Estimating Equations in R: `geex`

B. Saul

2017-07-07

M-estimation theory provides a framework for asympotic properties of estimators that are solutions to estimating equations. Regression methods such as Generalized Linear Models (GLM) and Generalized Estimating Equations (GEE) fit in this framework. Countless R packages implement specific applications of estimating equations. A common reason to use M-estimation is to compute the empirical sandwich variance estimator - an asymptotically Normal and "robust" covariance. Many packages compute this variance estimator automatically, and packages such as `sandwich` take the output of other modeling methods to compute this variance estimate.

`geex` aims to be provide a more general framework that any modelling method can use to compute point and variance estimates for parameters that are solutions to estimating equations. The basic idea:

- Analyst provides three things: (1) data, (2) instructions on how to split the data into independent units and (3) a function that takes unit-level data and returns a function in terms of parameters.
- `geex` computes point estimates and variance estimates for the parameters.

Basic Setup

I mostly follow the notation of Stefanski and Boos. I tried to keep notation in the code similar to mathematical notation.

Suppose we have m independent or nearly independent units of observations.

$$\sum_{i=1}^{m} \psi(O_i, \theta) = 0$$

Where ψ is vector of length p corresponding to the number of parameters in θ.

For notational ease, let $\psi(O_i, \theta) = \psi_i$ Let:

$$A_i = -\frac{\partial \psi(O_i, \theta)}{\partial \theta}$$

$$A = \sum_{i=1}^{m} A_i$$

$$B_i = \psi_i \psi_i^T$$

$$B = \sum_{i=1}^{m} B_i$$

$$\Sigma = A^{-1} B \{A^{-1}\}^T$$

Stefanski & Boos example 1

Example 1 illustrates calculation of sample mean and variance using estimating equations. I generate a data set with 100 observations drawn from a Normal(5, 2) distribution. Table translates the estimating equations into the `R` function needed for `geex`:

Table 1: Translating math to code

$$\psi(Y_i, \theta) = \begin{pmatrix} Y_i - \theta_1 \\ (Y_i - \theta_1)^2 - \theta_2 \end{pmatrix}$$

```
SB1_eefun <- function(data){
  function(theta){
    with(data,
      c(Y - theta[1],
       (Y - theta[1])^2 - theta[2] )
    )
  }
}
```

With the `eeFUN` function prepared, it is passed to `estimate_equations` along with the data, a character string naming the variable that identifies groups within the dataset, and starting values for the root finder.

```
estimates <- estimate_equations(
  eeFUN = SB1_eefun,
  data  = dt,
  rootFUN_control = list(start = c(1,1)))
```

```
## When units are not specified, each observation is considered independent.
```

\$geex *geex*estimates [1] 5.022282 2.510310

*geex*vcov [,1] [,2] [1,] 0.025103097 0.004584961 [2,] 0.004584961 0.155762273

\$cls *cls*parameters p1 p2 1 5.022282 2.51031

*cls*vcov [,1] [,2] [1,] 0.025103097 0.004584961 [2,] 0.004584961 0.155762273

Stefanski & Boos example 2

Example 2 illustrates calculation of a ratio estimator. I generate a data set with 100 observations where $Y \sim N(5, 2)$ and $X \sim N(2, 0.2)$. Table translates the estimating equations into the `R` function needed for `geex`:

Table 2: Translating math to code

$$\psi(Y_i, \theta) = \begin{pmatrix} Y_i - \theta_1 \\ X_i - \theta_2 \\ \theta_1 - \theta_3\theta_2 \end{pmatrix}$$

```
SB2_eefun <- function(data){
  function(theta){
    with(data,
      c(Y - theta[1],
        X - theta[2],
        theta[1] - (theta[3] * theta[2]) )
    )
  }
}
```

```
estimates <- estimate_equations(
  eeFUN = SB2_eefun,
```

```r
  data  = dt,
  rootFUN_control = list(start = c(1, 1, 1)))
```

```
## When units are not specified, each observation is considered independent.
```

$geex *geex*estimates [1] 5.025586 1.983351 2.533887

*geex*vcov [,1] [,2] [,3][1,] 0.0427723672 -0.0005003119 0.0222048960 [2,] -0.0005003119 0.0004626310 -0.0008433034 [3,] 0.0222048960 -0.0008433034 0.0122730329

$cls *cls*parameters p1 p2 p3 1 5.025586 1.983351 2.533887

*cls*vcov [,1] [,2] [,3][1,] 0.0432044114 -0.0005053656 0.0224291879 [2,] -0.0005053656 0.0004673041 -0.0008518216 [3,] 0.0224291879 -0.0008518216 0.0123970030

Stefanski & Boos example 3

Example 3 illustrates calculation of a ratio estimator. I generate a data set with 100 observations where $Y \sim N(5, 4)$. Table translates the estimating equations into the `R` function needed for `geex`:

Table 3: Translating math to code

$$\psi(Y_i, \theta) = \begin{pmatrix} Y_i - \theta_1 \\ X_i - \theta_2 \\ \sqrt{\theta_2} - \theta_3 \\ log(\theta_2) - \theta_4 \end{pmatrix}$$

```r
SB3_eefun <- function(data){
  function(theta){
    with(data,
      c(Y - theta[1],
       (Y - theta[1])^2 - theta[2],
       sqrt(theta[2]) - theta[3],
       log(theta[2]) - theta[4])
    )
  }
}
```

```r
estimates <- estimate_equations(
  eeFUN= SB3_eefun,
  data  = dt,
  rootFUN_control = list(start = c(1, 1, 1, 1)))
```

```
## When units are not specified, each observation is considered independent.
```

$geex *geex*estimates [1] 5.011650 16.502899 4.062376 2.803536

*geex*vcov [,1] [,2] [,3] [,4] [1,] 0.165028986 0.1115516 0.01372985 0.006759517 [2,] 0.111551626 5.3752170 0.66158537 0.325713512 [3,] 0.013729850 0.6615854 0.08142838 0.040089041 [4,] 0.006759517 0.3257135 0.04008904 0.019736746

$cls *cls*parameters p1 p2 p3 p4 1 5.01165 16.5029 4.062376 2.803536

*cls*vcov [,1] [,2] [,3] [,4] [1,] 0.165028986 0.1115516 0.01372985 0.006759517 [2,] 0.111551626 5.3752170 0.66158537 0.325713512 [3,] 0.013729850 0.6615854 0.08142838 0.040089041 [4,] 0.006759517 0.3257135 0.04008904 0.019736746