twexp and twgravity: Estimating exponential regression models with two-way fixed effects

Koen Jochmans
University of Cambridge
Cambridge, United Kingdom
kj345@cam.ac.uk

Vincenzo Verardi
Université de Namur
Namur, Belgium
vverardi@unamur.be

Abstract. We introduce the commands twexp and twgravity that implement the estimators developed in Jochmans (2017) for exponential regression models with two-way fixed effects. twexp is applicable to generic $n \times m$ panel data. twgravity is written for the special case where the data is a cross-section on dyadic interactions between n agents. A prime example of the latter is cross-sectional bilateral trade data, where the model of interest is a gravity equation with importer and exporter effects. Both twexp and twgravity can deal with data where n and m are large, that is, the case of many fixed effects. They make use of Mata and are very fast to execute.

Keywords: exponential regression, gravity model, panel data, two-way fixed effects.

Date: October 8, 2019

1 Introduction

The exponential-regression model finds wide application in the analysis of non-negative outcomes such as count data. It has also shown itself to be an attractive alternative to the log-linearized regression model. Indeed, following Santos Silva and Tenreyro (2006), constant-elasticity models are now routinely estimated from data in levels rather than logarithms. This paper presents two Stata routines to estimate exponential regressions with two-way fixed effects.

We consider double-indexed data on a non-negative outcome, y_{ij}, and a p-vector of regressors, $\boldsymbol{x}_{ij}$. The routine twexp is designed to estimate the slope vector $\boldsymbol{\gamma}$ in the $n \times m$ panel model

$$y_{ij} = e(\alpha_i + \beta_j + \boldsymbol{x}_{ij}^\top \boldsymbol{\gamma})\, \varepsilon_{ij}, \qquad \mathbb{E}(\varepsilon_{ij} | \boldsymbol{x}_{11}, \ldots, \boldsymbol{x}_{nm}) = 1, \tag{1}$$

where $i = 1, \ldots n$ and $j = 1, \ldots, m$ and we let $e(a) := \exp(a)$. Here, α_i and β_j are fixed effects and ε_{ij} is a latent disturbance. A slight variation to this is a cross-sectional data set in which we observe outcomes and regressors for the $n \times (n-1)$ pairwise interactions between agent $i = 1, \ldots n$ and $j \neq i$. This is different from the panel-data case as, here, we do not observe y_{ii} and $\boldsymbol{x}_{ii}$. The routine `twgravity` is designed to handle this case. Its name is derived from the leading example of such an application being the estimation of a gravity equation from a cross-section of bilateral trade flows. Here, the outcome is the directed trade flow from i to j, the regressors are measures of distance or (dis-)similarity between i and j, and α_i and β_j are exporter and importer effects, respectively.

The most popular estimator of (1) is the pseudo maximum-likelihood estimator (PMLE) that arises from treating the y_{ij} as conditionally-independent Poisson variates. If we introduce the shorthand

$$u_{ij}(\alpha_i, \beta_j, \boldsymbol{\gamma}) := y_{ij} - e(\alpha_i + \beta_j + \boldsymbol{x}_{ij}^\top \boldsymbol{\gamma}),$$

the PMLE solves the p first-order conditions for $\boldsymbol{\gamma}$,

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \boldsymbol{x}_{ij}\, u_{ij}(\alpha_i, \beta_j, \boldsymbol{\gamma}) = \mathbf{0},$$

jointly with the $n+m$ first-order conditions for the effects $\alpha_1, \ldots, \alpha_n$ and $\beta_1, \ldots, \beta_m$,

$$\sum_{j=1}^{m} u_{ij}(\alpha_i, \beta_j, \boldsymbol{\gamma}) = 0, \quad i = 1, \ldots, n, \quad \sum_{i=1}^{n} u_{ij}(\alpha_i, \beta_j, \boldsymbol{\gamma}) = 0, \quad j = 1, \ldots, m,$$

subject to a suitable normalization on the fixed effects, such as $\sum_{i=1}^{n} \alpha_i = \sum_{j=1}^{m} \beta_j$, for example. In spite of the presence of the growing number of nuisance parameters the estimator of $\boldsymbol{\gamma}$ is consistent and has a correctly-centered limit distribution when either n is large and m is small or when both n and m are large (and of a similar magnitude). Details on the theoretical properties are available in Wooldridge (1999) and Fernández-Val and Weidner (2016).

The pseudo-Poisson approach suffers from two drawbacks. The first is a numerical one. Indeed, the large amount of fixed effects implies that a simple approach that combines, say, `poisson` with $n+m$ dummy variables will be infeasible in many data sets. The routines `poi2hdfe` (Guimarães 2016) or `ppmlhdfe` (Correia et al. 2019) are designed especially to deal with this problem and are useful alternatives here. The second drawback is that the plug-in estimator of the covariance matrix of the above moment conditions is severly biased. The origin of the problem is again the estimation of the incidental parameters. Indeed, calculating the covariance matrix requires estimating terms involving

$$u_{ij}(\alpha_i, \beta_j, \boldsymbol{\gamma})^2$$

which requires estimates of the fixed effects. The latter are both numerous and estimated with low precision, creating an incidental-parameter bias in the estimated covariance matrix. The bias can be severe, as evidenced by the simulation results in Egger and Staub (2016), Jochmans (2017), and Pfaffermayer (2019). The practical implication of this is that the standard errors will often not be an accurate reflection of the statistical precision of the parameter estimates. Often they will be too small. Consequently, reported confidence interval will be too narrow and test procedures will overreject under the null.

Equation (1) is an important member of the class of multiplicative-error models. For such models moment conditions have been derived that are free of fixed effects (Charbonneau 2013, Jochmans 2017). They allow inference on $\boldsymbol{\gamma}$ to be separated from estimation of $\alpha_1, \ldots, \alpha_n$ and $\beta_1, \ldots, \beta_m$. `twexp` and `twgravity` implement estimators based on these moments. Both routines are designed to be computationally efficient and are very fast to implement. Hence, our routines should be a useful addition to the toolbox of empirical researchers working with count data and trade data. Furthermore, as the whole problem is free of nuisance parameters the standard errors do not suffer from an incidental-parameter bias.

2 Moment conditions and estimators

Consider (1) under the assumption that the errors are mutually independent. Then, using that

$$\mathbb{E}\left(\frac{y_{ij}}{e(\boldsymbol{x}_{ij}^\top \boldsymbol{\gamma})} \middle| \boldsymbol{x}_{11}, \ldots, \boldsymbol{x}_{nm} \right) = e(\alpha_i + \beta_j)$$

for all (i, j), we have

$$\mathbb{E}\left(\frac{y_{ij}}{e(\boldsymbol{x}_{ij}^\top \boldsymbol{\gamma})} \frac{y_{i'j'}}{e(\boldsymbol{x}_{i'j'}^\top \boldsymbol{\gamma})} - \frac{y_{ij'}}{e(\boldsymbol{x}_{ij'}^\top \boldsymbol{\gamma})} \frac{y_{i'j}}{e(\boldsymbol{x}_{i'j}^\top \boldsymbol{\gamma})} \middle| \boldsymbol{x}_{11}, \ldots, \boldsymbol{x}_{nm} \right) = 0 \tag{2}$$

for all i, i' and j, j'. This (conditional) moment condition for $\boldsymbol{\gamma}$ is free of incidental parameters. Equation (2) implies unconditional moment conditions that can form the basis of a method-of-moment (MM) estimator of $\boldsymbol{\gamma}$. Our Stata routines implement two of these estimators.

The first estimator, which we dub GMM1 below, uses the levels of the covariates,

$\boldsymbol{x}_{ij}$ as instruments. It is the solution to

$$\boldsymbol{s}_1(\boldsymbol{\gamma}) := \sum_{i=1}^{n}\sum_{i'=1}^{n}\sum_{j=1}^{m}\sum_{j'=1}^{m} \boldsymbol{x}_{ij}\left\{\frac{y_{ij}}{e(\boldsymbol{x}_{ij}^\top\boldsymbol{\gamma})}\frac{y_{i'j'}}{e(\boldsymbol{x}_{i'j'}^\top\boldsymbol{\gamma})} - \frac{y_{ij'}}{e(\boldsymbol{x}_{ij'}^\top\boldsymbol{\gamma})}\frac{y_{i'j}}{e(\boldsymbol{x}_{i'j}^\top\boldsymbol{\gamma})}\right\} = \boldsymbol{0}.$$

This is a system of p equations and is, therefore, just identified.[1] Consequently, the estimator is

$$\widehat{\boldsymbol{\gamma}}_1 := \arg\min_{\boldsymbol{\gamma}} \boldsymbol{s}_1(\boldsymbol{\gamma})^\top \boldsymbol{s}_1(\boldsymbol{\gamma}).$$

Under suitable regularity conditions $\widehat{\boldsymbol{\gamma}}_1$ is consistent and asymptotically normal. Its asymptotic variance has a sandwich form and can be estimated as $\boldsymbol{Q}_1^{-1}\boldsymbol{V}_1\boldsymbol{Q}_1^{-\top}$, where

$$\boldsymbol{Q}_1 := -\sum_{i=1}^{n}\sum_{i'=1}^{n}\sum_{j=1}^{m}\sum_{j'=1}^{m} \boldsymbol{x}_{ij}\left\{\frac{y_{ij}y_{i'j'}(\boldsymbol{x}_{ij}+\boldsymbol{x}_{i'j'})^\top}{e(\boldsymbol{x}_{ij}^\top\widehat{\boldsymbol{\gamma}}_{\boldsymbol{1}})e(\boldsymbol{x}_{i'j'}^\top\widehat{\boldsymbol{\gamma}}_1)} - \frac{y_{ij'}y_{i'j}(\boldsymbol{x}_{i'j}+\boldsymbol{x}_{ij'})^\top}{e(\boldsymbol{x}_{ij'}^\top\widehat{\boldsymbol{\gamma}}_1)e(\boldsymbol{x}_{i'j}^\top\widehat{\boldsymbol{\gamma}}_1)}\right\},$$

is the Jacobian of the empirical moments evaluated at the point estimator and the variance of the moments is estimated by

$$\boldsymbol{V}_1 := \sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{v}_{ij}\boldsymbol{v}_{ij}^\top,$$

where we define the p-vector $\boldsymbol{v}_{ij}$ as

$$4\sum_{i'\neq i}\sum_{j'\neq j}\{(\boldsymbol{x}_{ij}-\boldsymbol{x}_{ij'})-(\boldsymbol{x}_{i'j}-\boldsymbol{x}_{i'j'})\}\left\{\frac{y_{ij}}{e(\boldsymbol{x}_{i'j}^\top\widehat{\boldsymbol{\gamma}}_1)}\frac{y_{i'j'}}{e(\boldsymbol{x}_{ij'}^\top\widehat{\boldsymbol{\gamma}}_1)} - \frac{y_{ij'}}{e(\boldsymbol{x}_{ij}^\top\widehat{\boldsymbol{\gamma}}_1)}\frac{y_{i'j}}{e(\boldsymbol{x}_{i'j'}^\top\widehat{\boldsymbol{\gamma}}_1)}\right\}.$$

The use of $\boldsymbol{V}_1$ is needed to handle the fact that each observation appears in many of the summands that make up $\boldsymbol{s}_1(\boldsymbol{\gamma})$.

The second estimator we implement, GMM2, is

$$\widehat{\boldsymbol{\gamma}}_2 := \arg\min_{\boldsymbol{\gamma}} \boldsymbol{s}_2(\boldsymbol{\gamma})^\top \boldsymbol{s}_2(\boldsymbol{\gamma}),$$

which is of the same form as $\widehat{\boldsymbol{\gamma}}_1$ but solves the empirical moment equations

$$\boldsymbol{s}_2(\boldsymbol{\gamma}) := \sum_{i=1}^{n}\sum_{i'=1}^{n}\sum_{j=1}^{m}\sum_{j'=1}^{m} \boldsymbol{x}_{ij}\left\{\frac{y_{ij}}{e(-\boldsymbol{x}_{i'j}^\top\boldsymbol{\gamma})}\frac{y_{i'j'}}{e(-\boldsymbol{x}_{ij'}^\top\boldsymbol{\gamma})} - \frac{y_{ij'}}{e(-\boldsymbol{x}_{ij}^\top\boldsymbol{\gamma})}\frac{y_{i'j}}{e(-\boldsymbol{x}_{i'j'}^\top\boldsymbol{\gamma})}\right\} = \boldsymbol{0}.$$

1. As written here the moment equations of GMM1 can be set arbitrarily close to zero when the regressors are all non-negative by setting one of the elements of $\boldsymbol{\gamma}$ arbitrarily large. This can be resolved by transforming all regressors into deviations from their overall mean. Doing so does not alter the roots of the original estimating equation. Both of our Stata routines perform this normalization by default.

The large-sample behavior of this estimator parallels that of $\widehat{\gamma}_1$. The matrices $\boldsymbol{Q}_2$ and $\boldsymbol{V}_2$ needed to estimate the variance of the limit distribution are readily obtained. We omit further details here for brevity. There is an array of other possible estimators that can be derived from the conditional moment conditions above. Motivations for the estimators considered here are given in the supplementary material to Jochmans (2017).

The choice between the two estimators depends on the application at hand. The simulation results in Jochmans (2017) show that GMM2 tends to be more efficient than GMM1 in designs where the conditional variance increases with the conditional mean while GMM1 is relatively more precise in the other situations. In extensive numerical work we have found that GMM1 is extremely stable, making it very reliable. When the linear index $\boldsymbol{x}_{ij}^\top \boldsymbol{\gamma}$ can take on very large values the objective function of GMM2 can have multiple local maxima and regions over which it is fairly flat. This can be understood by noting that $\boldsymbol{s}_2(\boldsymbol{\gamma})$ can be obtained from $\boldsymbol{s}_1(\boldsymbol{\gamma})$ by multiplying through the latter's summand with $e((\boldsymbol{x}_{ij} + \boldsymbol{x}_{i'j'} + \boldsymbol{x}_{i'j} + \boldsymbol{x}_{ij'})^\top \boldsymbol{\gamma})$. This complicates numerical optimization using gradient-based methods such as the Newton algorithm that we use. Our code checks whether a global optimum has been reached by verifying whether the empirical moments are (up to tolerance) equal to zero at the solution and gives a warning if not. If this happens we suggest to experiment with different starting values or to switch to GMM1 in stead.

The large number of terms in $\boldsymbol{s}_1(\boldsymbol{\gamma})$ and $\boldsymbol{s}_2(\boldsymbol{\gamma})$ may suggest that evaluation of the objective function is time consuming, making estimation and inference based on them infeasible in large data sets (see, for example, the discussion in Egger and Staub 2016). This is not the case. Careful inspection and subsequent re-arrangement of terms reveals that evaluation of these equations is immediate in any matrix-based language (here, `Mata`). Additional details on this are provided in the appendix. The same is true for the Jacobian matrices $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ as well as for the variance estimators $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$. `twexp` and `twgravity` are written for balanced data sets. The implementation of our efficient computations would require adjustment to deal with gaps in the data. The exact form of the adjustment depends on the pattern of missingness of the data and is, therefore, not easily programmed in a generic manner. We note that merely dropping observations for which information is missing is not sufficient. This is because of the structure of the empirical moments, where each summand depends on quadruples of observations. One may, of course, decide to resort to brute-force evaluation of the criterion in such cases.

3 Stata commands

3.1 Command: twexp

The command `twexp` is designed for (balanced) $n \times m$ panel data sets.

Syntax

`twexp` has the following syntax:

`twexp` *depvar* [*indepvars*] *,* `indn`*(varname)* `indm`*(varname)* `model`*(option)* `init`*(vec)*

Here,

indn(varname) declares the cross-sectional dimension of the panel.

indm(varname) declares the time-series dimension of the panel.

model(option) determines whether $GMM1$ or $GMM2$ is implemented.

init(vec) specifies the starting value for the numerical optimization; the default is the zero vector.

A table in standard layout reports point estimates, standard errors, z-statistics and p-values for the null that the coefficient in question is equal to zero, and 95% confidence intervals for each of the coefficients. The vector of point estimates and their estimated covariance matrix can be recovered by typing `matrix list e(b)` and `matrix list e(V)`, respectively.

3.2 Command: twgravity

The command `twgravity` is designed for a cross-section on dyadic interactions between n agents. Agents do not interact with themselves, so y_{ii} and $\boldsymbol{x}_{ii}$ are not defined. This is like a panel model with $m = n - 1$. In the vectors and matrices defined in Section 2 this only requires modifying the range over which the sums go. To evaluate the criterion function efficiently, however, additional intervention is needed (see the discussion on gaps in the previous section). Therefore, a different Stata command is provided to deal with this case.

Syntax

`twgravity` has the same syntax as `twexp`:

`twgravity` *depvar* [*indepvars*] , `indn`*(varname)* `indm`*(varname)* `model`*(option)* `init`*(vec)*

Here, again,

indn(varname) identifies the first agent in the dyad.

indm(varname) identifies the second agent in the dyad.

model(option) determines whether $GMM1$ or $GMM2$ is implemented.

init(vec) specifies the starting value for the numerical optimization; the default is the zero vector.

The screen output has the same form as before.

4 Examples

4.1 Patents and R&D

We illustrate the use of `twexp` by looking at the relationship between the number of patent applications and R&D expenditure. We use the data of Hall et al. (1986). The data can be downloaded from the companion website of the textbook Cameron and Trivedi (2005) at

`http://cameron.econ.ucdavis.edu/mmabook/mmaprograms.html`,

however, they are not in Stata format. We load them into Stata by typing the following set of commands:

```
clear
infile CUSIP ARDSSIC SCISECT LOGK SUMPAT LOGR70 LOGR71 LOGR72 LOGR73 ///
LOGR74 LOGR75 LOGR76 LOGR77 LOGR78 LOGR79 PAT70 PAT71 PAT72 ///
PAT73 PAT74 PAT75 PAT76 PAT77 PAT78 PAT79 ///
using "http://cameron.econ.ucdavis.edu/mmabook/patr7079.asc"
* Use observation number as an identifier, not just CUSIP
gen id = _n
label variable id "id"

reshape long PAT LOGR, i(id) j(year)
```

The data is a balanced panel on 346 firms and spans the period 1970–1979; note that Cameron and Trivedi (2005) drop all observations for the period 1970-74 but we do not. For each of the firms we have data on the number of patents applied to (`PAT`) in each year (and were eventually granted) as well as the log of the amount (in 1972 U. S. dollars) spent on R&D during each year (`LOGR`). A summary of these data is as follows:

Variable	Obs	Mean	Std. Dev.	Min	Max
PAT	3,460	36.28439	74.46563	0	608
LOGR	3,460	1.229807	1.970524	-3.84868	7.06524

It is well established that it is important to control for firm-specific heterogeneity by the inclusion of firm fixed effects (Hausman et al. 1984). It also seems important to include a set of time dummies in the specification. These allow to control for aggregate shocks that affect all firms, such as the state of the economy and overall technological progress over time.

Estimating a two-way exponential regression of `PAT` on `LOGR` by means of GMM1 is done by typing

```
twexp PAT LOGR, indn(id) indm(year) model(GMM1)
```

and yields the following output.

Number of obs = 3460

PAT	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
LOGR	.4084421	.0457615	8.93	0.000	.3187521	.498133

The estimator GMM2 is computed by changing the *model* option. For efficiency we let the optimization start at the point estimated obtained by GMM1. To do so we first type `matrix start = e(b)` and next enter

```
twexp PAT RANDD, indn(id) indm(year) model(GMM2) init(start)
```

The output for GMM2 is

Number of obs = 3460

PAT	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
LOGR	.3241356	.0635514	5.10	0.000	.1995772	.448694

4.2 International trade

We use the model and data of Santos Silva and Tenreyro (2006) to illustrate the use of `twgravity`. The data set can be downloaded from

`http://personal.lse.ac.uk/tenreyro/lgw.html`.

The data is a cross-section on bilateral trade flows between 136 countries. The outcome variable is bilateral trade, measured in 1,000 U. S. dollars (`trade`). The regressors are all measures of distances between the importing and exporting country. They are (the logarithm of) geographical distance (`ldist`) and a set of dummies that aim to capture other factors of relatedness. These dummies indicate whether or not countries i and j share a common border (`border`), speak the same language (`comlang`), have a colonial history (`colony`), and take part in a common free-trade agreement (`comfrt_wto`). For each observation the variables `s1_im` and `s2_ex` identify the importer and exporter, respectively.

Variable	Obs	Mean	Std. Dev.	Min	Max
trade	18,360	172129.5	1829058	0	1.01e+08
ldist	18,360	8.785508	.7416775	4.876723	9.898691
border	18,360	.0196078	.1386522	0	1
comlang	18,360	.209695	.407102	0	1
colony	18,360	.1704793	.3760636	0	1
comfrt_wto	18,360	.0250545	.1562948	0	1

Estimating this model by GMM1 is done by typing

```
twgravity trade ldist border comlang colony comfrt_wto, indn(s2_ex)
   indm(s1_im) model(GMM1)
```

and completes in .81 seconds (using Stata/MP 15.1 on a MacBook 1.4HGz Intel Core i7 with 16GB RAM). The following output is reported.

Number of obs = 18360

trade	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
ldist	-.8165761	.0629112	-12.97	0.000	- .9398820	-.6932702
border	.4873677	.1361165	3.58	0.000	.2205795	.7541561
comlang	.2594789	.1300016	2.00	0.045	.0046756	.5142818
colony	.1648687	.1461561	1.13	0.256	- .1215973	.4513347
comfrt_wto	.3064196	.1250841	2.45	0.014	.0612548	.5515846

Changing the estimator used to GMM2 is done by typing

```
twgravity trade ldist border comlang colony comfrt_wto, indn(s2_ex)
   indm(s1_im) model(GMM2)
```

which terminates after 1.85 seconds with the following output.

```
                                               Number of obs =    18360

       trade       Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]

       ldist   -.7509313   .0567805   -13.23   0.000    -.8622191   -.6396436
      border    .1490604   .0771748     1.93   0.053    -.0021994    .3003202
     comlang    .4909294   .0929732     5.28   0.000     .3087052    .6731536
      colony    .2128996   .1212684     1.76   0.079    -.0247821    .4505813
  comfrt_wto    .3298556   .1249293     2.64   0.008     .0849987    .5747126
```

These results correspond to those reported in Table 5 of Jochmans (2017). To appreciate the computational speed, estimation by PMLE takes just under 16 seconds when using `poisson` with dummies, 3.87 second when using `poi2hdfe`, and 1.65 seconds when using `ppmlhdfe`.

5 Simulations

We use simulated data to further illustrate `twgravity`. The simulation design has two binary regressors. They are independent and take on the value one with probability .05 and .50, respectively. This makes the first regressor sparse. The coefficient on each regressor is set to unity. All fixed effects are set to zero and errors are drawn from a log-normal distribution such that their logs follow a standard-normal distribution. The regressors are drawn once and held fixed across the $5,000$ Monte Carlo replications. The errors are redrawn in each replication. The sample size was set to $n = 25$, yielding $25 \times 24 = 600$ observations at the dyad level. Simulation results for a variety of other designs and different sample sizes are reported in Jochmans (2017).

The first table below contains summary statistics for the three point estimators considered. BGMM11 refers to the GMM1 point estimator of the first coefficient and BGMM12 refers to the GMM1 point estimator of the second coefficient. This naming convention is also used for GMM2. BPPML1 and BPPML2 refer to the PMLE point estimates.

Variable	Obs	Mean	Std. Dev.	Min	Max
BGMM11	5,000	.9542519	.3584049	-.2982407	2.925951
BGMM12	5,000	1.002699	.109982	.5822676	1.549646
BGMM21	5,000	.9396433	.3955723	-.3508639	3.290722
BGMM22	5,000	.9997944	.1121814	.5487244	1.508134
BPPML1	5,000	.940787	.3754578	-.3382381	2.783273
BPPML2	5,000	1.002575	.1124283	.5688691	1.547408

GMM1 does best in terms of both bias and standard deviation but all estimators perform quite well. The average computational effort for GMM1, GMM2, and PMLE (each starting at a vector of zeros) was .1414 seconds, .1435 seconds, and .1780 seconds, respectively.

The next table provides corresponding summary statistics for the estimated standard errors for each estimator.

Variable	Obs	Mean	Std. Dev.	Min	Max
SEGMM11	5,000	.310138	.0805269	.1471121	.7484761
SEGMM12	5,000	.1115835	.0143905	.0828566	.2427083
SEGMM21	5,000	.3345285	.0903741	.1373527	.8006971
SEGMM22	5,000	.1157641	.0168373	.0827409	.4340205
SEPPML1	5,000	.2538752	.0547559	.1251421	.5346598
SEPPML2	5,000	.1025859	.0128109	.0756624	.2152773

It is of interest to compare the Monte Carlo standard deviation (in the previous table) to the average standard error (in the current table). The ratio of the latter to the former is .8654 and 1.0145 for GMM1, .8457 and 1.0319 for GMM2, and .67612 and .9125 for PMLE. Thus, the standard errors for pseudo-Poisson estimator are quite a bit too low, on average.

6 Conclusion

We have introduced the Stata routines `twexp` and `twgravity` for exponential-regression models with two-way fixed effects. These estimators are based on Jochmans (2017). They are fast to compute, even in large data sets, and yield reliable standard errors for inference.

Acknowledgments

Koen Jochmans gratefully acknowledges financial support from the European Research Council through Starting Grant n^o 715787. Vincenzo Verardi gratefully acknowledges financial support from the FNRS.

About the authors

Koen Jochmans is a Reader in the Faculty of Economics and a Fellow of Trinity Hall at the University of Cambridge. Vincenzo Verardi is Associate Researcher of the FNRS.

7 References

Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. Cambridge University Press.

Charbonneau, K. B. 2013. Multiple fixed effects in theoretical and applied econometrics. PhD thesis, Princeton University.

Correia, S., P. Guimarães, and T. Zylkin. 2019. PPMLHDFE: Stata module for Poisson pseudo-likelihood regression with multiple levels of fixed effects. Mimeo.

Egger, P. H., and K. E. Staub. 2016. GLM estimation of trade gravity models. *Empirical Economics* 50: 137–175.

Fernández-Val, I., and M. Weidner. 2016. Individual and time effects in nonlinear panel data models with large N, T. *Journal of Econometrics* 192: 291–312.

Guimarães, P. 2016. POI2HDFE: Stata module to estimate a Poisson regression with high-dimensional fixed effects. Mimeo.

Hall, B., Z. Griliches, and J. Hausman. 1986. Patents and R&D: Is there a lag? *International Economic Review* 27: 265–283.

Hausman, J. A., B. H. Hall, and Z. Griliches. 1984. Econometric Models for Count Data with an Application to the Patents-R&D Relationship. *Econometrica* 52: 909–938.

Jochmans, K. 2017. Two-way models for gravity. *Review of Economics and Statistics* 99: 478–485.

Pfaffermayer, M. 2019. Gravity models, PPML estimation and the bias of the robust standard errors. Forthcoming in *Applied Economics Letters*.

Santos Silva, J. M. C., and S. Tenreyro. 2006. The log of gravity. *Review of Economics and Statistics* 88: 641–658.

Wooldridge, J. M. 1999. Distribution-free estimation of some nonlinear panel data models. *Journal of Econometrics* 90: 77–97.

8 Appendix

Additional computational details for GMM1

Fix the value of $\boldsymbol{\gamma}$ and introduce the shorthands $e_{ij} := e(\boldsymbol{x}_{ij}^\top \boldsymbol{\gamma})$ and $u_{ij} := y_{ij}/e_{ij}$. First consider the pure panel data case. The (symmetrized) moment conditions for GMM1 are

$$\boldsymbol{s}_1(\boldsymbol{\gamma}) = \sum_{i=1}^{n}\sum_{i'=1}^{n}\sum_{j=1}^{m}\sum_{j'=1}^{m} \boldsymbol{x}_{ij}\, \{u_{ij}u_{i'j'} - u_{ij'}u_{i'j}\}\,.$$

Note that

$$\sum_{i=1}^{n}\sum_{i'=1}^{n}\sum_{j=1}^{m}\sum_{j'=1}^{m} \boldsymbol{x}_{ij}\, u_{ij}u_{i'j'} = \sum_{i=1}^{n}\sum_{j=1}^{m} \boldsymbol{x}_{ij}u_{ij} \sum_{i'=1}^{n}\sum_{j'=1}^{m} u_{i'j'} = \sum_{i=1}^{n}\sum_{j=1}^{m} \boldsymbol{x}_{ij}(u_{ij}\overline{u}),$$

where $\overline{u} := \sum_{i'=1}^{n}\sum_{j'=1}^{m} u_{i'j'}$ is the grand mean of the u_{ij}. Likewise,

$$\sum_{i=1}^{n}\sum_{i'=1}^{n}\sum_{j=1}^{m}\sum_{j'=1}^{m} \boldsymbol{x}_{ij}\, u_{ij'}u_{i'j} = \sum_{i=1}^{n}\sum_{j=1}^{m} \boldsymbol{x}_{ij} \sum_{i'=1}^{n} u_{i'j} \sum_{j'=1}^{m} u_{ij'} = \sum_{i=1}^{n}\sum_{j=1}^{m} \boldsymbol{x}_{ij}(\overline{u}_{i\cdot}\overline{u}_{\cdot j}),$$

where $\overline{u}_{i\cdot} := \sum_{j'=1}^{m} u_{ij'}$ and $\overline{u}_{\cdot j} := \sum_{i'=1}^{m} u_{i'j}$ are the means taken with respect to each of the two dimensions of the data. Consequently,

$$\boldsymbol{s}_1(\boldsymbol{\gamma}) = \sum_{i=1}^{n}\sum_{j=1}^{m} \boldsymbol{x}_{ij}\, \{u_{ij}\overline{u} - \overline{u}_{i\cdot}\overline{u}_{\cdot j}\}\,,$$

which is fast to evaluate in any matrix-based language. Expressions for the Jacobian matrix $\boldsymbol{Q}_1$ and for $\boldsymbol{v}_{ij}$ follow in the same manner. All these expressions are used in the implementation of `twexp`.

In `twgravity` self-links are ruled out, i.e., the observations $y_{ii}, \boldsymbol{x}_{ii}$ are not in the data. In this case the empirical moments for GMM1 become

$$\boldsymbol{s}_1(\boldsymbol{\gamma}) = \sum_{i=1}^{n}\sum_{i'\neq i}\sum_{j\neq i,i'}\sum_{j'\neq i,i',j} \boldsymbol{x}_{ij}\, \{u_{ij}u_{i'j'} - u_{ij'}u_{i'j}\}\,;$$

note the change in the range of the sums. It is convenient to *define* $y_{ii} = 0$ and $\boldsymbol{x}_{ii} = \boldsymbol{0}$. Then, in the same way as before,

$$\sum_{i=1}^{n}\sum_{i'\neq i}\sum_{j\neq i,i'}\sum_{j'\neq i,i',j} \boldsymbol{x}_{ij}u_{ij}u_{i'j'} = \sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{x}_{ij}u_{ij}(\overline{u} - \overline{u}_{\cdot i} - \overline{u}_{j\cdot} + u_{ji})$$

and

$$\sum_{i=1}^{n}\sum_{i'\neq i}\sum_{j\neq i,i'}\sum_{j'\neq i,i',j}\boldsymbol{x}_{ij}u_{i'j}u_{ij'}=\sum_{i=1}^{n}\sum_{j=1}^{n}\boldsymbol{x}_{ij}(\overline{u}_{i\cdot}\overline{u}_{\cdot j}-\check{u}_{ij})$$

where $\check{u}_{ij}:=\sum_{i'=1}^{n}u_{ii'}u_{i'j}$. Consequently, in this case we have

$$\boldsymbol{s}_1(\boldsymbol{\gamma})=\sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{x}_{ij}\;\{u_{ij}\overline{u}-\overline{u}_{i\cdot}\overline{u}_{\cdot j}\}-\sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{x}_{ij}\{u_{ij}(\overline{u}_{\cdot i}+\overline{u}_{j\cdot}-u_{ji})-\check{u}_{ij}\}$$

The additional term on the right-hand side compared to the corresponding expression above is a correction term for the absence of self links in the data. The Jacobian matrix and the covariance matrix of the moment conditions can again be obtained in a similar manner.

Additional computational details for GMM2

Fix the value of $\boldsymbol{\gamma}$ and introduce the shorthand $e_{ij}:=e(\boldsymbol{x}_{ij}^{\top}\boldsymbol{\gamma})$ First consider the pure panel data case. The (symmetrized) moment conditions for GMM2 are

$$\boldsymbol{s}_2(\boldsymbol{\gamma})=\sum_{i=1}^{n}\sum_{i'=1}^{n}\sum_{j=1}^{m}\sum_{j'=1}^{m}\boldsymbol{x}_{ij}\;\{y_{ij}y_{i'j'}e_{i'j}e_{ij'}-y_{ij'}y_{i'j}e_{ij}e_{i'j'}\}\,.$$

Here, defining the $n\times m$ matrices $(\boldsymbol{Y})_{ij}:=y_{ij}$ and $(\boldsymbol{E})_{ij}:=e_{ij}$ we can compactly write

$$\boldsymbol{x}_{ij}y_{ij}\sum_{i'=1}^{n}\sum_{j'=1}^{n}\varphi_{ij'}y_{i'j'}\varphi_{i'j}=\boldsymbol{x}_{ij}y_{ij}(\boldsymbol{E}\boldsymbol{Y}^{\top}\boldsymbol{E})_{ij},$$

$$\boldsymbol{x}_{ij}e_{ij}\sum_{i'=1}^{n}\sum_{j'=1}^{n}y_{ij'}e_{i'j'}y_{i'j}=\boldsymbol{x}_{ij}e_{ij}(\boldsymbol{Y}\boldsymbol{E}^{\top}\boldsymbol{Y})_{ij};$$

note that the terms on the right-hand side here are quadratic forms in $\boldsymbol{E}$ and $\boldsymbol{Y}$. Hence,

$$\boldsymbol{s}_2(\boldsymbol{\gamma})=\sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{x}_{ij}\left\{y_{ij}\,(\boldsymbol{E}\boldsymbol{Y}^{\top}\boldsymbol{E})_{ij}-e_{ij}(\boldsymbol{Y}\boldsymbol{E}^{\top}\boldsymbol{Y})_{ij}\right\},$$

which is again immediate to compute in any matrix-based language. When self-links are ruled out—again *defining* $y_{ii}=0$, $\boldsymbol{x}_{ii}=\boldsymbol{0}$, and now also setting $e_{ii}=0$, no further modification is needed for GMM2.