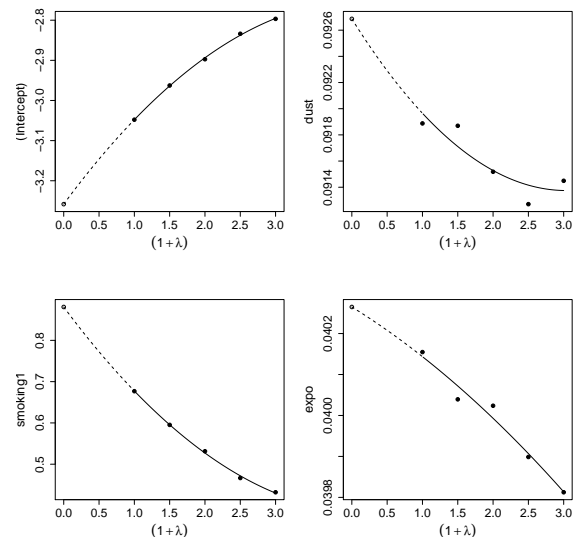


Figure 3: The effects of misclassification of smoking produced with `plot(mod.smoking,mfrow = c(2,2))` method.

More detailed information is available through the summary function, as shown in table 4 and the `plot` method which is shown in figure 3. Mind that the correction of the effect estimates *intercept*, *dust* and *expo* are rather small in absolut value, but that *smoking* is corrected rather strongly.

Recent research leads to the conclusion, that misclassification might as well occur in the response *cbr*. It is possible to correct for just the response or for both. In the following the code for correction of the response *cbr* and the variable *smoking* is shown

```
> mc.cbr <- matrix(c(0.8,0.2,0.1,0.9),nrow=2)
> dimnames(mc.cbr) <- list(levels(dat$cbr)
+                           ,levels(dat$smoking))
```

```

> summary(mod.smoking)

Call: mcsimex(model = naive, SIMEXvariable = "smoking", mc.matrix = mc.s)

Naive model:
glm(formula = cbr ~ dust + smoking + expo, family = binomial,
     data = dat, x = T, y = T)

Simex variable : smoking
Misclassification matrix:
  0  1
0 1 0.08
1 0 0.92

Number of iterations: 100

Residuals:
      Min.      1st Qu.      Median      Mean      3rd Qu.      Max.
-0.610900 -0.259600 -0.149800  0.006416 -0.057690  0.941900

Coefficients:

Asymptotic variance:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.258272   0.286575 -11.370 < 2e-16 ***
dust          0.092687   0.023967   3.867 0.000116 ***
smoking1      0.880861   0.240993   3.655 0.000268 ***
expo          0.040265   0.005942   6.777 1.89e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Jackknife variance:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.258272   0.280754 -11.605 < 2e-16 ***
dust          0.092687   0.023366   3.967 7.70e-05 ***
smoking1      0.880861   0.217463   4.051 5.42e-05 ***
expo          0.040265   0.006251   6.442 1.69e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Figure 4: Output of summary(mod.smoking)

```

> mc.cbr
  0  1
0 0.8 0.1
1 0.2 0.9
> mod.both <- mcsimex(naive
+   , mc.matrix =
+   list(smoking = mc.s, cbr = mc.cbr)
+   , SIMEXvariable = c("cbr", "smoking"))
> mod.both

```

```

Naive model:
glm(formula = cbr ~ dust + smoking + expo
     , family = binomial
     , data = dat, x = T, y = T)

```

```

SIMEX-Variables: cbr, smoking
Number of Simulations: 100

```

```

Coefficients:
(Intercept)      dust      smoking1      expo
-5.58519      0.16081      1.36969      0.07154

```

It is possible to model more complex kinds of misclassification e.g. dependent misclassification, by submitting instead to the function `mcsimex` of an misclassification matrix the name of a function, that returns the misclassified variables.

Summary

The package **simex** features easy to use functions for correcting estimation in regression models with measurement error or misclassification via the SIMEX- or MCSIMEX-method. It provides fast and easy means to produce plots that illustrate the effect of

measurement error or misclassification on parameters. Several additional functions are available that help with various problems concerning misclassification or measurement error.

Wolfgang Lederer
Ludwig-Maximilians-Universität München
Wolfgang.Lederer@googlegmail.com

Helmut Küchenhoff
Ludwig-Maximilians-Universität München
Kuechenhoff@stat.uni-muenchen.de

Bibliography

- R. J. Carroll, H. Küchenhoff, F. Lombard, and L. A. Stefanski. Asymptotics for the SIMEX Estimator in Nonlinear Measurement Error Models. *Journal of the American Statistical Association*, 91:242–250, 1996.
- J. R. Cook and L. A. Stefanski. Simulation-Extrapolation Estimation in Parametric Measurement Error Models. *Journal of the American Statistical Association*, 89:1314–1328, 1994.
- H. Küchenhoff, W. Lederer, and E. Lesaffre. Asymptotic Variance Estimation for the Misclassification SIMEX, 2006. submitted.
- H. Küchenhoff, S. M. Mwalili, and E. Lesaffre. A general method for dealing with misclassification in regression: the Misclassification SIMEX. *Biometrics*, 2005. Accepted, available online.
- L. A. Stefanski and J. R. Cook. Simulation-Extrapolation: The Measurement Error Jackknife. *Journal of the American Statistical Association*, 90: 1247–1256, 1995.