Survey bootstrap and bootstrap weights

Stas Kolenikov

Department of Statistics
University of Missouri-Columbia

SNASUG
July 25, 2008
The basic idea of the bootstrap

- Population distribution $F(\cdot)$ $\leftrightarrow$ sample $X_1, \ldots, X_n$ $\leftrightarrow$ empirical distribution function
  
  $$F_n(x) = \frac{1}{n} \sum I[X_i \leq x] \equiv \mathbb{E}_n I[X_i \leq x]$$

- Parameter $\theta = T(F)$, its estimate $\hat{\theta}_n = T(F_n)$

- Inference goal: assess sampling variability of $\hat{\theta}_n$ about $\theta$

- Bootstrap (Efron 1979): take samples of size $n$ with replacement $(X_1^{(r)}, \ldots, X_n^{(r)}), r = 1, \ldots, R$ from $F_n(\cdot)$, obtain parameter estimates $\tilde{\theta}^{(r)} = T(F_n^{(r)})$

- Exact bootstrap: all possible subsamples; Monte Carlo: random set of say $R = 1000$ replications

- An estimate of the distribution function of $\hat{\theta}_n$ is
  
  $$G_{n,R}(t) = \frac{1}{R} \sum_{r=1}^{R} I[\tilde{\theta}^{(r)} \leq t] \equiv \mathbb{E}_* I[\tilde{\theta}_* \leq t]$$
Bias, variance, CIs

- $\theta \leftrightarrow \hat{\theta}_n$ is like $\hat{\theta}_n \leftrightarrow \tilde{\theta}^{(r)}$

- Estimate of bias:
  \[
  \mathbb{B}[\hat{\theta}_n] = \mathbb{E}[\hat{\theta}_n - \theta] \approx \mathbb{E}_*[\tilde{\theta}_* - \hat{\theta}_n] = \mathbb{B}_B[\hat{\theta}_n] \approx \frac{1}{R} \sum_r (\tilde{\theta}_*^{(r)} - \hat{\theta}) \tag{1}
  \]

- Estimate of variance:
  \[
  \mathbb{V}[\hat{\theta}_n] = \mathbb{E}((\hat{\theta}_n - \mathbb{E}\hat{\theta}_n)^2 \approx \mathbb{E}_*(\tilde{\theta}_* - \mathbb{E}\tilde{\theta}_*)^2 = \mathbb{V}_B[\hat{\theta}_n] \approx \frac{1}{R} \sum_r (\tilde{\theta}_*^{(r)} - \tilde{\theta}_*)^2 \tag{2}
  \]

- Percentile CI:
  \[
  \text{Pr}[\hat{\theta}_n \leq t] \approx \mathbb{E}_* \mathbb{I}[\tilde{\theta}_* \leq t] \tag{3}
  \]

- Bias-corrected CI:
  \[
  \text{Pr}[\hat{\theta}_n \leq t] \approx \mathbb{E}_* \mathbb{I}[2\hat{\theta}_n - \tilde{\theta}_* \leq t] \tag{4}
  \]
Survey setting

- Complex survey designs include stratification, multiple stages of selection, unequal probabilities of selection, non-response and post-stratification adjustments, . . .

- Unless utmost precision is required (or sampling fractions are really large), it suffices to approximate the real designs by two-stage stratified designs with PSUs sampled with replacement:

  \[
  \text{svyset } \textit{psu} [\textit{pweight}=\textit{sampweight}], \textit{strata}(\textit{strata})
  \]

- Notation: \# strata = L, \# units in h-th strata = n_h, PSUs are indexed by i, SSUs are indexed by k, so the generic notation is \( x_{hik} \)
Variance estimation methods

- Taylor series linearization (Särndal, Swensson & Wretman 1992): the derivatives need to be obtained for each individual model; streamlined by `robust`

- Balanced repeated replication (McCarthy 1969): use half-samples of the data, estimate, repeat $R$ times, combine results using analogues of (1)–(2)
  Features: $\forall h = 1, \ldots, L n_h = 2, R = 4([L/4] + 1)$ by using Hadamard matrices

- Jackknife (Kish & Frankel 1974, Krewski & Rao 1981): throw one PSU out, estimate, combine results using analogues of (1)–(2)
  Features: # replications $R = n$, closest to linearization estimator, inconsistent for non-smooth functions

- Bootstrap (Rao & Wu 1988): resample $m_h$ units with replacement from the available $n_h$ units in stratum $h$
  Features: need internal scaling — best with Rao, Wu & Yue’s (1992) weights, although other schemes are available; choice of $m_h$; choice of $R$
Pros and cons of resampling estimators

+ Only need the software that does weighted estimation — no need for programming specific estimators for each model
+ No need to release the unit identifiers in public data sets
  - Computationally intensive
  - Non-response and post-stratification need to be performed on every set of weights
Comparisons of methods I


- Jackknife and linearization are asymptotically equivalent to higher order terms, coincide in certain situations, and have smaller biases than other methods.
- **Coverage:** bootstrap $\succ$ BRR $\succ$ jackknife $\succ$ linearization
- **Stability:** linearization $\succ$ jackknife $\succ$ BRR $\succ$ bootstrap
- Making the statistic pivotal (Fisher’s arctanh transform of correlation) improves coverage.
- Bootstrap is the best method for one-sided CIs. It is rarely the best one for variance estimation, but is applicable in a wider set of circumstances.
Comparisons of methods II

Shao (1996): “... the choice of the method may depend more on nonstatistical considerations, such as the feasibility of their implementation. ... Blindly applying the resampling methods may yield incorrect results”
Scaling of weights

• Rao & Wu (1988) showed that naïve bootstrap (resample $m_h$ PSUs with replacement from $h$-th stratum) is biased, producing variance estimates in $h$-th stratum that are understated by a factor of $(n_h - 1)/m_h$

• They proposed internal scaling: within each stratum, modify the pseudo-values, i.e., the estimates of the moments

• How can this be generalized to other nonlinear models?

• Rao, Wu & Yue (1992) proposed scaling of weights: if in $r$-th replication, the $i$-th unit in stratum $h$ is to be used $m_{hi}^{(r)}$ times, then the bootstrap weight is

\[ w_{hik}^{(r)} = \left\{ 1 - \left( \frac{m_h}{n_h - 1} \right)^{1/2} + \left( \frac{m_h}{n_h - 1} \right)^{1/2} \frac{n_h}{m_h} m_{hi}^{(r)} \right\} w_{hik} \]

where $w_{hik}$ is the original probability weight
bsweights syntax

bsweights \textit{prefix}, \texttt{reps(#)} \texttt{n(#)} [\texttt{balanced} \\
\texttt{replace} \texttt{calibrate(command @} \texttt{verbose} \texttt{quasi}} \\
\texttt{Monte Carlo options} ]

- \texttt{reps()} specifies the number of resampling replications
- \texttt{n()} specifies the number of units to be resampled from each stratum, or from the whole data set with no complex survey structure
- \texttt{balanced} specifies balanced bootstrap
- \texttt{calibrate} calls \texttt{command} substituting the name of the current replicate weight for @, and \texttt{verbose} shows the output of the calibrating command
- \texttt{replace} allows overwriting the existing set of weights
- \textit{QMC options} are \texttt{qmcstratified}, \texttt{qmcmatrix}, \texttt{shuffle} and \texttt{balance} referring to quasi-Monte Carlo based resampling variance estimators (Kolenikov 2007).
Sample size and # replications

What is a good choice of the resample size $m_h$?

- $0 < m_h \leq n_h - 1$, where the latter inequality is to maintain meaningful ranges
- Rao & Wu (1988): the optimal choice
  \[ m_h = \frac{(n_h - 2)^2}{(n_h - 1)} \]
corrects for the skewness of the estimate distribution when its variance is known

What is a good number of replicates?

- $R \geq$ degrees of freedom of the design $= \sum_h n_h - L$
- Rao & Wu (1988) found little gain in going beyond $R = 100$.
- The “industry standard” seems to be $R = 500$. 
Calibration

- Option `calibrate(call @)` allows to call an external program to perform additional adjustments on weights.
- The replication weight variables will be substituted for `@` in the above call.
- Subpopulation estimation: set weights outside the subpopulation = 0:

```plaintext
program define SubPopW
    gettoken weightvar condition : 0
    replace `weightvar' = 0 if !('condition')
end
bsweights bsw , ...calibrate(SubPopW @ black)
bs4rw , rw(bsw*) : ... [pw=weight*black]
```
Balanced bootstrap

- **First order balance**: each unit is resampled the same number of times (Davison, Hinkley & Schechtman 1986, Nigam & Rao 1996)
  - Reduces (simulation) variability of the bias estimate (by removing the linear part from it — adequate for linear or symmetric statistics)
  - Reduces the variability of the variance estimate somewhat; no discernible effect on coverage?
  - Achieved by permuting the vector of $R$ concatenated sample unit labels
- Efficient implementations: Gleason (1988)
- **Second order balance**: each pair of units is resampled the same number of times (Graham, Hinkley, John & Shi 1990)
- The usual bootstrap: discrepancy for either first or second order balance are $O_*(R^{-1/2})$
Balancing conditions

First order balance can be achieved by `bsweights`:

- Each unit in stratum $h$ is used the same number of times $k_h$
- Total number of units used in all replications:
  \[ k_h n_h = m_h R \]
- Balancing condition: \( \forall h : m_h R \) is a multiple of \( n_h \)
  - E.g., if \( n_h \) takes values 2, 3, 4 and 5, \( R \) must be a multiple of \( 3 \cdot 4 \cdot 5 = 60 \)

Second order balance: difficult to satisfy for an arbitrary design (except for BRR when \( \forall h n_h = 2 \), and jackknife).
Nigam & Rao (1996): constant \( n_h = 2k \), or \( n_h = 4k + 1, 4k + 3 \) which is a prime or a prime power.
QMC ideas in survey resampling

Quasi-Monte Carlo methods are widely used in computational mathematics and physics to approximate highly dimensional integrals (Niederreiter 1992)

- Regular deterministic sequences in $d$ dimensions
- Discrepancy: $O(A(d)n^{-1} \ln^d n)$ where $d$ is dimension, $n$ is the length of sequence
- This rate is better than the one for the usual Monte Carlo, $O_p(n^{-1/2})$, for $n \gg \exp(d)$
- Dimensionality curse: $A(d)$ is combinatorial in $d$
- Stratified version: each dimension $\mapsto$ each strata
- Matrix version: 2D sequence pointing at the units to be resampled
Examples

Do-file `bsw-example` provides examples of:

- basic bootstrap
- balanced bootstrap with fine-tuning the number of replicates \( R \) to achieve first order balance
- versions of QMC bootstrap
- calibrated weights
- estimation for subpopulation

Non-survey uses:

- eliminating simulation bias by balanced bootstrap
- weighted bootstrap
Stata or Mata?

- ado code: 230 lines
  - parsing options
  - choosing the method
  - `bsample` in the simplest case
  - rescaling the weights

- Mata code: 340 lines
  - balanced bootstrap
  - QMC resampling
  - allocating the samples
  - any other potentially applicable balanced designs
Limitations

What `bsweights` cannot do:

- Design effect — that is a post-estimation feature. One would need to save the relevant variance-covariance matrices and re-post them
- \( t \)-percentiles of jackknife-after-bootstrap

\[
D[t] = \frac{\hat{\theta} - \theta}{\sqrt{v_J}} \approx D[t^*] = \frac{\hat{\theta}^* - \hat{\theta}}{\sqrt{v_J^*}}
\]

Estimation feature rather than setting up pre-estimation weights: special coding of the jackknife passes within the bootstrapping routine

- Finite population corrections
- Missing and imputed data: re-impute missing values in each bootstrap sample (Shao 1996, Shao 2003)
- Other survey bootstrap schemes (BMM, BWO, RHSB)
What I covered was…

1. Resampling inference
2. Survey inference
3. bsweights
4. Examples
5. Conclusions
6. References
References I


References II


References III


References IV


Wishes and grumbles for \texttt{bs4rw}

- more “respect” to \texttt{svy} setting
- posting \texttt{e(V\_SRS)} for \texttt{estat} effects
- capacity of interacting with the current weights for imputation and/or subpopulation work
- explicit \texttt{subpop} option: zero out the weights outside the subpopulation