Estimating Multi–Way Fixed Effect Models with `reghdfe`

Sergio Correia, Duke University

2016 Stata Conference, Chicago Illinois

sergio.correia@gmail.com

http://scorreia.com/software/reghdfe/

https://github.com/sergiocorreia/reghdfe

Introduction

`reghdfe` implements the estimator from:

- Correia, S. (2016). Linear Models with High-Dimensional Fixed Effects: An Efficient and Feasible Estimator. *Working Paper*

Borrows heavily from previous contributions, many from the Stata camp (`reg2hdfe`, `a2reg`, `gpreg`)

Use it to control for unobservables that stay constant within an economic unit (workers, firms, exporters, importers, etc.)

Applications in many fields: accounting (DeHaan et al 2015), finance (Gormley et al 2015), labor (Guimarães et al 2015), trade (Mayer 2016), etc.

Estimator

Linear Fixed Effect Models — Problem

We want to compute the least squares estimates $\hat{\beta}$ of

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{D}\boldsymbol{\alpha} + \boldsymbol{\varepsilon}$$

- $\mathbf{D} = \left[\, \mathbf{D}_1 \ \mathbf{D}_2 \ \cdots \ \mathbf{D}_F \,\right]$ consists of F indicator matrices
- If $F = 1$, this collapses to a standard fixed effect regression (`xtreg`, `areg`)
- Can't use dummies because $\left[\, \mathbf{D}_2 \ \cdots \ \mathbf{D}_F \,\right]$ is too large

Linear Fixed Effect Models — Solution Strategy

Steps:

1. Compute the residuals of $\mathbf{y}$ and $\mathbf{X}$ against $\mathbf{D}$:

$$\tilde{\mathbf{y}} = \mathbf{M_D y}$$

$$\tilde{\mathbf{X}} = \mathbf{M_D X}$$

2. Apply the Frisch–Waugh–Lovell Theorem:

$$\hat{\boldsymbol{\beta}} = \left(\tilde{\mathbf{X}}'\tilde{\mathbf{X}}\right)^{-1} \tilde{\mathbf{X}}'\tilde{\mathbf{y}}$$

Thus, we can just focus on one variable at a time: $\tilde{\mathbf{y}}$

Linear Fixed Effect Models — Solution Strategy

To obtain $\hat{\mathbf{y}} = \mathbf{M_D y}$, find an $\hat{\boldsymbol{\alpha}}$ that satisfies the normal equations

$$\mathbf{D}'\mathbf{e} = 0 \quad , \quad \mathbf{e} \stackrel{\text{def}}{=} \mathbf{y} - \mathbf{D}\hat{\boldsymbol{\alpha}}$$

In plain English:

> *For every level g of every fixed effect f the mean of the residuals must be zero:*

$$\overline{e_i} = 0 \quad , \quad i \in \mathcal{I}(f, g)$$

Note: We don't care if $\hat{\boldsymbol{\alpha}}$ is unique

Outline of the Algorithm

1. Divide and conquer: apply FWL to work on one variable at a time
2. Apply Method of Alternating Projections (MAP)
3. Accelerate MAP with conjugate gradient
4. Insights from graph theory: exactly the same problem as solving a Graph Laplacian

MAP - Definition

$$\lim_{n\to\infty}||(\mathbf{M}_1 \cdot \mathbf{M}_2 \ldots \mathbf{M}_F)^n\mathbf{y} - \mathbf{M}_{12\ldots F}\mathbf{y}|| = 0$$

Suggests iteration:

$$\mathbf{y}_{k+1} = \underbrace{(\mathbf{M}_1 \cdot \mathbf{M}_2 \ldots \mathbf{M}_F)}_{\text{Linear Transform } \mathbf{T}}\mathbf{y}_k$$

MAP - Example (1/2)

```
sysuse auto, clear

// Benchmark
areg price gear length i.trunk, absorb(turn)
```

MAP - Example (2/2)

```
foreach var in price gear length { // FWL Step
  forval i = 1/10 { // MAP Step
    foreach fe in turn trunk {
      qui areg ‘var’, absorb(‘fe’)
      predict double resid, resid
      drop ‘var’
      rename resid ‘var’
    }
  }
}
regress price gear length, dof(38) nocons
```

MAP - Problem #1

Bauschke et al (2003):

> *[...] The main practical drawback of the MAP appears to be that it is often slowly convergent [...] Franchetti and Light and Bauschke, Borwein, and Lewis have given examples showing that the convergence [...] can be arbitrarily slow!*

It can be very, very slow!

(In particular when the underlying fixed effects are *poorly connected*)

MAP - Problem #1

	y	id1	id2
1	1	0	0
2	0	0	1
3	0	1	1
4	0	1	2
5	0	2	2
6	0	2	3
7	0	3	3
8	0	3	4
9	0	4	4
10	0	4	5

Figure 1: This dataset will turn your PC into a heater in the winter

MAP - Solution #1

Guimarães & Portugal (2010) and Gaure (2013) apply accelerations that are related to steepest descent

$$\mathbf{y}_{k+1} = t\,\underbrace{(\mathbf{M}_1 \cdot \mathbf{M}_2 \ldots \mathbf{M}_F)}_{\text{Linear Transform } \mathbf{T}}\,\mathbf{y}_k + (1-t)\mathbf{y}_k$$

Often improve speeds significantly, but ...

MAP - Problem #2

Bauschke et al (2003):

> *[...] perhaps surprisingly, we show that the acceleration scheme may actually be slower than the MAP [...]!*

Hernández-Ramos et al (2011):

> *[...] the steepest descent method is known for its slowness in the presence of ill-conditioned problems [...]*

MAP - Solution #2

- Why apply steepest descent and not conjugate gradient?
- Because CG requires a symmetric transform and $T \stackrel{\text{def}}{=} \mathbf{M}_1 \cdot \mathbf{M}_2 \ldots \mathbf{M}_F$ is not symmetric
- Solution: follow Hernández-Ramos et al (2011) and make it symmetric:

$$T^{\text{Sym}} \stackrel{\text{def}}{=} \mathbf{M}_1 \cdot \mathbf{M}_2 \ldots \mathbf{M}_F \ldots \mathbf{M}_2 \cdot \mathbf{M}_1$$

$$T^{\text{Cim}} \stackrel{\text{def}}{=} (\mathbf{M}_1 \cdot \mathbf{M}_2 \ldots \mathbf{M}_F)/F$$

- Theoretical advantages (monotonic convergence) and practical ones (as fast as other methods for easy problems, significantly faster for ill-defined ones)

Not fast enough for some applications, can we speed it even more? Yes!

Link with Graph Theory

- Let's rewrite the two–way fixed effect model as a graph:
- If CEO j has only worked at firm k:

  $\sum_{i \in j} y_i - n_j \hat{\alpha}_j - n_j \hat{\gamma}_k = 0$

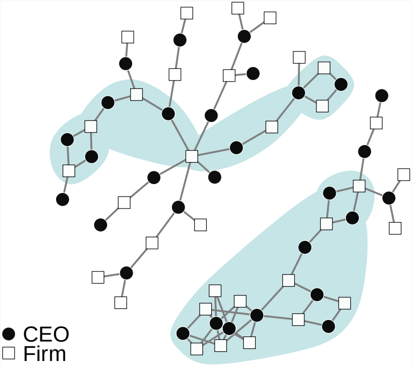

Figure 2: Graph of CEO–Firm Connections

Link with Graph Theory

- Solving a two–way fixed effects problem is *exactly* the same problem as solving $\mathbf{Lx} = \mathbf{b}$ where $\mathbf{L}$ is a Laplacian matrix
- Spielman & Teng (2004), Kelner et al (2013):
    - Laplacian systems can now be solved in nearly–linear time, instead of in $O(n^{2.36})$!
    - This is a fundamental breakthrough in graph theory and numerical optimization, and we can apply it to solve our model
- Can also apply other insights from graph theory (e.g. graph condition number)

Link with Graph Theory

However:

- Solver has a very complex implementation
- Suffers from cache locality problems (Hoske et al 2015, Boman et al 2016)
- What's the point of an $O(n)$ solver if Stata requires multiple sorts? $O(n \log n)$
- Solution: use a better sorting algorithm (see `ftools` package)

Implementation

reghdfe

```
sysuse auto
ssc install reghdfe
reghdfe price weight, absorb(turn trunk foreign)
```

reghdfe

```
. reghdfe price weight, absorb(turn trunk foreign)
(dropped 9 singleton observations)
(converged in 13 iterations)

HDFE Linear regression                          Number of obs   =         65
Absorbing 3 HDFE groups                         F(   1,      38) =      25.43
                                                Prob > F        =     0.0000
                                                R-squared       =     0.6563
                                                Adj R-squared   =     0.4211
                                                Within R-sq.    =     0.4009
                                                Root MSE        =  2203.9341

------------------------------------------------------------------------------
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      weight |   4.978642   .9872606     5.04   0.000     2.980037    6.977246
-------------+----------------------------------------------------------------
    Absorbed |         F(25, 38) =      1.879   0.037             (Joint test)
------------------------------------------------------------------------------

Absorbed degrees of freedom:
---------------------------------------------------------------+
 Absorbed FE |  Num. Coefs.  =   Categories  -   Redundant     |
-------------+-------------------------------------------------|
        turn |           13             13              0      |
       trunk |           12             13              1      |
     foreign |            1              2              1 ?    |
---------------------------------------------------------------+
? = number of redundant parameters may be higher
```

Figure 3: reghdfe screenshot

Design Principles: Simplicity

```
a2reg price gear, individual(turn) unit(foreign)
    indeffect(FE1) uniteffect(FE2)

reg2hdfe price gear, id1(turn) id2(trunk) fe1(FE1)
    fe2(FE2) uniteffect(FE2)

gpreg price gear, ivar(turn) jvar(trunk) ife(FE1)
    jfe(FE2)

felsdvregdm price gear, ivar(turn) jvar(trunk)
    peff(FE1) feff(FE2)
```

These are wonderful packages, but can we do better? (See The Zen of Python, Python for Humans, etc.)

Design Principles: Simplicity

```
reghdfe price gear, a(turn trunk, save)
```

Design Principles: Powerful Under the Hood

IV Regressions:

```
reghdfe price (gear=length), a(turn trunk)
```

Multi–way clustering:

```
reghdfe price gear, a(turn trunk)
vce(cluster turn foreign)
```

Additional VCE methods:

```
reghdfe price gear, a(turn t) vce(cluster
turn t, bw(2) kernel(parzen))
```

Design Principles: Powerful Under the Hood

Supports most standard Stata features:

```
reghdfe L.price i.foreign [aw=length],
a(turn trunk)
```

Heterogeneous slopes:

```
reghdfe price weight, a(turn##c.gear)
```

```
reghdfe price weight, a(turn##c.(gear
length) trunk)
```

Design Principles: Powerful Under the Hood

Save users' time:

```
reghdfe price gear, absorb(turn#trunk)
cluster(turn#foreign)
```

Also: implemented in heavily optimized Mata code (`reghdfe` is faster than `areg` and `xtreg` even for one set of fixed effects!)

Design Principles: Don’t Reinvent the Wheel

Most features come from the Stata community: see

`reghdfe, version`

- `ivreg2` or `ivregress` for IV/GMM models
- `avar` for VCE estimation
- `tuples` for MWC
- `group3hdfe` to compute degrees–of–freedom
- Learned *a lot* from `reg2hdfe`, `a2reg`, etc.
- Supports `esttab`: `viewsource estfe.ado`

Design Principles: Don't Let Users Shoot Themselves in the Foot

Same principle behind `use ..., clear`

Warn about several gotchas:

- Drop singleton groups, which might affect VCE estimates
- Compute conservative degrees–of–freedom
- Present alternatives to overall R2, which might be misleading

Improvements and Extensions (1)

- Fixed effects are not identified; researchers are using it incorrectly; alternatives?
- Can we provide better VCE estimates? (e.g. Cattaneo et al 2016)
- What if every obs. has a varying number of fixed effects? (board of directors)

Improvements and Extensions (2)

- `lsmr` estimator from Matthieu Gomez
- `ftools` allows significant speedups in Stata with large datasets (based on optimizations by Python's Pandas)
- Publicize collected benchmark datasets

Also see

- Detailed manual
- Github bug tracker

Thank you!