Robust Standard Errors in Small Samples

Michal Kolesár

February 23, 2021

Contents

Description 1

Methods 3

Variance estimate ...................................... 4

Degrees of freedom correction .......................... 5

Description

This package implements small-sample degrees of freedom adjustments to robust and cluster-robust
standard errors in linear regression, as discussed in Imbens and Kolesár [2016]. The implementation
can handle models with fixed effects, and cases with a large number of observations or clusters
1.

library(dfadjust)

To give some examples, let us construct an artificial dataset with 11 clusters

```r
set.seed(7)
d1 <- data.frame(y = rnorm(1000), x1 = c(rep(1, 3), rep(0, 997)), x2 = c(rep(1, 150), rep(0, 850)), x3 = rnorm(1000), cl = as.factor(c(rep(1:10, each = 50), rep(11, 500))))
```

Let us first run a regression of `y` on `x1`. This is a case in which, in spite of moderate data size, the
effective number of observations is small since there are only three treated units:

```r
r1 <- lm(y ~ x1, data = d1)
```

## No clustering
dfadjustSE(r1)

```r
#> Coefficients:
#> Estimate HC1 se HC2 se Adj. se df p-value
#> (Intercept) 0.00266 0.0311 0.031 0.0311 996.00 0.932
#> x1 0.12940 0.8892 1.088 2.3743 2.01 0.957
```

We can see that the usual robust standard errors (HC1 se) are much smaller than the effective

---

1We thank Ulrich Müller for suggesting to us the lemma below.
standard errors (Adj. se), which are computed by taking the HC2 standard errors and applying a degrees of freedom adjustment.

Now consider a cluster-robust regression of $y$ on $x_2$. There are only 3 treated clusters, so the effective number of observations is again small:

```r
r1 <- lm(y ~ x2, data = d1)
# Default Imbens-Kolesár method
dfadjustSE(r1, clustervar = d1$cl)
#>
#> Coefficients:
#> Estimate HC1 se HC2 se Adj. se df p-value
#> (Intercept) -0.0236 0.0135 0.0169 0.0222 4.94 0.288
#> x2 0.1778 0.0530 0.0621 0.1157 2.43 0.124
# Bell-McCaffrey method
dfadjustSE(r1, clustervar = d1$cl, IK = FALSE)
#>
#> Coefficients:
#> Estimate HC1 se HC2 se Adj. se df p-value
#> (Intercept) -0.0236 0.0135 0.0169 0.0316 2.42 0.4547
#> x2 0.1778 0.0530 0.0621 0.1076 2.70 0.0983
```

Now, let us run a regression of $y$ on $x_3$, with fixed effects. Since we’re only interested in $x_3$, we specify that we only want inference on the second element:

```r
r1 <- lm(y ~ x3 + cl, data = d1)
dfadjustSE(r1, clustervar = d1$cl, ell = c(0, 1, rep(0, r1$rank - 2)))
#>
#> Coefficients:
#> Estimate HC1 se HC2 se Adj. se df p-value
#> Estimate 0.0261 0.0463 0.0595 0.0928 3.23 0.778
dfadjustSE(r1, clustervar = d1$cl, ell = c(0, 1, rep(0, r1$rank - 2)), IK = FALSE)
#>
#> Coefficients:
#> Estimate HC1 se HC2 se Adj. se df p-value
#> Estimate 0.0261 0.0463 0.0595 0.0928 3.23 0.778
```

Finally, an example in which the clusters are large. We have 500,000 observations:

```r
d2 <- do.call("rbind", replicate(500, d1, simplify = FALSE))
d2$y <- rnorm(length(d2$y))
r2 <- lm(y ~ x2, data = d2)
summary(r2)
#>
#> Call:
#> lm(formula = y ~ x2, data = d2)
#>
#> Residuals:
```
Methods

This section describes the implementation of the Imbens and Kolesár [2016] and Bell and McCaffrey [2002] degrees of freedom adjustments.

There are $S$ clusters, and we observe $n_s$ observations in cluster $s$, for a total of $n = \sum_{s=1}^{S} n_s$ observations. We handle the case with independent observations by letting each observation be in its own cluster, with $S = n$. Consider the linear regression of a scalar outcome $Y_i$ onto a $p$-vector of regressors $X_i$, 

$$Y_i = X'_i \beta + u_i, \quad E[u_i \mid X_i] = 0.$$ 

We’re interested in inference on $\ell' \beta$ for some fixed vector $\ell \in \mathbb{R}^p$. Let $X, u,$ and $Y$ denote the design matrix, and error and outcome vectors, respectively. For any $n \times k$ matrix $M$, let $M_s$ denote the $n_s \times k$ block corresponding to cluster $s$, so that, for instance, $Y_s$ corresponds to the outcome vector in cluster $s$. For a positive semi-definite matrix $M$, let $M^{1/2}$ be a matrix satisfying $M^{1/2} M^{1/2} = M$, such as its symmetric square root or its Cholesky decomposition.

Assume that 

$$E[u_s u'_t \mid X] = \Omega_s, \quad \text{and} \quad E[u_s u'_t \mid X] = 0 \quad \text{if} \ s \neq t.$$ 

Denote the conditional variance matrix of $u$ by $\Omega$, so that $\Omega_s$ is the block of $\Omega$ corresponding to cluster $s$. We estimate $\ell' \beta$ using OLS. In R, the OLS estimator is computed via a QR decomposition,
We estimate \( V \), which allows us to compute the pseudo-inverse if it is singular, as is the case, for example, if the cluster size \( n \) is large. We therefore use the following result, suggested to us by Ulrich Müller, which allows us to compute \( a_s \) by computing a spectral decomposition of a \( p \times p \) matrix.

- Let \( Q'_sQ_s = \sum_{i=1}^{p} \lambda_is r_is' r_is \) be the spectral decomposition of \( Q'_sQ_s \). Then \( A_s = \sum_{i: \lambda_i \neq 1} (1 - \lambda_i)^{-1/2} Q_s r_is' r_is' Q'_s \) satisfies \( A_s(I - Q'_sQ_s)A_s = I \). This follows from the fact that \( I - Q'_sQ_s \) has eigenvalues \( 1 - \lambda_is \) and eigenvectors \( Q_s r_is \) and hence its pseudoinverse is \( \sum_{i: \lambda_i \neq 1} (1 - \lambda_i)^{-1/2} Q_s r_is' r_is' \).

Using the lemma, we can compute \( a_s \) efficiently as:

\[
a_s = \sum_{i: \lambda_i \neq 1} (1 - \lambda_i)^{-1/2} Q_s r_is' r_is' Q'_sQ_s \ell = Q_s D_s \ell, \quad D_s = \sum_{i: \lambda_i \neq 1} \lambda_i (1 - \lambda_i)^{-1/2} r_is' r_is'.
\]
Degrees of freedom correction

Let $G$ be an $n \times S$ matrix with columns $(I - QQ')_s a_s$. Then the Bell and McCaffrey [2002] adjustment sets the degrees of freedom to

$$f_{BM} = \frac{\text{tr}(G'G)^2}{\text{tr}((G'G)^2)}.$$

Since $(G'G)_{st} = a_s'(I - QQ')_s (I - QQ')_t a_t = a_s(1\{s = t\} - Q_s Q'_t) a_t$, the matrix $G'G$ can be efficiently computed as

$$G'G = \text{diag}(a'_s a_s) - B B' \quad B_{sk} = a'_s Q_{sk}.$$

Note that $B$ is an $S \times p$ matrix, so that computing the degrees of freedom adjustment only involves $p \times p$ matrices:

$$f_{BM} = \frac{(\sum_s a'_s a_s - \sum_{s,k} B_{sk}^2)^2}{\sum_s (a'_s a_s)^2 - 2 \sum_{s,k} (a'_s a_s) B_{sk}^2 + \sum_{s,t} (B'_s B_t)^2}.$$

If the observations are independent, we compute $B$ directly as $B \leftarrow a*Q$, and since $a_i$ is a scalar, we have

$$f_{BM} = \frac{(\sum_i a_i^2 - \sum_{sk} B_{sk}^2)^2}{\sum_i a_i^4 - 2 \sum_i a_i^2 B'_i B_i + \sum_{i,j} (B'_i B_j)^2}.$$

The Imbens and Kolesár [2016] degrees of freedom adjustment instead sets

$$f_{IK} = \frac{\text{tr}(G'\hat{\Omega}G)^2}{\text{tr}((G'\hat{\Omega}G)^2)},$$

where $\hat{\Omega}$ is an estimate of the Moulton [1986] model of the covariance matrix, under which $\Omega_s = \sigma^2 \epsilon I_{n_s} + \rho \epsilon_{1_{n_s}} \epsilon_{1_{n_s}}'$. Using simple algebra, one can show that in this case,

$$G'\hat{\Omega}G = \sigma^2 \epsilon \text{diag}(a'_s a_s) - \sigma^2 \epsilon B B' + \rho (D - BF')(D - BF')',$$

where

$$F_{sk} = \epsilon_{1_{n_s}} Q_{sk}, \quad D = \text{diag}(a'_s \epsilon_{1_{n_s}})$$

which can again be computed even if the clusters are large. The estimate $\hat{\Omega}$ replaces $\sigma^2$ and $\rho$ with analog estimates.

References


