

Guimarães, Paulo; Portugal, Pedro

Working Paper

A simple feasible alternative procedure to estimate models with high-dimensional fixed effects

IZA Discussion Papers, No. 3935

Provided in Cooperation with:

IZA – Institute of Labor Economics

Suggested Citation: Guimarães, Paulo; Portugal, Pedro (2009) : A simple feasible alternative procedure to estimate models with high-dimensional fixed effects, IZA Discussion Papers, No. 3935, Institute for the Study of Labor (IZA), Bonn, <https://nbn-resolving.de/urn:nbn:de:101:1-2009020957>

This Version is available at:

<https://hdl.handle.net/10419/35698>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

IZA DP No. 3935

A Simple Feasible Alternative Procedure to Estimate Models with High-Dimensional Fixed Effects

Paulo Guimarães
Pedro Portugal

January 2009

A Simple Feasible Alternative Procedure to Estimate Models with High-Dimensional Fixed Effects

Paulo Guimarães

*University of South Carolina,
CEFUP and IZA*

Pedro Portugal

*Banco de Portugal,
Universidade Nova de Lisboa and IZA*

Discussion Paper No. 3935

January 2009

IZA

P.O. Box 7240
53072 Bonn
Germany

Phone: +49-228-3894-0
Fax: +49-228-3894-180
E-mail: iza@iza.org

Any opinions expressed here are those of the author(s) and not those of IZA. Research published in this series may include views on policy, but the institute itself takes no institutional policy positions.

The Institute for the Study of Labor (IZA) in Bonn is a local and virtual international research center and a place of communication between science, politics and business. IZA is an independent nonprofit organization supported by Deutsche Post World Net. The center is associated with the University of Bonn and offers a stimulating research environment through its international network, workshops and conferences, data service, project support, research visits and doctoral program. IZA engages in (i) original and internationally competitive research in all fields of labor economics, (ii) development of policy concepts, and (iii) dissemination of research results and concepts to the interested public.

IZA Discussion Papers often represent preliminary work and are circulated to encourage discussion. Citation of such a paper should account for its provisional character. A revised version may be available directly from the author.

ABSTRACT

A Simple Feasible Alternative Procedure to Estimate Models with High-Dimensional Fixed Effects

In this paper we describe an alternative iterative approach for the estimation of linear regression models with high-dimensional fixed-effects such as large employer-employee data sets. This approach is computationally intensive but imposes minimum memory requirements. We also show that the approach can be extended to non-linear models and potentially to more than two high dimensional fixed effects.

JEL Classification: C01, C81

Keywords: high dimensional fixed effects, linked employer-employee data

Corresponding author:

Paulo Guimarães
University of South Carolina
Moore School of Business
1705 College Street
Columbia, SC 29208
USA
E-mail: guimaraes@moore.sc.edu

1 Introduction

The increasing availability of large micro-level data sets has spurred interest on methods for estimation of models with high-dimensional fixed-effects. Researchers in several fields such as economics, sociology, and political science among others, find the introduction of fixed effects a particularly appealing way of controlling for unobserved heterogeneity that is shared among groups of observations. In this case it becomes possible to account for all inter-group variability by adding to the set of regressors dummy variables that absorb group specific heterogeneity. This approach has the advantage of allowing for the existence of general patterns of correlation between the unobserved effects and the other regressors. In practice, when estimating a model with a single fixed-effect (i.e. a factor in the analysis of covariance) one is not required to actually add the group dummy variables to the set of regressors. This is particularly convenient when dealing with high-dimensional fixed effects, that is, in a situation when the number of groups (dummy variables) is very large. For several common procedures such as linear regression, Poisson and logit regression the fixed effect can be eliminated from the model making it possible to obtain estimates for the coefficients of the relevant regressors without having to introduce the group dummy variables in the model. For other nonlinear models it is still possible to avoid the explicit introduction of dummy variables to account for the fixed effect by modifying the iterative algorithm used to solve the maximum-likelihood problem [see Greene (2004)]. However, there is no simple solution when there is more than one high-dimensional fixed effect. A notable example are the large employer-employee panel data sets commonly used in the labor economics literature. When studying relations in the labor market researchers often want to simultaneously account for two sources of unobserved heterogeneity, the firm and the worker. Explicit introduction of dummy variables is not an option because the number of units (groups) for either firms or workers is too large. Other well known examples of large data sets with obvious sources of unobserved heterogeneity are panel data sets of patient claims data - here the potential sources of heterogeneity are the patient, the doctor and the hospital; and panel data sets on student performance - the potential sources of heterogeneity being the students, the teachers, and the school.

Abowd et al. (1999) tackled the problem of accounting for two high-dimensional fixed effects in the linear regression problem. In a widely cited paper, the authors proposed several methods that provide approximate solutions to this problem.¹ Later, in an unpublished paper Abowd et al. (2002) presented an iterative algorithm that leads to the exact solution for the least square estimation of the model with two

¹For a discussion on the implementation of these methods in Stata see Andrews et al. (2006).

fixed effects. The user-written command `a2reg` is a Stata implementation of this algorithm by Amine Ouazad.² In a recent article published in this journal Cornelissen (2008) presented a new user written command, `felsdvreg`, which consists of a memory saving procedure for estimation of a linear regression model with two high-dimensional fixed effects.

In our own work, Carneiro et al. (2008), we were faced with the problem of estimating a linear regression model with 26 variables and two high-dimensional fixed effects (firm and worker) using a linked employer-employee data set with over 26 million observations. Implementation of the user-written commands discussed above in a computer with 8 Gigabytes of RAM and running Stata MP for Windows failed because of memory limitations. However, using a simple to implement iterative procedure we were able to estimate the regression coefficients and the fixed effects themselves. The approach is computationally intensive but it has the advantage of imposing minimal memory requirements. In this paper we present a detailed discussion of the method proposed in Carneiro et al. (2008) and show how it can be extended to non-linear models and possibly to applications with more than two high-dimensional fixed effects.

2 The Linear Regression Model

2.1 One fixed effect

To start with consider the conventional linear regression model setup:

$$y_i = \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_k x_{ki} + \epsilon_i \quad (1)$$

or, more compactly,

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon. \quad (2)$$

Application of the least squares method results in a set of equations (the normal equations) given below:

$$\left[\begin{array}{l} \frac{\partial SS}{\partial \beta_1} = \sum_i x_{1i}(y_i - \beta_1 x_{1i} - \beta_2 x_{2i} - \dots - \beta_k x_{ki}) = 0 \\ \frac{\partial SS}{\partial \beta_2} = \sum_i x_{2i}(y_i - \beta_1 x_{1i} - \beta_2 x_{2i} - \dots - \beta_k x_{ki}) = 0 \\ \dots \\ \frac{\partial SS}{\partial \beta_k} = \sum_i x_{ki}(y_i - \beta_1 x_{1i} - \beta_2 x_{2i} - \dots - \beta_k x_{ki}) = 0 \end{array} \right] \quad (3)$$

²Apparently `a2reg` should only be used when there are no nested fixed effects.

These equations have a closed form solution, the least squares estimator, given by the well-known formula:

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{Y} . \quad (4)$$

However, the above formula is one of several alternatives to find the solution to (3). For instance, we can solve for $\hat{\beta}$ using a partitioned iterative algorithm. An example of such algorithm is shown below:

- Initialize $\beta_1^{(0)}, \beta_2^{(0)}, \dots, \beta_k^{(0)}$
- Solve for $\beta_1^{(1)}$ as the solution to $\frac{\partial SS}{\partial \beta_1} = \sum_i x_{1i}(y_i - \beta_1 x_{1i} - \beta_2^{(0)} x_{2i} - \dots - \beta_k^{(0)} x_{ki}) = 0$
- Solve for $\beta_2^{(1)}$ as the solution to $\frac{\partial SS}{\partial \beta_2} = \sum_i x_{2i}(y_i - \beta_1^{(1)} x_{1i} - \beta_2 x_{2i} - \dots - \beta_k^{(0)} x_{ki}) = 0$
- ...
- Repeat until convergence.

Algorithms such as this are discussed in Smyth (1996). This algorithm is known as the "zigzag" or full Gauss-Seidel algorithm. According to Smyth this algorithm produces a stable but slow iteration depending on the correlation of the parameters. In this particular case use of an iterative algorithm to solve the normal equations is highly inefficient. However, we should note that this implementation has the advantage of not requiring the explicit calculation of the inverse of the $\mathbf{X}'\mathbf{X}$ matrix.

Consider now what happens if we include a set of dummy variables to account for a fixed effect in the regression. In that case,

$$\mathbf{Y} = \mathbf{Z}\beta + \mathbf{D}\alpha + \epsilon \quad (5)$$

where \mathbf{Z} is the matrix of explanatory variables with $N \times k$ dimension and \mathbf{D} is the $N \times G_1$ matrix of "dummy" variables. Now, we can write the normal equations as:

$$\begin{bmatrix} \mathbf{Z}'\mathbf{Z} & \mathbf{Z}'\mathbf{D} \\ \mathbf{D}'\mathbf{Z} & \mathbf{D}'\mathbf{D} \end{bmatrix} \begin{bmatrix} \beta \\ \alpha \end{bmatrix} = \begin{bmatrix} \mathbf{Z}'\mathbf{Y} \\ \mathbf{D}'\mathbf{Y} \end{bmatrix} \quad (6)$$

which can be arranged to show

$$\begin{bmatrix} \mathbf{Z}'\mathbf{Z}\beta + \mathbf{Z}'\mathbf{D}\alpha = \mathbf{Z}'\mathbf{Y} \\ \mathbf{D}'\mathbf{Z}\beta + \mathbf{D}'\mathbf{D}\alpha = \mathbf{D}'\mathbf{Y} \end{bmatrix} . \quad (7)$$

Solving each set of equations independently yields,

$$\begin{bmatrix} \beta = (\mathbf{Z}'\mathbf{Z})^{-1} \mathbf{Z}' (\mathbf{Y} - \mathbf{D}\alpha) \\ \alpha = (\mathbf{D}'\mathbf{D})^{-1} \mathbf{D}' (\mathbf{Y} - \mathbf{Z}\beta) \end{bmatrix}. \quad (8)$$

The above partition of the normal equations suggests a convenient iteration strategy. To obtain the exact least squares solution one can simply alternate between estimation of β and estimation of α . But the key thing to note is that we no longer have to worry about the dimensionality of \mathbf{D} . The expression $(\mathbf{D}'\mathbf{D})^{-1} \mathbf{D}'$ used on the estimation of α translates into a simple group average of the residuals of the regression of \mathbf{Y} on \mathbf{Z} . On the other hand, the expression $\mathbf{D}\alpha$ that shows up on the equation for β is a column vector containing all the elements of α . Estimation of β consists of a simple linear regression of a transformed \mathbf{Y} on \mathbf{Z} . In our implementation instead of transforming \mathbf{Y} we will keep \mathbf{Y} as the dependent variable and add $\mathbf{D}\alpha$ as an additional covariate. When the estimation procedure converges the coefficient on $\mathbf{D}\alpha$ must equal one and the vector $\mathbf{D}\alpha$ will contain all the estimated coefficients for the group dummy variables. With this approach we avoided inversion of a potentially large matrix that would be required if we had simply added \mathbf{D} to the set of regressors. As an illustration of this approach we use the Stata data set *nlswork* and replicate the coefficient estimates obtained with the fixed effect estimator for the linear regression model as shown in page 400 of the [XT] Stata Manual version 10.

```
. webuse nlswork
(National Longitudinal Survey. Young Women 14-26 years of age in 1968)
. generate age2 = age^2
(24 missing values generated)
. generate ttl_exp2 = ttl_exp^2
. generate tenure2 = tenure^2
(433 missing values generated)
. generate double fe1=0
. local rss1=0
. local dif=1
. local i=0
. while abs('dif')>epsdouble() {
2. quietly {
3. regress ln_w age* ttl_exp* tenure* not_smsa south fe1
4. local rss2='rss1'
5. local rss1=e(rss)
6. local dif='rss2'-'rss1'
7. capture drop yhat
8. predict double yhat
9. generate double temp1=ln_w-yhat+_b[fe1]*fe1
10. capture drop fe1
11. egen double fe1=mean(temp1), by(idcode)
12. capture drop temp1
13. local i='i'+1
14. }
```



```

15. }
. di "Total Number of Iterations --> " 'i'
Total Number of Iterations --> 131
. regress ln_w age* ttl_exp* tenure* not_smsa south fe1

```

Source	SS	df	MS			
Model	4437.88225	9	493.098028	Number of obs =	28093	
Residual	1976.12232	28083	.070367209	F(9, 28083) =	7007.50	
				Prob > F =	0.0000	
				R-squared =	0.6919	
				Adj R-squared =	0.6918	
Total	6414.00457	28092	.228321393	Root MSE =	.26527	

ln_wage	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
age	.0359987	.0024266	14.83	0.000	.0312424	.0407549
age2	-.000723	.0000394	-18.34	0.000	-.0008002	-.0006457
ttl_exp	.0334668	.0016297	20.53	0.000	.0302724	.0366611
ttl_exp2	.0002163	.0000873	2.48	0.013	.0000452	.0003873
tenure	.0357539	.0013781	25.94	0.000	.0330528	.038455
tenure2	-.0019701	.0000939	-20.98	0.000	-.0021541	-.0017861
not_smsa	-.0890108	.0036161	-24.62	0.000	-.0960985	-.0819231
south	-.0606309	.0033127	-18.30	0.000	-.067124	-.0541379
fe1	1	.0051626	193.70	0.000	.989881	1.010119
_cons	1.037301	.0340432	30.47	0.000	.9705747	1.104027

As implied earlier, the estimated coefficients are identical to those obtained using [XT] `xtreg` with the `fe` option. Note also that the regression includes an additional variable, `fe1`, with a coefficient of one. This variable was created during estimation and contains the estimates of the fixed effect.

2.2 Two fixed effects

Suppose now that instead of a single high-dimensional fixed-effect we have two high-dimensional fixed-effects. That is, we now intend to estimate the following model

$$\mathbf{Y} = \mathbf{Z}\beta + \mathbf{D}_1\alpha + \mathbf{D}_2\gamma + \epsilon, \quad (9)$$

where \mathbf{D}_1 is $N \times G_1$ and \mathbf{D}_2 is $N \times G_2$ and both G_1 and G_2 have high dimensionality. As discussed earlier, in this particular case estimation of the linear regression model is complicated. However, implementation of the partitioned algorithm discussed above is straightforward. Proceeding as we did before, we can solve the normal equations as

$$\begin{bmatrix} \beta = (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'(\mathbf{Y} - \mathbf{D}_1\alpha - \mathbf{D}_2\gamma) \\ \alpha = (\mathbf{D}'_1\mathbf{D}_1)^{-1}\mathbf{D}'_1(\mathbf{Y} - \mathbf{Z}\beta - \mathbf{D}_2\gamma) \\ \gamma = (\mathbf{D}'_2\mathbf{D}_2)^{-1}\mathbf{D}'_2(\mathbf{Y} - \mathbf{Z}\beta - \mathbf{D}_1\gamma) \end{bmatrix} \quad (10)$$

Iterating between these sets of equations provides us with the exact least squares solution. All we have to do is compute several linear regressions with k explanatory variables and compute group means of residuals. To illustrate the approach we

modify the above algorithm and apply it to the ancillary data set that accompanies the `felsdvreg` command developed by Thomas Cornelissen. As before, we introduce $\mathbf{D}_1\alpha$ and $\mathbf{D}_2\gamma$ as additional regressors instead of subtracting them from the dependent variable.

```

. use felsdvsimul, clear
. generate double temp=0
. generate double fe1=0
. generate double fe2=0
. local rss1=0
. local dif=1
. local i=0
. while abs('dif')>epsdouble() {
2. quietly {
3. regress y x1 x2 fe1 fe2
4. local rss2='rss1'
5. local rss1=e(rss)
6. local dif='rss2'-'rss1'
7. capture drop res
8. predict double res, res
9. replace temp=res+_b[fe1]*fe1, nopromote
10. capture drop fe1
11. egen double fe1=mean(temp), by(i)
12. replace temp=res+_b[fe2]*fe2, nopromote
13. capture drop fe2
14. egen double fe2=mean(temp), by(j)
15. local i='i'+1
16. }
17. }
. di "Total Number of Iterations --> " `i'
Total Number of Iterations --> 694
. regress y x1 x2 fe1 fe2

```

Source	SS	df	MS			
Model	9385.65759	4	2346.4144	Number of obs =	100	
Residual	2076.72508	95	21.860264	F(4, 95) =	107.34	
Total	11462.3827	99	115.781643	Prob > F =	0.0000	
				R-squared =	0.8188	
				Adj R-squared =	0.8112	
				Root MSE =	4.6755	

y	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
x1	1.029258	.162224	6.34	0.000	.707203	1.351314
x2	-.709482	.1604068	-4.42	0.000	-1.02793	-.3910342
fe1	1	.0876916	11.40	0.000	.8259101	1.17409
fe2	1	.0738805	13.54	0.000	.8533287	1.146671
_cons	-1.995179	.4739307	-4.21	0.000	-2.936051	-1.054308

As hinted above, the estimates for the model coefficients are identical to the least squares results with all dummy variables included that is reported in Cornelissen (2008). We should note that the algorithm took 694 iterations to converge. This is one of the drawbacks of this approach. Fortunately, as discussed below, there is

substantial room for improvement. One obvious simplification is to sweep out one of the fixed effects by subtracting the group mean from all variables. By doing this we only need to deal with one fixed effect. This means that with minor modifications the code shown above can be used to estimate a model with 3 high dimensional fixed effects!

In Carneiro et al. (2008) we estimated a conventional Mincerian wage equation with two high dimensional fixed effects. Our data source is the *Quadros do Pessoal*, a mandatory employment survey collected yearly by the Portuguese Ministry of Labor and Social Security. The dataset comprised a total of 26,085,101 observations spanning from 1986 to 2005. In our estimation we wanted to account for firm, worker and year fixed effects. With 541,229 firms and 7,155,898 workers employing dummy variables for one of the fixed effects was not an option. But in the following example we show the output result of one specification with 26 covariates and the two high dimensional fixed effects estimated using the approach outlined above.

```
. regress ln_real_hw school* age* ten* yr* fe1 fe2
```

Source	SS	df	MS	Number of obs =26085101		
Model	7623399.13	28	272264.255	F(28,26085072)=	.	
Residual	606248.16	26085072	.023241192	Prob > F	= 0.0000	
				R-squared	= 0.9263	
				Adj R-squared	= 0.9263	
Total	8229647.29	26085100	.315492265	Root MSE	= .15245	

ln_real_hw	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
school2	.0086473	.0001819	47.54	0.000	.0082908	.0090038
school3	.0326273	.0001993	163.69	0.000	.0322366	.0330179
school4	.0482319	.000202	238.71	0.000	.0478359	.0486279
school5	.1539935	.0002913	528.64	0.000	.1534226	.1545645
school6	.2373886	.000246	964.89	0.000	.2369064	.2378708
age	.0257784	.0000212	1216.00	0.000	.0257369	.02582
agesq	-.0002749	2.77e-07	-993.19	0.000	-.0002754	-.0002744
tenure	.0007567	9.38e-07	806.43	0.000	.0007549	.0007586
tensq	-1.49e-06	2.59e-09	-573.79	0.000	-1.49e-06	-1.48e-06
yr1987	.0504275	.0002091	241.22	0.000	.0500178	.0508373
yr1988	.0659034	.0002072	318.08	0.000	.0654973	.0663095
yr1989	.0800703	.0002049	390.75	0.000	.0796686	.0804719
yr1991	.1895915	.0002019	938.86	0.000	.1891957	.1899872
yr1992	.2346536	.0002009	1167.74	0.000	.2342597	.2350474
yr1993	.2418796	.0002008	1204.65	0.000	.241486	.2422731
yr1994	.2669475	.0001995	1338.10	0.000	.2665565	.2673386
yr1995	.2793258	.0001964	1422.32	0.000	.2789409	.2797107
yr1996	.2905584	.0001971	1474.26	0.000	.2901721	.2909447
yr1997	.3383318	.0001937	1746.45	0.000	.3379521	.3387115
yr1998	.3827365	.000194	1973.12	0.000	.3823563	.3831166
yr1999	.4240073	.0001925	2203.00	0.000	.4236301	.4243845
yr2000	.433117	.0001911	2266.20	0.000	.4327424	.4334916
yr2002	.450015	.0001921	2342.35	0.000	.4496385	.4503916
yr2003	.4531523	.0001913	2368.54	0.000	.4527773	.4535273
yr2004	.4784224	.0001903	2514.21	0.000	.4780495	.4787954
yr2005	.4877544	.0001886	2586.27	0.000	.4873848	.4881241
fe1	1	.0001086	9212.19	0.000	.9997872	1.000213

fe2	1	.0001162	8602.81	0.000	.9997722	1.000228
_cons	-.6465547	.0004291	-1506.62	0.000	-.6473958	-.6457136

2.3 Estimation of the Standard Errors

So far the discussion has focused on estimation of the β coefficients. To provide the standard errors associated with the estimator of β we would need to estimate

$$V(\hat{\beta}) = \sigma^2 (\mathbf{X}'\mathbf{X})^{-1} , \quad (11)$$

which again raises the problem of inverting the $\mathbf{X}'\mathbf{X}$ matrix. An alternative solution to estimate the elements of $V(\hat{\beta})$ is to use the known relation:

$$V(\hat{\beta}_j) = \frac{\sigma^2}{N s_j^2 (1 - R_{j.123\dots}^2)} . \quad (12)$$

where s_j^2 is the sample variance associated with the x_j variable and $R_{j.123\dots}^2$ is the coefficient of determination obtained from a regression of x_j on all other remaining explanatory variables. Estimation of σ^2 is easy because the final regression that provides the estimates of β has the correct sum of squared residuals (SSR). The remaining difficulty is the computation of the number of degrees of freedom associated with SSR. Knowing the dimension of \mathbf{X} is not sufficient because some of the coefficients for the fixed effects may not be identifiable. Abowd et al. (2002) developed an algorithm that can be used for determining the number of identifiable effects. The algorithm is used to identify "mobility groups" but it turns out that the total number of "mobility groups", which we denote by M , is also the number of non-identified coefficients. Hence, the degrees of freedom associated with SSR equal $N - k - G_1 - G_2 + M$. Computation of $R_{j.123\dots}^2$ is no longer a problem because we know how to estimate a model with high-dimensional effects. However, this may be time consuming because it would require estimation of a regression with two high dimensional fixed effects for each of the regressors. Fortunately, there is an alternative strategy that will produce faster results. Simply put, the idea consists of estimating the model in two steps. In the first step we expurgate the two fixed effects from all variables in the model. This involves running a linear regression of each individual variable on the high-dimensional effects only and storing the residuals. In the second step we run the regression of interest and the regressions required by (12) using the stored residuals of the variables obtained in the first step instead of the original variables. Because we are not dealing with the high-dimensional fixed effects, the

regressions in the second step are very easy to implement. One reason why this approach works well is because the calculations in the first step are relatively simple. We can see from (10) that in this case the algorithm involves only the computation of means. In the next example we use again the ancillary data set that accompanies the `felsdvreg` user-written command and show how to compute the (corrected) standard errors in a regression with two high-dimensional fixed effects. In the Stata code shown below we speed up the algorithm by following the common practice of sweeping one of the fixed effects by demeaning the variables.

```

. use felsdvsimul, clear
. gen double temp=0
. gen double fe2=0
. gen double lastfe2=0
. foreach var of varlist y x1 x2 {
2. quietly {
3. local i=0
4. local dif=1
5. capture drop mean
6. egen double mean=mean(`var'), by(i)
7. replace `var'=`var'-mean
8. while abs(`dif')>epsfloat() {
9. replace lastfe2=fe2, nopromote
10. capture drop mean
11. egen double mean=mean(fe2), by(i)
12. capture drop fe2
13. egen double fe2=mean(`var'+mean), by(j)
14. replace temp=sum(reldif(fe2,lastfe2)), nopromote
15. local dif=temp[_N]/_N
16. local i='i'+1
17. }
18. noisily di "Total Number of Iterations for `var' --> " `i'
19. gen double `var'_res=`var'-fe2+mean
20. }
21. }
Total Number of Iterations for y --> 17
Total Number of Iterations for x1 --> 19
Total Number of Iterations for x2 --> 18
. regress y_res x1_res x2_res, nocons

```

Source	SS	df	MS			
Model	1033.49121	2	516.745607	Number of obs =	100	
Residual	2076.72505	98	21.1910719	F(2, 98) =	24.39	
Total	3110.21626	100	31.1021626	Prob > F	= 0.0000	
				R-squared	= 0.3323	
				Adj R-squared	= 0.3187	
				Root MSE	= 4.6034	

y_res	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
x1_res	1.029258	.1805092	5.70	0.000	.6710437	1.387473
x2_res	-.709482	.1757232	-4.04	0.000	-1.058199	-.3607649

Notice that the iterative procedure was much faster, taking less than 20 iterations for each variable. In this example there are 100 observations, 6 "mobility groups",

$G_1 = 15$ and $G_2 = 20$ and two regressors, x_1 and x_2 . This means that there are 69 degrees of freedom. We can now proceed to calculate the standard errors associated with the coefficients. We will do it only for the variable x_1 because the calculations are similar for x_2 .

```
. local s2='e(rss)'/69
. quietly sum x1
. local N=r(N)
. local varx1=r(Var)
. local TSSx1='varx1'*('N'-1)
. quietly regress x1_res x2_res
. local RSSx1=e(rss)
. local R2x1=1-'RSSx1'/'TSSx1'
. local sebx1=sqrt('s2'/((('N'-1)*'varx1'*(1-'R2x1'))))
. di "The standard error associated with x1 is --> 'sebx1'"
The standard error associated with x1 is --> .2151235245693864
```

The estimates for the coefficients of the fixed effects can be easily recovered by implementing the same iterative procedure to the residuals that are obtained when we subtract the effect of x_1 and x_2 from y .

```
. qui regress y_res x1_res x2_res, nocons
. gen double dum=y-_b[x1_res]*x1-_b[x2_res]*x2
. capture drop mean
. egen double mean=mean(dum), by(i)
. qui replace dum=dum-mean
. local i=0
. local dif=1
. while abs('dif')>epsfloat() {
2. qui replace lastfe2=fe2, nopromote
3. capture drop mean
4. egen double mean=mean(fe2), by(i)
5. capture drop fe2
6. egen double fe2=mean(dum+mean), by(j)
7. qui replace temp=sum(reldif(fe2,lastfe2)), nopromote
8. local dif=temp[_N]/_N
9. local i='i'+1
10. }
```

```
. di "Total Number of Iterations for fe2 --> " 'i'
Total Number of Iterations for fe2 --> 17
```

```
. quietly replace dum=dum-fe2
. egen double fe1=mean(dum), by(i)
. regress y x1 x2 fe1 fe2, nocons
```

Source	SS	df	MS	
Model	3887.8086	4	971.952151	Number of obs = 100
Residual	2076.72505	96	21.6325526	F(4, 96) = 44.93
				Prob > F = 0.0000
				R-squared = 0.6518
				Adj R-squared = 0.6373
Total	5964.53365	100	59.6453365	Root MSE = 4.6511

y	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
x1	1.029258	.1715101	6.00	0.000	.6888133 1.369703
x2	-.7094819	.1701882	-4.17	0.000	-1.047303 -.3716611
fe1	1	.192855	5.19	0.000	.617186 1.382814
fe2	1	.0895005	11.17	0.000	.8223429 1.177657

We confirm that the estimated coefficients are correct by adding the variables `fe1` and `fe2` to the linear regression of y on x_1 and x_2 . As expected the estimated β coefficients are the correct ones and the coefficients associated with the variables `fe1` and `fe2` equal one.

Subtracting the influence of the fixed effects from each variable and working only with the residuals has some advantages compared to the process shown earlier entailing direct estimation of the full regression with all the fixed effects added. First, the simple regressions in step 1 are likely to converge at a faster rate. Second, it is possible to test different specifications of the model using only the residuals from the variables without the need to deal with the high dimensional fixed effects. And third, when dealing with very large data sets we can substantially reduce memory requirements because during step 1 we only need to load in memory the variable being handled and the group identifiers for the fixed effects. In fact we could do even better because the solution to the algorithm is performed independently across "mobility groups" meaning that it would be possible to load each "mobility group" in memory separately.

3 Extension to Non-linear Models

In this section we show that the iterative approach outlined earlier for the linear regression model can be extended to non-linear models. Lets first consider a typical Poisson regression model with expected value given by

$$E(y_i) = \lambda_i = \exp(\mathbf{x}'_i \beta) . \tag{13}$$

We know that the maximum-likelihood estimators are obtained as the solution to:

$$\begin{aligned} \frac{\partial \ln L}{\partial \beta} &= \sum_{i=1} y_i \mathbf{x}_i - \sum_{i=1} \mathbf{x}_i \exp(\mathbf{x}'_i \beta) \\ &= \sum_{i=1} (y_i - \exp(\mathbf{x}'_i \beta)) \mathbf{x}_i = \mathbf{0} \end{aligned}$$

If one of the regressors is a dummy variable, say \mathbf{d}_j , then its estimated coefficient, say α_j , has a closed form solution given by:

$$\exp(\alpha_j) = \mathbf{d}'_j \mathbf{y} \times [\mathbf{d}'_j \exp([\mathbf{x}'_i \beta]_{(j)})]^{-1} \quad (14)$$

where the subscript (j) in the argument of the exponential function is used to note that \mathbf{d}_j is excluded from the argument. The above expression suggests a simple iterative strategy, much like the one that was used for the linear regression. We can alternate between estimation of a Poisson regression with k explanatory variables and calculation of the estimates for all the coefficients of the fixed effects using (14). These estimates can be kept in a single column vector. To show this algorithm at work we replicate the estimates of the Poisson model with fixed effects which appears in page 348 of the [XT] Stata Manual Version 10 as an illustration of the [xt] `xtpoisson` command with the option `fe`. Note that the example shown in the manual includes an exposure variable, `service`, which we incorporated in the algorithm.

```
. webuse ships, clear
. qui poisson acc op_75_79 co_65_69 co_70_74 co_75_79, exposure(service) nolog
. keep if e(sample)
(6 observations deleted)
. bys ship: egen sumy=total(acc)
. gen double off=0
. gen double temp=0
. local dif=1
. local ll1=0
. local i=0
. while abs('dif')>epsdouble() {
2. qui poisson acc op_75_79 co_65_69 co_70_74 co_75_79, off(off) nocon
3. local ll2='ll1'
4. local ll1=e(ll)
5. capture drop xb
6. predict double xb, xb
7. qui replace temp=xb-off+log(service), nopromote
8. capture drop sumx
9. bys ship: egen double sumx=total(exp(temp))
10. qui replace off=log(sumy/sumx)+log(service), nopromote
11. local dif='ll2'-'ll1'
12. local i='i'+1
13. }
. di "Total Number of Iterations --> " `i'
Total Number of Iterations --> 103
. poisson acc op_75_79 co_65_69 co_70_74 co_75_79, nocon off(off) irr nolog
Poisson regression                Number of obs   =       34
                                Wald chi2(4)      =      287.40
Log likelihood = -68.280771        Prob > chi2    =       0.0000
```

accident	IRR	Std. Err.	z	P> z	[95% Conf. Interval]
op_75_79	1.468831	.1634895	3.45	0.001	1.180941 1.826903

co_65_69	2.008003	.2086755	6.71	0.000	1.637969	2.46163
co_70_74	2.26693	.2814137	6.59	0.000	1.777342	2.89138
co_75_79	1.573695	.3184952	2.24	0.025	1.058397	2.339875
off	(offset)					

Extending the algorithm to two fixed effects is straightforward. We use the same data set as before but instead of including the dummy variables for year of construction (co_65_69, co_70_74 and co_75_79) we treat the year of construction as a fixed effect, that is, we let the variable yr_con identify a second fixed effect. Now the algorithm is implemented without an exposure variable. For comparability purposes we first run a Poisson regression including dummy variables for both fixed effects.

```
. webuse ships, clear
. local dif=1
. xi: poisson acc op_75_79 i.yr_con i.ship, nolog
i.yr_con      _Iyr_con_1-4      (naturally coded; _Iyr_con_1 omitted)
i.ship        _Iship_1-5        (naturally coded; _Iship_1 omitted)
Poisson regression                                Number of obs   =       34
                                                    LR chi2(8)      =      475.45
                                                    Prob > chi2     =       0.0000
Log likelihood = -118.47588                        Pseudo R2      =       0.6674
```

accident	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
op_75_79	.2928003	.1127466	2.60	0.009	.071821 .5137796
_Iyr_con_2	.5824489	.1480547	3.93	0.000	.2922671 .8726308
_Iyr_con_3	.4627844	.151248	3.06	0.002	.1663437 .7592251
_Iyr_con_4	-.1951267	.2135749	-0.91	0.361	-.6137258 .2234724
_Iship_2	1.79572	.1666196	10.78	0.000	1.469151 2.122288
_Iship_3	-1.252763	.3273268	-3.83	0.000	-1.894312 -.6112142
_Iship_4	-.9044563	.2874597	-3.15	0.002	-1.467867 -.3410457
_Iship_5	-.1462833	.2351762	-0.62	0.534	-.6072202 .3146537
_cons	1.308451	.1972718	6.63	0.000	.9218049 1.695096

```
. keep if e(sample)
(6 observations deleted)
. bys ship: egen sumy1=total(acc)
. bys yr_con: egen sumy2=total(acc)
. gen double off=0
. gen double temp=0
. gen double temp1=0
. gen double temp2=0
. gen double fe1=0
. gen double fe2=0
. local ll1=0
. local i=0
. while abs('dif')>epsdouble() {
2. qui poisson acc op_75_79, nocons offset(off) nolog
3. local ll2='ll1'
4. local ll1=e(ll)
5. capture drop xb
```

```

6. predict double xb, xb
7. qui replace temp1=xb-off+fe2, nopromote
8. capture drop sumx
9. bys ship: egen double sumx=total(exp(temp1))
10. qui replace fe1=log(sumy1/sumx), nopromote
11. qui replace temp2=xb-off+fe1, nopromote
12. capture drop sumx
13. bys yr_con: egen double sumx=total(exp(temp2))
14. qui replace fe2=log(sumy2/sumx), nopromote
15. qui replace off=fe1+fe2
16. local dif='l12'-'l11'
17. local i='i'+1
18. }

. di "Total Number of Iterations --> " `i'
Total Number of Iterations --> 46

. poisson acc op_75_79, nocons offset(off) nolog
Poisson regression
Log likelihood = -118.47588
Number of obs = 34
Wald chi2(1) = 18.60
Prob > chi2 = 0.0000

```

accident	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
op_75_79	.2928003	.0678844	4.31	0.000	.1597493 .4258513
off	(offset)				

Application of the algorithm to Poisson regression was straightforward because we could find a closed form solution for the coefficients associated with the fixed effects. However, in most non-linear regression models the fixed effects do not have a closed-form solution. As shown in Guimarães (2004) models from the multinomial logit family such as logit, multinomial logit and conditional logit can all be estimated using Poisson regression meaning that the above algorithm could be used for these cases. The inexistence of a closed-form solution for the coefficients of the fixed-effects does not invalidate use of the "zigzag" algorithm but it requires the use of a numerical optimization routine to solve for the coefficients of the fixed-effects. This may slow down the algorithm considerably. As an example of this approach we show an application of the "zigzag" algorithm to estimate a negative binomial model with fixed effects.³

```

. webuse ships, clear
. xi: nbreg acc op_75_79 co_65_69 co_70_74 co_75_79 i.ship, nolog
i.ship      _Iship_1-5      (naturally coded; _Iship_1 omitted)
Negative binomial regression
Log likelihood = -88.445258
Number of obs = 34
LR chi2(8) = 41.45
Dispersion = mean
Prob > chi2 = 0.0000
Pseudo R2 = 0.1898

```

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

³The "fixed effects negative binomial model" ([XT] `xtnbreg` with the `fe` option) is not equivalent to a negative binomial model with dummy variables added for fixed effects (see Guimarães (2008)).

op_75_79	.3324104	.328116	1.01	0.311	-.3106852	.975506
co_65_69	.8380919	.4378077	1.91	0.056	-.0199955	1.696179
co_70_74	1.658684	.4850461	3.42	0.001	.708011	2.609357
co_75_79	.8604224	.5955773	1.44	0.149	-.3068876	2.027732
_Iship_2	2.35359	.4701847	5.01	0.000	1.432045	3.275135
_Iship_3	-1.104561	.5214874	-2.12	0.034	-2.126657	-.082464
_Iship_4	-.9606946	.4905212	-1.96	0.050	-1.922098	.0007092
_Iship_5	-.077889	.4780747	-0.16	0.871	-1.014898	.8591201
_cons	.4230202	.5218569	0.81	0.418	-.5998006	1.445841
/lnalpha	-.7372302	.3814595			-1.484877	.0104166
alpha	.4784372	.1825044			.2265302	1.010471

Likelihood-ratio test of alpha=0: chibar2(01) = 60.06 Prob>=chibar2 = 0.000

```
. keep if e(sample)
(6 observations deleted)
. egen id=group(ship)
. qui sum id
. local maxg=r(max)
. local dif=1
. local ll1=0
. local i=0
. gen double off1=0
. gen double off2=0
. while abs('dif')>epsdouble() {
2. qui nbreg acc op_75_79 co_65_69 co_70_74 co_75_79, nocons offset(off1) nol
> og
3. local lna=log(e(alpha))
4. constraint define 1 [lnalpha]_cons='lna'
5. local ll2='ll1'
6. local ll1=e(ll)
7. capture drop xb
8. predict double xb, xb
9. qui replace off2=xb-off1, nopromote
10. forval j=1/'maxg' {
11. qui nbreg acc if id=='j', offset(off2) constraint(1) nolog
12. qui replace off1=_b[_cons] if e(sample), nopromote
13. }
14. local dif='ll2'-'ll1'
15. local i='i'+1
16. }
. di "Total Number of Iterations -->" 'i'
Total Number of Iterations -->122
. nbreg acc op_75_79 co_65_69 co_70_74 co_75_79, offset(off1) nocons nolog
Negative binomial regression          Number of obs   =       34
Dispersion   = mean                  Wald chi2(4)    =       76.36
Log likelihood = -88.445258          Prob > chi2     =       0.0000
```

accident	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
op_75_79	.3324104	.3022232	1.10	0.271	-.2599362 .924757
co_65_69	.838092	.3036241	2.76	0.006	.2429996 1.433184
co_70_74	1.658684	.3187183	5.20	0.000	1.034008 2.283361
co_75_79	.8604225	.4997525	1.72	0.085	-.1190744 1.839919

off1	(offset)			
/lnalpha	-.7372302	.3659848	-1.454547	-.0199132
alpha	.4784373	.1751007	.2335061	.9802837

Likelihood-ratio test of alpha=0: $\chi^2(01) = 88.87$ Prob>= $\chi^2 = 0.000$

Following an approach similar to the one used for the negative binomial model it should be possible to extend the algorithm to other non-linear models. The algorithm should work well with models that have globally concave log-likelihood functions such as the ones discussed here. Finally, we should note that with the "zigzag" algorithm we obtain the correct value of the log-likelihood function. This means that we can very easily implement statistical tests based upon likelihood ratios.

4 Conclusion

In this paper we successfully explored the implementation of the full Gauss-Seidel algorithm to estimate regression models with high-dimensional fixed-effects. The main advantage of this procedure is the ability to estimate a very large number of two-way fixed effects under minimal memory requirements. Generalizing the procedure to non-linear regression models is straightforward, in particular if there is a closed-form solution for the fixed-effect. Extending the procedure to three-way fixed effects (for example: worker, firm, and match-specific effects) is also feasible, and is likely to prove very fruitful.

We do not claim, however, that our procedure is a superior estimation strategy. Quite to the contrary, the "zigzag" algorithm can be very slow and researchers should use more efficient estimation techniques whenever available. We know that the linear regression model with two high-dimensional fixed effects is estimated much more efficiently with the user-written command `felsdvreg` the same way that `xtpoisson` is the better approach to estimate a Poisson model with a single high dimensional fixed-effect. Nevertheless, there are circumstances when the "zigzag" algorithm may prove useful, namely when existing approaches do not work because of hardware (memory) limitations or when there are no other known ways of estimating the model. As we mentioned earlier the estimation strategy outlined in this paper is time consuming but it does have the advantage of imposing minimum memory requirements and being simple to implement. There are many ways to improve the speed of the algorithms discussed above and research is needed to figure out how to improve them. Some obvious things to try is to write more efficient Stata code (possibly Mata), to work on obtaining good starting values and to use some acceleration technique for

the algorithm. This should not be hard to accomplish because the estimates of fixed effects tend to converge monotonically and it should be possible to use the information from the last iterations to adjust the trajectory of the fixed effect estimates and thus obtain faster convergence. Anyway, such enterprise is beyond the scope of this paper.

References

- Abowd, J. M., R. H. Creedy, and F. Kramarz. 2002. Computing Person and Firm Effects Using Linked Longitudinal Employer-Employee Data.
- Abowd, J. M., F. Kramarz, and D. N. Margolis. 1999. High Wage Workers and High Wage Firms. *Econometrica* 67: 251–334.
- Andrews, M., T. Schank, and R. Upward. 2006. Practical Fixed-effects Estimation Methods for the Three-way Error-components Model. *Stata Journal* 6: 461–481.
- Carneiro, A., P. Guimarães, and P. Portugal. 2008. Real Wages and the Business Cycle: Accounting for Worker and Firm Heterogeneity. Unpublished manuscript.
- Cornelissen, T. 2008. The Stata Command `felsdsvreg` to Fit a Linear Model with Two High-Dimensional Fixed Effects. *Stata Journal* 8: 170–189.
- Greene, W. 2004. The Behaviour of the Maximum Likelihood Estimator of Limited Dependent Variable Models in the Presence of Fixed Effects. *Econometrics Journal* 7: 98–119.
- Guimarães, P. 2004. Understanding the Multinomial-Poisson Transformation. *Stata Journal* 4: 265–273.
- . 2008. The Fixed Effects Negative Binomial Model Revisited. *Economics Letters* 99: 63–66.
- Smyth, G. 1996. Partitioned Algorithms for Maximum Likelihood and other non-linear Estimation. *Statistics and Computing* 6: 201–216.