Comparing the Horvitz-Thompson estimator and the Hajek estimator

January 21, 2011

Consider a finite population $U = \{1, 2, \dots, N\}$. Suppose $y_k, k \in U$ are values of the variable of interest in the population. We wish to estimate the total $\sum_{k=1}^{N} y_k$ based on a sample s taken from the population U. Assume that the sample is taken according to a sampling scheme having inclusion probabilities $\pi_k = Pr(k \in s)$. When the π_k is proportional to a positive quantity x_k available over U, and s has a predetermined sample size n, then

$$\pi_k = \frac{n x_k}{\sum_{i=1}^{N} x_i},$$

and the sampling scheme is said to be probability proportional to size (πps). Under this scheme, the Hájek estimator of the population total is defined by

$$\hat{y}_{Hajek} = N \frac{\sum_{k \in s} y_k/\pi_k}{\sum_{k \in s} 1/\pi_k}.$$

Särndal, Swenson, and Wretman (1992, p. 182) give several reasons for regarding the Hájek as 'usually the better estimator' comparing to the Horvitz-Thompson estimator

$$\hat{y}_{HT} = \sum_{k \in s} y_k/\pi_k :$$

a) the $y_k - \bar{y}_U$ tend to be small,

b) sample size is not fixed,

c) π_k are weakly or negatively correlated with the y_k.

Monte Carlo simulations are used here to compare the accuracy of both estimators for a sample size (or expected value of the sample size) equal to 20. Four cases are considered:

Case 1. y_k is constant for $k = 1, \dots, N$; this case corresponds to the case a) above;

Case 2. Poisson sampling is used to draw a sample s; this case corresponds to the case b) above;

Case 3. y_k are generated using the following model: $x_k = k, \pi_k = n x_k / \sum_{i=1}^{N} x_i, y_k = 1/\pi_k$; this case corresponds to the case c) above;

Case 4. y_k are generated using the following model: $x_k = k, y_k = 5(x_k + \epsilon_k), \epsilon_k \sim N(0, 1/3)$; in this case the Horvitz-Thompson estimator should perform better than the Hájek estimator.

Tillé sampling is used in Cases 1, 3 and 4. Poisson sampling is used in Case 2. The `belgianmunicipalities` dataset is used in Cases 1 and 2 with $x_k = Tot04_k$. In Case 2, the variable of interest is TaxableIncome. The mean square error (MSE) is computed using simulations for each case and estimator. The Hájek estimator should perform better than the Horvitz-Thompson estimator in Cases 1, 2 and 3.

```
> data(belgianmunicipalities)
> attach(belgianmunicipalities)
> n = 20
> pik = inclusionprobabilities(Tot04, n)
> N = length(pik)
```

Number of simulations (for an accurate result, increase this value to 10000):

```
> sim = 10
> ss = ss1 = array(0, c(sim, 4))
```

Defines the variables of interest:

```
> cat("Case 1\n")
> y1 = rep(3, N)
> cat("Case 2\n")
> y2 = TaxableIncome
> cat("Case 3\n")
> x = 1:N
> pik3 = inclusionprobabilities(x, n)
> y3 = 1/pik3
> cat("Case 4\n")
> epsilon = rnorm(N, 0, sqrt(1/3))
> pik4 = pik3
> y4 = 5 * (x + epsilon)
```

Simulation and computation of the Horvitz-Thompson estimator and Hájek estimator:

```
> ht = numeric(4)
> hajek = numeric(4)
> for (i in 1:sim) {
+     cat("Simulation ", i, "\n")
+     cat("Case 1\n")
+     s = UPtille(pik)
+     ht[1] = HTestimator(y1[s == 1], pik[s == 1])
+     hajek[1] = Hajekestimator(y1[s == 1], pik[s ==
+         1], N, type = "total")
```

```
+     cat("Case 2\n")
+     s1 = UPpoisson(pik)
+     ht[2] = HTestimator(y2[s1 == 1], pik[s1 ==
+         1])
+     hajek[2] = Hajekestimator(y2[s1 == 1], pik[s1 ==
+         1], N, type = "total")
+     cat("Case 3\n")
+     ht[3] = HTestimator(y3[s == 1], pik3[s ==
+         1])
+     hajek[3] = Hajekestimator(y3[s == 1], pik3[s ==
+         1], N, type = "total")
+     cat("Case 4\n")
+     ht[4] = HTestimator(y4[s == 1], pik4[s ==
+         1])
+     hajek[4] = Hajekestimator(y4[s == 1], pik4[s ==
+         1], N, type = "total")
+     ss[i, ] = ht
+     ss1[i, ] = hajek
+ }
```

Computation of the MSE and the ratio $\frac{MSE_{HT}}{MSE_{Hajek}}$:

```
> tv = c(sum(y1), sum(y2), sum(y3), sum(y4))
> for (i in 1:4) {
+     cat("Case ", i, "\n")
+     cat("The Horvitz-Thompson estimator under simulations:",
+         mean(ss[, i]), " and the true value:",
+         tv[i], "\n")
+     MSE1 = var(ss[, i]) + (mean(ss[, i]) - tv[i])^2
+     cat("MSE Horvitz-Thompson estimator:", MSE1,
+         "\n")
+     cat("The Hajek estimator under simulations:",
+         mean(ss1[, i]), " and the true value:",
+         tv[i], "\n")
+     MSE2 = var(ss1[, i]) + (mean(ss1[, i]) - tv[i])^2
+     cat("MSE Hajek estimator:", MSE2, "\n")
+     cat("Ratio of the two MSE:", MSE1/MSE2, "\n")
+ }
```

```
Case  1
The Horvitz-Thompson estimator under simulations: 1660.93  and the true value: 1767
MSE Horvitz-Thompson estimator: 223296.1
The Hajek estimator under simulations: 1767  and the true value: 1767
MSE Hajek estimator: 4.021017e-26
Ratio of the two MSE: 5.553224e+30
Case  2
The Horvitz-Thompson estimator under simulations: 131762702830  and the true value: 12112848168
MSE Horvitz-Thompson estimator: 8.810173e+20
The Hajek estimator under simulations: 124547881503  and the true value: 121128481686
```

```
MSE Hajek estimator: 5.124866e+20
Ratio of the two MSE: 1.719103
Case  3
The Horvitz-Thompson estimator under simulations: 20034652  and the true value: 60436.25
MSE Horvitz-Thompson estimator: 4.013596e+14
The Hajek estimator under simulations: 1927317  and the true value: 60436.25
MSE Hajek estimator: 3.500854e+12
Ratio of the two MSE: 114.6462
Case  4
The Horvitz-Thompson estimator under simulations: 892528.1  and the true value: 868897.9
MSE Horvitz-Thompson estimator: 561635431
The Hajek estimator under simulations: 86552.94  and the true value: 868897.9
MSE Hajek estimator: 612177882976
Ratio of the two MSE: 0.0009174383
```