Estimating Equations in R: `geex`

B. Saul

2017-07-03

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

$$\psi(Y_i, \theta) = \begin{pmatrix} Y_i - \theta_1 \\ (Y_i - \theta_1)^2 - \theta_2 \end{pmatrix}$$	<pre>SB1_eefun <- function(data){
	  function(theta){
	    with(data,
	      c(Y - theta[1],
	       (Y - theta[1])^2 - theta[2])
	    )
	  }
	}</pre>

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,
                                units = 'id',
                                roots = c(1,1))
```

Table 2: " Comparing estimates from closed form versus geex "

	$\hat{\theta}$	$\hat{\Sigma}$
Closed form	$\begin{pmatrix} 5.2363 & 3.7570 \end{pmatrix}$	$\begin{pmatrix} 0.0376 & 0.0133 \\ 0.0133 & 0.2476 \end{pmatrix}$
geex	$\begin{pmatrix} 5.2363 & 3.7570 \end{pmatrix}$	$\begin{pmatrix} 0.0376 & 0.0133 \\ 0.0133 & 0.2476 \end{pmatrix}$
Decimal of difference	$\begin{pmatrix} Inf & 16 \end{pmatrix}$	$\begin{pmatrix} 14 & 14 \\ 14 & 13 \end{pmatrix}$

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 3: Translating math to code

$$\psi(Y_i, \theta) = \begin{pmatrix} Y_i - \theta_1 \\ X_i - \theta_2 \\ \theta_1 - \theta_3\theta_2 \end{pmatrix}$$	<pre>SB2_eefun <- function(data){
	  function(theta){
	    with(data,
	      c(Y - theta[1],
	        X - theta[2],
	        theta[1] - (theta[3] * theta[2]))
	    )
	  }
	}</pre>

```r
estimates <- estimate_equations(eeFUN = SB2_eefun,
                                data  = dt, units = 'id',
                                roots = c(1, 1, 1))
```

Table 4: " test "

	$\hat{\theta}$	$\hat{\Sigma}$
Closed form	$\begin{pmatrix} 5.4986 & 2.0031 & 2.7451 \end{pmatrix}$	$\begin{pmatrix} 0.0437 & 0.0002 & 0.0215 \\ 0.0002 & 0.0004 & -0.0005 \\ 0.0215 & -0.0005 & 0.0114 \end{pmatrix}$
geex	$\begin{pmatrix} 5.4986 & 2.0031 & 2.7451 \end{pmatrix}$	$\begin{pmatrix} 0.0433 & 0.0002 & 0.0213 \\ 0.0002 & 0.0004 & -0.0005 \\ 0.0213 & -0.0005 & 0.0113 \end{pmatrix}$
Decimal of difference	$\begin{pmatrix} Inf & Inf & Inf \end{pmatrix}$	$\begin{pmatrix} 4 & 6 & 4 \\ 6 & 6 & 6 \\ 4 & 6 & 4 \end{pmatrix}$

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 5: Translating math to code

$$\psi(Y_i, \theta) = \begin{pmatrix} Y_i - \theta_1 \\ X_i - \theta_2 \\ \sqrt{\theta_2} - \theta_3 \\ log(\theta_2) - \theta_4 \end{pmatrix}$$	`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
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, units = 'id',
                                roots = c(1, 1, 1, 1))
```

Table 6: " Example 3 "

	$\hat{\theta}$	$\hat{\Sigma}$
Closed form	$\begin{pmatrix} 5.0117 & 16.5029 & 4.0624 & 2.8035 \end{pmatrix}$	$\begin{pmatrix} 0.1650 & 0.1116 & 0.0137 & 0.0068 \\ 0.1116 & 5.3752 & 0.6616 & 0.3257 \\ 0.0137 & 0.6616 & 0.0814 & 0.0401 \\ 0.0068 & 0.3257 & 0.0401 & 0.0197 \end{pmatrix}$
geex	$\begin{pmatrix} 5.0117 & 16.5029 & 4.0624 & 2.8035 \end{pmatrix}$	$\begin{pmatrix} 0.1650 & 0.1116 & 0.0137 & 0.0068 \\ 0.1116 & 5.3752 & 0.6616 & 0.3257 \\ 0.0137 & 0.6616 & 0.0814 & 0.0401 \\ 0.0068 & 0.3257 & 0.0401 & 0.0197 \end{pmatrix}$
Decimal of difference	$\begin{pmatrix} Inf & Inf & Inf & 16 \end{pmatrix}$	$\begin{pmatrix} 13 & 12 & 13 & 13 \\ 12 & 10 & 12 & 12 \\ 13 & 12 & 12 & 13 \\ 13 & 12 & 13 & 13 \end{pmatrix}$