User Guideline for *No More Limited Mobility Bias: Exploring the Heterogeneity of Labor Markets*

Miren Azkarate-Askasua and Miguel Zerecero

October 8, 2024

This document provides a guideline for using our bootstrap correction method for quadratic objects in the parameters and explains the main features of the functions involved in the bootstrap correction. We provide an example of a labor market with low mobility to test the bootstrap correction of the quadratic objects from *No More Limited Mobility Bias: Exploring the Heterogeneity of Labor Markets*.

1 Function overview

The code running the example correction is in the file `main_example.m` located in the main folder. This script initializes parallelization and defines the variables needed to call the function `correction_all`, which implements the chosen type of bootstrap correction. The `correction_all` function is located in the `src` folder.

Below, we explain the main features of the function performing the bootstrap correction, which were either written by us or borrowed from other sources, as noted:

Listing 1: Function for bootstrap correction

```
function [plugin, delta, corrected, decomp_pi, decomp_b, dimensions, NT] =
    correction_all(y, id, firmid, other_fe, mat_controls, group,...
                        n_lev, n_boot, type_hc, type_leave,...
                        cluster, ind_light, mytol, LdM_mom, year,...
                        filename, ind_export, v_filename_group)
```

The `correction_all` function is the central component that collects user-defined variables and parameters, sets the missing parameters, and calls the appropriate sub-functions based on the type of bootstrap selected.

IMPORTANT: It is necessary that the person-year data is sorted by worker identifiers and years. In Stata this is done by using the command `xtset id year`.

Mandatory Inputs

1. **y**:
   The outcome variable.
   Dimension: $N^* \times 1$, where N^* is the number of person-year observations.

2. **id**:
   A vector of worker identifiers.
   Dimension: $N^* \times 1$.

3. **firmid**:
   A vector of firm identifiers.
   Dimension: $N^* \times 1$.

Optional Inputs

1. **other_fe**:
   Additional fixed effects beyond worker and firm fixed effects. These fixed effects are normalized to avoid collinearity. The connected set and the leave-one-out connected set are computed using `id` and `firmid`.
   Dimension: $N^* \times L$, where L is the number of additional fixed effects.

2. **mat_controls**:
   Matrix of other covariates.
   Dimension: $N^* \times K_1$, where K_1 is the number of other covariates.

3. **group**:
   A vector or matrix indicating group identifiers for each observation. If the estimation of quadratic objects needs to be split by groups, this variable must be provided. Multiple groupings can be included if a matrix is supplied.
   Dimension: $N^* \times G$, where G is the number of groupings.

4. **n_lev**:
   Number of simulations for leverage estimation. Required when `type_hc` is set to 'hc_2', 'hc_u', 'hc_u_match', or 'hc_u_clus'.
   Default: 300 (must be a natural number).

5. **n_boot**:
   Number of bootstrap simulations, controlling the precision of the estimation.
   Default: 300 (must be a natural number).

6. **type_hc**:
   Specifies the estimator for the error covariance matrix. Possible values are:

   - 'hom': for homoscedastic errors.
   - 'hc_0', 'hc_1', 'hc_2', 'hc_u': diagonal matrix; heteroscedastic errors. The different variance estimates, $\widehat{\sigma}_i$, for the error of observation i with OLS residual $\widehat{\varepsilon}_i$ and outcome variable y_i are:
     - 'hc_0': $\widehat{\sigma}_i = \widehat{\varepsilon}_i^2$.
     - 'hc_1': $\widehat{\sigma}_i = \frac{NT}{NT-K}\widehat{\varepsilon}_i^2$, where K is the total number of covariates; NT the size of the data in the final sample.
     - 'hc_2': $\widehat{\sigma}_i = \frac{\widehat{\varepsilon}_i^2}{1-P_{ii}}$, where P_{ii} is the leverage of observation i.
     - 'hc_u': $\widehat{\sigma}_i = \frac{y_i\widehat{\varepsilon}_i}{1-P_{ii}}$, where P_{ii} is the leverage of observation i.
   - 'hc_u_match', 'hc_u_clus': Leave-cluster-out variance estimate as explained in Section 3.1. These options assume error dependence within matches or user-specified clusters (see `cluster` input below).

   Default: 'hc_u_match'.

7. **type_leave**:
   Specifies the data selection procedure for leave-one-out operations. Possible values are:

   - 'obs': Leave an observation out.
   - 'worker': Leave a worker out.
   - 'match': Leave a match out.

   Default behavior depends on the `type_hc` setting:

   - For `type_hc = 'hc_u'`, the default is 'obs'.
   - For `type_hc = 'hc_u_match'`, the default is 'match'.
   - For `type_hc = 'hc_u_clus'`, the default is 'clus'.

   If 'none' is chosen, no data is left out, which is consistent with `type_hc` values 'hom', 'hc_0', and 'hc_1'.

8. **cluster**:
   A vector indicating the cluster identifier for each observation. Required when `type_hc = 'hc_u_clus'`.

Dimension: $N^* \times 1$.

9. **ind_light**:
   Indicator to manage memory usage and keep a memory-light environment. When set to 1, the code clears variables from the base workspace to minimize memory consumption.
   Default: 0.

10. **mytol**:
    Tolerance parameter for the preconditioned conjugate gradient method (`pcg`).
    Default: 10^{-6}.

11. **LdM_mom**:
    Indicator to compute the quadratic objects related to the average coworker fixed effect. If set to 1, the `year` input becomes mandatory.
    Default: 0.

12. **year**:
    A vector of year indicators, required if `LdM_mom` is set to 1.
    Dimension: $N^* \times 1$.

13. **filename**:
    A string defining the prefix for the filenames of the exported CSV tables. If set to 'example', the exported files will be named:

    - `example_plugin_estimates.csv`
    - `example_corrected_estimates.csv`
    - `example_var_decomp_plugin.csv`
    - `example_var_decomp_corrected.csv`

    A relative path can be specified by including it in the filename, e.g., `filename = './results/example'` will store the files in the `results` subfolder.

14. **ind_export**:
    Indicator to export CSV tables.
    Default: 0 (no export).

15. **v_filename_group**:
    A cell array of strings to specify names for each grouping when exporting CSV files. If not

provided, and there are multiple groupings, the output filenames will be automatically suffixed with 'g1', 'g2', etc.
Example: If `filename = 'example'` and two groupings are supplied, this input could be set to { 'group1', 'group2' }, resulting in filenames such as `example_group1.csv` and `example_group2.csv`.

Outputs

1. **plugin**:
   Plug-in estimates of the quadratic objects. If the user provided a `group` variable, the plug-in estimates are also computed per group and returned as a cell.

2. **delta**:
   Bootstrap estimate of the bias of the quadratic objects. If the user provided a `group` variable, the bias is also estimated per group and returned as a cell.

3. **corrected**:
   Bootstrap-corrected quadratic objects. If the user provided a `group` variable, the correction is also estimated per group and returned as a cell.

4. **decomp_pi**:
   Variance decomposition using the plug-in estimates in levels and as explained shares. If the user provided a `group` variable, the decomposition is also performed per group and returned as a cell.

5. **decomp_b**:
   Variance decomposition using the bootstrap-corrected estimates in levels and as explained shares. If the user provided a `group` variable, the decomposition is also performed per group and returned as a cell.

6. **dimensions**:
   Dimensions of workers and firm fixed effects taking into account the normalizations. It gives the total number of workers and firms used in the final sample.

7. **NT**:
   Number of person-year observations in the final sample. `NT` can be below N^* if the user did not provide a connected set or a leave-one-out connected set.

8. **Stored CSVs**:

   If the user sets `ind_export` equal to 1, the function stores four CSV tables: plug-in estimates of the quadratic objects, corrected estimates, and their respective variance decompositions. If `filename` is not provided, the tables are stored in the working directory.

2 Working examples

In the following, we summarize the different ways to implement the bootstrap-corrected estimation of the quadratic objects for various user-decided variable combinations.

2.1 Minimal

The box below shows a minimal working example where the user provides the mandatory inputs (`y`, `id`, and `firmid`) and other variables that will be residualized, such as `other_fe` and `mat_controls`. Given this specification, **correction_all** takes the default options (leave-match-out sample selection procedure and clustering at the match level) and will not print CSV tables. **correction_all** then calls **correction_match**, which is the function estimating the correction.

Listing 2: Minimal example

```
[plugin, delta, corrected, decomp_pi, decomp_b, dimensions, NT] = correction_all(y, id
    , firmid, other_fe, mat_controls);
```

If the user provides the `group` variable, **correction_all** calls **correction_match_multi_group** to estimate the correction.

If the user would like to estimate the same model but print the CSV tables, the call should be as below, setting `ind_export` equal to 1. In such a case, it is not necessary to recover the output of the function as the results will be stored in the CSV files.

Listing 3: Minimal example with printing

```
correction_all(y, id, firmid, other_fe, mat_controls, [], [], [], ...
    [], [], [], [], [], [], [],...
    filename, ind_export, []);
```

2.2 Choosing the Covariance Matrix Estimate

If the user wants to change the covariance matrix estimate from the default (leave-match-out variance estimate), the variable `type_hc` must be adjusted accordingly. If the user sets `type_hc` to 'hc_u_clus', the `cluster` variable must be provided. In this case, the function estimating the correction is **correction_cluster**. If the `group` variable is defined, the function for the correction becomes **correction_cluster_multi_group**.

Listing 4: Leave-cluster-out

```
[plugin, delta, corrected, decomp_pi, decomp_b, dimensions, NT] = correction_all(y, id
    , firmid, other_fe, mat_controls, [], [], [], ...
    type_hc, [], cluster, [], [], [], [],...
    filename, ind_export, []);
```

If the user opts for the assumption of homoscedastic errors or chooses to cluster at the observation level by setting `type_hc` accordingly, the function estimating the correction will be **correction**. For grouped decomposition, where the `group` variable is provided, the function for the correction will be **correction_multi_group**.

2.3 Memory efficient estimation

The code allows the option to remove the variables from the main workspace by setting *ind_light* equal to 1. This is a useful option when encountering memory issues with large datasets where the computer might be running out of RAM.

WARNING: The function only deletes variables from Matlab's base workspace. This means that for this option to work properly, the function **correction_all** must be called from the base workspace and not within any other function.

The example below takes the default values and does not print CSVs.

Listing 5: Memory efficient estimation

```
[plugin, delta, corrected, decomp_pi, decomp_b, dimensions, NT] = correction_all(y, id
    , firmid, other_fe, mat_controls, [], [], [], ...
    [], [], [], ind_light, [], [], [],...
    [], [], []);
```

2.4 Sample Selection

To assess the impact of different covariance matrix estimators while maintaining consistent sample restrictions, set `type_leave` to 'match'. Subsequently, you can run multiple estimations by varying `type_hc` to 'hom', 'hc_u', and 'hc_u_match', for example. The box below illustrates how to implement the sample selection procedure to leave the match out and estimate the covariance matrix with clustering at the observation level.

Listing 6: Sample selection

```
[plugin, delta, corrected, decomp_pi, decomp_b, dimensions, NT] = correction_all(y, id
    , firmid, other_fe, mat_controls, [], ...
    [], [], 'hc_u', 'match',...
    [], ind_light, [], [], [],...
    [], [], []);
```

2.5 Other moments

All the main correction functions (**correction**, **correction_multi_group**, **correction_match**, **correction_match_multi_group**, **correction_cluster**, **correction_cluster_multi_group**) have versions when the user wants to compute quadratic objects related to the average coworker fixed effect. This is done by setting `LdM_mom` equal to 1 and requires to provide a `year` variable.

Listing 7: Coworker moments

```
[plugin, delta, corrected, decomp_pi, decomp_b, dimensions, NT] = correction_all(y, id
    , firmid, other_fe, mat_controls, [], [], [], ...
    [], [], [], [], [], LdM_mom, year,...
    filename, ind_export, []);
```

2.6 Fully flexible

A fully flexible call of the function is done by specifying all the function arguments which can done as follows.

Listing 8: Flexible

```
[plugin, delta, corrected, decomp_pi, decomp_b, dimensions, NT] = correction_all(y, id
    , firmid, other_fe, mat_controls, group,...
                        n_lev, n_boot, type_hc, type_leave,...
                        cluster, ind_light, mytol, LdM_mom, year,...
                        filename, ind_export, v_filename_group);
```

3 Simulation

The folder `data` contains a CSV file with the simulated labor market with low mobility and roughly 2.2 million person-year observations. The code `simulate_example_large.R` in the data folder replicates the simulation. The logarithm of wages are simulated as:

$$\log w_{it} = \theta_i + \psi_{J(i,t)} + q_{it}\gamma + \varepsilon_{it}, \tag{1}$$

where the function $J(i,t)$ gives the identity of the unique firm that employs worker i at time t, θ_i is a worker fixed effect, $\psi_{J(i,t)}$ is the firm $J(i,t)$ fixed effect, and q_{it} are time varying observables (age and age squared), and ε_{it} is the error term.

3.1 Estimation example

Below we reproduce the output to the Command Window of Matlab when running the minimal example above.

```
-----------------------------------
```

```
1. MINIMAL EXAMPLE WITHOUT PRINTING
-----------------------------------
General options: Standard corrections with leave match out variance
{'Setup: leave option is match. Covariance matrix is assumed block diagonal'}

Size initial data: 2187268
Relabeling ids...
---------- Connected set ----------
# of firms: 49630
# connected sets: 2942
Largest connected set contains 46619 firms

Relabeling ids again...
Size connected set data: 2142819
Type of data selection procedure: match
---------- Pruning ----------
Size after removing one-timers: 2142819
Elapsed time is 1.645687 seconds.
Size of pruned data: 2025565
Building matrix of dummies for the additional fixed effects...
Done!
Residualizing controls and/or extra fixed effects
USER WARNING: function -ichol()- with diagcomp = 0.1 failed!
USER WARNING: function -ichol()- with diagcomp = 0.1663 failed!
USER WARNING: function -ichol()- with diagcomp = 0.24945 failed!
pcg converged at iteration 56 to a solution with relative residual 9.6e-07.
Done!
Share of mover ids: 0.17981
Collapsing observations of movers at the cluster level
Elapsed time is 2.307348 seconds.

---------- Initial AKM ----------
Building preconditioner for Laplacian Matrix...
Done!
pcg converged at iteration 13 to a solution with relative residual 9.2e-07.
plugin estimates:
var_worker    var_firm    cov_worker_firm      N_obs          N_id       N_firmid
----------    --------    ---------------    ----------    ----------    --------

2.5989       2.0378        -0.00116        2.0256e+06    5.7903e+05     40987
```

```
---------- Covariance matrix estimation ----------

---------- Leverage estimation ----------
Start to estimate the leverages...
Done!
Elapsed time is 20.464770 seconds.

---------- Diagnostic ----------
Problems in leverage estimation: 0
Elapsed time is 0.006018 seconds.

---------- Bootstrap ----------
Elapsed time is 42.107157 seconds.
corrected estimates:
var_worker    var_firm    cov_worker_firm      N_obs          N_id       N_firmid
----------    --------    ---------------    ----------    ----------    --------

2.0041        1.7071         0.30873        2.0256e+06    5.7903e+05     40987


---------- Variance decomposition ----------
Variance decomposition of plugin:
type_decomp    var_y    var_worker    var_firm    2*cov_worker_firm    var_resid
-----------    -----    ----------    --------    -----------------    ---------

{'levels' }    5.328      2.5989       2.0378           -0.00232         0.69357
{'percent'}        1     0.48779      0.38247        -0.00043544         0.13017

Variance decomposition of corrected:
type_decomp    var_y    var_worker    var_firm    2*cov_worker_firm    var_resid
-----------    -----    ----------    --------    -----------------    ---------

{'levels' }    5.328      2.0041       1.7071           0.61745          0.99929
{'percent'}        1     0.37615      0.32041           0.11589          0.18756

FINISHED!
```