Università degli Studi di Napoli Federico II
Scuola delle Scienze Umane e Sociali
Quaderni
12

# GRETL 2019
# Proceedings of the International Conference on the GNU Regression, Econometrics and Time Series Library

Naples, 13-14 June 2019

Editors

Francesca Di Iorio and Riccardo Lucchetti


Federico II University Press

fedOA

Università degli Studi di Napoli Federico II
Scuola delle Scienze Umane e Sociali
Quaderni
12

GRETL 2019

# Proceedings of the International Conference on the Gnu Regression, Econometrics and Time Series Library

Naples, 13-14 June 2019

Editors

Francesca Di Iorio and Riccardo Lucchetti

Federico II University Press

fedOA Press

GRETL 2019 : Proceedings of the International Conference on the Gnu Regression, Econometrics and Time Series Library : Naples, 13-14 June 2019 / editors Francesca Di Iorio, Riccardo Lucchetti. – Napoli : FedOAPress, 2019. – (Scuola di Scienze Umane e Sociali. Quaderni ; 12).

Accesso alla versione elettronica:
http://www.fedoabooks.unina.it

ISBN: 978-88-6887-057-7
DOI: 10.6093/978-88-6887-057-7
ISSN Collana: 2499-4774



Università degli Studi di Napoli Federico II
Centro di Ateneo per le Biblioteche “Roberto Pettorino”
Piazza Bellini 59-60
80138 Napoli, Italy
http://www.fedoapress.unina.it/

Published in Italy

# Contents

# Foreword

This volume contains the peer-reviewed contributions presented at the 6th International GRETL Conference - GRETL 2019 - held at the Department of Political Sciences of the University of Naples Federico II, (13-14 June 2019).

The Gretl Conference has come a long way. Since the first GRETL Conference, held in Bilbao in 2009, the project has witnessed tremendous development in several directions. The user base has increased, the developer pool has become larger, the program features have multiplied. The present proceedings book is a testimony of this process.

The Conference featured the distinguished keynote lecture by Jörg Breitung (University of Köln) who significantly contributed to making the Conference successful with his inspiring presentations.

I would also like to take this opportunity to express my gratitude to the members of the Scientific Committee: Stephen Pollock and Claudia Pigini for their helpful support. I am also very grateful to the members of the Organizing Committee: Francesca Di Iorio (chair), Stefania Capecchi, Carmela Cappelli, Giuliana Perretti, Rosaria Simone that contributed to the success of GRETL 2019 and worked actively for its organization.

Finally, I wish to acknowledge the scientific sponsorship of the Italian Econometric Association (SIdE) and the financial support of the Department of Political Sciences and the University of Naples Federico II.

*Riccardo Lucchetti*
*Chair of the Scientific Committee*

# Replication of the UK inflation model by Hendry (2001) using BACE approach

Marcin Błażejowski*, PawełKufel**, Jacek Kwiatkowski***

*Abstract:* In this paper, we revisit the well-known UK inflation model by Hendry (*Journal of Applied Econometrics* 2001, 16:255-275. doi: 10.1002/jae.615). We replicate the results in a narrow sense using the gretl and PcGive programs. In a wide sense, we extend the study of model uncertainty using the Bayesian averaging of classical estimates (BACE) approach to compare model reduction strategies. Allowing for the investigation of other specifications, we confirm the same set of significant determinants but find that Hendrys' model is not the most probable.

*Keywords:* BACE, gretl, Model uncertainty, Reduction strategy

## *1. Introduction*

This paper concerns a replication of a model of UK inflation, 1875-1991, by Hendry (2001) based on data provided by JAE services at (`http://qed.econ.queensu.ca/jae/2001-v16.3/hendry`). To replicate Hendrys' procedure for modeling inflation in the UK in a narrow sense, we used the gretl and PcGive/Autometrics. Our extension, in a wide sense, of Hendrys' work employed the Bayesian averaging of classical estimates (BACE) approach proposed by Sala-i-Martin *et al.* (2004) to compare model reduction strategies and the variable selection procedure.

When we consider the large number of variables, it is difficult to decide which model is the most appropriate for analyzing the dependencies, i.e., to find the optimal set of variables in terms of goodness of fit measures. Using BACE, we can obtain the most probable set of determinants along with posterior parameter estimates based on the whole model space instead of making decisions based only on a single model. This approach is an alternative to the

*WSB University in Torun, Poland, marcin.blazejowski@wsb.torun.pl
**WSB University in Torun, Poland, pawel.kufel@wsb.torun.pl
***Nicolaus Copernicus University in Torun, Poland, jkwiat@uni.torun.pl

earlier and familiar Bayesian model averaging (BMA) (see Fernandez *et al.* 2001; Ley *et al.* 2012), from which it differs, first of all, by having less restrictive prior assumptions of parameters. Sala-i-Martin *et al.* (2004) showed that the BMA approach may be understood as limiting case of Bayesian analysis in the situation where prior information is "dominated" by the data. The parameter estimates are averaged across all possible combinations of models obtained by means of OLS. In our case, the BACE analysis was performed in the BACE 1.1 package (see Błażejowski *et al.* 2018).

*2. Data*

In our paper and replication files, we used the same data definitions as in Hendry (2001) with the following exceptions:

1. *Profit markup* $(\pi_t^*)$ was taken directly from the `jaedfh4.dat` file (part of the `dfhdata.zip` archive); this variable exists as "`pistarn`" in the JAE archive.

2. *Short-long spread* $(R_{s,t} - R_{l,t} + 0.006)$ was named $S_t$, similar to Clements *at al* (2008).

3. *Excess demand* $\left(y_t^d\right)$ was taken directly from the `jaedfhm.dat` file (part of the `dfhdata.zip` archive); this variable exists as "`gdpd`" in the JAE archive.

4. *The real exchange rate* was defined as $e_{r,t} = p_t - p_{\pounds,t} - 0.52$. We found an inaccuracy in the paper by Hendry (2001) and data definitions in the JAE archive. The calculation of $e_{r,t} = p_t - p_{\pounds,t} + 0.52$ (equation (3) in Hendry (2001)) is misleading with the form of calculating $e_{r,t}$ in the formula for "`pistarn`" (`readme.h.txt` file) and refers to subtracting (not adding) the intercept value (0.52).

5. According to formulas in the JAE archive (`readme.h.txt` file), the variable *Unit labor costs in constant prices* was defined as $c_t^* = c_t - p_t + 0.006 \times (trend - 69.5) + 2.37$. This exception was due to an inconsistency between the JAE archive and Hendrys' paper, where it was defined as $c_t^* = (c_t - p_t)^*$.

## 3. Research scenario

To replicate the Hendry (2001) results in a narrow sense, we proceed as follows. In the case of the initial model for all 52 variables, we received identical output to that in the original model (`Model: GUM52`; residual standard deviation $\hat{\sigma} = 1.21\%$, Schwartz Criterion $SC = -7.3$). After excluding indicators from the initial model, we also received exact results (`Model: GUMnoIndicators`; $\hat{\sigma} = 2.5\%$, $SC = -6.63$). In the next step, we added dummy variables $I_b, I_l, I_m$ concerning outliers in particular years to (`Model: GUMnoIndicators`), and we obtained the same results as in the paper (`Model: GUMfirst Reduction`; $\hat{\sigma} = 1.16\%$, $SC = -8.08$). In the next step, the dummy variables $I_b, I_l, I_m$ were substituted by one overall index, $I_d$, and once again, we obtained the same results (`Model: GUMsecondReduction`; $\hat{\sigma} = 1.15\%$, $SC = -8.16$). Finally, we expressed the general model in terms of $\pi^*_{t-1}$ with indicators restricted to $I_d$ (`Model: GUMfinal`; $\hat{\sigma} = 1.15\%$, $SC = -8.33$). At this point, we had the following specification:

$$\begin{aligned}\Delta p_t = f(&\Delta p_{t-1}, y^d_{t-1}, m^d_{t-1}, n^d_{t-1}, U^d_{t-1}, S_{t-1}, R_{l,t-1}, \Delta p_{e,t}, \Delta p_{e,t-1}, \Delta U_{r,t-1},\\ &\Delta w_{t-1}, \Delta c_{t-1}, \Delta m_{t-1}, \Delta n_{t-1}, \Delta R_{s,t-1}, \Delta R_{l,t-1}, \Delta p_{o,t-1}, I_{d,t}, \pi^*_{t-1}; \varepsilon_t)\end{aligned} \tag{1}$$

After the reduction of model (1) at a 1% significance level, we obtained model (6) in Hendry (2001) and (`Model: FinalModel`; $\hat{\sigma} = 1.14\%$, $SC = -8.66$) in our notation. Detailed results are available in the Table 1. According to results in Table 1, we found minor differences in the coefficient estimates for four variables—$y^d_{t-1}$, $\Delta m_{t-1}$, $\Delta R_{s,t-1}$, and $\Delta p_{t-1}$—and two differences in standard errors for $S_{t-1}$ and $\Delta p_{t-1}$. The remaining coefficients and the model statistics were identical.

In his paper, Hendry used the PcGets automatic model selection procedure with a 1% significance level for the model (1) to check the correctness of the simplification. We repeated this automatic model selection procedure using Autometrics for model (1), and we obtained exactly the same estimates as in `Model: GUMfinal` in gretl (i.e., with slight differences compared to model (6) in Hendrys' paper). We suppose that these differences are due to

*Table 1. Comparison of Hendrys' and the replication results*

| | Hendry (2001) | Replication |
|---|---|---|
| $y^d_{t-1}$ | 0.180<br>(0.032) | 0.184<br>(0.032) |
| $\Delta m_{t-1}$ | 0.187<br>(0.028) | 0.182<br>(0.028) |
| $S_{t-1}$ | −0.834<br>(0.088) | −0.834<br>(0.087) |
| $\Delta R_{s,t-1}$ | 0.618<br>(0.106) | 0.619<br>(0.106) |
| $\pi^*_{t-1}$ | −0.186<br>(0.024) | −0.186<br>(0.024) |
| $\Delta p_{e,t}$ | 0.265<br>(0.025) | 0.265<br>(0.025) |
| $I_{d,t}$ | 0.038<br>(0.002) | 0.038<br>(0.002) |
| $\Delta p_{o,t-1}$ | 0.041<br>(0.010) | 0.041<br>(0.010) |
| $\Delta p_{t-1}$ | 0.267<br>(0.027) | 0.268<br>(0.026) |
| $R^2$ | 0.975 | 0.975 |
| $\hat{\sigma}$ | 1.14% | 1.14% |
| *SC* | −8.66 | −8.66 |

diverse computer architectures (64 bit and 32 bit).

### *4. BACE results*

To verify the correctness of Hendrys' variable selection strategy, we used the BACE approach (replication in a wide sense). This procedure enables searching the whole model space and selecting the most probable regressions. The BACE analysis was performed for the set of $k = 20$ variables (including the intercept) defined in `Model: GUMfinal` (model (1)), and so the total number of possible models was $2^k = 1,048,576$. The BACE approach enables calculations of the averages of the posterior means and standard deviations of parameters as well as posterior inclusion probabilities (PIP). The posterior inclusion probability is the probability that, conditional on the data but unconditional with respect to the model space, the independent variable is relevant

in explaining $\Delta p_t$ (see Doppelhofer *at al* 2009; Koop *at al* 2007). PIP is calculated as the frequency of appearance of a given variable in all considered models. The BACE results, obtained after 1,000,000 Monte Carlo iterations, are presented in table 2.

*Table 2. Posterior inclusion probabilities and posterior estimates of regression coefficients obtained by BACE*

| | PIP | Avg. Mean | Avg. Std. Dev. |
|---|---|---|---|
| $\pi^*_{t-1}$ | 1.000000 | −0.186844 | 0.025828 |
| $I_{d,t}$ | 1.000000 | 0.037903 | 0.001573 |
| $\Delta p_{e,t}$ | 1.000000 | 0.264119 | 0.025146 |
| $S_{t-1}$ | 1.000000 | −0.856166 | 0.090581 |
| $\Delta p_{t-1}$ | 1.000000 | 0.279046 | 0.033585 |
| $y^d_{t-1}$ | 0.999996 | 0.193686 | 0.036891 |
| $\Delta R_{s,t-1}$ | 0.999949 | 0.609606 | 0.114351 |
| $\Delta m_{t-1}$ | 0.999936 | 0.173201 | 0.029831 |
| $\Delta p_{o,t-1}$ | 0.987283 | 0.038862 | 0.011714 |
| $U^d_{t-1}$ | 0.610013 | −0.041815 | 0.040875 |
| $n^d_{t-1}$ | 0.194672 | 0.000631 | 0.001692 |
| $R_{l,t-1}$ | 0.151855 | 0.006635 | 0.022907 |
| $\Delta R_{l,t-1}$ | 0.126007 | 0.026201 | 0.111685 |
| $\Delta p_{e,t-1}$ | 0.105244 | 0.002085 | 0.011372 |
| $m^d_{t-1}$ | 0.104247 | −0.000513 | 0.004491 |
| $\Delta U_{r,t-1}$ | 0.097311 | −0.002368 | 0.024480 |
| *const* | 0.095136 | 0.000021 | 0.000643 |
| $\Delta c_{t-1}$ | 0.090481 | −0.000170 | 0.010168 |
| $\Delta n_{t-1}$ | 0.089092 | 0.000461 | 0.004619 |
| $\Delta w_{t-1}$ | 0.085100 | −0.000306 | 0.012303 |

According to the results in table 2, the set of variables used in the BACE analysis can be divided into 3 groups: highly probable determinants ($\pi^*_{t-1}$, $I_{d,t}$, $\Delta p_{e,t}$, $S_{t-1}$, $\Delta p_{t-1}$, $y^d_{t-1}$, $\Delta R_{s,t-1}$, $\Delta m_{t-1}$, $\Delta p_{o,t-1}$) with $PIP \geq 0.987$, medium probable ($U^d_{t-1}$) with $PIP = 0.61$ and lowly probable ($n^d_{t-1}$, $R_{l,t-1}$, $\Delta R_{l,t-1}$, $\Delta p_{e,t-1}$, $m^d_{t-1}$, $\Delta U_{r,t-1}$, $const$, $\Delta c_{t-1}$, $\Delta n_{t-1}$, $\Delta w_{t-1}$) with $PIP \leq 0.195$. Our results were consistent with Hendrys' paper because the highly

probable determinants according to the BACE approach were the same as in model (6). This result confirms that the "conservative" model reduction strategy was relevant in the case of modeling UK inflation. Moreover, our results confirmed the inconclusive inference on the relevance of $U^d_{t-1}$, i.e., the excess labor demand (with $PIP = 0.61$, it could not be classified as a highly probable determinant).

In addition to the posterior characteristics presented in table 2, the BACE approach allows models to be ranked according to their posterior probabilities. Table 3 presents the coefficient estimates and model statistics for the top 10 models. The total probability of these models was 50.8%. The most probable model $M_1$ had the posterior probability 0.2. The second probable model $M_2$, with probability 0.095, was model (6) in Hendry (2001) and `FinalModel` in our notation. In addition, $M_1$ fit the data better then $M_2$ based on the following statistics: $\overline{R}^2_{M_1} > \overline{R}^2_{M_2}$, $\hat{\sigma}_{M_1} < \hat{\sigma}_{M_2}$ and $SC_{M_1} < SC_{M_2}$.

These two best models differ only by the variable $U^d_{t-1}$, i.e., the excess labor demand. Although the posterior probability of the highest ranked model $M_1$ was more than twice as large as that for the second model $M_2$, an inference based only on $M_1$ omits 80% of the total information contained in the entire model space. As a consequence, the average coefficient estimates presented in table 2 were different than coefficient estimates for the `FinalModel` in table 1. The greatest differences were noticed for the following variables: $\Delta p_{t-1}$, $y^d_{t-1}$, $\Delta m_{t-1}$ and $\Delta p_{o,t-1}$.

Taking our results into consideration, we confirmed the simplification problem about the relevance of $U^d_{t-1}$, as in Hendrys' paper. If we use all available information contained in the whole model space, $U^d_{t-1}$ will be classified as a medium determinant variable with $PIP = 0.61$. The "conservative" model reduction strategy dropped $U^d_{t-1}$, leading to $M_2$, while the "liberal" strategy leads to $M_1$, which includes $U^d_{t-1}$. Furthermore, setting the target size to "medium" (2.5% significance level) in Autometrics also leads to $M_1$. The posterior probabilities of the other models $P(M_3), \ldots, P(M_{10})$ were less than 0.038. These models differ from the two best models only by the least probable variables, and they did not contribute substantial information in this case.

*Table 3. Coefficient estimates and model statistics for top 6 models*

| Model | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ |
|---|---|---|---|---|---|---|
| $P(M_j)$ | 0.200 | 0.095 | 0.037 | 0.035 | 0.026 | 0.024 |
| $\pi^*_{t-1}$ | $-0.187$ (***) | $-0.186$ (***) | $-0.194$ (***) | $-0.196$ (***) | $-0.168$ (***) | $-0.177$ (***) |
| $I_{d,t}$ | 0.038 (***) | 0.038 (***) | 0.038 (***) | 0.038 (***) | 0.037 (***) | 0.038 (***) |
| $\Delta p_{e,t}$ | 0.265 (***) | 0.265 (***) | 0.263 (***) | 0.262 (***) | 0.262 (***) | 0.263 (***) |
| $S_{t-1}$ | $-0.857$ (***) | $-0.834$ (***) | $-0.882$ (***) | $-0.872$ (***) | $-0.833$ (***) | $-0.854$ (***) |
| $\Delta p_{t-1}$ | 0.287 (***) | 0.268 (***) | 0.288 (***) | 0.272 (***) | 0.264 (***) | 0.283 (***) |
| $y^d_{t-1}$ | 0.188 (***) | 0.184 (***) | 0.216 (***) | 0.223 (***) | 0.191 (***) | 0.192 (***) |
| $\Delta R_{s,t-1}$ | 0.625 (***) | 0.619 (***) | 0.572 (***) | 0.547 (***) | 0.635 (***) | 0.633 (***) |
| $\Delta m_{t-1}$ | 0.178 (***) | 0.182 (***) | 0.167 (***) | 0.167 (***) | 0.162 (***) | 0.168 (***) |
| $\Delta p_{o,t-1}$ | 0.037 (***) | 0.041 (***) | 0.041 (***) | 0.045 (***) | 0.042 (***) | 0.038 (***) |
| $U^d_{t-1}$ | $-0.069$ (**) | | $-0.062$ (**) | | | $-0.062$ (**) |
| $n^d_{t-1}$ | | | 0.003 (–) | 0.004 (–) | | |
| $R_{l,t-1}$ | | | | | 0.051 (–) | 0.028 (–) |
| $R^2$ | 0.976 | 0.975 | 0.977 | 0.975 | 0.976 | 0.977 |
| $\overline{R}^2$ | 0.974 | 0.973 | 0.975 | 0.974 | 0.974 | 0.974 |
| $\hat{\sigma}$ | 1.11% | 1.14% | 1.11% | 1.13% | 1.11% | 1.11% |
| *SC* | $-8.67$ | $-8.66$ | $-8.65$ | $-8.64$ | $-8.64$ | $-8.64$ |

(***) significance at 1%, (**) significance at 5%, (–) insignificance at 10%, $\overline{R}^2$ stands for the adjusted $R^2$, and $P(M_j)$ denotes the posterior model probability of model $M_j$.

## *5. Conclusions*

Replication of Hendrys' model for UK inflation in a narrow sense was performed in two programs (gretl and PcGive/Autometrics) and brought exactly the same results, although they were slightly different than the original. In the replication in a wide sense, we used BACE as an automatic model reduction strategy. Taking into account the whole model space, we obtained the same set of determinants as in Hendrys' paper, although his `FinalModel` was

the second one in the ranking, and it was more than two times less probable then the most likely model containing the additional variable $U_{t-1}^{d}$. Hendrys' model omitted over 90% of the information contained in all possible models, which leads to different coefficients estimates. Referring directly to the findings in Hendry (2001), inference based on just one model may lead to slightly different conclusions than inference based on the whole model space.

*Acknowledgements:* We are grateful to Riccardo Lucchetti for helpful comments.

## *References*

Błażejowski M., Kwiatkowski J. (2018) Bayesian Averaging of Classical Estimates (BACE) for gretl, *Gretl working papers* 6, Università Politecnica delle Marche, Dipartimento di Scienze Economiche e Sociali, Ancona (Italy).

Clements M.P., Hendry D.F. (2008) Forecasting Annual UK Inflation Using an Econometric Model over 1875–1991, in: Rapach D.E., Wohar M. E. (eds) *Forecasting in the Presence of Structural Breaks and Model Uncertainty*, Emerald Group Publishing Limited, Chapter 1, 3-39.

Doornik J.A., Hendry D.F. (2013) *Empirical Econometric Modelling - PcGive 14*, Timberlake Consultants Ltd, London.

Doppelhofer G., Weeks M. (2009) Jointness of growth determinants, *Journal of Applied Econometrics*, 24, 209-244.

Fernández C., Ley E., Steel M.F.J. (2001) Benchmark priors for bayesian model averaging, *Journal of Econometrics*, 100(2), 381-427.

Hendry D.F. (2001) Modelling UK Inflation, 1875–1991, *Journal of Applied Econometrics* , 16, 255-275.

Hendry D.F., Krolzig H.M. (2001) *Automatic Econometric Model Selection Using PcGets*, Timberlake Consultants Press, London.

Koop G., Poirier D.J., Tobias J.L. (2007) *Bayesian Econometric Methods*, Cambridge University Press, New York.

Ley E., Steel M.F. (2012) Mixtures of g-priors for Bayesian model averaging with economic applications, *Journal of Econometrics*, 171, 251-266.

Sala-i-Martin X., Doppelhofer G., Miller R.I. (2004) Determinants of Long-Term Growth: A Bayesian Averaging of Classical Estimates (BACE) Approach, *The American Economic Review*, 94(4), 813-835.

# Model selection for modeling the demand for narrow money in transitional economies

Marcin Błażejowski*, PawełKufel**, Jacek Kwiatkowski***,
Tadeusz Kufel****, Magdalena Osińska*****

*Abstract:* The aim of this study was to verify the stability of monetary systems in selected emerging economies. The United Kingdom's economy was used as a benchmark. The Baumol-Tobin and Friedman monetary models were used as the theoretical basis for the empirical error-correction models. A Bayesian averaging of classical estimates (BACE) approach was used to incorporate model uncertainty and select the best model. The results show that the monetary systems in 6 of the 11 economies were stable in the long run and that a set of factors changed in the short run. The robustness of the model selection based on the BACE procedure was strongly confirmed.

*Keywords:* Model uncertainty, BACE, Jointness, Robust variables selection, gretl.

## *1. Introduction*

This paper was motivated by the question of whether Milton Friedman and Anna Schwartz's model of the demand for money (Friedman *et al.* 1982) is appropriate for contemporary transition economies. This idea comes directly from the works of Hendry *et al.* (1991), who analyzed the specifications of several money demand models for the United Kingdom (UK) and the USA. The aim of this paper is to consider both economic and econometric issues. The first aspect, closely related to motivation, focuses on the question of whether economies in transition are affected by the set of money demand factors that was proposed by Friedman *et al.* (1982) and which of them are robust despite of volatile surroundings. The second aspect, which is related to using

*WSB University in Torun, Poland, marcin.blazejowski@wsb.torun.pl
**WSB University in Torun, Poland, pawel.kufel@wsb.torun.pl
***Nicolaus Copernicus University in Torun, Poland, jkwiat@uni.torun.pl
****Nicolaus Copernicus University in Torun, Poland, tadeusz.kufel@umk.pl
****Nicolaus Copernicus University in Torun, Poland, emo@umk.pl

an econometric methodology, is to evaluate both the probability of each factor included in the model and the probability of the entire model specification.

*2. Methodological background*

Building a model with a large number of explanatory variables can potentially lead to decision-making problems. One potential solution to overcome this issue is using the BACE approach, which enables the measurement of the importance of particular potential determinants. This method was suggested by Sala *et al.* (2004) and is is a rough approximation of the earlier Bayesian model averaging (BMA) technique presented by Fernandez *et al.* (2001). BACE is performed with noninformative priors. As a consequence of the estimation method, BACE uses the Schwarz model selection criterion, so the posterior weights of the estimated models are proportional to the natural logarithm of the likelihood function corrected for degrees of freedom. Based on the BACE results, we can also calculate one more useful characteristic: a jointness measure. According to Ley *et al.* (2007), jointness is the posterior odds ratio of the models including both $x_i$ and $x_j$ explanatory variables versus that of the models that include them separately.

*3. Model specification and data characteristics*

In this paper, we use two different model specifications in three variants each for selected transition economies and the UK, which plays the role of the benchmark. Among the transition economies, we consider two groups:

- new member states of the European Union coming from Central and Eastern Europe, such as the Czech Republic (CZE), Hungary (HUN) and Poland (POL);
- economies struggling to develop very fast: Brazil (BRA), India (IND), Indonesia (IDN), Mexico (MEX), Russia (RUS), Turkey (TUR) and South Africa (ZAF).

Money demand is defined here as the demand for narrow money and is measured as aggregate M1. The rationale for the selection of this aggregate comes

from the fact that it contains the same monetary categories across the entire sample for all economies being investigated. According to Hendry (1995), the narrow money category is appropriate when the stability in the long run is checked. The sample covers the years 1995-2017, using quarterly observations. Using this time frame ensures the comparability of both the data and the results. From 2008-2009, all the economies experienced deep economic recession; a dummy variable for this period was also employed. The following macroeconomic time series were collected from the OECD.Stat database:

- $GDP_t$ – expenditure approach;
- $P_t$ – GDP price deflator;
- $M_t$ – narrow money aggregate;
- $IMP_t$ – imports of goods and services;
- $R_t$ – short-term interest rate;
- $imR_t$ – immediate interest rates.

Based on the original time series, the following variables were calculated. Real TFE according to formula: $Y_t = (GDP_t + IMP_t)/P_t$, which is equivalent to TFE, as defined by Hendry *at al* (1991). Then, the following interest rate was defined as $dR_t = R_t - imR_t$, which is the premium of holding money in three-month deposits. This variable corresponds to Friedman's differential yield on money (Friedman *at al* 1982). Additionally, for the period of low short-term interest rates, the following dummy was used:

$$R08_t = \begin{cases} R_t, & \text{from 2008Q2 to 2013Q4,} \\ 0, & \text{in other periods.} \end{cases}$$

We also defined dummy variables for regional crises. Following Hendry *et al.* (1991), we allow two alternative assumptions regarding parameter $\delta$. If $\delta = 0.5$, the Baumol-Tobin square-root model for the transaction demand for cash is applied (Baumol *et al.* 1962; Tobin 1956) and the case when $\delta = 1.0$ corresponds to Friedman's quantity theory Friedman ). Hendry *et al.* (1991)

found that for the UK, Friedman's model should be applied, while in the case of the United States, the Baumol-Tobin model should be used. Taking the above into account, an error-correction term can be defined as:

$$ECM_t = \begin{cases} m_t - p_t - \frac{1}{2}y_t, & \text{for Baumol-Tobin model (specification type 1),} \\ m_t - p_t - y_t, & \text{for Friedman model (specification type 2).} \end{cases} \tag{1}$$

The interest rates may be included in different ways. In our research, *IR* is a set of 4 different combinations of interest rate measures. It takes one of the following specification forms:

$$IR_t = \begin{cases} \sum_{s=0}^{4} \gamma_{1,s} R_{t-s} + \sum_{s=0}^{4} \gamma_{2,s} imR_{t-s} + \sum_{s=0}^{4} \gamma_{3,s} \Delta R08_{t-s}, & \text{'a'}, \\ \sum_{s=0}^{4} \gamma_{1,s} \Delta R_{t-s} + \sum_{s=0}^{4} \gamma_{2,s} \Delta imR_{t-s} + \sum_{s=0}^{4} \gamma_{3,s} \Delta R08_{t-s}, & \text{'b'}, \\ \sum_{s=0}^{4} \gamma_{1,s} dR_{t-s} + \sum_{s=0}^{4} \gamma_{2,s} \Delta R08_{t-s}, & \text{'c'}, \\ \sum_{s=0}^{4} \gamma_{1,s} \Delta dR_{t-s} + \sum_{s=0}^{4} \gamma_{2,s} \Delta R08_{t-s}, & \text{'d'}. \end{cases} \tag{2}$$

These four specifications are in line with those in Friedman *et al.* (1982) and Hendry *et al.* (1991). In our research, the dummies for 3 months of interest rates were used beginning with the second quarter of 2008 until the last quarter of 2013, which corresponds to a low interest rate period. Taking into account the relations 1 and 2 we have 8 possible forms of a general unrestricted model for each analyzed country. We employed the BACE 1.1 package (Błażejowski *et al.* 2018).

## *4. Empirical results*

In this section, the empirical results obtained using the research strategy described in sections 2 and 3 are presented and discussed. Because the error-correction term is included in the model and we have 8 possible specifications for each country, we have defined the minimum conditions for *ECM* variable that must be met by the posterior results for a given specification to be taken into account in the next steps: the sign of the mean value of coefficient estimate must be negative and, at the same time, the minimum value of *PIP*, which is interpreted as uncertainty measures, must exceed 2/3 (0.66).
The results show that for Brazil, Russia, Indonesia and India we cannot find

a specification that meets the minimum conditions for the *ECM* variable defined above, while Hungary, the only one *ECM* variable with $PIP > 0.66$ has a positive sign. This apparent instability results from massive financial problems experienced in 2008 and the immense rescue package that Hungary received from several institutions: 6.5 billion euros from the European Union, 12.5 billion euros from the International Monetary Fund and 1 billion euros from the World Bank (Csáki 2013). On the other hand, for 6 countries, more than 1 specification satisfies these conditions. For those countries, the ranked total probability of the models was used as the criterion for selecting the best specification. We can state that for the analyzed economies in transition (without the UK) in 4 cases, specification type 1 (Baumol-Tobin's model) is preferred, and in 6 cases, specification type 2 (Friedman's model) outperforms. Moreover, specification type 'd' is selected for 9 cases, with the exception of only Russia, where type 'b' is selected. This result means that the dynamics of the premium of holding money for 3 months ($\Delta dR_t$) is an appropriate measure of the interest rate for modeling the demand for narrow money in the analyzed economies. The type 'a' and 'c' specifications seem to be inadequate. It is worth mentioning that the results for modeling the money demand for the UK are in line with the results presented in Hendry *et al.* (1991). Although numerous external and internal shocks in the UK economy have occurred since Hendry's model was developed, the proposed model selection procedure confirms that it is still valid: the most likely specification is Friedman's model incorporating a 'spread or net opportunity cost' of holding money (our specification 2d). This result can be interpreted as a confirmation of the accuracy of our approach because the UK served as our benchmark.
The output can be summarized as follows. First, we noticed that two alternative model specifications denoted as 1 and 2 were supported by the data. The Baumol-Tobin model was confirmed for the Czech Republic, Poland and Turkey. On the other hand, Friedman's quantity theory was successfully implemented for Mexico, Indonesia, ZAF and the UK. This result means that in the long run, the difference in the proportionality of the demand for money is distinct at both the theoretical and empirical levels.

The main finding is that in such countries as Brazil, Russia, Indonesia and India, we cannot confirm the stability of the monetary systems measured

in terms of aggregate M1 because the values of *ECM* are usually negative but rather unlikely (although the *PIP* for the *ECM* variable for Russia equals 0.641 and for Indonesia, equals 0.647). For Hungary, the sign of the *ECM* is positive, which excludes it from consideration. In the remaining countries, stability was fully confirmed. The highest level of stability observed across all specifications (apart from 1a and 2a) is for South Africa, where the probability exceeds 0.88. For the Czech Republic and the UK, six specifications are confirmed. For Poland and Turkey, four specifications were valid: were 1c, 1d, 2c, and 2d. For Mexico, specifications 1d and 2d confirmed the stability of the monetary system. When the long-run specifications are compared, one can notice that specification 2 (Friedman model) outperformed specification 1 (Baumol-Tobin model).

In summary, for five cases, we cannot fully confirm the stability of the monetary system measured in terms of aggregate M1, although, the valued for Russia and Indonesia are close to the limit value of *PIP*. The monetary systems represented by narrow money are stable in the transitional economies: the Czech Republic, Poland, Mexico, Turkey and South Africa. Stability is also confirmed for the UK. A further analysis is conducted on the concomitance of the factors in different specifications (jointness). An analysis of the jointness results leads to the following conclusions:

- There are no complementary pairs for India.
- For Indonesia, $\Delta p_{t-s}$ coexists with the interest rate and $\Delta mp_{t-s}$; in Poland, $\Delta p_{t-s}$ with $\Delta mp_{t-s}$; and in Brazil, the 2008 financial crisis coexists with $\Delta y_{t-s}$.
- For Turkey, the dummy $Cr\_Tur01_t$ coexists with $\Delta y_{t-s}$, $\Delta p_{t-s}$, $R_{t-s}$ and $imR_{t-s}$.
- For the UK, the following variables have individual impacts: $ECM_{t-1}$ with $\Delta p_{t-s}$, $Cr\_Fin_t$ and $Cr\_Asia97_t$. On the other hand, in each specification, 3 variables occur together: $\Delta p_{t-s}$, $Cr\_Fin_t$, $Cr\_Asia97_t$. It can be easily seen that $R08_{t-s}$, $R_{t-s}$, $imR_{t-s}$, and $\Delta y_{t-s}$ remained unrelated with the other variables.

- The variable $\Delta p_{t-s}$ always pairs with other variables, with the exception of Brazil.
- The most pairs can be observed for the UK, Mexico and Turkey.
- $ECM_{t-1}$ coexists in pairs in the Czech Republic, South Africa and UK.

In the cases where complementary pairs of variables are detected, the joint explanatory power of such pairs is greater than if they are considered individually. This type of analysis supports the interpretation of the results of the short-run model. For example, for Russia $\Delta p_{t-s}$, $\Delta mp_{t-s}$, and $\Delta R_{t-s}$ appear in specification 1b, which is most likely. This result means that that three factors are responsible for the short-run dynamics of the demand for narrow money in Russia. The results are similar for the Czech Republic, Poland, Indonesia, Mexico, Turkey, South Africa and the UK. The results for India and Brazil show that the relations between the variables are dubious, which confirms the results from the BACE. For Hungary, this occurs for only one pair of complementary variables, but this result is not stable in the long run, when one of the complementary variables is a dummy variable and the results show occasional relationships.

*5. Robustness analysis*

To confirm the empirical findings, we performed a robustness check. Since the analysis addresses variable and model selection issues, we decided to apply Ockham's razor rule. In our analysis, the prior average model size was set to $\mathrm{E}(\Xi) = k/2$ (where $k$ is the number of variables in a given GUM). This means that we do not prefer any specification, so all possible models are equally probable. For the BACE approach, the use of Ockham's razor rule is very simple, and the only change we have to make is to set the prior average model size to a reasonably small value to penalize the large models (in terms of the number of variables). If the resulting average size of the posterior model is similar for both normal and small values of $\mathrm{E}(\Xi)$, the empirical results are robust.

In all cases, for $\mathrm{E}(\Xi) = k/2$ (uniform prior on model space), the values of the average size of the posterior model are smaller than the corresponding

values of the average size of the previous model. This result means that the most parsimonious specifications are preferred, and the BACE results are in line with Ockham's razor rule. Moreover, the differences between the values of the average size of the posterior model for different $E(\Xi)$ are small. The maximum difference is equal to to 2.76, but the median difference is equal to 0.92 and the mode difference is 0.68. When the values of Pearson's correlation coefficients between the corresponding values of *PIP* in normal and small values of $E(\Xi)$ are compared, they are very close to 1 in all cases. The same conclusions are true for the means of the parameters' estimates in the same models. This means that the empirical results are strongly robust.

*Acknowledgements:* We gratefully acknowledge financial support from the National Science Center in Poland (contract/grant number UMO-2016/21/B/HS4/01970).

## *References*

Baumol W.J. (1952) The Transactions Demand for Cash: An Inventory Theoretic Approach, *The Quarterly Journal of Economics*, 66(4), 545-556.

Błażejowski M., Kwiatkowski J. (2018) Bayesian Averaging of Classical Estimates (BACE) for gretl, *Gretl working papers* 6, Università Politecnica delle Marche, Dipartimento di Scienze Economiche e Sociali, Ancona, Italy.

Csáki G. (2013) IMF loans to Hungary, 1996-2008, *Public Finance Quarterly*, 58(1), 95.

Fernández C., Ley E., Steel M.F.J. (2001) Benchmark priors for bayesian model averaging, *Journal of Econometrics*, 100(2), 381-427.

Friedman M. (1956) *Studies in the quantity theory of money*, University of Chicago Press, Chicago.

Friedman M., Schwartz A.J. (1982) *Monetary Trends in the United States and the United Kingdom: Their Relations to Income, Prices, and Interest Rates*, University of Chicago Press, Chicago.

Hendry D.F., Ericsson N.R. (1991) Modeling the demand for narrow money in the United Kingdom and theUnited States, *European Economic Review*, 35(4), 833-881.

Hendry S. (1995) Long-Run Demand for M1, *WP* 95-11, Bank of Canada, Ottawa, Canada.

Ley E., Steel M.F. (2007) Jointness in Bayesian Variable Selection with Applications to Growth Regression, *Journal of Macroeconomics*, 29(3), 476-493.

Sala-i-Martin X., Doppelhofer G., Miller R.I. (2004) Determinants of Long-Term Growth: A Bayesian Averaging of Classical Estimates (BACE) Approach, *The American Economic Review*, 94(4), 813-835.

Tobin J. (1956) The Interest-Elasticity of Transactions Demand For Cash, *The Review of Economics and Statistics*, 38(3), 241-247.

# Estimating segmented regression in gretl

Andrea Bucci*, Andrea Faragalli**, Edlira Skrami***,
Marianxhela Dajko****, Flavia Carle***** Luigi Ferrante******

*Abstract:* In many fields, including biology, epidemiology, and toxicology, response variables are often related to one or more explanatory variables in a segmented way, *i.e.* two or more straight lines are connected at given change-points. A researcher may be interested in estimating the parameters of such regression and the location of the change-point(s). To do so, a linearization of the non-linear term may be implemented to obtain a linear model easy to estimate. Thus, we created a Gretl package capable of estimating the parameters and the breakpoint(s) in a piecewise regression. Finally, we used it to estimate chronological age in a Bayesian calibration framework.

*Keywords:* Segmented regression.

## *1. Introduction*

In analysing the relationship between a response variable and a set of covariates, sometimes it happens that this relationship presents one or more break-points, *i.e.* points where the relationship changes unexpectedly. In such cases, linear regression does not accurately fit the data, given the presence of the so-called segmented or broken-line relation (see Figure 1 for an example of segmented line).

When a relationship is supposed to be piecewise linear, it is necessary to define the number of break-points, $\psi = \{\psi_j, \dots \psi_m\}$, and the location of them. However, the presence of two or more straight lines connected makes the estimation quite difficult because the log-likelihood becomes piecewise differentiable, see Feder (1975). Consequently, Muggeo (2003) introduced a

*Università Politecnica delle Marche, a.bucci@staff.univpm.it
**Università Politecnica delle Marche, a.faragalli@staff.univpm.it
***Università Politecnica delle Marche, e.skrami@staff.univpm.it
****Università Politecnica delle Marche, m.dajko@staff.univpm.it
*****Università Politecnica delle Marche, f.carle@staff.univpm.it
******Università Politecnica delle Marche, l.ferrante@staff.univpm.it

method to estimate broken line model's coefficients by the linearization of the piecewise regression.

*2. Estimating Segmented Regression*

Let $Y$ be a response variable,and $X$ a set of covariates, with a non-linear term for a covariate $Z$, $g(Z;\psi_j)$, the following model:

$$y = \beta_1 + \beta_2 X + \sum_{j=1}^{m} \beta_{3,j} \cdot g(Z;\psi_j) \tag{1}$$

can be written as follows if a number of $m$ breakpoints is assumed:

$$y = \beta_1 + \beta_2 X + \sum_{j=1}^{m} \beta_{3,j}(Z-\psi_j)_+ \tag{2}$$

if $g(\cdot)$ is assumed to be a segmented relation where $(Z-\psi_j)_+$ is specified as

$$(Z-\psi_j)_+ = \begin{cases} (Z-\psi_j) & \text{if} \quad Z > \psi_j \\ 0 & \text{if} \quad Z \leq \psi_j \end{cases} \tag{3}$$

then, the log-likelihood of such model is not differentiable when $Z = \psi_j$.

As a consequence, the maximum likelihood estimation of $\beta$ should be obtained by linearizing $(Z-\psi_j)_+$ through first order Taylor expansion around $\psi_j^{(0)}$, such that the non-linear term $g(\cdot)$ may be written as

$$g(Z;\psi_j) = (Z-\psi_j)_+ \approx (Z-\psi_j^{(0)})_+ + (\psi_j - \psi_j^{(0)})(-1)I(Z > \psi_j^{(0)})$$

where$(-1)I(Z > \psi_j^{(0)})$ is the first derivative of $(Z-\psi_j)_+$ assumed in $\psi_j^{(0)}$ and $I(\cdot)$ is the index function, such that $I(Z > \psi_j^{(0)}) = 1$ if $Z$ is greater than the parameter $\psi_j^0$, and 0 otherwise.

The estimation process of parameters through maximum likelihood should be the following:

1. Fix $\psi_j^{(0)}$ and $\delta > 0$.

2. At iteration $k$, calculate

$$U_{j,i}^{(k)} = (Z_i - \psi_j^{(k)})_+ \quad \text{and} \quad V_{j,i}^{(k)} = -I(Z_i > \psi_j^{(k)}).$$

3. Substitute in Equation (2) the nonlinear term $(Z - \psi_j)_+$ with $U_j^{(k)}$ and $V_j^{(k)}$ as follows

$$y_i = \beta_1^{(k)} + \beta_2^{(k)} X_i + \sum_{j=1}^{m} \left( \beta_{3,j}^{(k)} U_{j,i}^{(k)} + \varphi_j^{(k)} V_{j,i}^{(k)} \right) \tag{4}$$

where

$$\varphi_j^{(k)} = \beta_{3,j}^{(k)} (\psi_j - \psi_j^{(k)}). \tag{5}$$

4. Estimate the parameters in Equation (4) and calculate

$$\psi_j^{(k)} = \frac{\hat{\varphi}_j^{(k-1)}}{\hat{\beta}_{3,j}^{(k-1)}} + \psi_j^{(k-1)}. \tag{6}$$

5. Repeat from Step 2 until convergence, *i.e.* $\max \mid \psi_j^{(k)} - \psi_j^{(k-1)} \mid < \delta$, at $K$-th iteration. The maximum likelihood estimates for $\beta$ are given by:

$$\hat{\beta} = \left( \beta_1^{(K)}, \beta_2^{(M)}, \beta_{3,1}^{(K)}, \ldots \beta_{3,m}^{(K)} \right)$$

In this way, the non-differentiable model (2) may be treated as a linear model with covariates $W = \{X, U_1, \ldots, U_m, V_1, \ldots, V_m\}$ and parameters, $\theta = \{\beta_1, \beta_2, \beta_{3,1}, \ldots, \beta_{3,m}, \varphi_1, \ldots, \varphi_m\}$. From Equation (5) it is clear that improvements of estimates of the breakpoint depend on the ratio $\frac{\hat{\varphi}_j}{\hat{\beta}_{3,j}}$. It is worth noticing that this ratio, given that $\hat{\beta}_{3,j}$ should be different from zero, never goes to infinity; see Muggeo (2003). We implemented a function package, in Gretl, which allows to obtain estimates of coefficients in a broken-line model and location of break-point(s). In Section 3, we further applied such package on a simulated dataset to obtain estimates of chronological age from a dental

maturity index in a Bayesian framework.

*3. Example*

Following Ferrante *et al.* (2015), we used a Bayesian calibration model with Normal probability to estimate the unknown simulated chronological age. In such context, we specified the expected value of a distribution as a segmented relation (`SNBC`) with a single breakpoint, such that

$$\mu(y, \beta) = \beta_1^{(K)} + \beta_2^{(K)} y + \beta_3^{(K)} U^{(K)} + \varphi^{(K)} V^{(K)} \tag{7}$$

where $U^{(K)} = (y - \psi^{(K)})_+$ and $V^{(K)} = -I(y > \psi^{(K)})$. The probability model (see Ferrante *et al.*, 2015) becomes

$$p(x \mid y, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ - \frac{\left(x - \beta_1^{(K)} - \beta_2^{(K)} y - \beta_3^{(K)} U^{(K)} - \varphi^{(K)} V^{(K)}\right)^2}{2\sigma^2} \right\}. \tag{8}$$

From this model, we estimated chronological age on a simulated dataset and compared the estimation results with a set of competing models. We showed that the calibrating method implementing segmented regression exhibited lower MAE, and RMSE compared to competing models. In addition, IQR was lower for the segmented-based model, `SNBC`. These results were valid regardless of which break-point had been chosen.

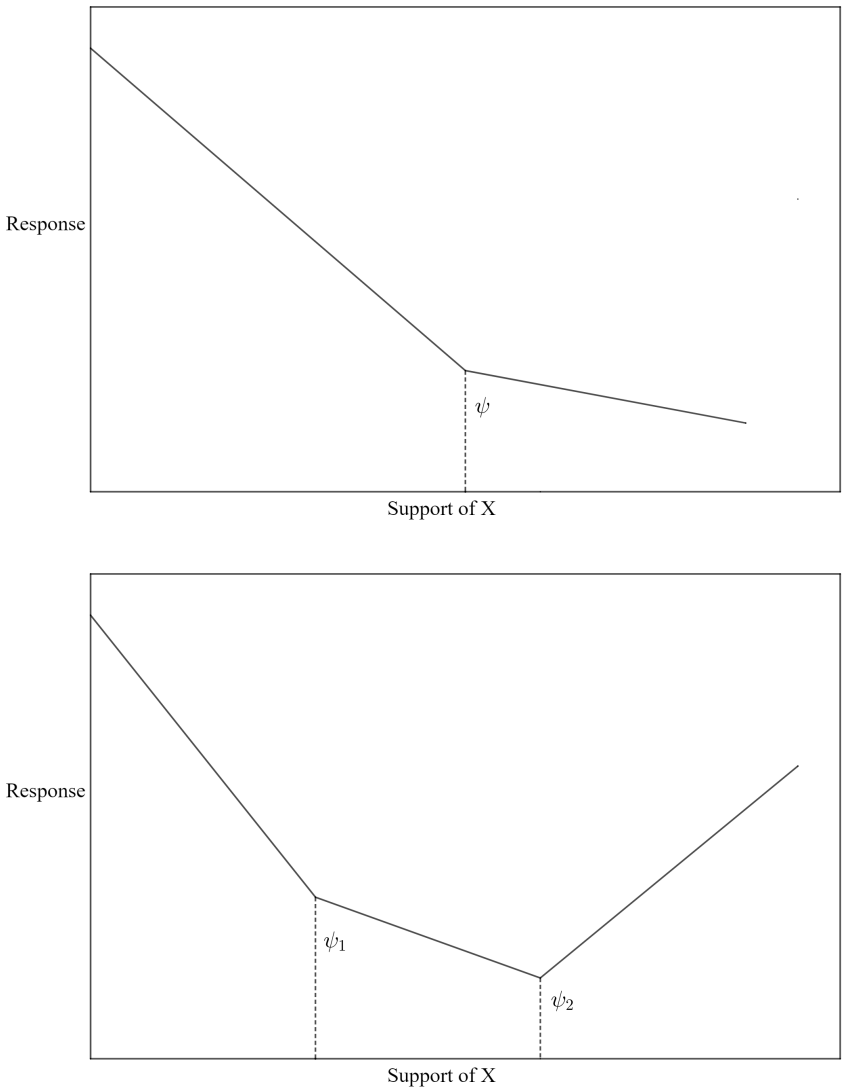



*Figure 1. Two possible segmented relations.*

*Table 1. Performance of the considered models using simulated data.*

| Method | MAE | RMSE | $IQR_{ERR}$ | $MCI_{95\%}$ |
|---|---|---|---|---|
| $\psi = 5$ | | | | |
| `SNBC` | 0.778 | 1.103 | 0.923 | 3.886 |
| `LNBC` | 0.940 | 1.169 | 1.513 | 5.529 |
| `Linear regression` | 1.014 | 1.241 | 1.719 | 5.844 |
| $\psi = 8$ | | | | |
| `SNBC` | 0.586 | 0.834 | 0.805 | 2.646 |
| `LNBC` | 0.758 | 0.908 | 1.389 | 3.390 |
| `Linear regression` | 0.770 | 0.942 | 1.505 | 3.523 |

Where `LNBC` implies the use of a linear relationship in the location parameter, linear regression is a simple regression model where the chronological age is the dependent variable.

## *4. Conclusions*

Estimating piecewise regression can be tricky for the presence of the change-point(s). Following Muggeo (2003), we proved that a linear model may be derived from it. Then, we implemented a package in Gretl which allows to estimate the coefficients and the location of the breakpoint in a segmented regression. We used it to estimate chronological age in a Bayesian context, proving the superior performance in terms of estimation accuracy.

## *References*

Betts M., Forbes G., Diamond A. (2007) Thresholds in songbird occurrence in relation to landscape structure, *Conservation Biology*, 21, 1046-1058.

Muggeo V.M.R. (2003) Estimating regression models with unknown break-points, *Statistics in Medicine*, 22, 3055-3071.

Muggeo V.M.R. (2008) segmented: An R package to Fit Regression Models with Broken-Line Relationships, *R News*, 8, 20-25.

Ulm K. (1991) A statistical methods for assessing a threshold in epidemiological studies, *Statistics in Medicine*, 10, 341-349.

# Spatial models in gretl: the SPM package

Chiara Casoli*, Luca Pedini**, Francesco Valentini***

*Abstract:* This article presents a new package for estimating cross-sectional spatial models in gretl, named SPM. The package can handle three types of models: Spatial Autoregressive Models (SAR), Spatial Durbin Models (SDM) and Spatial Error Models (SEM). The first integrates the canonical linear model by including spatial lags of the dependent variable, the second also includes spatial lags of independent variables, and the last examines spatially autoregressive errors. Computation of the Hessian matrix is performed in both analytical and mixed ways. Some speed-up procedures for the computation of the log-determinant term are implemented and compared. Finally, results of the proposed package are compared with those of some software alternatives, namely Matlab Spatial Econometric Toolbox, Stata module sp and R packages spatialreg and spdep.

*Keywords:* Cross Sectional Spatial Models, Maximum Likelihood Estimation, Software Comparison.

## *1. Introduction*

Spatial econometrics deals with spatial data, representing particular kinds of observations associated with geographical areas. These could include countries, regions or more in general, points with a particular position in space. The intuition is that space matters, requiring *ad hoc* models able to take into account the existence of some spatial relationships among variables. Because of the two concepts of *spatial dependence* and of *spatial spillovers*, the standard assumptions of linear models no longer hold. Spatial dependence makes values of one particular area dependent on values of the near locations - with nearer areas more important than farther areas - thus implying a simultaneous data generating process.

There are nowadays several tools for dealing with spatial models, but none

* Università Politecnica delle Marche, c.casoli@pm.univpm.it
** Università Politecnica delle Marche, l.pedini@pm.univpm.it
*** Università Politecnica delle Marche, f.valentini@pm.univpm.it

was yet included in gretl. With this work we aim at filling this gap, introducing also some speed up procedures for large dataset. There are indeed two aspects which could severely affect the procedure' speed: the computation of the analytical Hessian matrix, and of the log-determinant term - entering in the likelihood function - which implies a huge computational burden. The proposed SPM gretl package can handle cross-section estimation of Spatial Autoregressive, Spatial Durbin and Spatial Error models via Maximum Likelihood. For what concerns the Hessian matrix, we provide both the pure analytical and the mixed solution as proposed in LeSage and Pace (2009). The computation of the log-determinant term in the profile likelihood $\ln |I_n - \rho W|$ is performed analytically, exploiting the LU factorisation, and with two approximations, namely Monte Carlo and Chebyshev polynomials.

## *2. The Spatial Autoregressive process and the Weight matrix*

The concept of spatial dependence, which makes values of one particular area dependent on values of the near locations - with nearer areas more important than farther areas - would imply a simultaneous data generating process. Unfortunately, it is useless to formalise a model taking into account *all* the possible spatial relationships among variables, especially for a large dataset: with $n$ observations, the number of potential dependence relations which could arise is equal to $n^2 - n$. The over-parametrisation issue was solved by Ord (1975), who proposed a more parsimonious specification able to summarise the concept of spatial dependence and the degree of interactions between regions into only two terms: the parameter $\rho$ and the *spatial weights*, $w_{ij}$. Formalising this structure, we obtain a DGP which is called *spatial autoregressive process*[1]:

$$y_i = \alpha + \rho \sum_{j=1}^{n} w_{ij} y_j + \varepsilon_i, \tag{1}$$

in which $y_i$ is the observation of a variable in the $i$-th region and $\varepsilon_i$ is a normally distributed error term with zero mean and variance $\sigma^2$. The summation

[1] We will follow the notation of LaSage and Pace (2009).

$\sum_{j=1}^{n} w_{ij}y_j$ defines the *spatial lag*, representing a linear combination of values of $y$ built from neighbour areas of the observation $i$. The collection of the spatial weights $w_{ij}$ allows to build the *spatial weight matrix* $W$, of dimension $n \times n$, which describes all the spatial relationships among areas.
The $W$ matrix plays a crucial role in the estimation phase and must be constructed properly, satisfying some properties. Firstly, $w_{ii} = 0$, meaning that the spillover from $i$ to itself will not be computed. Then, $w_{ij} > 0$ only if there exists a neighbour relationship between the two regions, otherwise it is set to $0$; in practice, matrix $W$ is often sparse. The matrix could also be asymmetric, so it could happen that $w_{ij} \neq w_{ji}$. Finally, the spatial weights should be "exogenous", meaning that the phenomenon under investigation should not be correlated with spatial weights[2]. In the SPM package, $W$ will be imported from other sources and it is important to normalise it for computational purposes: this normalisation implies that the maximum eigenvalue should be set equal to 1. This can be achieved by imposing the sum of each row equal to 1, defining a *row stochastic* spatial weight matrix. In SPM, this is done by default.

### *3. The Spatial Autoregressive model and the Spatial Durbin Model*

The Spatial Autoregressive Model (SAR) and the Spatial Durbin Model (SDM) will be jointly analysed, as it is possible to see the latter as an extension of the former. The SAR model includes indeed spatial lags of the dependent variable only, whereas the SDM adds also spatial lags of the covariates. Generalising, and defining the explanatory variables as $Z = [\iota_n X\ WX]$ and the related parameters $\delta = [\alpha\ \beta\ \theta]'$, it is possible to write:

$$y = \rho W y + Z\delta + \varepsilon \tag{2}$$

$$y = (I_n - \rho W)^{-1}(Z\delta + \varepsilon) \tag{3}$$

$$\varepsilon \sim N(0, \sigma^2 I_n),$$

[2] For further information on the construction of the spatial weight matrix, see the existing literature on the topic (Anselin 1988, LeSage and Pace 2009, Arbia 2014).

where equation (2) denotes the SAR if $Z = [\iota_n X]$ or the SDM if $Z = [\iota_n X\ WX]$, and equation (3) the related DGP.

Estimation of parameters $\rho$, $\delta$ and $\sigma^2$ can be implemented via Maximum Likelihood[3]. The log-likelihood function is given by:

$$\ln L = -(n/2)\ln(\pi\sigma^2) + \ln|I_n - \rho W| - \frac{e'e}{2\sigma^2} \tag{4}$$

$$e = y - \rho W y - X\beta$$

$$\rho \in (min(\omega)^{-1}, max(\omega)^{-1}), \tag{5}$$

in which $\omega$ contains the eigenvalues of the spatial weights matrix. If $W$ has been scaled such to have the maximum eigenvalue equal to 1, it is possible to restrict the interval such that $\rho \in (min(\omega)^{-1}, 1)$. As already mentioned, the way in which $W$ is built has several consequences for estimation purposes. Another computational difficulty may arise while computing the log-determinant in equation (4); Section 5 will deepen the matter and provide some solutions to achieve good results with gains in computational time. The optimisation problem can be easily handled using the concentrated log-likelihood (equation (6)) as a function of the only parameter $\rho$, as in LeSage and Pace (2009). $\delta$ and $\sigma^2$ can be consequently derived as a function of the estimated $\rho$. This can be summarised in:

$$\ln L_\rho = c + \ln|I_n - \rho W| - (n/2)\ln[(e_0 - \rho e_d)'(e_0 - \rho e_d)] \tag{6}$$

$$e_0 = y - Z\delta_0 \tag{7}$$

$$e_d = Wy - Z\delta_d \tag{8}$$

$$\delta_0 = (Z'Z)^{-1}Z'y \tag{9}$$

$$\delta_d = (Z'Z)^{-1}Z'Wy, \tag{10}$$

in which $c$ is a constant term, $\delta_0$, $e_0$, $\delta_d$ and $e_d$ are computed *ex ante* from two auxiliary regressions of $y$ and $Wy$ on $Z$ respectively. The Maximum Likelihood estimates of parameters $\hat{\delta}$, $\hat{\sigma^2}$ and the associated disturbances variance-covariance matrix $\hat{\Omega}$ are given by: $\hat{\delta} = \delta_0 - \hat{\rho}\delta_d$, $\hat{\sigma^2} = n^{-1}(e_0 - \hat{\rho}e_d)'(e_0 - \hat{\rho}e_d)$

[3] However, note that, on the contrary of standard linear models, in those including spatial lags the interpretation of parameters is not straightforward.

and $\hat{\Omega} = \hat{\sigma^2}[(I_n - \hat{\rho}W)'(I_n - \hat{\rho}W)]^{-1}$. Finally, to calculate standard errors and the related $t$ statistics, the variance-covariance matrix of the parameters is necessary. Following the maximum likelihood framework, it is equal to $-(H)^{-1}$, where $H$ denotes the Hessian matrix. However, straightforward evaluation of the analytical Hessian or information matrix element $\partial^2 L/\partial\rho^2$ involves computing a trace term which contains the $n \times n$ matrix $A = (I_n - \rho W)^{-1}$, specifically equal to $tr(WAWA + WAA'W')$. This trace term becomes problematic for large $n$, so a "mixed" analytical-numerical strategy, as named in LeSage and Pace (2009), is adopted as an alternative. This method simply exploits the second derivatives of the univariate profile likelihood with respect to $\rho$ which arise after the optimisation, $\partial^2 L_\rho/\partial\rho^2$.

## *4. The Spatial Error Model*

The Spatial Error Model (SEM) contains spatial dependences in the *disturbances*, as shown in equation (11), with (12) being the DGP.

$$y = X\beta + u \tag{11}$$

$$u = \lambda W u + \varepsilon$$

$$y = X\beta + (I_n - \lambda W)^{-1}\varepsilon \tag{12}$$

$$\varepsilon \sim N(0, \sigma^2 I_n).$$

The full log-likelihood is given by:

$$\ln L = -(n/2)\ln(\pi\sigma^2) + \ln|I_n - \lambda W| - \frac{e'e}{2\sigma^2} \tag{13}$$

$$e = (I_n - \lambda W)(y - X\beta).$$

Again, it is possible to concentrate the log-likelihood, as a function of the only parameter $\lambda$, and then recover $\beta$ and $\sigma^2$; unlike the previous case, however, $e(\lambda)'e(\lambda)$ is not a simple quadratic form of the parameters, but is derived from moment matrices as in (15)

$$\ln L_\lambda = c + \ln|I_n - \lambda W| - (n/2)\ln(e(\lambda)'e(\lambda)) \tag{14}$$

$$
\begin{aligned}
A_{XX}(\lambda) &= X'X - \lambda X'WX - \lambda X'W'X + \lambda^2 X'W'WX \qquad (15)\\
A_{Xy}(\lambda) &= X'y - \lambda X'Wy - \lambda X'W'y + \lambda^2 X'W'Wy \\
A_{yy}(\lambda) &= y'y - \lambda y'Wy - \lambda y'W'y + \lambda^2 y'W'Wy \\
\beta(\lambda) &= A_{XX}(\lambda)^{-1} A_{Xy}(\lambda) \\
e(\lambda)'e(\lambda) &= A_{yy}(\lambda) - \beta(\lambda)' A_{XX}(\lambda)\beta(\lambda).
\end{aligned}
$$

The values for $\hat{\beta}$ and $\hat{\sigma^2}$ can be recovered, again, straightforwardly (LeSage and Pace (2009)). The variance-covariance matrix is computed, as in the previous case, in both analytical and mixed ways, with the only difference that for the SEM, the exploited second derivative of the profile likelihood involves parameter $\lambda$ instead of $\rho$. The comparison of the different Hessian calculations will be carried out in Section 6, where also the various approaches to deal with the log-determinant will be analysed.

*5. Log-determinants and computational advantages*

Speed up procedures for the estimation of spatial models are often implemented, given the computational burden of several components such as the log-determinant term, especially when $n$ is large. In SPM we implemented routines that compute two well-known approximations of $\ln|I_n - \rho W|$, specifically Monte Carlo and Chebyshev polynomials[4].
The Monte Carlo approximation exploits the fact that the log-determinant equals the trace of the logarithm of the matrix $(I_n - \rho W)$:

$$\ln|I_n - \rho W| = tr[\ln(I_n - \rho W)].$$

Given this starting point, two results follow. Firstly, the logarithm of the matrix can be expressed as an infinite series of the kind: $\ln(I_n - \rho W) = -\sum_{i=1}^{\infty} \frac{\rho^i W^i}{i}$. Secondly, since the trace operator is linear, we can rewrite the problem as: $tr[\ln(I_n - \rho W)] = -\sum_{i=1}^{\infty} \frac{\rho^i tr(W^i)}{i}$.

It is now possible to approximate the log-determinant through a finite,

[4] For further technicalities see LeSage and Pace (2009).

lower-order series of order $q$, obtaining:

$$\ln|I_n - \rho W| \approx -\sum_{i=1}^{q} \frac{\rho^i tr(W^i)}{i}$$

However, generating all the powers of the matrix $W$ and computing their trace is inefficient from a computational point of view. It can be proven that we can estimate the trace of the matrices by a quadratic form $\tilde{T}^i = u'W^i u$, where $u$ is a random $n \times 1$ vector drawn from a Gaussian distribution. In order to increase the computational accuracy, $\tilde{T}^i$ is recomputed $m$ times replacing the random draw of $u$. Then, averaging over the $m$ iterations, we end up with the following estimated quantity:

$$\ln|I_n - \rho W| \approx -\frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{q} \frac{\rho^i \tilde{T}_j^i}{i}.$$

For what concerns instead the Chebyshev approximation, it exploits the possibility of calculating the log-determinant via Chebyshev polynomials of degree $q$, with the advantage of being computed recursively as follows:

$$tr[\ln(I_n - \rho W)] \approx \sum_{j=1}^{q+1} c_j tr(T_{j-1}(W)) - \frac{n}{2}c_1.$$

In particular, $c_j$ are the Chebyshev coefficients and are defined as:

$$c_j(\rho) = \frac{2}{q+1}\sum_{k=1}^{q+1} f(x_k)\cos\left(\frac{\pi(j-1)(k-\frac{1}{2})}{q+1}\right)$$
$$x_k = \cos\left(\frac{\pi(k-\frac{1}{2})}{q+1}\right)$$
$$f(x) = \ln(1-\rho x);$$

$T_j(W)$ are the Chebyshev polynomials, expressed as:

$$\begin{aligned} T_0(W) &= I_n \\ T_1(W) &= W \\ T_{q+1}(W) &= 2WT_q(W) - T_{q-1}(W) \quad q \geq 1. \end{aligned}$$

### *6. Performance of different computational techniques*

The previous Section introduced the variety of available options concerning the computation of the log-determinant term. By use of a Monte Carlo simulation, this Section aims to compare the actual impact of those techniques on the time required for the estimation procedure of a SAR model. The simulation design is very simple: data are generated by a *spatial autoregressive process* as in Equation (3) where the dependent variable $y$ and the error term $\varepsilon$ are drawn from a Gaussian distribution. We fixed the constant term $\alpha = 0.5$ and the regression coefficient $\beta = 1$. The parameter $\rho$ assumes values in $\{0, 0.25, 0.5, 0.75\}$. We assume that all the observations ($n = 10000$) lay in a linear space and the weight matrix $W$ is built as a first-order contiguity matrix. The number of replications is $N = 50$.

*Table 1. Average computational time (in seconds), $n = 10000$*

| | $\rho = 0$ | | $\rho = 0.25$ | | $\rho = 0.50$ | | $\rho = 0.75$ | |
|---|---|---|---|---|---|---|---|---|
| | Analytical | Mixed | Analytical | Mixed | Analytical | Mixed | Analytical | Mixed |
| ldet() | 692.11 | 665.06 | 923.39 | 890.02 | 977.87 | 942.01 | 1188.5 | 1139.4 |
| MC(30, 50) | 370.40 | 335.25 | 374.10 | 335.81 | 376.55 | 335.91 | 371.20 | 315.24 |
| MC(50, 50) | 487.33 | 452.84 | 498.05 | 455.54 | 495.11 | 451.41 | 478.68 | 417.32 |
| MC(50, 100) | 792.41 | 767.11 | 789.75 | 752.63 | 766.90 | 743.22 | 730.93 | 698.52 |
| Chebyshev(5) | 225.11 | 199.98 | 230.46 | 201.64 | 242.65 | 204.94 | 252.89 | 205.19 |
| Chebyshev(10) | 310.23 | 285.56 | 315.12 | 283.97 | 339.01 | 301.78 | 338.01 | 290.59 |

Table 1 reports the average time required by the estimation procedure with six different techniques[5] (by row) run for eight different scenarios (by column). Concerning the notation, ldet() is the gretl function that computes the

[5] Reported methods produce estimated parameters and std. errors that differ at the fifth decimal point.

log-determinant exploiting the LU factorisation and represents the pure analytical method. The Monte Carlo approximation is denoted by MC(·,·) where the first item is the length of the approximating series and the second the number of inner replications. Finally, Chebyshev polynomial and the related order are given by Chebyshev(·). For each value of $\rho$ we report the CPU-time in seconds according to whether the Hessian is computed as "pure analytical" or via the "mixed" strategy.

As expected, ldet() is the slowest solution.
Regarding the approximations, Chebyshev polynomials behave at best. In facts, a polynomial of order five is sufficient to perfectly replicate the analytical results saving up 70% of the time. The MC(·,·) approximation works properly but it is sensitive to the order of the series and the number of inner replications we consider. Finally, the usage of the "mixed" Hessian seems to slightly speed up the estimating procedure.

Further investigations are needed to understand the performance of the techniques developed in SPM, namely it would be interesting to discuss how robust are these preliminary results with different, more complex specifications of the spatial linkages among observations.

## *7. Empirical application with comparison of results*

In order to compare the proposed package with the existing alternatives, we provide an empirical example exploiting Maximum Likelihood Estimation within different software packages, in the spirit of Bivand and Piras (2015). Specifically, we refer to **spdep** and **spatialreg** in R[6], to **Spatial Econometrics Toolbox** in Matlab[7] and to the command **spregress**, comprehended in the **sp** module, in Stata[8]. We exploit the publicly disposable data used in Anselin (1988), reporting observations for 49 contiguous Planning Neighborhoods in the city of Columbus, Ohio, for the year 1980. The three variables are CRIME, INC and HOUSE, which denote the aggregation of burglaries and vehicle thefts (in thousand USD in the neighbourhood), income (in thousand

[6] https://CRAN.R-project.org/package=spdep and https://CRAN.R-project.org/package=spatialreg
[7] https://www.spatial-econometrics.com/
[8] https://www.stata.com/manuals/spspregress.pdf

USD) and housing values (in thousand USD), respectively.

*Table 2. SAR model (ML) estimation. Dependent variable: CRIME. Standard errors in parentheses.*

| | gretl | R | Stata | Matlab |
|---|---|---|---|---|
| Intercept | 38.065 | 38.064460 | 38.06446 | 38.2720 |
| | (6.5519) | (6.551942) | (6.607553) | (6.5661) |
| INC | -0.97346 | -0.973458 | -0.973458 | -0.9777 |
| | (0.27115) | (0.271153) | (0.275893) | (0.2713) |
| HOUSE | -0.23279 | -0.232790 | -0.232790 | -0.2331 |
| | (0.078048) | (0.078048) | (0.078012) | (0.0781) |
| $\rho$ | 0.58697 | 0.586961 | 0.586964 | 0.5830 |
| | (0.10440) | (0.104396) | (0.106897) | (0.1049) |
| $\sigma^2$ | 74.031 | 74.03103 | 74.03103 | 74.1258 |
| | (15.153) | (15.15279) | (15.16223) | - |

*Table 3. SEM (ML) estimation. Dependent variable: CRIME. Standard errors in parentheses.*

| | gretl | R | Stata | Matlab |
|---|---|---|---|---|
| Intercept | 58.231 | 58.231002 | 58.231 | 58.2307 |
| | (6.2420) | (6.242020) | (6.252355) | (6.2451) |
| INC | -0.85775 | -0.857755 | -0.857755 | -0.8576 |
| | (0.28714) | (0.287144) | (0.305209) | (0.2871) |
| HOUSE | -0.23807 | -0.238072 | -0.238072 | -0.2381 |
| | (0.079773) | (0.079773) | (0.079774) | (0.0798) |
| $\lambda$ | 0.74878 | 0.748771 | 0.748776 | 0.7490 |
| | (0.11143) | (0.109814) | (0.123736) | (0.1098) |
| $\sigma^2$ | 78.099 | 78.09908 | 78.09908 | 78.0899 |
| | (16.635) | (16.39337) | (16.55526) | - |

The $W$ matrix is constructed as row-stochastic using two direction coordinates $x$ and $y$ to produce spatial contiguity within the matrix. Here we focus

on the parameters estimation for the SAR model and the SEM. As reported in Table 2 and Table 3, the estimated parameters and the relative standard errors are very close. Small differences can be explained with the different optimisation algorithms employed by each software.

## 8. Concluding remarks

The SPM package provides a tool for the estimation of cross-sectional spatial models via Maximum Likelihood in gretl. The models which SPM handle are the Spatial Autoregressive Model, the Spatial Durbin model and the Spatial Error model. Among the main features of the new gretl package there are the possibility of computing the Hessian matrix in the canonical analytical way or via a mixed analytical-numerical solution, and some different techniques provided for the computation of the log-determinant term entering in the likelihood function. These methodologies, other than the analytical computation, include two approximations, exploiting Monte Carlo and Chebyshev polynomials, respectively. A simulation experiment allows us to compare the different techniques in terms of speed, and in this respect the better procedure is found to be an approximation via a Chebyshev polynomial of order five, which allows to save up to 70% of CPU-time with respect to the pure analytical solution. Finally, an empirical example on crime in different neighbourhoods in the city of Columbus, Ohio, in 1980, allows us to compare the performance of the SPM gretl package with the existing software alternatives, namely Matlab Spatial Econometric Toolbox, Stata command spregress and R packages spatialreg and spdep. The estimated parameters and relative standard errors are very close, suggesting a good performance of the proposed procedure from a computational and accuracy point of view.

## References

Anselin L. (1988) *Spatial Econometrics: Methods and Models*, Springer, Dordrecht (NL).

Arbia G. (2014) *A primer for spatial econometrics: with applications in R*, Palgrave-McMillian, UK.

Bivand R., Anselin L., Berke O., Bernat A., Carvalho M., Chun Y., Lewin-Koh N. (2011) spdep: Spatial dependence: weighting schemes, statistics and models, *R package*

*version 0.30-30*, `http://ftp.auckland.ac.nz/software/CRAN/src/contrib/Descriptions/spdep.html`

Bivand R., Piras G. (2015) Comparing implementations of estimation methods for spatial econometrics. *Journal of Statistical Software*, 63(18).

LeSage J. P. (1999) Spatial econometrics using MATLAB, *Manuscript, Deprtment of Economics, University of Toledo, Ohio* www.econ.utoledo. edu.

LeSage J., Pace R.K. (2009) *Introduction to spatial econometrics*, Chapman and Hall/CRC, Bocha Raton, USA.

Ord K. (1975) Estimation methods for models of spatial interaction, *Journal of the American Statistical Association*, 70(349), 120-126.

Pace R.K., Barry R. (1997) Quick computation of spatial autoregressive estimator, *Geographical analysis*, 29(3), 232-247.

# SVMs in gretl: the case of ordinal data

Allin Cottrell*

*Abstract:* An account is given of recent improvements in gretl's support for machine learning via Support Vector Machines. An extended example of prediction of an ordinal outcome (ranking of wine quality) is presented. Ranking SVMs are shown to offer more accurate prediction than several alternative methods.

*Keywords:* Machine Learning, SVMs, Ranking.

## *1. Introduction*

This article advances prior discussion of machine learning in gretl in two main ways. It reports on some recent improvements to gretl's SVM support, and it illustrates use of machine learning for ordinal data, complementing examples using continuous and categorical data in Cottrell (2019).

We confine ourselves to a very brief account of SVMs here. The interested reader is referred to Mullainathan and Spiess (2017), Smola and Schlöpf (2004), Cottrell (2019). SVMs, for which the classic reference is Vapnik (1998), subsume earlier machine learning techniques such as decision trees and neural nets with up to one hidden layer. Unlike the latter methods, SVMs solve a well-defined optimization problem a case of quadratic programming and are not liable to get trapped at a local optimum (Lin and Li, 2008; Berwick, 2003). Their original application was to binary classification but the method has been extended to cover multinomial classification, regression (where we have a continuous dependent variable rather than categorical "labels") and, more recently, ranking or ordinal outcomes. Nonlinear prediction functions are supported via the so-called "kernel trick" (Jordan, 2004), which projects the data into a higher dimensional space.

To facilitate understanding of what follows we offer a few words on the workflow associated with SVMs. The object is to predict out of sample. To

*Wake Forest University, cottrell@wfu.edu

that end one divides the available data into training and testing subsets. In the simplest variant, one trains an SVM on the training data then uses it to generate predictions for the testing data; if the predictive performance of the model on the latter is good enough one is then encouraged to use it on actually unseen data.

More commonly, however, training involves cross validation. An SVM typically includes a (potentially very large) number of coefficients whose value is set via a training function (as provided by LIBSVM, for example). But these coefficients, and therefore the predictions, usually depend on at least one hyperparameter, which must be set either by some rule of thumb or "optimally". Optimization of the hyperparameter value(s) is generally performed by cross validation, which involves dividing the training data into $k$ subsets or "folds". The algorithm is then, in pseudo-code:

```
for each hyperparameter value (or set of such values)
   for each fold i
      train the SVM on k-1 folds, excluding fold i
      compute predictions for fold i
   end
end
```

Once this is complete one has a set of predictions for every observation in the training set, for each hyperparameter specification. One then selects the "best" specification, according to a chosen loss function, and trains the SVM on the entire training set. (Note that the testing data are not, and *must not be*, referenced in this entire procedure.) Only then do we use the SVM to predict for the testing data.

## *2. SVMs in gretl*

gretl's support for SVMs is based on LIBSVM, written by Chih-Chung Chang and Chih-Jen Lin[1], which according to the website svms.org appears to be the leading SVM software as of this writing. Our SVM "plugin" uses a modified version of LIBSVM 3.23 (released July, 2018). The modifications

[1] See `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

include enhanced random number generation, parallelization via OpenMP, and support for rank regression (of which more below).

As explained in Cottrell (2019) gretl's `svm` function is set up to perform the whole procedure described above, that is, training with cross validation followed by prediction for both the training and testing data, although the steps can be unbundled if desired. Stripped of optional arguments associated with unbundling, the function has the following simple signature

```
series svm(list L, bundle parms)
```

where `L` is a list holding the dependent variable followed by the explanatory or predictive variables and `parms` is a bundle containing parameters. The return value is a series containing predictions from the SVM.

The parameters that can be set in the `parms` bundle are detailed in Cottrell (2019); we comment below on those that we used in the context of our empirical example.

## *3. Empirical example: wine quality*

The example offered here draws on Cortez *et al* (2009). The dependent variable is a ranking: the perceived quality of Vinho Verde wines from Portugal, on a scale of 0 (very bad) to 10 (very good) this is the median of at least three scores given by professional tasters. The independent variables or "features" (see Table 1) are measures of several physicochemical properties of the wines. There are two sets of data: red wines ($n = 1599$) and white wines ($n = 4898$). The research question is: how well can one predict the rating of a wine by tasters, based solely on its objectively measurable characteristics[2]?

We decided to run a "horse race" between two standard econometric methods (OLS and Ordered Probit) and three variants of SVM (classification, regression and ranking). This approach evinces an agnostic view of the dependent variable: ostensibly it's a ranking (which could be seen as calling for ordered probit on the standard econometric side, or SVM ranking) but might one usefully treat it as a ratio scale (OLS, SVM regression)? Or might it work better just to treat it as a case of classification? We'll see.

[2] Replication files (data and scripts) for this exercise are available at `http://gretl.sourceforge.net/svm-files/`.

*Table 1. Physicochemical properties of the wines*

| | | |
|---|---|---|
| 1 | `fixed_acidity` | Fixed acidity, g(tartaric acid)/dm$^3$ |
| 2 | `volatile_acidity` | Volatile acidity, g(acetic acid)/dm$^3$ |
| 3 | `citric_acid` | Citric acid, g/dm$^3$ |
| 4 | `residual_sugar` | Residual sugar, g/dm$^3$ |
| 5 | `chlorides` | Chlorides, g(sodium chloride)/dm$^3$ |
| 6 | `free_SO2` | Free sulfur dioxide, mg/dm$^3$ |
| 7 | `total_SO2` | Total sulfur dioxide, mg/dm$^3$ |
| 8 | `density` | Density, g/cm$^3$ |
| 9 | `pH` | pH |
| 10 | `sulphates` | Sulphates, g(potassium sulphate)/dm$^3$ |
| 11 | `alcohol` | Alcohol, percent by volume |

Compared with Cortez *et al* (2009) we're exploring a larger set of models, but we adopted a relatively simple, *ad hoc*, method of specification. Based on gretl's `bkw` test for collinearity we excluded some potential explanatory variables implicated in relations of near-linear dependence, but included (for OLS and Ordered Probit) quadratic terms that appeared as highly significant in gretl's test for nonlinearity.

The variables included in the red wine specification were those numbered 2, 5, 7, 9, 10, 11 and the square of sulphates. For the white wines we used 1, 2, 4, 6, 8, 9, 10, 11 and the square of alcohol. In the SVM specifications we omitted the squared terms since SVMs can handle nonlinearity without this sort of nudge.

In comparing methods of prediction it is of course necessary to specify a loss function. Given that the dependent variable is a discrete ranking, for the most part we relied upon "percent correct". This is straightforward for classification and ranking methods; for continuous regression methods we rounded predictions to the nearest integer. In addition we computed mean absolute deviation (MAD) and calculated a "tails correct" percentage. The latter is motivated by the fact that most of the wines received a middling quality score of 5 or 6 (82 percent of the red wines, 75 percent of the whites), so perhaps a prediction method could distinguish itself by its ability successfully to predict scores out of that range. ("Tail" scores are therefore defined as those less than

5 or greater than 6.)

We show below the hansl script for SVM regression using the red wine dataset. The other scripts used to produce the results shown in section 5 are very similar. Comments follow.

```
open winequality-red.gdt --quiet
set seed 54321
series sorter = normal()
dataset sortby sorter
list All = quality volatile_acidity chlorides total_SO2 pH \
  alcohol sulphates
scalar ntrain = 1066
svm_type = "eps-SVR"
bundle parms = defbundle("n_train", ntrain, "search", 1)
  parms.consecutive = 1
  parms.quiet = 0
  parms.epsilon = 0.06
  parms.svm_type = svm_type
  parms.use_mpi = 1
series yhat = round(svm(All, parms))
include assess.inp
assess_prediction(quality, yhat, 5, 6, ntrain, svm_type)
```

Note that the script above begins by randomizing the order of the data (just in case there's any systematic pattern to the observations). The same seed is used in randomization for all methods. We then flag 1066 observations (about two thirds) for training. Since this case employs SVM regression we select the $\epsilon$-SVR type. The other options have the following meanings:

| | |
|---|---|
| `search = 1` | Search for hyperparameter values with cross validation using default grid (see Cottrell, 2019). |
| `consecutive = 1` | Take the cross validation "folds" as consecutive blocks of observations (since the dataset is already randomized). |
| `quiet = 0` | Print details of hyperparameter search. |
| `epsilon = 0.06` | SVR-specific: set threshold below which prediction errors are costed at zero. |
| `use_mpi = 1` | Use MPI for cross validation (see below). |

Prior to version 2019c of gretl the `svm_type` parameter had to be given as an ID number; that's still accepted but it's now possible to use a mnemonic label, which is read on a case-insensitive basis: `C-SVC` for (unordered) classification; either `eps-SVR` or `nu-SVR` for SVM regression; or `C-rnk` for SVM ranking (see section 4).

As regards the `use_mpi` flag, we have to digress briefly on the subject of parallelization of SVM calculations. There are two main ways in which parallelization can speed things up significantly. First, with a trained model in hand, calculation of predicted values per observation can be divided between threads using OpenMP. Gretl implements this, following a suggestion on the LIBSVM website[3]. Second, an even more substantial speed-up can be achieved when parameter search via cross validation is performed, by dividing combinations of data-fold and hyperparameter values between MPI processes. This can also be done in gretl, in an automatic manner when the `use_mpi` flag is set in the parameter bundle.

gretl has supported MPI since 2017, but until very recently it has been up to the user to organize a division of labour between processes; the automatic division supported by the `svm` function is a new departure. Naturally, the automatic method is subject to certain limitations; in particular, it works in "local" mode only (exploiting multiple cores, if present, on the user's machine). If you wish to use multiple machines across a network, you will have to use gretlmpi (or an `mpi` block within a gretl script) yourself, see Cottrell (2019a). But there is some control available to the user in the automatic version. If you set the parameter `use_mpi` to 1, gretl invokes one MPI process per physical core on the host machine; if you set it to an integer value greater than 1, gretl takes this as the number of processes to be launched.

Still in the context of parameters passed to the `svm` function, informed readers may be wondering: what about selection of an SVM kernel? Indeed, a kernel must be specified when we call LIBSVM, but in the absence of an explicit choice by the user gretl selects a default that is likely to be best in most applications for classification and regression, the Gaussian RBF (Radial Basis Function) kernel.

[3] See `https://www.csie.ntu.edu.tw/~cjlin/libsvm/faq.html#f432`.

Cottrell (2019) gives an account of the other supported options.

This may be a suitable place to point out that what we did in the present study, namely, apply a single, common randomization of the datasets, leaves something to be desired. Ideally, one would repeat the whole procedure with $N \gg 1$ different randomizations of the data and report mean or median performance of the prediction methods. That will have to await a follow-up study.

## *4. SVM ranking methods*

Before proceeding to presentation and discussion of results it's necessary to say a little about SVM methods specifically geared to ranking, or in other words an ordinal dependent variable.

Ranking or ordinal regression is not built-in functionality in LIBSVM, but various methods have been proposed in the machine learning literature. The approach of Frank and Hall (2001) involves sequential application of a binary classifier. The idea (or perhaps a simplification thereof) is as follows: Is outcome $i$ predicted to be greater than zero or not? If the prediction is "not" we count it as a 0, otherwise we ask: is it greater than 1 or not? And so on, for each rank and each observation. This is simple, but not necessarily consistent. It could be, for example, that a case that evaluates as "not greater than zero" would nonetheless evaluate as "greater than 1" if it were taken to the next round. We tried the Frank and Hall method, but since (in our experiments) it was clearly dominated by that of Li and Lin (see below), the results are not reported here.

More systematic methods for ranking via SVM have been proposed by Li and Lin (2006) and Cardoso and Pinto da Costa (2007). Of these methods, which appear to be quite similar, we have chosen to implement in gretl that of Li and Lin, who made their LIBSVM-based code publicly available[4]. The gretl identifier for this SVM type is `C-rnk`. Li and Lin also propose some new kernels in addition to those in standard LIBSVM (see Lin and Li, 2008), and we have now included these in gretl's SVM implementation. Their "perceptron" kernel (gretl label, `perc`) seems to be particularly suited for effective ranking. This kernel has the advantage of being "scale-free": this means that

[4] See `http://www.work.caltech.edu/~htlin/program/libsvm/`.

there is one less hyperparameter to worry about in cross validation than with the commonly used RBF kernel, namely the scale factor $\gamma$. Search is therefore a good deal faster. For this reason gretl selects `perc` as the default kernel when the `C-rnk` ranking SVM is specified.

*5. Results*

Tables 2 and 3 summarize the results we obtained, for the red wines and white wines respectively. The first two rows per table show the econometric methods, Ordinary Least Squares and Ordered Probit; the last three show the SVMs: unordered classification, regression and ranking. (For the definition of "correct tails" see section 3 above.)

A first observation is that OLS and Ordered Probit did about equally well (OLS marginally superior on the red wines, OP marginally superior on the whites, except in respect of "correct tails"). A second is that all the SVM methods did better than the standard econometric methods. And a third is that Li and Lin's $C$-rnk SVM is the clear winner in both tables.

*Table 2. Results on testing data, red wines*

| method | % correct | correct tails | MAD |
|---|---|---|---|
| OLS | 60.60 | 13.98 | 0.428 |
| OP | 60.20 | 13.98 | 0.432 |
| $C$-SVC | 61.35 | 26.04 | 0.437 |
| $\epsilon$-SVR | 63.04 | 23.96 | 0.418 |
| $C$-rnk | 65.48 | 28.13 | 0.375 |

It would be a mistake to make over-confident claims on the basis of a single example, but here are some tentative conclusions with regard to the SVM variants.

First, SVM *classification* seems to be quite robust. It's certainly not always best for ordinal data (in this example it's never best) but it seems to reliably beat standard regression methods, even if narrowly. Second, SVR *regression*, while it does quite well on the red wines, is apparently less robust. In our example it comes out ahead of standard econometric methods but its perfor-

*Table 3. Results on testing data, white wines*

| method | % correct | correct tails | MAD |
|---|---|---|---|
| OLS | 51.07 | 15.90 | 0.541 |
| OP | 52.65 | 11.05 | 0.526 |
| $C$-SVC | 61.95 | 49.06 | 0.448 |
| $\epsilon$-SVR | 55.15 | 24.53 | 0.490 |
| $C$-rnk | 65.16 | 50.67 | 0.380 |

mance on the white wine data is disappointing compared with other SVMs; perhaps a wider or deeper hyperparameter search would help. Third, the Li and Lin *ranking* SVM, using their perceptron kernel, does an outstanding job of prediction on the wine quality data. On both the red and white wines it comes out top on all three criteria: percent correct overall, Mean Absolute Deviation, and percent correct in the tails of the distributions.

Suppose you were wondering: is there any advantage to prediction of a ranked outcome via SVM over prediction via standard econometric methods? Then we'd have to say our answer is Yes. And if you were wondering what's the best SVM method for this purpose, our provisional answer would be, that of Li and Lin.

*References*

Berwick R. (2003) An idiot's guide to support vector machines, MIT Department of Computational Linguistics, `http://www.svms.org/tutorials/Berwick2003.pdf`.

Cardoso J., Pinto da Costa J. (2007) Learning to classify ordinal data: the data replication method, *Journal of Machine Learning Research*, 8, 1393-1429.

Cortez P., Cerdeira A., Almeida F., Matos T., Reis J. (2009) Modeling wine preferences by data mining from physicochemical properties, *Decision Support Systems*, 47, 547-553.

Cottrell A. (2019) *Gretl + SVM*, gretl documentation at `https://sf.net/projects/gretl/files/manual/`.

Cottrell A. (2019a) *Gretl + MPI*, gretl documentation at `https://sf.net/projects/gretl/files/manual/`.

Frank E., Hall M. (2001) A simple approach to ordinal classification, in: De Raedt L., Flach P. (eds): *Machine Learning: ECML 2001*, Springer, 145-156.

Jordan M. (2004) The kernel trick, lecture notes, UC Berkeley Department of Electrical Engineering and Computer Sciences, `https://people.eecs.berkeley.edu/~jordan/courses/281B-spring04/lectures/lec3.pdf`

Li L., Lin H-T. (2006) Ordinal regression by extended binary classification, *NIPS'06 Proceedings of the 19th International Conference on Neural Information Processing Systems*, MIT Press, 865–872.

Lin H-T., Li L. (2008) Support vector machinery for infinite ensemble learning, *Journal of Machine Learning Research*, 9, 285-312.

Mullainathan S., Spiess J. (2017) Machine learning: an applied econometric approach, *Journal of Economic Perspectives*, 31, 87-106.

Smola A., Schlöpf B. (2004) A tutorial on support vector regression, *Statistics and Computing*, 14, 199-222.

Vapnik V. (1998) *Statistical Learning Theory*, New York, Wiley.

# Bootstrap inference in Instrumental Variables with gretl

Riccardo Ievoli*, Lucio Palazzo**

*Abstract:* In this paper we implement bootstrap in conventional instrumental variables estimations using gretl and compare the performance of several methods through simulation study and an empirical dataset coming from the seminal paper of Acemoglu et al. (2001).

*Keywords:* Instrumental Variables, Bootstrap methods

## *1. Introduction*

The main purpose of this work is to introduce bootstrap in Instrumental Variable (IV) linear models using gretl, in order to provide a fast and useful toolkit for economists and practitioners. Bootstrap methods can help to improve inference in IV estimation, conducted through conventional Two Stage Least Square (TSLS) estimator and associated t-test. Four types of bootstrap are implemented in order to obtain confidence sets and p-values associated to the t/Wald statistic for the null hypothesis $H_0 : \beta = 0$. The first method consists in resampling directly rows of the data, while three others are based on the resampled residuals and original estimates of the parameters.
The paper is organized as follows: bootstrap methods in IV are summarized in Section 2 while small-scale Monte Carlo exercise and Empirical application are respectively in Section 3 and Section 4. Finally, Section 5 presents some concluding remarks.

## *2. Bootstrap Methods in Instrumental Variables*

In this Section we discuss non–parametric, parametric and semi–parametric bootstraps for IV linear models. We focus on a linear model with only one

*University of Ferrara, riccardo.ievoli@unife.it
**University of Salerno, l.palazzo@unisa.it

endogenous regressor:

$$\mathbf{y} = \mathbf{x}\beta + \mathbf{u}; \quad (1)$$
$$\mathbf{x} = \mathbf{Z}\boldsymbol{\pi} + \boldsymbol{v}, \quad (2)$$

where $\mathbf{y}$ is the $n \times 1$ vector of the outcome variable, $\mathbf{x}$ is an endogenous regressor, $\mathbf{Z}$ is a $n \times k$ matrix of $k \geq 1$ instruments where $k$ is fixed, $\beta$ represents the scalar parameter of interest and vector $\boldsymbol{\pi}$ includes $k$ nuisance parameters measuring the effects of instruments on the endogenous explanatory variable. Errors are assumed to be $(u_i, v_i)' \sim \text{iid}(\mathbf{0}, \boldsymbol{\Sigma})$, $\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_u^2 & \rho\sigma_u\sigma_v \\ . & \sigma_v^2 \end{pmatrix}$, where $\rho = Cor(u_i, v_i)$ measures the level of endogeneity. Instruments are deemed relevant and exogenous, i.e. $E(Z_i u_i) = 0 = E(Z_i v_i)$, $E(Z_i x_i) \neq 0$. Thus, conventional TSLS estimator is defined as:

$$\hat{\beta}_n^{TSLS} = \left(\mathbf{x}' P_Z \mathbf{x}\right)^{-1} \left(\mathbf{x}' P_Z \mathbf{y}\right); \quad \text{where} \quad P_Z = \mathbf{Z}(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}', \quad (3)$$

and is simplified in the so called IV estimator when $k = 1$, expressed as:

$$\hat{\beta}_n^{IV} = \frac{\sum_{i=1}^n y_i z_i}{\sum_{i=1}^n x_i z_i}. \quad (4)$$

First stage coefficients in $\boldsymbol{\pi}$ are estimated through Ordinary Least Squares (OLS), i.e. $\hat{\pi}_n = (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{x}$. Model introduced in equations (1), (2) and also estimator in (3) and (4) can be generalized in order to include more than one endogenous regressors and control (exogenous) variables (see also Stock et al., 2002) in both equations.

The first bootstrap applied in IV setting is non–parametric and is denoted as "pair bootstrap" (Freedman, 1984) which consists in sampling (with replacement) the rows of original data matrix: $D_n = (\mathbf{y}, \mathbf{x}, \mathbf{Z})'$. After this resampling, the bootstrap counterpart of TSLS becomes:

$$\hat{\beta}_T^{TSLS*} = \left(\mathbf{x}^{*\prime} P_{Z^*} \mathbf{x}^*\right)^{-1} \mathbf{x}^{*\prime} P_{Z^*} \mathbf{y}^*, \quad \text{where} \quad P_{Z^*} = \mathbf{Z}^*(\mathbf{Z}^{*\prime}\mathbf{Z}^*)^{-1}\mathbf{Z}^{*\prime}. \quad (5)$$

As pointed out by Flores-Lagunes (2007), this method must be slightly modified to avoid a particular issue; basically, pair bootstrap does not guarantee

orthogonality between the TSLS residuals and the instruments.

Therefore, we consider two methods based on the combination of bootstrapped residuals, denoted with the symbols $(u_i^*, v_i^*)'$, and original estimates of the parameters, i.e. $\hat{\beta}_n$ and $\hat{\pi}_n$. Bootstrapped data $D_n^* = (y_i^*, x_i^*)$ can be constructed as:

$$y_i^* = x_i^* \hat{\beta}_n + u_i^*; \qquad x_i^* = Z_i \hat{\pi}_n + v_i^*,$$

and the bootstrapped TSLS estimator becomes:

$$\hat{\beta}_n^{TSLS*} = (\mathbf{x}^{*\prime} P_Z \mathbf{x}^*)(\mathbf{x}^{*\prime} P_Z \mathbf{y}^*). \tag{6}$$

Two different techniques can be used to obtain $(u_i^*, v_i^*)'$; the so called parametric bootstrap is based on quantities sampled from the following distribution:

$$\begin{pmatrix} u_i^* \\ v_i^* \end{pmatrix} \sim NIID\left(0, \widehat{\Sigma}\right); \quad \text{where} \quad \widehat{\Sigma} = \begin{bmatrix} \hat{\sigma}_u^2 & \hat{\sigma}_{uv} \\ \hat{\sigma}_{uv} & \hat{\sigma}_u^2 \end{bmatrix}.$$

Furthermore, in the semi-parametric residual bootstrap, residuals are directly sampled from their Empirical Distribution Function (EDF):

$$(u_i^*, v_i^*)' \sim \text{EDF}(\hat{u}_i, \hat{v}_i)', \quad \text{where} \quad \hat{u}_i = y_i - \hat{\beta}_n x_i; \quad \hat{v}_i = x_i - Z_i \hat{\pi}_n,$$

and are rescaled to have mean equal to zero[1] and being *orthogonal* to the instruments.

Between presented methods, only the pair bootstrap is considered valid even if the disturbances are not (jointly) homoskedastic. Moreover, residual bootstrap could be modified with the so called "wild" residuals, based on a transformation of the disturbances (Wu, 1986). The method consists in resampling (with replacement) from the following distribution:

$$(u_i^*, v_i^*)' \sim \text{EDF}(\hat{u}_i \xi_i, \hat{v}_i \xi_i)', \qquad \text{where } E(\xi_i) = 0 \text{ and } V(\xi_i) = 1.$$

We use the Rademacher distribution, i.e. $\xi_i = \{-1, 1\}$ with both probabilities

[1] We remark that including a constant term in both equations ensures that $\bar{u} = \bar{v} = 0$

equal to 1/2, but there are other possible choices.
Previous methods may be straightforward applied in hypothesis testing without imposing the null in bootstrap data generating process (DGP); we consider bootstrap counterpart of t/Wald statistic for the null hypothesis $H_0 : \beta = 0$, expressed as follows:

$$\tau_n^* = \sqrt{n}\hat{\omega}_n^{*-1}(\hat{\beta}_n^* - \hat{\beta}_n), \tag{7}$$

where $\hat{\omega}^*/\sqrt{n}$ is the bootstrap counterpart of $\hat{\omega}/\sqrt{n}$, i.e. the standard error of $\hat{\beta}_n$ expressed in (3) or (4). The associated bootstrap p–value, based on the assumptions that limiting distribution of $\tau_n^*$ is symmetric, is computed as:

$$p^* = B^{-1}\sum_{b=1}^{B}(\left|\tau_{n,b}^*\right| \geq |\tau_n|), \tag{8}$$

representing the proportion of bootstrap statistics greater than $\tau_n = n^{1/2}\hat{\omega}_n^{-1}\hat{\beta}_n$. The null is safely rejected if $p^* \leq \alpha$, where $\alpha$ is the I type error level.

Moreover, bootstrapped residuals can be also modified in order to impose the null hypothesis in the bootstrap DGP, setting $\beta = 0$, as follows:

$$\hat{u}_{0,i} = y_i - 0 \cdot x_i; \quad \hat{v}_i = x_i - Z_i\hat{\pi}_n,$$

where $\hat{u}_{0,i}$ is the vector of residuals from the structural equation, induced by imposing the null hypothesis. Hence, bootstrap series are re–constructed using sampled residuals from: $(u_i^*, v_i^*)' \sim \text{EDF}(\hat{u}_{0,i}, \hat{v}_i)'$, and bootstrapped data $D_n^*$: $x_i^* = Z_i\hat{\pi}_n + v_i^*; y_i^* = u_i^*$. Thus, the bootstrap counterpart of the t–statistic is computed in the following way:

$$\tau_{0,n}^* = \sqrt{n}\hat{\omega}_{0,n}^{*-1}\hat{\beta}_{0,n}^* \tag{9}$$

where $\hat{\beta}_{0,n}^*$ and its standard error $n^{-1/2}\hat{\omega}_{0,n}^*$ are estimated through TSLS. Bootstrap p–value is computed as in expression (8) using $\tau_{0,n}^*$ instead of $\tau_n^*$. This method is also called *restricted* residual bootstrap and can be valid in presence of heteroskedasticity applying "wild" residuals in the bootstrap DGP.

We also consider two types of bootstrap-based confidence sets having con-

fidence level of $1-\alpha$: the first is called *percentile* and is computed as:

$$CI^*_{P,1-\alpha} = (\hat{\beta}^*_{n,\alpha/2}, \hat{\beta}^*_{n,1-\alpha/2}), \tag{10}$$

where $\hat{\beta}^*_{n,j}$ represent the $j$–percentile obtained through $B$ replication of bootstrapped TSLS. The latter confidence interval is denoted as *t–boot* and it is expressed as follows:

$$CI^*_{t,1-\alpha} = (\hat{\beta}_n - \tau^*_{n,\alpha/2} \cdot \hat{\omega}_n/\sqrt{n}, \hat{\beta}_n - \tau^*_{n,1-\alpha/2} \cdot \hat{\omega}/\sqrt{n}), \tag{11}$$

where $\tau^*_{n,\alpha}$ and $\tau^*_{n,1-\alpha}$ are the estimated $\alpha/2$ and $1-\alpha/2$ quantiles of the distribution of $\tau^*_n$, obtained through $B$ replications, previously defined in expression (7).

## *3. Simulation Study*

In this section we develop a small–scale Monte Carlo exercise considering just–identified IV model to observe performance of proposed methods in terms of empirical size of bootstrapped t–statistic and coverage of bootstrapped confidence intervals introduced in Section 2. We generate $1000$ dataset of size $n = (50, 100)$ where $y_i = \beta x_i + u_i;\ x_i = \pi z_i + v_i$. The value of $\beta$ is set equal to zero and first stage coefficient is a function of the first stage $R^2_f$: $\pi = \sqrt{R^2_f(1-R^2_f)^{-1}}$, where $R^2_f = \{0.1, 0.5\}$ represents two scenarios in terms correlation between instrument and regressor. The $z_i$ is drawn from $N(0,1)$ and errors are generated through following probability law: $(u_i, v_i)' \sim N_2(0, \mathbf{\Sigma})$. $\mathbf{\Sigma} = \begin{pmatrix} 1 & \rho \\ \cdot & 1 \end{pmatrix}$. Parameter $\rho$ takes values equal to $\{0.5, 0.9\}$, related to moderate and high endogeneity for the explanatory variable. Number of bootstrap replications is $B = 399$ and nominal level is $\alpha = 0.05$. We use `tsls` function and random number generator from gretl. All the analysis are developed using a I7 processor with RAM$= 16.0$ GB. Since we generate homoskedastic and uncorrelated errors we do not apply wild bootstrap methods in this simulation study.

Table 3 contains results regarding empirical size of t/Wald statistics; consider four different combinations of first stage goodness of fit $R^2_f$ and endogeneity

$\rho$. Bootstrapped methods always perform better than the asymptotic approximation, especially with small sample ($n = 50$) and when instrument is poorly correlated with endogenous regressor ($R_f^2 = 0.1$). Regarding residual bootstrap, *restricted* ($\text{Res}_r$) presents better size than *unrestricted* ($\text{Res}_u$) when $R_f^2$ is low. In general, parametric (also due to our DGP) and residual methods outperform the pair one. Table 2 shows coverage of confidence intervals; *percentile* and *t-boot* intervals are indicated respectively with "P" and "t". Moreover, bootstrapped confidence intervals can even perform worse than the asymptotic one in some scenarios. In particular, t–boot confidence sets may perform badly in case of poor correlation between instrument and endogenous regressor, while Percentile method associated to Pair bootstrap presents good coverage even if $R_f^2 = 0.1$.

Finally, a slightly analysis of computational costs is summarized in Figure 1 for a dataset of $n = 200$ obtained through the same simulation design. Parametric bootstrap is the fastest method among the proposed ones, while pair and residual bootstrap present similar computational costs in terms of seconds. Residual wild bootstrap remains little bit computational demanding, exceeding 1 second for a number of bootstrap replications greater than $B = 800$. We remark that computational costs can dramatically increase in overidentified models, when $k > 1$.

## *4. Empirical Application*

The seminal paper of Acemoglu et al. (2001) is a well-known example of instrumental variable estimation in Political Economics. The sample size[2] consists of $n = 64$ ex–European colonies while the outcome variable of interest is the logarithm of income per capita in 1995 (on the purchasing power parity basis), and the deemed endogenous regressor is an averaged index of risk protection against government appropriation of assets between 1985 and 1995. Authors suggest to use, as instrument, the logarithm of mortality referring to European settlers during the colonization period. The simplest model,

[2] date are free available at `https://economics.mit.edu/faculty/acemoglu/data/ajr2001`

*Table 1. Empirical size of t/Wald test for the null $\beta = 0$*

$n = 50$

| $R^2$ | $\rho$ | Asymptotic | Parametric | Pair | $\text{Res}_u$ | $\text{Res}_r$ |
|---|---|---|---|---|---|---|
| 0.1 | 0.5 | 0.071 | 0.044 | 0.077 | 0.074 | 0.056 |
| 0.1 | 0.9 | 0.128 | 0.066 | 0.105 | 0.100 | 0.068 |
| 0.5 | 0.5 | 0.068 | 0.050 | 0.058 | 0.067 | 0.060 |
| 0.5 | 0.9 | 0.079 | 0.043 | 0.065 | 0.049 | 0.062 |

$n = 100$

| $R^2$ | $\rho$ | Asympt. | Param. | Pair | $\text{Res}_u$ | $\text{Res}_r$ |
|---|---|---|---|---|---|---|
| 0.1 | 0.5 | 0.075 | 0.051 | 0.089 | 0.059 | 0.044 |
| 0.1 | 0.9 | 0.107 | 0.061 | 0.069 | 0.075 | 0.060 |
| 0.5 | 0.5 | 0.059 | 0.032 | 0.057 | 0.049 | 0.048 |
| 0.5 | 0.9 | 0.074 | 0.047 | 0.068 | 0.042 | 0.042 |

without any control variable, takes the following specification:

$$y_i = \alpha + \beta x_i + u_i; \qquad x_i = \tau + \pi z_i + v_i$$

where $y_i$ is the logarithm of DGP, $x_i$ and $z_i$ are respectively the Risk index and the logarithm of mortality previoulsy discussed, $\beta$ is the scalar parameter of interest and $\pi$ is associated to the instrument. OLS estimate is $\beta_n^{OLS} = 0.52$ (standard error is 0.061), but there are arguments suggesting endogeneity of this explanatory variable (Durbin Wu Hausman *exogeneity* test is equal to 22.24, p.value $< 0.001$). First stage results, $\hat{\pi}_n = -0.607(0.12)$, $F = 22.4$ and $R_f^2 = 0.27$, show that proposed instrument is relevant, especially because F statistic exceeds both the critical values of Stock and Yogo (2005) and the empirical threshold of 10. The estimates of $\beta$ is $\hat{\beta}_n^{IV} = 0.945$ (0.157) and asymptotic confidence interval is $(0.637; 1.25)$. We run a high number of bootstrap replications $B = 99\,999$ and present results in Table 1 applying methods of Section 2. We notice that percentile confidence intervals obtained from the pair bootstrap are wider compared to other methods, while they seem comparable in terms of length and substantially differs from those obtained with conventional asymptotic approximation. In fact, Figure 1 shows that the

*Table 2. Coverage of confidence intervals (95%)*

| $n = 50$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $R^2$ | $\rho$ | Asympt. | Par$_P$ | Par$_t$ | Pair$_P$ | Pair$_t$ | Res$_P$ | Res$_t$ |
| 0.1 | 0.5 | 0.960 | 0.990 | 0.936 | 0.959 | 0.752 | 0.968 | 0.939 |
| 0.1 | 0.9 | 0.907 | 0.978 | 0.763 | 0.973 | 0.732 | 0.965 | 0.798 |
| 0.5 | 0.5 | 0.954 | 0.950 | 0.940 | 0.932 | 0.907 | 0.911 | 0.948 |
| 0.5 | 0.9 | 0.947 | 0.931 | 0.930 | 0.933 | 0.899 | 0.879 | 0.932 |
| $n = 100$ | | | | | | | | |
| $R^2$ | $\rho$ | Asympt. | Par$_P$ | Par$_t$ | Pair$_P$ | Pair$_t$ | Res$_P$ | Res$_t$ |
| 0.1 | 0.5 | 0.959 | 0.980 | 0.919 | 0.953 | 0.813 | 0.964 | 0.927 |
| 0.1 | 0.9 | 0.939 | 0.945 | 0.803 | 0.972 | 0.794 | 0.934 | 0.817 |
| 0.5 | 0.5 | 0.948 | 0.947 | 0.943 | 0.944 | 0.935 | 0.937 | 0.954 |
| 0.5 | 0.9 | 0.947 | 0.938 | 0.939 | 0.945 | 0.941 | 0.925 | 0.947 |

distribution of bootstrap estimator$(\hat{\beta}^{IV*}_{n,1}, \ldots, \hat{\beta}^{IV*}_{n,B})$, obtained through residual method, presents some (positive) skewness. The p-value of t-test is minor than 0.001 for all proposed bootstrap methods, confirming that $\beta$ is statistically different from zero. In terms of computational costs, we report time (in seconds) to compute bootstrap p–value of $\tau^*_n$ (CPU) statistics and this value is less than 30 seconds except for the wild bootstrap case, consistently with results of Section 3.

*Table 3. Bootstrap Inference for Acemoglu et al. (2001)*

| Method | $CI^*_{P,0.95}$ | $CI^*_{t,0.95}$ | p–val$(\tau^*_n)$ | p–val$(\tau^*_{0,n})$ | CPU(s) |
|---|---|---|---|---|---|
| Par.B | (0.680; 1.384) | (0.716; 1.328) | 0.00004 | / | 12.3 |
| Pair.B | (0.695; 1.641) | (0.739; 1.402) | 0.00013 | / | 21.1 |
| Res.B | (0.709; 1.401) | (0.736; 1.341) | 0.00003 | 0.0000 | 26.7 |
| Wild.B | (0.708; 1.403) | (0.738; 1,342) | 0.00010 | 0.0000 | 36.6 |

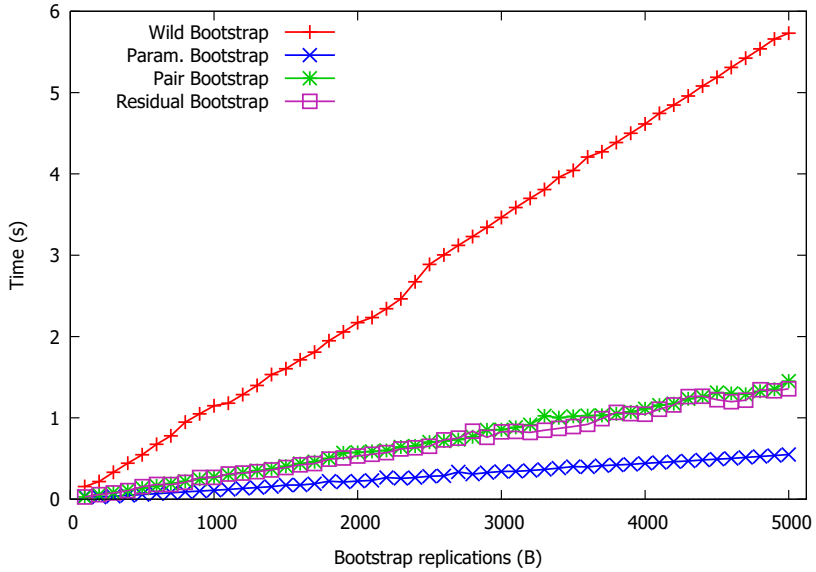



*Figure 1. Computational Time of different bootstraps* ($n = 200$)

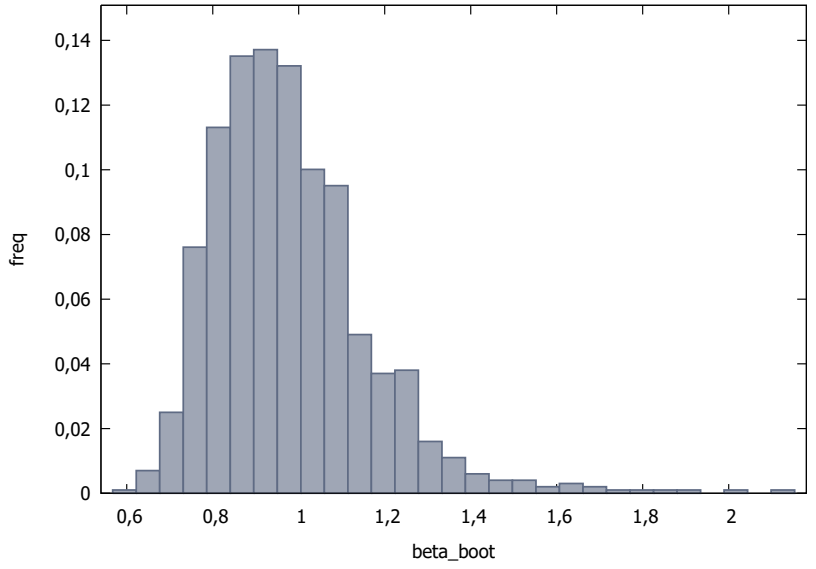



*Figure 2. Histogram for* $\hat{\beta}_n^{IV*}$ *(residual bootstrap)*

## *5. Concluding Remarks*

In this work we implement bootstrap techniques for IV estimation in *Gretl*. These methods can perform better than asymptotic approximation when the simple size is relatively small, improving especially the empirical size of t/Wald test. Presented methods can be immediately adapted to partially robust LIML estimator using function `liml` instead of tsls. They can also be applied in presence of $m > 1$ and in case of control variables in both equations. We remark that bootstrap may produce misleading results, performing even worse than conventional asymptotic approximation, when instruments are poorly correlated with the endogenous regressors (often denoted as "weak"), when overidentification is severe and when instruments are not completely *exogenous*. To overcome some of these issues, Davidson and Mackinnon (2010) introduced new residual–based bootstrap methods producing better performance under weak instruments, while Wang and Kaffo (2016) developed a new bootstrap method, valid even under many instruments. We leave implementation of these robust methods for further researches.

## *References*

Acemoglu D., Johnson S., Robinson, J.A. (2001) The colonial origins of comparative development: An empirical investigation, *American economic review*, 91(5), 1369-1401.

Davidson R., MacKinnon J. G. (2010) Wild bootstrap tests for IV regression, *Journal of Business & Economic Statistics*, 28(1), 128-144.

Freedman D. (1984) On bootstrapping two-stage least-squares estimates in stationary linear models, *The Annals of Statistics*, 12(3), 827-842.

Flores-Lagunes A. (2007) Finite sample evidence of IV estimators under weak instruments, *Journal of Applied Econometrics*, 22(3), 677-694.

Stock J.H., Wright J.H., Yogo M. (2002) A survey of weak instruments and weak identification in generalized method of moments, *Journal of Business & Economic Statistics*, 20(4), 518-529.

Stock J.H., Yogo M. (2005) Testing for weak instruments in linear IV regression, in Andrews D.W.K., Stock J.H. (eds.), *Identification and Inference for Econometric Models: Essays in Honor of Thomas J. Rothenberg.*, Cambridge University Press, 80-108.

Wang W., Kaffo M. (2016) Bootstrap inference for instrumental variable models with many weak instruments, *Journal of Econometrics*, 192(1), 231-268.

Wu C.F.J. (1986) Jackknife, bootstrap and other resampling methods in regression analysis, *the Annals of Statistics*, 14(4), 1261-1295.

# BMA for the GLM in gretl

Riccardo Lucchetti*, Luca Pedini**

*Abstract:* In this article, we lay down the foundations for a gretl function package that provides Bayesian Model Averaging (BMA) for the Generalised Linear Model via a sampling technique known as Reversible Jump Markov Chain Monte Carlo (RJMCMC). Implementation issues are discussed, with a view to ensuring maximum efficiency when the model space is large, particularly so by exploiting as much as possible the possibility of parallelising the algorithm vis the Message Passing Interface (MPI) standard. Apart from bringing about a substantial reduction in computation time, parallelisation also provides an effective way of checking whether convergence of the Markov chain has occurred.

*Keywords:* MPI, BMA, Generalised linear model.

## *1. Introduction*

As well known, the Generalised Linear Model (GLM) is a statistical framework that includes as special cases several models widely used in econometric practice. The distribution of dependent variable is assumed to belong to the exponential family, with density function $f(y)$:

$$f(y_i) = \exp\left[\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi)\right]$$

and that the conditional expectation of $y_i$ given a set of $k$ covariates $x_i$, $\mathrm{E}\left[y_i|x_i\right]$ is a continuous transform of a linear combination:

$$l(\mathrm{E}\left[y_i|x_i\right])) = \eta_i = x_i'\beta \tag{1}$$

*Università Politecnica delle Marche, r.lucchetti@univpm.it
**Università Politecnica delle Marche, l.pedini@pm.univpm.it

where $l(\cdot)$ is known as the *link function*[1]. Maximum likelihood estimation of GLMs can be carried out, in a frequentist framework, via Iterative Weighted Least Squares on the transformed variable $z_i = \eta_i + (y_i - \mu_i)\frac{\partial \eta_i}{\partial \mu_i}$, where the weights $w_i$ are defined as:

$$w_i = \left[ \frac{\partial^2 b(\eta_i)}{\partial \eta_i^2} \left( \frac{\partial \eta_i}{\partial \mu_i} \right)^2 \right]^{-1}$$

The above was adapted in Gamerman (1997) to a Bayesian setup: by using a normal prior on $\beta \sim N(m_0, V_0)$, then the following sampling scheme can be used:

1. Set as initialization $\beta_0$;

2. At the *i-th* iteration, draw $\beta^{(i)}$ from the proposal density $q(\beta|\beta^{(i-1)}) = N(m^{(i)}, V^{(i)})$, where:

$$V^{(i)} = (V_0^{-1} + X'W_{i-1}X)^{-1} \tag{2}$$
$$m^{(i)} = V^{(i)}(V_0^{-1}m_0 + X'W_{i-1}z_{i-1}); \tag{3}$$

3. Accept the new draw with probability $\alpha(\beta^{(i-1)}, \beta^{(i)})$, defined via a standard Metropolis-Hastings scheme as:

$$\alpha(\beta^{(i-1)}, \beta^{(i)}) = \min\left[ \frac{f(\beta^{(i)}|y)q(\beta^{(i-1)}|\beta^{(i)})}{f(\beta^{(i-1)}|y)q(\beta^{(i)}|\beta^{(i-1)})}; 1 \right]$$

   where $f(\beta^{(i)}|y) \propto f(y|\beta^{(i)})f(\beta^{(i)})$ and $q(\beta^{(i)}|\beta^{(i-1)})$ is a normal density evaluated at $\beta^{(i)}$ with mean and variance, respectively, equal to (3) and (2).

The algorithm above makes it possible to sample from the posterior density of $\beta$ and study it via simulation.

[1] Special cases of $l(\cdot)$ include, for example, the identity function for OLS or the natural logarithm for the Poisson regression model.

This extension makes it possible to adapt Bayesian Model Averaging (BMA) to the GLM via a generalisation of the canonical Metropolis-Hastings approach.[2] Given a set of candidate models $M_1, M_2, \ldots, M_m$, the main object of BMA is the computation of the mixture distribution:

$$\mathrm{P}\left[\beta|y\right] = \sum_{i=1}^{m} \mathrm{P}\left[\beta|M_i, y\right] \mathrm{P}\left[M_i|y\right]$$

and its related moments $\mathrm{E}\left[\beta|y\right]$ and $\mathrm{V}\left[\beta|y\right]$, where $\mathrm{P}\left[\beta|y\right]$ and $\mathrm{P}\left[\beta|M_i, y\right]$ represent respectively the marginalised (over models) and the model specific posterior distribution of the parameter of interest $\beta$, whereas $\mathrm{P}\left[M_i|y\right]$ the posterior model probability.

Unlike linear models, however, no analytical solution exists for both the parameter and model posteriors under the standard prior set-up, leading to the impossibility of applying common MCMC-based BMA[3].

## 2. *Reversible Jump Markov Chain Monte Carlo*

The solution we adopt here is based on the usage of Reversible Jump Markov Chain Monte Carlo (RJMCMC) proposed by Green (1995), which is a modification of the MCMC technique for model exploration where parameters and models are sampled jointly.

In a Metropolis-Hastings MCMC sampler the parameter of interest drawn across iterations has fixed dimension: in case of standard MCMC BMA, for instance, the parameter is a point in the $k$-dimensional lattice on $\{0, 1\}$, representing a generic model $M_i$ whose entries are proxies for variable inclusion (if 1) or exclusion (if 0), given $k$ potential covariates.

RJMCMC, instead, generalises Metropolis-Hastings by allowing to sample parameters with different dimensions, in our case the couple $(M_i, \beta_i)$, where $M_i$ is again a point in the lattice, but $\beta_i$ are model specific parameters whose dimension varies.

[2] Implementation of BMA for linear models in gretl is discussed in Błażejowski (2015).

[3] Common practice involves the use of maximum-likelihood estimators as proxies for posterior moments, and the use of Laplace or BIC approximations for posterior model distributions, which is questionable; see for example Amini *et al.* (2011).

In particular, RJMCMC can be briefly described as follows: at each step, a specification is proposed from a model transitional kernel; the related parameters are not sampled directly from a proposal distribution, but taken from previous step via an *ad-hoc* function. Since the size of the parameter vector may differ between iterations, a *matching* variable is used in the following way[4]: the transformation function from $(\beta_i, M_i)$ to $(\beta_j, M_j) = g(\beta_i, M_i)$ is

$$\beta_j = g(\beta_i, M_i, u_i) = \mu_j + B_j \upsilon \tag{4}$$

where $B$ is the Cholesky factor of $V_j$, $\mu_j$ and $V_j$ the mean and variance of $\beta_j$ and $\upsilon$ is defined as:

$$\upsilon = \begin{cases} [RB_i^{-1}(\beta_i - \mu_i)]^{k_j} & \text{if } k_j < k_i \\ RB_i^{-1}(\beta_i - \mu_i) & \text{if } k_j = k_i \\ R \begin{pmatrix} B_i^{-1}(\beta_i - \mu_i) \\ u \end{pmatrix} & \text{if } k_j > k_i \end{cases}$$

with $k$ as the number of variables, $R$ a random permutation matrix; the notation $[...]^{k_j}$ indicates the first $k_j$ elements of the vector and finally $u$, a $k_j - k_i$ vector of random numbers with density $f()$, (usually, a standard normal variate).

The parameter $\beta$ is taken as a multivariate normal, which gets standardised first, and then corrected via the mean and covariance matrix of the new model, where equations (3) and (2) can be used for the posterior moment estimation. The distributional choice may be objectionable, but Green (2003) argues that this choice is a good compromise between efficiency of the chain and simplicity.

The probability of acceptance is:

$$\rho = \min \left[ \frac{\mathrm{P}\left[\beta_j, M_j | y\right] q(M_i | M_j)}{\mathrm{P}\left[\beta_i, M_i | y\right] q(M_j | M_i)} \frac{|B_j|}{|B_i|} G; 1 \right] \tag{5}$$

with $\mathrm{P}\left[\beta_j, M_j | y\right]$ as the joint posterior distribution for the couple $(\beta_j, M_j)$

[4] See, among others, Green (2003), Green (2009), Lamnisos *et al.* (2009), Lamnisos *et al.* (2013).

and $q(M_j|M_i)$ as the model transitional kernel, where we implicitly assume its independence from the sampling of $\beta$, and:

$$G = \begin{cases} f(u) & \text{if } k_j < k_i \\ 1 & \text{if } k_j = k_i \\ f(u)^{-1} & \text{if } k_j > k_i \end{cases}$$

As for the prior distributions, we assume normality for the $\beta$ parameters,

$$\beta_i|M_i \sim N(\mu_{0,i}, V_{0,i})$$

while for the model prior we use a Binomial distribution:

$$P(M_i) = \prod_{j=1}^{k} \pi_j^{\delta_{ij}} (1 - \pi_j)^{1-\delta_{ij}}$$

where given $k$ total variables, $0 \leq \pi_j \leq 1$ is the prior probability that the *j-th* variable is significant and $\delta_{ij}$ is an indicator of the variable inclusion.

The algorithm can be summarised as follows (see Lamnisos *et al.*, 2013 ):

1. Set the initial $\beta_i$ for the model $M_i$, (usually, the full specification).
2. Propose a new model $M_j$ from a transitional kernel $q(M_j|M_i)$ and compute its $\beta_j$ as in (4);
3. Accept the move with probability (5), otherwise propose a resampling of $\beta_i$ in $M_i$ via one iteration of the Gamerman procedure.
4. Repeat from 2 until convergence.

## *3. Parallelisation*

The issues in parallelising MCMC algorithms are well known: parallelisation leads to a speed-up in sampling, as long as the convergence rate is fast and the proportion of burn-in period is small compared to the total amount of iterations.

A plausible guideline is provided by Gelman and Rudin (1992), Brooks and Gelman (1998), who introduce some helpful indices for monitoring the convergence rate of each chain, and the possible advantages of using additional CPUs.

However, parallelisation could also be helpful for the exploration of the parameter space: for example, in the case of a multimodal distribution, a single chain can get stuck in a local maximum, thereby leaving other regions of the parameter unexplored unless the number of iterations is enormous; in these cases, running the MCMC algorithm in parallel instances (possibly with different starting points) can be an extremely effective technique.

This aspect is quantitatively analysed in Gelman and Rubin (1992) and Brooks and Gelman (1998). Gelman and Rubin (1992) analyse the scenario when a univariate random variable $x$ is simulated on in $i = 1 \dots c$ cores over $j = 1 \dots n$ times, together with $\bar{X}$ as an unbiased estimator for $\mathrm{E}[x]$. The between variance $B/n$ and the within variance $W$ are defined as:

$$B = \frac{n}{c-1} \sum_{i=1}^{c} (\bar{X}_i - \bar{X})^2 \tag{6}$$

$$W = \frac{1}{c(n-1)} \sum_{i=1}^{c} \sum_{j=1}^{n} (x_{ji} - \bar{X}_i)^2 \tag{7}$$

where $\bar{X}_i = \frac{1}{n} \sum_{j=1}^{N} x_{ji}$ and $\bar{X} = \frac{1}{c} \sum_{i=1}^{c} \bar{X}_i$, in obvious notation.

The parameter $\sigma^2$ can be estimated now via[5] $\hat{\sigma}^2 = \frac{n-1}{n} W + \frac{B}{n}$, so:

$$\hat{V} = \hat{\sigma}^2 + \frac{B}{cn} \tag{8}$$

The Gelman-Rubin convergence measure is given by:

$$\hat{R} = \frac{\hat{V}}{W}; \tag{9}$$

[5] A correction term due to the sampling variability of $\bar{X}$ is generally added, i.e. $\frac{B}{cn}$This is the variance of the sample mean $\bar{X}$, given by $\frac{1}{c}$ times the sample variance of $\bar{X}_i$, $B/n$.

where $\hat{R}$ is close to 1 upon convergence.

Equation (9) can be further improved by taking into account sampling variability in the variance estimates: Brooks and Gelman (1998) propose

$$\hat{R}_c = \frac{df + 3}{df + 1}\hat{R}$$

where $df = \frac{2\hat{V}}{Var(\hat{V})}$

Alternative, more robust, statistics are available, such as the so-called *interval-based* $\hat{R}$ (based on quantiles) and the *empirical central moment* $\hat{R}_s$ (based on higher-order moments).

An extension in a multivariate set-up is given in Brooks and Gelman (1998), where a multivariate generalisation of $\hat{R}$ is proposed: define the matrices

$$\mathbf{W} = \frac{1}{c(n-1)} \sum_{j=1}^{c} \sum_{i=1}^{n} (\mathbf{x}_{ij} - \bar{\mathbf{x}}_j)(\mathbf{x}_{ij} - \bar{\mathbf{x}}_j)'$$

$$\mathbf{B} = \frac{n}{c-1} \sum_{j=1}^{c} (\bar{\mathbf{x}}_j - \bar{\mathbf{x}})(\bar{\mathbf{x}}_j - \bar{\mathbf{x}})'$$

as multivariate versions of (6) and (7); then the new convergence statistics is:

$$\tilde{R} = \frac{n-1}{n} + \frac{c+1}{c}\lambda \tag{10}$$

where $\lambda$ is the maximum eigenvalue of $\mathbf{W}^{-1}\mathbf{B}/n$.

## *4. Implementation*

The RJMCMC is implemented as a gretl package named bmaglm, composed by several private functions which deal with specific part of the procedure and a main public one. The package provides the user a quite flexible choice of glm link functions, priors for both parameters and models and model transitional kernels.

In order to parallelise the Markov Chains, the MPI framework is used (see Snir *et al.* 1996); the strategy adopted for identifying and storing information

into sampled models is based on the representation of each model as a binary vector (a point in the lattice on $\{0, 1\}$), so there is a one-to-one correspondence between models and integer numbers.

In order to construct a compound gretl object whose elements are the sampled specifications, we use a bundle of bundles where each element of the bundle is a model whose key is the *hexadecimal* representation of the integer. The hexadecimal notation is adopted to circumvent potential issues related to integer precision when the model space is huge.

## *5. Empirical illustration*

In order to illustrate the package, we provide a simple example on the famous Mroz (1987) dataset: a Probit estimation on female labour force participation in 1975. The dataset contains information about 753 women, where our binary dependent variable is named `LFP`, which is equal to $1$ in case of labour participation; $0$ otherwise.

The set of regressors used is:

- `KL6`, the number of children under the age of 6;
- `WA`, wife's age;
- `WE`, wife's education attainments, in years;
- `HA`, husband's age;
- `HE`, husband's education attainments;
- `HW`, husband's hourly wage;
- `MTR`, marginal tax rate facing the wife;
- `UN`, unemployment rate in the country of residence;
- `CIT`, dummy variable - 1 if living in a a large city, 0 otherwise;
- `AX`, actual years of wife's previous labour experience.

The output of a standard Probit estimation is the following:

```
Model 1: Probit, using observations 1-753
Dependent variable: LFP
Standard errors based on Hessian

            coefficient   std. error      z      p-value
  ---------------------------------------------------------
  const       0.238379     0.0523348     4.555    5.24e-06 ***
  KL6        -0.813479     0.117719     -6.910    4.83e-12 ***
  WA         -0.0604316    0.0145616    -4.150    3.32e-05 ***
  WE          0.117676     0.0299991     3.923    8.76e-05 ***
  HA         -0.00364750   0.0141875    -0.2571   0.7971
  HE         -0.0522828    0.0235334    -2.222    0.0263   **
  HW         -0.0897814    0.0198192    -4.530    5.90e-06 ***
  MTR        -5.57322      1.04103      -5.354    8.62e-08 ***
  UN          0.00303028   0.0170686     0.1775   0.8591
  CIT         0.0555055    0.117783      0.4713   0.6375
  AX          0.0693860    0.00752985    9.215    3.12e-20 ***

Mean dependent var   0.568393   S.D. dependent var   0.495630
McFadden R-squared   0.239902   Adjusted R-squared   0.218537
Log-likelihood      -391.3541   Akaike criterion     804.7083
Schwarz criterion    855.5730   Hannan-Quinn         824.3039
```

Let us assume the following set up for the function: a diffuse prior on the constant term $\beta_0 \sim N(0, 100)$, a prior on the parameter defined as

$$\beta_i \sim N(\underline{0}, n(\tilde{X}_i{}' \tilde{X}_i)^{-1})$$

where $n$ is the total number of observations; $\tilde{X}_i$ is the matrix of demeaned regressors in model $M_i$ and a uniform prior distribution for models. The number of iterations and burn-in are set respectively to $100000$ and $10000$.

```
-------------------------------------------------------
Bayesian Model Averaging with Generalized Linear Model
-------------------------------------------------------
Overall sampling statistics
            mean        se       pip    c_mean      c_se
const    0.23710   0.05323   1.00000   0.23710   0.05323
  KL6   -0.80971   0.11833   1.00000  -0.80971   0.11833
   WA   -0.06165   0.00857   0.99981  -0.06166   0.00853
   WE    0.09412   0.03687   0.95603   0.09844   0.03156
   HA   -0.00011   0.00316   0.04889  -0.00230   0.01414
   HE   -0.01685   0.02706   0.34091  -0.04944   0.02318
   HW   -0.09045   0.01911   0.99998  -0.09045   0.01911
  MTR   -5.33488   1.04899   1.00000  -5.33488   1.04899
   UN    0.00034   0.00417   0.05039   0.00677   0.01737
  CIT    0.00214   0.02820   0.05861   0.03643   0.11099
   AX    0.07011   0.00759   1.00000   0.07011   0.00759
-----------------------------------
Best specifications (P>0.10):

Model_00000399: P(M|D)=0.527433
const KL6 WA WE HW MTR AX

Model_000003b9: P(M|D)=0.286444
const KL6 WA WE HE HW MTR AX
```

The columns `mean` and `se` identify the model averaging posterior mean and standard errors, whereas `pip` is the probability of inclusion, i.e. the number of times the covariate appears in model specifications. `c_mean` and `c_se` are, instead, the posterior mean and standard errors conditioned on inclusion.

Comparing the two outputs, it is possible to note how the most significant variables detected by the standard Probit model are identified by the BMA too, through the probability of inclusion. Not significant ones, instead, have their coefficients shrunk (according to the `pip`). The above top specifications identified, which account for most part of the posterior model probability, are also the ones recognized as best ones using Frequentist measures such as Information Criteria.

Finally, in order to ascertain the importance of parallelisation, the same

*Table 1. Parallelisation results*

| threads | c=1 | c=2 | c=4 | c=8 |
|---|---|---|---|---|
| Iterations per core | 100000 | 50000 | 25000 | 12500 |
| Elapsed time (sec) | 684.467 | 332.396 | 201.559 | 104.618 |
| BG statistic | | 1.030 | 1.014 | 1.021 |
| $P(M\|D)$ | | | | |
| Model 399 | 0.527 | 0.525 | 0.520 | 0.526 |
| Model 3b9 | 0.286 | 0.298 | 0.293 | 0.287 |

experiment is run splitting the MCMC iterations[6] across $2, 4$ and $8$ cores: results are displayed in Table 1. The multivariate Brook and Gelman statistics of convergence is provided alongside with the posterior model probability of best models: as can be seen, parallelisation does not affect negatively convergence as the statistics approaches $1$ in all the scenarios and the posterior model probabilities are very similar. As for CPU time, using $4$ and $8$ cores leads to a huge improvement, allowing for saving up to the $70\%$ and $80\%$, respectively, of the single-threaded case.

## *6. Conclusion*

Reversible Jump Markov Chain Monte Carlo is a valuable instrument for performing BMA in the GLM. An implementation in gretl is proposed here, with special focus on the computational aspects: the parallelisation of the Markov Chain algorithm leads to major improvements in both CPU time and exploration of model spaces.

[6] A flexible burn-in of $10\%$ is assumed.

## *References*


Amini S.M., Parmeter C.F. (2011) Bayesian model averaging in R, *Journal of Economic and Social Measurement*, 36, 253-287.

Błażejowski M., Kwiatkowski J. (2015). Bayesian model averaging and jointness measures for gretl, *Journal of Statistical Software*, 68, 1-24.

Brooks S.P. and Gelman, A. (1998) General methods for monitoring convergence of iterative simulations, *Journal of computational and graphical statistics*, 7, 434-455.

Gamerman D. (1997) Sampling from the posterior distribution in Generalized Linear Mixed Models, *Statistics and Computing*, 7, 57-68.

Gelman A., Rubin D.B. (1992) Inference from iterative simulation using multiple sequences, *Statistical science*, 7, 457-472.

Green P.J. (1995) Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, *Biometrika*, 82, 711-732.

Green P.J. (2003) Trans-dimensional Markov Chain Monte Carlo, *Oxford Statistical Science Series*, 179-198.

Green P. J., Hastie D.I. (2009) Reversible jump MCMC, *Genetics*, 155, 1391-1403.

Lamnisos D., Griffin J.E., Steel M.F. (2009) Transdimensional sampling algorithms for Bayesian variable selection in classification problems with many more variables than observations, *Journal of Computational and Graphical Statistics*, 18, 592-612.

Lamnisos D., Griffin J.E., Steel M.F. (2013) Adaptive Monte Carlo for Bayesian variable selection in regression models, *Journal of Computational and Graphical Statistics*, 22, 729-748.

Mroz T.A. (1987) The sensitivity of an empirical model of married women's hours of work to economic and statistical assumptions, *Econometrica*, 55, 765-799.

Snir M., Otto S.W., Huss-Lederman S., Walker D.W., Dongarra J. (1996) *MPI: The complete reference*, MIT Press, Cambridge USA.

# Model uncertainty in Propensity Score Matching

Riccardo Lucchetti*, Luca Pedini**, Claudia Pigini***

*Abstract:* Propensity Score Matching (PSM) is a popular approach to evaluate treatment effects in observational studies. While model selection for the PS estimation is often naive in practice, the choice of variables in this step is crucial because the related treatment effect estimate is highly dependent on it. We propose dealing with such *model uncertainty* by Bayesian Model Averaging (BMA) for the PS model and three different Model Averaging treatment effect estimators are proposed. We propose an empirical application based on the 2014 Italian tax credit reform (the so-called "Renzi bonus"). We show that model uncertainty importantly affects the estimated treatment effects and how the proposed BMA-based estimators help to drastically reduce it. Both BMA and PSM routines have been implemented in gretl.

*Keywords:* Propensity Score Matching, Bayesian Model Averaging, 2014 Italian tax credit reform.

## *1. Introduction*

Since the seminal paper of Rosenbaum and Rubin (1983), Propensity Score Matching (PSM) has become a standard methodology for treatment evaluation problems in observational studies, as it offers a plausible solution to compute the "counterfactual" given the treated units. It is based on building an untreated comparison group with observational characteristics $X$ close to the treated one, i.e. *matching* treated with untreated units.

In particular, treated and untreated individuals are matched on the conditional probability of being treated given $X$, i.e. the Propensity Score (PS) $p(X) = p(D = 1|X)$, where $D$ is the treatment assignment indicator. The PS is usually obtained by estimating probit or logit models.

The underlying binary choice model, however, is often built devoting little attention on the choice of explanatory variables: common routines perform

*Università Politecnica delle Marche, r.lucchetti@univpm.it
**Università Politecnica delle Marche, l.pedini@pm.univpm.it
***Università Politecnica delle Marche, c.pigini@univpm.it

the estimation using all available covariates (full model), but over-parametrization may lead to violations to PS assumptions (common support and balancing condition) as well as to a larger variance of the estimator. Parsimonious models, on the other hand, are equally problematic: omitting relevant variables can heavily affect the treatment effect estimator.

Clearly, a problem of *model uncertainty* arises because different treatment effects estimates can be obtained according to the possible specifications of the PS model. We tackle this issue by means of Model Averaging techniques: Bayesian Model Averaging (BMA) has proven to be the main solution to account for uncertainty in model selection in the literature, and here we propose three extension to PSM scenarios.

The proposed methodology is applied to evaluate the economic impact of the Italian tax credit reform (Decree Law 66/2014), which introduced a monthly wage increase of about 80€ for all employees with an annual gross income between 8145€ and 26000€. A tax reduction is supposed to encourage household consumption, however the effectiveness of tax credit policy is a debated topic in the literature (Shapiro and Slemrod 2003a,b; 2009) In particular, we replicate the approach proposed by Neri et al (2017), based on *Propensity Score Difference-in-Differences* estimation and, in doing so, we also show that PSM estimates can be quite sensitive to the specification of the PS model.

## *2. Model uncertainty in Propensity Score*

The idea of model uncertainty in PSM can be easily expressed as follows: with $k$ variables, $2^k$ binary models are available and we can assume that, among them, there is one model or a set of models, $M^*$, that are not known but reflect all of the desirable conditions and lead to different treatment effect estimates. We could then average model specific estimates according to a weight reflecting the model probability to be the closest to reality, with the benefit of avoiding any choice of a single specification, which can be seen as a guess about $M^*$.

With PSM being a "sequential" procedure, which starts with the PS computation, then involves the matching and finally computes the treatment effect,

we can directly attach the probability of a PS model to the related treatment effect estimate, so a simple example of a model averaging estimator is

$$\hat{\gamma} = \sum_{i}^{2^k} \hat{\gamma}_{M_i} \omega_{M_i}, \tag{1}$$

where $\hat{\gamma}$ and $\hat{\gamma}_{M_i}$ identify the model averaging treatment effect estimator and the model specific treatment effect estimator (related to the Propensity Score model $M_i$), respectively; $\omega_{M_i}$ is the weight attached to each model.

The Bayesian Model Averaging counterpart of equation (1) is

$$\hat{\gamma} = E(\gamma|y, X, D) = \sum_{i}^{2^k} E(\gamma|y, X_i, D, M_i) P(M_i|X_i, D), \tag{2}$$

where the model averaging treatment effect is substituted by the posterior mean of the parameter $\gamma$, $E(\gamma|y, X, D)$ with $y$ as the outcome variable, and $\hat{\gamma}_{M_i}$ by the related model specific posterior mean $E(\gamma|y, \mathbf{x}_i, D, M_i)$. Finally, the weight $\omega_{M_i}$ is now equal to $P(M_i|\mathbf{x}_i, D)$, which is the posterior model probability of the *i-th* PS model. Equation (2) represents a model averaging treatment effect estimator *tout court* and, depending on how $E(\gamma|y, \mathbf{x}_i, D, M_i)$ is computed, different estimators may be defined.

An alternative approach to equation (2) is the so-called *plug-in* estimator: PSM is performed in the usual manner (i.e. without explicitly averaging treatment effects across different models), but instead of computing the PS on simple probit or logit estimates $\hat{\beta}$ of the linear predictor $X\beta$, the model averaging posterior mean of the parameter of interest $\beta$ in the binary model is used. We define the model averaging posterior mean of $\beta$ as

$$E(\beta|y) = \sum_{i=1}^{2^k} E(\beta|y, M_i) P(M_i|y), \tag{3}$$

where $E(\beta|y, M_i)$ is the model specific posterior mean.

Since BMA is commonly concerned with linear models, its application to a binary specification is not obvious. Binary models pose the problem of not

analytical formulae for both $P(M_i|y)$ and $E(\beta|y, M_i)$, which make the *standard* BMA routines unfeasible. In order to solve the problem, a more general framework is requested, and the solution proposed exploits the Reversible Jump Markov Chain Monte Carlo (RJMCMC) design by Green (1995). In RJMCMC, the parameter of interest $\beta$ and the related model $M_i$ are sampled jointly, partially avoiding, in this way, the afore-mentioned problem of analytical formulae and leading to a more immediate and simpler computation of $P(M_i|y)$ and $E(\beta|y)$.

In summary, the Model Averaging treatment effect estimators here proposed are:

- "BMA mean", which identifies the plug-in estimator based on a propensity score computed via the above model averaging posterior mean;
- "BMA Frequentist", which is based on equation (2), where the posterior model probability are provided by the RJMCMC, whereas $E(\gamma|y, \mathbf{x}_i, D, M_i)$ is obtained via the standard treatment effect on the PS model $M_i$ using simple probit estimation;
- "BMA full", which exploits (2) as well, but computes $E(\gamma|y, \mathbf{x}_i, D, M_i)$ taking into account the different sampled PS (obtained from the corresponding parameter $\beta$ in the binary model) for each $M_i$, across the RJMCMC iterations.

### *3. Empirical illustration: the 2014 Italian tax credit reform*

Tax rebate policies are common stabilizing instruments applied by policymakers to reduce the impact of the business cycle, with the particular aim of inducing an increase in the propensity to consume.

The 2014 Italian tax credit was not an exception from this pattern: according to Government estimates, its introduction via Decree Law 66/2014 implied a total transfer of $5.9$€ billion to households, equal to $0.4\%$ of GDP. From a technical viewpoint, the reform introduced a monthly wage increase of about 80€ for all employees with an annual gross income between 8145€ and 26000€; eligibility was defined on an individual basis, so that households could have more than one member benefiting from the bonus.

The topic of tax rebates in the economic literature is, however, very debated. According to the *Life Cycle-Permanent Income Hypothesis*, any transitory policy has not effect on consumption. However, if *liquidity constraints* and behavioral factors such as "mental accounting" or "myopia" are considered, a positive effect may be recovered.

Empirical works are equally divergent: on the one hand, "time series" approaches based on structural break analyses by Modigliani et al. (1977), Blinder (1981), and Poterba (1988) show unclear effects of the policy; on the other hand a "micro-data" analysis approach focused on the treatment effect estimation recovers positive effects (Wilcox, 1989; Parker, 1999; Souleles, 2002; Johnson, 2006) but, at the same time, shows a high sensitivity to model specifications (Heim, 2007).

The analysis here proposed belongs to this second strand of literature and, following Neri et al (2017), a *Propensity Score Difference-in-Difference* (Brozowski, 2007; Stuart, 2014) approach is used to evaluate the effect of the 2014 tax credit on durables and non-durables consumption. The identification of the treatment effect with the Difference-in-Difference approach is based on the common trend assumption, which is often violated in practice.
In such contexts, PSM allows to build treatment and control groups matched on the basis of the PS, which alleviates potential sources of pre-treatment heterogeneity.
Two different datasets from the Survey on Household Income and Wealth (SHIW) issued by the Bank of Italy are used: cross-section data from the 2014 Survey, containing information about the tax credit; panel data for the $2012 - 2014$ period, from which we draw information for the pre- and post-treatment comparison. The estimation of the PS is performed using a Probit model based on $4458$ households that were simultaneously observed in 2012 and 2014, $864$, of which were declared eligible for the bonus.
Following Neri et al (2017), the chosen covariates are mainly sets of dummy variables which reflect demographic and socio-cultural characteristics as well as the economic condition such as employment status, income, level of education, age, and geographical area of residence[1].

[1] Individual variables refers to the head of the household.

Matching is then performed via nearest-neighbor without replacement, with a caliper of $0.01$[2]. The treatment estimates (non-winsorized and winsorized to the bottom/top at $1\%$) for monthly food, car and other durables consumption using the *full* model are shown in Table 1. Since the effects related to many

*Table 1. Full model estimates*

| | Food | Cars | Other durables |
|---|---|---|---|
| Full Model | $2.56/6.19$ | $13.98/3.93$ | $16.64/10.32^{*}$ |
| | $(21.85/20.53)$ | $(22.51/18.70)$ | $(10.49/6.24)$ |

explanatory variables turn out to be not statistically significant, performing the same procedure on only highly significant variables leads to the results displayed in Table 2. Clearly, a model uncertainty issue appears: Table 3

*Table 2. Reduced model estimates*

| | Food | Cars | Other durables |
|---|---|---|---|
| Only sign. variables | $20.25/20.11$ | $25.00/11.07$ | $11.35/9.01$ |
| | $(21.64/20.24)$ | $(25.08/20.01)$ | $(10.06/6.34)$ |

shows the treatment effects estimates obtained by applying the three Model Averaging estimators here proposed. All the estimators point toward a positive treatment effect, but produce different results, in particular "BMA mean" shows the highest values, whereas "BMA full" the smallest ones; "BMA Frequentist" is halfway between the two. Modifying caliper or data ordering induces similar conclusions, even though "BMA full" seems to appear as the most robust to changes as opposed to "BMA mean" which is the most variable.

To further illustrate the problem of model uncertainty, it is possible to transpose it in terms of the kernel density estimation[3] of the treatments: fo-

[2] As for the order of units, the choice is the minimum distance order (Austin, 2014). Common support hypothesis and balancing properties are verified too.

[3] A Gaussian kernel is used.

*Table 3. Model Averaging estimates*

| | Food | Cars | Other durables |
|---|---|---|---|
| BMA (mean) | 14.36/17.44 | 25.00/16.80 | 17.73*/13.08** |
| | (21.28/20.19) | (24.36/19.53) | (9.64/6.40) |
| BMA (Frequentist) | 11.63/12.68 | 22.75/9.82 | 13.99/10.36 |
| | (22.13/20.92) | (26.04/20.53) | (10.59/7.04) |
| BMA (full) | 8.05/9.48 | 19.07/8.63 | 13.89/10.35 |
| | (22.11/20.76) | (26.07/20.90) | (11.26/7.47) |

cusing the attention on food consumption and assuming all models as *equally* probable lead to the density function represented with red lines in Figure 1; introducing, however, a weighing scheme for model specific treatment based on the posterior model probabilities (blue lines) induces a reduction in the probability of the "tails" of the distribution. In particular, it is possible to notice how the probability of models with negative treatment effects is strongly reduced; moreover the distribution becomes clearly peaked around the most probable specifications leading to a more concentrated distribution.

## *4. Conclusion*

In this article we have explored the effect of BMA in Propensity Score matching: the choice of variables which should be included in the PS estimation is often ignored, but the consequences can be severe. Model averaging has been proposed as a plausible solution which avoids problems of misspecification; in particular, three different techniques concerning BMA have been used: BMA mean, BMA Frequentist and BMA full.

Using data from the Bank of Italy SHIW, a similar analysis to the one proposed by Neri et al (2017) for the Italian 2014 tax rebate has been performed, with the aim of verifying the validity of BMA method: in particular, taking into account model uncertainty may help to strongly reduce the arbitrariness in this model choice, where different PS models lead to not negligible differences in the treatment effect evaluation.

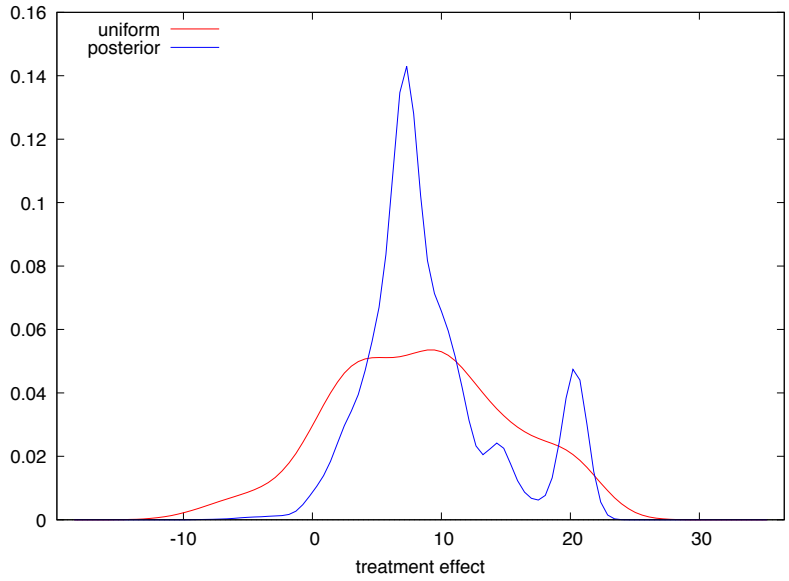



*Figure 1. Kernel density estimation (Gaussian kernel) of treatment effects (food) across model specification: not using BMA (red) vs. using BMA (blue)*

However, to further improve this conclusion additional analyses are required such as applying different matching methods for PS and devising robustness checks in a simulated scenario.

## *References*

Austin P.C. (2014) A comparison of 12 algorithms for matching on the propensity score, *Statistics in medicine*, 33(6), 105-1069.

Blinder A.S. (1981) Temporary income taxes and consumer spending, *Journal of Political Economy*, 89(1), 26-53.

Brzozowski M. (2007) Welfare reforms and consumption among single mother households: evidence from Canadian provinces, *Canadian Public Policy*, 33(2), 227-250.

Green P.J. (1995) Reversible jump markov chain monte carlo computation and bayesian model determination, *Biometrika*, 82(4), 711-732.

Heim B.T. (2007) The effect of tax rebates on consumption expenditures: evidence from state tax rebates, *National Tax Journal*, 685-710.

Johnson D.S., Parker J.A., Souleles N.S. (2006) Household expenditure and the income tax rebates of 2001, *American Economic Review*, 96(5), 1589-1610.

Modigliani F., Steindel C., Hymans S.H., Juster F.T. (1977) Is a tax rebate an effective tool for stabilization policy?, *Brookings Papers on Economic Activity*, 1977(1), 175-209.

Neri A., Rondinelli C., Scoccianti F. (2017) Household spending out of a tax rebate: Italian "80€ tax bonus", *Bank of Italy: Occasional Papers*

Parker J.A. (1999) The reaction of household consumption to predictable changes in social security taxes, *American Economic Review*, 89(4), 959-973.

Poterba J. M. (1988) Are consumers forward looking? Evidence from fiscal experiments, *The American Economic Review*, 78(2), 413-418.

Rosenbaum P.R., Rubin D.B. (1983) The central role of the propensity score in observational studies for causal effects, *Biometrika*, 70(1), 41-55.

Shapiro M.D., Slemrod J. (2003a) Consumer response to tax rebates, *American Economic Review*, 93(1), 381-396.

Shapiro M.D., Slemrod J. (2003b) Did the 2001 tax rebate stimulate spending? evidence from taxpayer surveys, *Tax policy and the economy*, 17, 83-109.

Shapiro M.D., Slemrod J. (2009) Did the 2008 tax rebates stimulate spending?, *American Economic Review*, 99(2), 374-79.

Souleles N.S. (2002) Consumer response to the Reagan tax cuts, *Journal of Public Economics*, 85(1), 99-120.

Stuart E.A., Huskamp H.A., Duckworth K., Simmons J., Song Z., Chernew M.E., Barry C.L. (2014) Using propensity scores in difference-in-differences models to estimate the effects of a policy change, *Health Services and Outcomes Research Methodology*, 14(4), 166-182.

Wilcox D.W. (1989) Social security benefits, consumption expenditure, and the life cycle hypothesis, *Journal of Political Economy*, 97(2), 288-304.

# Generalized Dynamic Factor Models. Estimation and forecasting using gretl

Riccardo Lucchetti*, Ioannis A. Venetis**

*Abstract:* We describe the methods behind a suit of functions - to be incorporated in the DFM package - that deal with generalized dynamic factor models (GDFM) estimation, forecasting and construction of impulse response functions. Examples are presented using a widely cited dataset of 132 U.S macroeconomic time series and a dataset of 81 U.S sectoral labour market series (employment). Prior to the inclusion of the GDFM functions in the DFM package, all code (package and example scripts) is available by the authors upon request.

*Keywords:* Generalized Dynamic Factor Models, GDFM

## *1. The generalized dynamic factor model - GDFM*

The Generalized Dynamic Factor Model (GDFM) as introduced in Forni *et al* (2000) and Forni and Lippi (2001) assumes we work with a $T \times n$ panel of observations $x_{i,t}, i = 1, ..., n, t = 1, ..., T$ and admits the form

$$x_{i,t} = \chi_{i,t} + \xi_{i,t} \tag{1}$$

$$\chi_{i,t} = b_{i,1}(L)\, u_{1,t} + b_{i,2}(L)\, u_{2,t} + ... + b_{i,q}(L)\, u_{q,t} \tag{2}$$

where $\chi_{i,t}$ is the *"common component"* and $\xi_{i,t}$ is the *"idiosyncratic component"*, $L$ stands for the lag operator. In vector notation, let $\mathbf{x}_t = (x_{1,t}, \ldots, x_{n,t})'$, $t \in \mathbf{Z}$,

$$\mathbf{x}_t = \boldsymbol{\chi}_t + \boldsymbol{\xi}_t$$
$$\boldsymbol{\chi}_t = \mathbf{B}(\mathbf{L})\mathbf{u}_t$$

The vector $\mathbf{u}_t = (u_{1,t}, \ldots, u_{q,t})'$, $t \in \mathbf{Z}$, of *"common shocks"* is a $q$-dimensional unobservable orthonormal white noise process orthogonal to $\boldsymbol{\xi}_t$,

* Università Politecnica delle Marche, r.lucchetti@univpm.it
** University of Patras, Department of Economics, ivenetis@upatras.gr

$t \in \mathbf{Z}$, that implies $\chi_{i,t}$ and $\xi_{j,t}$ are orthogonal at any lead and lag for all $i, j \in \mathbf{N}$.)

Each common component results from the action of a small number of $q$ unobserved shocks weighted differently (different lag structures) for each series $x_{i,t}$, $i = 1, ..., n$, $t = 1, ..., T$.

Initially, in Forni *et al* (2000), the filters $b_{i,j}(L)$ are one-sided in $L$, that is,

$$b_{i,j}(L)\, u_{j,t} = \sum_{s=0}^{+\infty} b_{i,j,s} \cdot u_{j,t-s}$$

and their coefficients $\{b_{i,j,s}\}_0^{+\infty}$ are square summable, i.e., $\sum_{s=0}^{+\infty} b_{i,j,s}^2 < +\infty$. One-sidedness is necessary for the structural interpretation of (1) and (2). However, in Forni and Lippi (2001), Forni *et al* (2004) and subsequent work, two-sided filters (with square summable coefficients) are allowed without affecting unique determination of the dimension of $\mathbf{u}_t$ or the reconstruction of $\chi_{i,t}$ and $\xi_{i,t}$.

This is an approximate factor model as the idiosyncratic components are allowed to be weakly cross-correlated.

The model is estimated by employing principal components in the frequency-domain and an interesting feature is that the common component is allowed to have an infinite MA representation, so as to accommodate for both autoregressive AR, MA and ARMA type responses to common shocks (factors).

As Forni *et al* (2015) mention: *"Apart for some minor features, most factor models considered in the literature are particular cases of the so-called Generalized Dynamic Factor Model (GDFM)..."*

The estimation problem consists in recovering the unobserved common and idiosyncratic components $\chi_{i,t}$, $\xi_{i,t}$, the common shocks $u_{j,t}$ and the filters $b_{i,j}(L)$, from a finite realization of the multivariate process $x_{i,t}$ as both the cross-sectional and time dimensions tend to infinity.

The main `GDFM` package function is `GDFM_Setup (list xlist, int q, int m, int h, bool cVerbose[0])` with inputs:

`xlist` : a list with the observable (stationary) variables $x_{i,t}$, $i = 1, ..., n$, $t = 1, .., T$. Data mean-variance standardization is performed by the

function;

`q` : number of assumed static factors; a positive integer (optional; default: 1);

`m` : covariogram truncation parameter; a positive integer (optional; default: $m = floor(\sqrt{T})$);

`h` : number of frequencies $\theta_h$ at which the spectral density matrix is estimated; a positive integer (optional; default: $h = m$);

`cVerbose` : Boolean, print details (optional; default: no)

It returns the **model bundle** that contains a $T \times n$ standardized data matrix: `X`, a string array with variable names: `Xnames`, and a number of useful model parameters such as $T$, $n$, $q$, $m$, $h$, $M = 2m + 1$, $H = 2h + 1$.

The `matlab` codes provided by Matteo Barigozzi in his website were extremely helpful and used not only as a guide to build the code for this subpart of the package but also to numerically test the results (see `http://www.barigozzi.eu` and `http://www.barigozzi.eu/gdfm.zip`). Of course, any errors are our responsibility

*2. Two-sided estimation and one-sided estimation and forecasting*

The two-sided approach of Forni *et al* (2000) produces common component estimates $\hat{\chi}_{i,t}^{2S} = \sum_{k=-m}^{m} \underline{K}_{i,k} \mathbf{x}_{t-k}$ and although more efficient than principal components (PC) estimation of the factor space (common components) is not directly suitable for prediction of the observable series as the common component estimate $\hat{\chi}_{i,t}^{2S}$ deteriorates for $t$ close to 1 or $T$.

Forni *et al* (2005) put forth an one-sided estimation procedure to efficiently (compared to principal components) forecast the observable series through $\hat{x}_{i,t+h} = \hat{\chi}_{i,t+h}^{1S}$, $h \geq 1$ and as a by-product an estimate $\hat{\chi}_{i,t}^{1S}$ of the common components is also obtained.

Both, $\hat{\chi}_{i,t}^{1S}$ and $\hat{\chi}_{i,t+h}^{1S}$ are consistent in the sense that, as the cross-section size $n$ and the number of time observations $T$ tend to infinity, the estimate $\hat{\chi}_{i,t}^{1S}$ converges to $\chi_{i,t}^{1S}$ and the predictor $\hat{\chi}_{i,t+h}^{1S}$ tends in probability to the population-optimal predictor.

In matrix notation, let $\mathbf{B(L)} = \mathbf{C(L)[D(L)]^{-1}}$, where $\mathbf{C}(L) = \mathbf{C_0} + \mathbf{C_1}L + \cdots + \mathbf{C_s}L^s$ is an $n \times q$ matrix polynomial in the lag operator $L$ with finite maximum $s > 0$ and $\mathbf{D}(L) = \mathbf{I_q} - \mathbf{D_1}L - \cdots - \mathbf{D_S}L^S$ is a $q \times q$ matrix polynomial with $S < s + 1$.

If we denote by $\mathbf{f}_t$ the dynamic factors generated by the $q$ dynamic shocks, $\mathbf{f}_t = [\mathbf{D(L)}]^{-1}\mathbf{u}_t$ or $\mathbf{D(L)}\mathbf{f}_t = \mathbf{u}_t$ and $\mathbf{x}_t = \mathbf{C}(L)\mathbf{f}_t + \boldsymbol{\xi}_t$.

Now, by letting $\mathbf{F}_t = \left(\mathbf{f}_t', \mathbf{f}_{t-1}' \ldots \mathbf{f}_{t-s}'\right)'$ and $\mathbf{C} = (\mathbf{C}_0 \ldots \mathbf{C}_s)$ we obtain a static factor model

$$\mathbf{x}_t = \mathbf{C} \cdot \mathbf{F}_t + \boldsymbol{\xi}_t \tag{3}$$

with $r = q\,(s + 1)$ static factors $\mathbf{F}_t$ with spectral density of rank $q$.

The one-sided estimator of FHLR05 produces $\hat{\boldsymbol{\chi}}_t^{1S} = \widehat{\mathbf{C} \cdot \mathbf{F}_t}$ and its $h-$steps ahead forecast $\hat{\boldsymbol{\chi}}_{t+h}^{1S}$.

The following example script is part of the largest example script available upon request `GDFMexamples_Estimation_FHLR00_FHLR05.inp` and it is based on the Stock and Watson (2005) dataset of 132 U.S macroeconomic series. It can serve as a starting point for those interested in the previously described procedures:

```
clear
set verbose off
include GDFM.inp
# See Table 1 and section 6 in Bai and Ng (2013)
open sw2005datatC.gdtb -q
list L1 = CES002c IPS10c sFYGT1c
list L2 = dataset
list xlist = L1 || L2
q = 3
verbocity = 1
# Setup
bsetup = GDFM_Setup(xlist, q, , , verbocity)

bFHLR00 = GDFM_FHLR00(&bsetup)

bFHLR05 = GDFM_FHLR05(&bsetup)
```

## 3. Unrestricted estimation

In Forni *et al* (2015) and Forni *et al* (2017) the most general approach is put forth. In detail, without loss of generality and for the simplicity of notation, assume, that $n$ is an integer multiple of $(q+1)$, that is, $n = m(q+1)$ for some $m \in \mathbf{N}$ and let $\boldsymbol{\chi}_{n,t} = (\chi_{1,t}, \chi_{2,t}, \cdots, \chi_{n,t})'$.

The *"unrestricted"* GDFM (vector notation) is given by

$$\begin{aligned} \mathbf{x}_{n,t} &= \boldsymbol{\chi}_{n,t} + \boldsymbol{\xi}_{n,t} \\ \boldsymbol{\chi}_{n,t} &= \underset{n\times q}{\mathbf{B}}(\mathbf{L})\mathbf{u}_{n,t} \end{aligned}$$

where $\mathbf{B}(\mathbf{L})$ is a rational square-summable one-sided filter and $\mathbf{u}_{n,t}$ is orthonormal white noise.

Under the assumptions in Forni *et al* (2015), there exist an $m(q+1) \times m(q+1)$ block-diagonal matrix of one-sided filters $\mathbf{A}_n(L)$ with $m$ diagonal blocks $\mathbf{A}^{(i)}(L)$ of dimension $(q+1) \times (q+1)$ such that the VAR operators $\mathbf{A}(L) = (\mathbf{I}_n - \mathbf{A}_n(L))$ are fundamental for $\boldsymbol{\chi}_{n,t}$

$$\mathbf{A}(L)\boldsymbol{\chi}_{n,t} = \mathbf{H}_n\mathbf{u}_{n,t} \tag{4}$$

with $\mathbf{H}_n = (\mathbf{H}_n^{1\prime}, \cdots, \mathbf{H}_n^{m\prime})'$ is a full-rank $m(q+1) \times q$ matrix of constants and sub-matrix $\mathbf{H}_n^m$ is $(q+1) \times q$.

Pre-multiply the observations vector with $\mathbf{A}(L) = (\mathbf{I}_n - \mathbf{A}_n(L))$ and let $\mathbf{A}(L)\mathbf{x}_{n,t} = \mathbf{y}_{n,t}$, then a static factor model is obtained (for $\mathbf{y}_{n,t}$)

$$\mathbf{y}_{n,t} = \mathbf{H}_n\mathbf{u}_{n,t} + \mathbf{A}(L)\boldsymbol{\xi}_{n,t} \tag{5}$$

The corresponding package function provides parameter estimates $\hat{\mathbf{B}}(\mathbf{L}) = [\hat{\mathbf{A}}(L)]^{-1} \cdot \hat{\mathbf{R}} \cdot \hat{\mathbf{H}}_{\mathbf{q}}$ (IRF's), shocks $\hat{\mathbf{u}}_{n,t}$, common components

$$\hat{\boldsymbol{\chi}}_t^{UR} = \hat{\mathbf{B}}_0 \cdot \hat{\mathbf{u}}_t + \hat{\mathbf{B}}_1 \cdot \hat{\mathbf{u}}_{t-1} + \hat{\mathbf{B}}_2 \cdot \hat{\mathbf{u}}_{t-2} + \cdots + \hat{\mathbf{B}}_{19} \cdot \hat{\mathbf{u}}_{t-19}$$

and forecasts for the common components at horizon $h$

$$\hat{\boldsymbol{\chi}}_{T+h}^{UR} = \hat{\mathbf{B}}_h \cdot \hat{\mathbf{u}}_T + \hat{\mathbf{B}}_{h+1} \cdot \hat{\mathbf{u}}_{T-1} + \hat{\mathbf{B}}_{h+2} \cdot \hat{\mathbf{u}}_{T-2} + \cdots$$

The following extract from the example script `GDFMexamples_FHLZ17.inp` is based on the Stock and Watson (2005) dataset of 132 U.S macroeconomic series and can serve as a starting point for those interested in the previously described method. The full script includes (among others) a plot of estimated impulse response functions of the first three target variables to a shock in the first or more dominant factor.

```
clear
set verbose off
include GDFM.inp

open sw2005datatC.gdtb -q
list L1 = CES002c IPS10c sFYGT1c
list L2 = dataset
list xlist = L1 || L2
q = 3
p = 2 # Number of VAR lags
verbocity = 1

bsetup = GDFM_Setup(xlist, q, , , verbocity)

bFHLZ17 = GDFM_FHLZ17(&bsetup, p, , ,)
```

*4. Choosing the number of dynamic factors*

Finally, the package includes two functions that produce summary statistics (a scree plot and a factor variance contribution plot across frequencies) as well as the Hallin and Liška (2007) information criterion approach to determine the number $q$ of common shocks in GDFM settings. The following two scripts,

`GDFMexamples_summary.inp`

and

`GDFMexample_Hallin_Liska_2007.inp`

produce a number of plots that aid researchers in their initial GDFM specification quest. In this example, we consider the 81 U.S sectoral employment growth series employed by Fiorentini *et al* (2018) as our data panel,

```
clear
set verbose off
include GDFM.inp

open Labor.gdt -q
list xlist = dataset - CES0000000001
q = 3
m = floor( 0.75*sqrt( rows({xlist}) ) )
h = m
plotoption = 1

bsetup = GDFM_Setup(xlist, q, m, h, plotoption)
bsummary = GDFM_summary(&bsetup)
```

while the Stock and Watson (2005) dataset is utilized in the last example,

```
clear
set verbose off
include GDFM.inp
set seed 333888333

open sw2005datatC.gdtb -q
smpl full
list xlist = dataset
scalar qmax = 8 # Upper bound on the number of dynamic shocks
scalar plotoption = 1 # Set 1 to display plot(s)

bsetup = GDFM_Setup(xlist, , , , plotoption)

bHL2007 = GDFM_HL2007(&bsetup,qmax)
```

## *5. Conclusions and future developments*

The set of scripts we presented here reproduce most of the methods that Forni, Hallin, Lippi and various co-authors have been proposing over the past years. It remains to be seen whether these function will be integrated into the existing package DFM, which deals with dynamic factor models estimated in the time domain rather than in the frequency domain, or will have to be packaged as a standalone entity.

## *References*

Bai J., Ng S. (2013) Principal components estimation and identification of static factors, *Journal of Econometrics*, 176, 1, 18-29.

Fiorentini G., Galesi A., Sentana E. (2018) A spectral EM algorithm for dynamic factor models, *Journal of Econometrics*, 205, 1, 249-279.

Forni M., Hallin, M., Lippi M., Reichlin L. (2000) The generalized dynamic-factor model: Identification and estimation, *Review of Economics and Statistics*, 82, 4, 540-554.

Forni M., Hallin, M., Lippi M., Reichlin L. (2004) The generalized dynamic factor model consistency and rates, *Journal of Econometric*s, 119, 2, 231-255.

Forni M., Hallin, M., Lippi M., Reichlin L. (2005) The generalized dynamic factor model, *Journal of the American Statistical Association*, 100, 471, 830-840.

Forni M., Hallin M., Lippi M., Zaffaroni P. (2015) Dynamic factor models with infinite-dimensional factor spaces: One-sided representations, *Journal of Econometrics*, 185, 2, 359-371.

Forni M., Hallin M., Lippi M., Zaffaroni P. (2017) Dynamic factor models with infinite-dimensional factor space: Asymptotic analysis, *Journal of Econometrics*, 199, 1, 74-92.

Forni M., Lippi M. (2001) The generalized dynamic factor model: representation theory, *Econometric Theory*, 17, 6, 1113-1141.

Hallin M., Liška R. (2007) Determining the number of factors in the general dynamic factor model, *Journal of the American Statistical Association*, 102, 478, 603-617.

Stock J.H., Watson M.W. (2005) Implications of dynamic factor models for var analysis, *NBER Working Paper 11467*, 1-65.

# Benchmarking and temporal disaggregation with related indicators in gretl

Alberto Maronilli*, Tommaso Di Fonzo**

*Abstract:* Benchmarking (Denton, 1971) and temporal disaggregation with related indicators (Chow and Lin, 1971) are well established statistical procedures, available in both commercial and free econometric software. The 2019c release of Gretl does not support Denton's benchmarking and performs temporal disaggregation according to Chow and Lin in a limited way, returning poor and debatable results. In this paper the two procedures are described and the *chowlin()* command of Gretl is critically reviewed. Then the function package *tbench* is presented, which allows Gretl users to perform benchmarking and temporal disaggregation in a more effective way.
*Keywords:* Time series, Benchmarking, Temporal disaggregation

## *1. Introduction*

In time series analysis, temporal benchmarking consists in adjusting a preliminary, high frequency time series to have temporal consistency with a lower frequency version of the same variable, usually measured from a different data source (Dagum and Cholette, 2006). This statistical procedure is used to correct inconsistencies between two different estimates by combining (and preserving as much as possible) (i) the dynamic profile of the preliminary (unbenchmarked) time series, and (ii) the levels of the low-frequency benchmarks.

Temporal disaggregation is strictly related to, and in fact generalizes, benchmarking, since high-frequency estimated data are derived from low-frequency ones and, if available, one or more related high-frequency series.

In both cases, however, given a 'low' frequency (annual, quarterly) observed time series, the 'high' frequency (quarterly, monthly) series has to be estimated and the result must be coherent with the available temporally aggregated data. According to Chow and Lin (1971), depending on the nature of the variable to be disaggregated we refer to temporal disaggregation

* University of Padua, alberto.maronilli@studenti.unipd.it
** University of Padua, tommaso.difonzo@unipd.it

as either *distribution* of flows or averages (indices) or *interpolation* of stock variables. A strictly related issue to be considered in this framework is the *extrapolation*, that is the estimation of high frequency figures when the relevant low-frequency observation is not observed (for example, in the course of the current year, when the annual benchmark is not yet available).

In literature there is a variety of well established procedures to deal with benchmarking and temporal disaggregation of time series (Chen *et al* 2018, and the references therein). For space reason, in this paper the focus is on the Modified Denton Proportionate First Differences (*PFD*) procedure (Denton, 1971, Cholette, 1984), and on the regression-based temporal disaggregation procedure by Chow and Lin (1971). In fact, these two procedures are the most widely used by data producers and statistical agencies (IMF, 2017, Eurostat, 2018). However, the package $tbench$ for Gretl offers other benchmarking procedures as well.

## 2. Notation

In the description of the procedures, the following notation is used:

$s$ : number of high-frequency values for each benchmark value (aggregation order: $s = 3$ if quarterly data is to be converted into monthly values, $s = 4$ for annual-to-monthly estimation, and $s = 12$ for the annual-to-monthly case);

$n$ : number of high-frequency periods;

$N$ : number of low-frequency periods. It is usually assumed $n = s \cdot N$ (i.e., the low frequency and the high-frequency periods exactly cover the same time span). When $n > s \cdot N$ we face benchmarking/temporal disaggregation *and* extrapolation;

$y_t$ : the unknown high-frequency series to be estimated, $t = 1, \ldots, n$;

$y_{0T}$ : the known low-frequency benchmark series of the variable of interest, $T = 1, \ldots, N$;

$p_t$ : the known preliminary high-frequency series (to be used in benchmarking), $t = 1, \ldots, n$;

$x_{j,t}$ : the related high-frequency series (to be used in temporal disaggregation), $j = 1, \ldots, q$, $t = 1, \ldots, n$.

The temporal aggregation constraints between the target variable and the available benchmark can be expressed in matrix form as

$$y_0 = Cy \tag{1}$$

where $y_0$ and $y$ are $(N \times 1)$ and $(n \times 1)$ vectors, respectively, and the $(N \times n)$ matrix $C$ takes different forms depending on the case we are dealing with:

$$C^{distribution} = \begin{bmatrix} 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 1 & \cdots & 1 \end{bmatrix}$$

$$C^{index} = \frac{1}{s} C^{distribution}$$

$$C^{interpolation} = \begin{bmatrix} 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 1 \end{bmatrix}$$

$$C^{extrapolation} = \begin{bmatrix} & & & 0 & 0 \\ & C^{dist/ind/inter} & & \vdots & \vdots \\ & & & 0 & 0 \end{bmatrix}$$

## *3. Modified Denton PFD benchmarking*

The adjusted (benchmarked) estimates are obtained according to a movement preservation principle on proportional levels, which is considered a good approximation of the 'true' movement preservation principle, based on the growth rates (this issue is discussed by Di Fonzo and Marini, 2012).

Following the proposal by Cholette (1984), where the first differences are

correctly calculated, the modified Denton's PFD benchmarked estimates are the solution to a linearly constrained quadratic optimization problem, where a loss function given by the sum of the squared first differences of the proportional adjustments is minimized with respect to $y_t$:

$$\sum_{t=2}^{n}\left(\frac{y_t - p_t}{p_t} - \frac{y_{t-1} - p_{t-1}}{p_{t-1}}\right)^2 \equiv \sum_{t=2}^{n}\left(\frac{y_t}{p_t} - \frac{y_{t-1}}{p_{t-1}}\right)^2. \tag{2}$$

Using matrix notation, the problem can be stated as

$$\underset{y}{min}(y-p)'M(y-p) \qquad s.t.\ \ y_0 = Cy, \tag{3}$$

where $M = \hat{p}^{-1}\Delta'\Delta\hat{p}^{-1}$, $\hat{p} = \text{diag}(p)$ is a diagonal $(n \times n)$ matrix, and $\Delta$ is the $(n-1 \times n)$ first difference matrix:

$$\Delta = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}.$$

The PFD benchmarked estimates are contained in the $(n \times 1)$ vector $y^{PFD}$, solution to the linear system (Di Fonzo and Marini, 2012)

$$\begin{bmatrix} M & C' \\ C & 0 \end{bmatrix}\begin{bmatrix} y^{PFD} \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ y_0 \end{bmatrix}, \tag{4}$$

where $\lambda$ is a $(N \times 1)$ vector of Lagrange multipliers.

### *4. Chow and Lin regression-based temporal disaggregation*

This procedure assumes a regression model with stationary $AR(1)$ disturbances between the true (and unobserved) high frequency observations $y_t$ and a set of related series observed at the same 'high' frequency, $x_{1,t}, ..., x_{q,t}$:

$$\begin{aligned} y_t &= \textstyle\sum_{j=1}^{q}\beta_j x_{j,t} + u_t \\ u_t &= \rho u_{t-1} + \varepsilon_t, \quad |\rho| < 1. \end{aligned} \tag{5}$$

This model can include 'true' related series (indicators) and deterministic effects as well, whose combination is used to estimate the high-frequency dynamic profile of the target variable.

Since $y_t$ is unobserved, model (5) cannot be estimated. In order to derive the Best Linear Unbiased Estimator (BLUE) of the unknown $y_t$'s in line with the benchmarks $y_{0T}$'s, Chow and Lin (1971) considers the temporal aggregation of model (5),

$$y_{0T} = \sum_{j=1}^{q} \beta_j x_{0j,T} + u_{0T}, \tag{6}$$

where $x_{0j,T}$ is the $j$-th temporally aggregated indicator series, and $u_{0T}$ is the result of the temporal aggregation of the high-frequency $AR(1)$ disturbances.

In matrix notation, denote $X$ the $(n \times q)$ matrix containing the $q$ high-frequency related series, $X_0 = CX$ its temporally aggregated counterpart, $V$ the $(n \times n)$ covariance matrix of the $AR(1)$ vector of disturbances $u$, and $\beta$ the $(q \times 1)$ vector of the unknown regression coefficients. If $\rho$ was known, matrix $V$ would be known up to a multiplicative factor (the variance of the white noise $\varepsilon_t$). In this case Chow and Lin (1971) show that the BLUE of the high-frequency series $y$ in line with the benchmark $y_0$ is given by:

$$\hat{y} = X\hat{\beta} + VC'V_0^{-1}(y_0 - X_0\hat{\beta}) \tag{7}$$

where $V_0 = CVC'$ and $\hat{\beta} = (X_0'V_0^{-1}X_0)^{-1}X_0'V_0^{-1}y_0$.

The estimated series consists of two components: the former coming from the regression ($X\hat{\beta}$) and the latter from the residuals of the auxiliary temporally aggregated regression model (6) smoothed through matrix $VC'V_0^{-1}$.

Chow and Lin (1971) propose to estimate the autoregressive parameter $\rho$ exploiting the relationship between this parameter and the first order autoregressive coefficient of the temporally aggregated disturbances. This relationship is indeed invertible in $(-1,\ 1)$ in the quarterly-to-monthly disaggregation case, which is considered by Chow and Lin in their paper, but unfortunately this is not true in general. For example, this relationship is not invertible in the annual-to-quarterly case (Bournay and Laroque, 1979), making this estimation procedure unfeasible for the purpose in hand.

The generally adopted solution is to estimate $\rho$ and $\beta$ through either Max-

imum Likelihood (assuming gaussian white noise $\varepsilon_t$) or Feasible Generalized Least Squares, where $\rho$ can be easily estimated through simple numerical procedures. Furthermore, it is generally recognized that, to better preserve the original movements of the related series and to avoid introducing spurious fluctuations in the final estimated values, only positive values of $\rho$ should be looked for (i.e., $\rho \in [0,\ 1)$).

*5. The chowlin() function*

In the 2019c release of Gretl the $chowlin()$ function performs temporal disaggregation of a time series using related indicators. It is our opinion that this function, which is also called for by Gretl when the user wishes to expand the dataset to a higher frequency, suffers some limitations and problematic issues.

More precisely, (i) $chowlin()$ supports only temporal disaggregation of an index variable, while flows variable (obtained as sum of the high-frequency values) and stock variables are not considered options, (ii) extrapolation is not supported, and (iii) the only considered time conversions are annual-to-quarterly ($s = 4$) and quarterly-to-monthly ($s = 3$), while annual-to-monthly ($s = 12$) is missing.

Furthermore, at least two issues do not allow the function to perform an effective temporal disaggregation of the target series: (i) the inclusion of two deterministic variables (both linear and quadratic trends) in the $x$'s of the regression model (5), and (ii) the estimation procedure of $\rho$.

As for the forced inclusion of two deterministic trends as regressors, the user of the function $chowlin()$ cannot discard any or all of three deterministic variables (constant, linear and quadratic trend), even when a 'true' related series is used. This seems inappropriate since the variables to be included as regressors should depend on the features of the target series, and the decision to include either a linear or a quadratic trend should be left to the user only.

The estimation procedure for $\rho$ raises two concerns: (i) it is based on the original proposal by Chow and Lin (1971), whose weakness (possible non-invertibility of the relationship between $\rho$ and its temporally aggregate counterpart) has been stressed in section 4, (ii) by default the search range

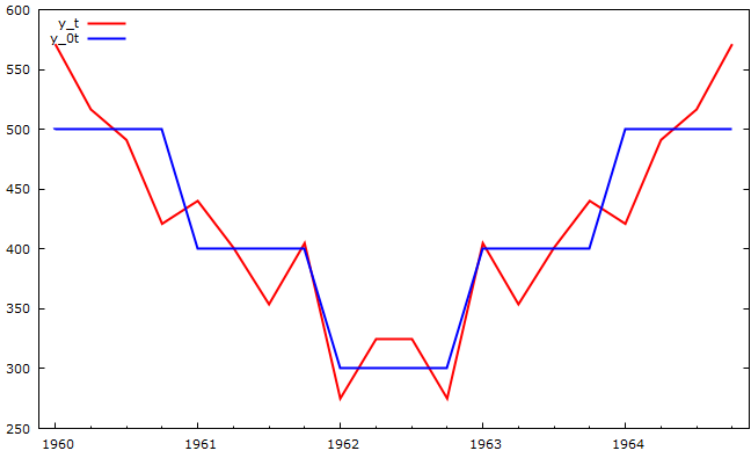



*Figure 1. Results of the temporal disaggregation of a simple series using chowlin() without related indicators*

is $(-1,\ 1)$: since the user cannot manage this option, this fact could create unpleasant (and sometimes counterintuitive) results.

This last drawback is shown in figure 1, which reports the quarterly estimates produced by $chowlin()$ for a very simple annual benchmark ($y_{0t}$ which consists of the values 500, 400, 300, 400, 500): in absence of related series which suggest short-term movements of this kind, the noticeable fluctuations in the disaggregated series $y_t$ are an unjustified artifact caused by a negative estimate of $\rho$.

## 6. The tbench package

Grounded on the considerations made so far, the $tbench$ package comes with some new and distinctive features aimed at improving the benchmarking and temporal disaggregation procedures offered by Gretl:

- The values of the benchmark series to be disaggregated can represent the sum or the average of the disaggregated values (flows variable) or the first or the last element of the subperiods (stock variable).
- $tbench$ offers the choice between different aggregation orders ($s = 3$, $s = 4$, $s = 12$).

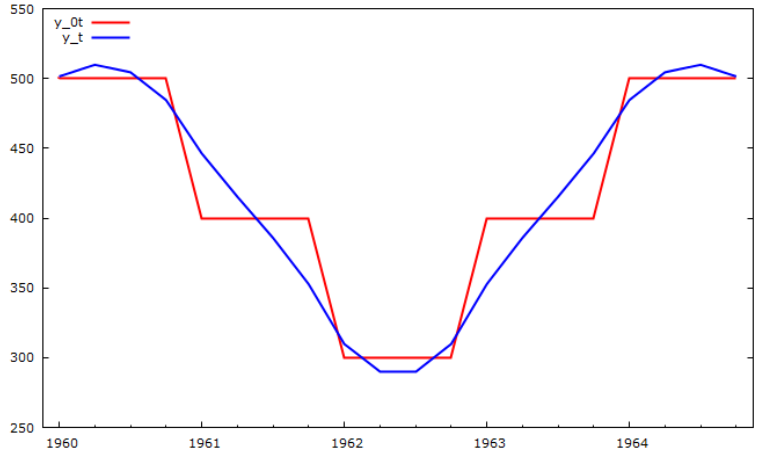



*Figure 2. Results of the temporal disaggregation of a simple series using the Chow and Lin procedure implemented in the tbench package (a constant and an index variable are used as related indicators)*

- Extrapolation is supported.
- Besides the Modified Denton PFD and Chow and Lin procedures, the user can choose between other simple benchmarking techniques, like pro-rata benchmarking and the additive first and second differences variants of the Denton's benchmarking procedure.
- $tbench$ offers a series of graphs useful to get a visual inspection of the results, in order to appreciate the adequacy of the chosen benchmarking/temporal disaggregation procedure. For example, the user can look at the graph of the time series of the growth rates of both the preliminary and the estimated series, and at their scatterplots.

Figure 2 shows the results of the temporal disaggregation of the simple benchmark series $y_{0t}$ seen in figure 1 according to the Chow and Lin procedure implemented in $tbench$, and using a constant and a linear trend (index variable) as related indicators. It clearly appears that, compared to figure 1, now the estimated series $y_t$ does not present unpleasant short-term fluctuations and has a smooth dynamics, as expected.

Regarding the implementation of the package, it should be noted that, with the exception of midas list, in Gretl the user cannot manage series of different frequencies in the same dataset. Since the focus of the benchmarking is on the estimated high frequency series, in order to work properly the $tbench$ package needs a dataset at the same frequency of the preliminary series. In this dataset the benchmark series must be present in repeated form. This can be easily obtained (i) by creating a working dataset using the 'true' (not expanded) benchmark series, (ii) then by expanding the working dataset by choosing to repeat the lower frequency values, and finally (iii) by adding the preliminary or related series to the expanded dataset.

The procedures offered by the $tbench$ package are: naive benchmarking, pro-rata benchmarking, several Modified Denton variants (additive first differences, additive second differences, proportional first differences, proportional second differences), Cholette and Dagum (1994) benchmarking procedure, Chow and Lin temporal disaggregation with manual selection of $\rho$, Chow and Lin with ML estimation of $\rho$, and the regression based temporal disaggregation procedure by Fernández (1981).

The package also comes with a set of functions allowing the users to work with matrix variables, which are more flexible for the user and permit to perform the original version of the Denton benchmarking procedure and to choose aggregation orders other than 3, 4 and 12. It is planned to use this framework in the future to implement other benchmarking and temporal disaggregation procedures.

## *References*

Bournay J., Laroque G. (1979) Réflexions sur la méthode d'élaboration des comptes trimestriels, *Annales de l'INSEE*, 36, 3-30.

Chen B., Di Fonzo T., Mushkudiani N. (2018) Preface to the Special Issue on 'Benchmarking, Temporal Disaggregation, and Reconciliation of Systems of Time Series', *Statistica Neerlandica*, 72, 402-405.

Cholette P.A. (1984) Adjusting sub-annual series to yearly benchmarks, *Survey Methodology*, 10, 35-49.

Cholette P.A., Dagum E.B. (1994) Benchmarking time series with autocorrelated survey errors, *International Statistical Review* 62, 365-377.

Chow G.C., Lin A. (1971) Best Linear Unbiased Interpolation, Distribution, and Extrapolation of Time Series by Related Series, *The Review of Economics and Statistics*, 53, 372-375.

Dagum E.B., Cholette P.A. (2006) *Benchmarking, Temporal Distribution and Reconciliation Methods for Time Series*, New York, Springer.

Denton F.T. (1971) Adjustment of monthly or quarterly series to annual totals: An approach based on quadratic minimization, *Journal of the American Statistical Association*, 66, 99-102.

Di Fonzo T, Marini M. (2012) Benchmarking time series according to a growth rates preservation principle, *Journal of Economic and Social Measurement*, 37, 225-252.

Eurostat (2018) *ESS guidelines on temporal disaggregation, benchmarking and reconciliation*, Luxembourg, Publications Office of European Union (`https://ec.europa.eu/eurostat/web/products-manuals-and-guidelines/-/KS-06-18-355`).

Fernández R.B. (1981) Methodological Note on the Estimation of Time Series, *Review of Economic and Statistics*, 63, 471-478.

IMF (2017), *Quarterly National Accounts manual*, Washington, International Monetary Fund (`https://www.imf.org/external/pubs/ft/qna/`).

# Integer Autoregressive modeling: a new gretl routine

Lucio Palazzo*

*Abstract:* This paper focuses on a family of observation-driven models for autoregressive discrete-valued data, called INAR models. The main purpose of the project is to write and document a set of *Gretl* Econometric Software functions that perform time series estimation of one-lagged univariate INAR models with Poisson and Negative Binomial marginals.

*Keywords:* INAR models, Poisson, Negative Binomial.

## *1. Introduction*

Classical ARMA models, defined for stationary real–valued processes, show a lot of attractive properties but such models cannot be applied when the observations are categorical in nature or quantitative but fairly small and is not possible to approximate them to a continuous distribution, since the multiplication of an integer by a real number usually results in a non–integer value. For this reason several models for discrete valued data have been proposed, e.g. high–order Markov Chains of Pegram (1980), discrete autoregressive moving average models (DARMA) of Jacobs and Lewis (1983), a generalized linear model approach for dependent data by Zeger (1988) and multivariate count time series models by Pedeli and Karlis (2013).
There are two basic approaches to handle non–Gaussian data exhibiting serial dependence: it is possible to distinguish between observation driven and parameter driven processes. In parameter–driven models, parameters vary over time as dynamic processes with uncorrelated errors. In an observation–driven model current parameters are deterministic functions of lagged dependent variables as well as contemporaneous and lagged exogenous variables. On top of the latter class, Integer Autoregressive (INAR) processes introduced by Alzaid and Al–Osh (1988) represent the framework from which we depart to build a new Gretl routine in order to compute MLE of two specific classes

*University of Salerno, lpalazzo@unisa.it

of one-lagged INAR models.
The rest of the paper is organized as follows. Section 2 briefly describes INAR($p$) models and their main features. Section 3 introduces the proposal while in Section 4 the effectiveness of the algorithms is illustrated by running a Monte Carlo simulation and evaluating the computational cost of the functions. Relevant points of discussion are finally summarized in Section 5.

## *2. Autoregressive Modeling of Discrete Valued Time Series*

One of the possible ways to formulate a well posed model for stationary sequences of integer–valued random variables consists of replacing the scalar multiplication by a different operator with similar properties, called Binomial Thinning operator, introduced by Steutel and van Harn (1979). INAR models follow this philosophy. Let $X$ a non–negative integer valued random variable and $\alpha \in (0, 1)$ a real constant value. The *Binomial Thinning* operator is defined as the random variable such that:

$$\alpha * X \stackrel{def.}{=} \sum_{i=1}^{X} Y_i = Y_1 + Y_2 + \ldots + Y_X \tag{1}$$

where the counting series $\{Y_t\}$ are i.i.d. non–negative integer valued random variables, independent of $X$, with range $\{0, 1\}$ and having probabilities $P(\ Y_t\ =\ 1\ )\ =\ \alpha$ and $P(Y_t = 0) = 1 - \alpha$. This operator is defined as a random sum of i.i.d. random variables $\{Y_t\}$, with $Y_t \sim \text{Ber}(\alpha)$, independent of $X$, such that $\text{E}(Y_t) = \alpha$ and $\text{Var}(Y_t) = \alpha(1 - \alpha)$.
The random variable defined by thinning operator involves two random components: in fact, it represents a stochastic sum of i.i.d. stochastic processes where $X$ expresses the (random) number of i.i.d. random variables $Y_i$ involved into summation. If the process $X$ is deterministic, i.e. $X = n$, then:

$$\alpha * X | (X = n) = \sum_{i=1}^{n} Y_i = Y_1 + Y_2 + \ldots + Y_n \tag{2}$$

and, since $\{Y_t\}$ is a sequence of i.i.d. Bernoulli random variables, the binomial thinning operator, given $X$, leads to the Binomial Law: the random

variable given in equation (2) has Bin$(n, \alpha)$ distribution.
Let $\{\varepsilon_t\}$ be an i.i.d. nonnegative integer–valued process having finite mean $\mu_\varepsilon$ and variance $\sigma^2_\varepsilon$ and let $\alpha \in (0, 1)$. Given the binomial thinning operator defined in equation (1), a nonnegative integer–valued process $X_t$ is said to be an INAR(1) if

$$X_t = \alpha * X_{t-1} + \varepsilon_t \tag{3}$$

Although there are several ways to define an INAR processes having $p \geq 2$ lags depending on the joint distribution among thinning operators, the model given by Du and Li (1991) is the most studied in literature. INAR($p$) models involve the summations of $p$ mutually independent thinning operations, being at the same time independent of the innovation process $\varepsilon_t$.
There is a strong relationship between INAR($p$) and classical AR($p$) processes, in fact they share the same autocovariance structure and belong to a broader class of conditional autoregressive models (CLAR, see Grunwald et al. (2000)). Stationarity conditions are satisfied when the the roots of the polynomial $z^p - \alpha_1 z^{p-1} - \ldots - \alpha_{p-1} z - \alpha_p$ lay inside the unit circle and, in case of an INAR(1) model, such conditions hold whenever $\alpha \in (0, 1)$.
Parameter estimation methods include Yule–Walker (YW), Conditional Least Squares (CLS) and Conditional Maximum Likelihood (CML). The latter is based on convolution of the arrivals and the results of each thinning operation, providing consistency, asymptotic normality and asymptotic efficiency.
Transition probabilities of INAR(1) models correspond to the convolution between Binomial and innovation's distributions

$$\mathrm{P}(X_t = x_t | X_{t-1} = x_{t-1}) = \sum_{i=0}^{\min\{x_t, x_{t-1}\}} \binom{x_t}{i} \alpha^i (1-\alpha)^{x_{t-1}-i} \, \mathrm{P}(\varepsilon_t = x_t - i) \tag{4}$$

Then, using the above formula it is possible to compute the conditional likelihood function as

$$L(\boldsymbol{\theta}; \boldsymbol{X} | X_1) = \prod_{t=2}^{T} \mathrm{P}(X_t | X_{t-1}) \tag{5}$$

where $\boldsymbol{\theta}$ is the vector of parameters to be estimated and $\boldsymbol{X} = \{X_1, \ldots, X_T\}$.

In this framework the Poisson distribution plays an important role: an INAR(1) is distributed as a Poisson if and only if innovations process has a Poisson distribution, cf. McKenzie (1985). Unfortunately this relationship holds only in this specific case. In particular, given an INAR(1) process with Negative Binomial marginal distribution, Leonenko et a. (2007) proved that innovation process follows a not common random variable called Negative Binomial-Geometric.
Aside from the thinning parameter $\alpha$, the other values involved in an INAR(1) model depend on the innovation process. In a P-INAR(1) model there is only an additional parameter, $\lambda$, corresponding to the average number of events occurring in a specific interval. Besides, an NB–INAR(1) process requires the estimation of $\gamma$, operating as shape parameter, and $\beta$, a scale parameter.

### *3. Poisson and Negative Binomial Marginals: Discussion and Proposal*

Poisson distribution is one of the most widely–used counting processes presenting several unique features, especially in an INAR framework. However, in real data applications overdispersion is often encountered for various reasons and Poisson does not allow for the variance to be adjusted independently of the mean, suggesting the use of different processes that may provide a better fit. Negative Binomial and Poisson distributions have the same support, that is $\mathbb{N}_0$, but the former is more flexible because it includes an additional parameter allowing the variance to be greater than the mean, which often improves model fitting to data. Conditional Maximum Likelihood of a P–INAR(1) model is relatively simple since the derivation of first order conditions yields to a single equation:

$$\sum_{i=2}^{T} x_t = \alpha \sum_{i=2}^{T} x_{t-1} + (T-1)\lambda \tag{6}$$

Estimates are obtained making explicit either $\alpha$ or $\lambda$ in equation (6), substituting them in first order conditions and iterating the resulting equations. However, for the NB–INAR(1) CMLE can only be found via numerical global optimization since no closed–form solution is available. Part of the proposal involves the discovery that the probability mass function of the innovations

can be reduced as terms of the Hypergeometric function, as follows:

$$\begin{aligned} \mathrm{P}(\varepsilon_t = k) = & \sum_{i=0}^{\infty} \binom{i+k-1}{k} \left(\frac{\beta}{\beta+\alpha}\right)^i \left(\frac{\alpha}{\beta+\alpha}\right)^k \cdot \binom{\gamma+i-1}{i} \alpha^\gamma (1-\alpha)^i \\ = & \ \gamma\alpha^\gamma \left(\frac{\alpha}{\beta+\alpha}\right)^k (1-\alpha)^{\frac{\beta}{\beta+\alpha}} \, {}_2F_1\left(\gamma+1, k+1, 2, (1-\alpha)\frac{\beta}{\beta+\alpha}\right) \end{aligned}$$

for all $k \in \mathbb{N}_0$, where ${}_2F_1(\cdot)$ is the Hypergeometric function defined as the sum of hypergeometric series.

**Algorithm 1.**

*Poisson and Negative Binomial INAR(1) Parameter Estimation*

```
1:  Input
2:  X                                                    ▷ Data
3:  par0                                                 ▷ Starting Parameters
4:  procedure COND.PROB(param, Xt, Xt_1)                 ▷ Transition Probabilities
5:      alpha ← param[1]                                 ▷ Thinning Parameter
6:      M ← xmin(Xt, Xt_1)
7:      jj ← seq(0,M)'
8:      p1 ← pdf(b, alpha, Xt_1, jj)                     ▷ pmf of Binomial r.v.
9:      p2 ← pdf(innovations, param, Xt-jj)              ▷ pmf of Innovations
10:     ret ← p2'p1
11:     return ret
12: end procedure
13: procedure LOGLIK.FUN(par,X)                          ▷ Objective function
14:     n ← $nobs
15:     cond_prob ← mshape(NA, n, 1)
16:     for t = 2...T do
17:         cond_prob[t] ← COND.PROB (par, X[t], X[t-1])
18:     end for
19:     return log(cond_prob)
20: end procedure
21: mle LOGLIK.FUN (par0, X)                             ▷ MLE maximizer
```

**end**

Therefore a set of *Gretl* Econometric Software functions is introduced in order to perform time series estimation of one–lagged univariate INAR models with Poisson and Negative Binomial marginals.

Pseudo code depicted in Algorithm 1 explains briefly the procedure involving the numerical computation of CML estimates related with both P–INAR(1) and NB–INAR(1) models.

## *4. Simulation Study*

In this section a small–scale Monte Carlo experiment is carried out generating synthetic data at each step from a P–INAR(1) and an NB–INAR(1) processes. Then we computed parameter estimates along with their respective standard errors. Experiments were run with Gretl (2018d) software and a total of 500 simulations were performed. The setting scheme consists of fixing the $\alpha$ parameter at value $\alpha = 0.7$, letting vary the series length in $n = 200, 400, 800$. The P–INAR(1) case is depicted in table 1, while for the NB-INAR(1) model the scale parameter $\beta$ is fixed and $\gamma$ varies in $\{5, 10\}$, as shown in table 2. In conclusion, in figure 1 is depicted a summary of the computational cost of both the CML procedures. Conditional Maximum Likelihood estimation of model parameters is performed in each replication and in each table are compared Monte Carlo mean, Monte Carlo standard error and standard deviation of the Monte Carlo mean. In the Negative Binomial case Nelder Mead simplex method has been adopted as procedure to find the maximum value of the log–likelihood. When $\gamma = 5$, the mean of estimated parameters of the NB–INAR(1) are close to the true values and, as expected, the bias reduces when sample size increases. The same happens with the mean of estimated standard error and the Monte Carlo standard deviation, for each parameter the two estimates are close each other. But it is worth to notice that the $\gamma$ estimator produces high standard errors in each setting and, in general, Conditional Maximum Likelihood estimator shows an evident bias for the parameter $\gamma$ with increasing values, even with large sample sizes. This case is related to distributions with a probability concentration far from the left boundary of the support, thus zero event occurrences are rare. But, in the same framework, estimation of $\alpha$ and scale parameter $\beta$ are close

to the true values and standard errors results are comparable with the other configurations. Moreover, it follows that when the value of shape parameter increases, the skew of the marginal distribution changes less and less while the low count events become very rare.

*Table 1. Monte Carlo summary of CML estimates, P–INAR(1) model.*

| | | $\alpha$ | $\lambda$ | $\alpha$ | $\lambda$ | $\alpha$ | $\lambda$ |
|---|---|---|---|---|---|---|---|
| True Parameter | | 0.7 | 1 | 0.7 | 3 | 0.7 | 5 |
| n=200 | MC mean | 0.6989 | 1.0071 | 0.6986 | 3.0172 | 0.6970 | 5.0136 |
| | MC s.d. | 0.0353 | 0.1186 | 0.0316 | 0.3215 | 0.0312 | 0.5210 |
| | s.d. MC | 0.0337 | 0.1187 | 0.0317 | 0.3244 | 0.0313 | 0.5286 |
| n=400 | MC mean | 0.6981 | 1.0051 | 0.7008 | 2.9837 | 0.6986 | 5.0163 |
| | MC s.d. | 0.0241 | 0.0837 | 0.0224 | 0.2187 | 0.0208 | 0.3481 |
| | s.d. MC | 0.0236 | 0.0835 | 0.0221 | 0.2251 | 0.0219 | 0.3690 |
| n=800 | MC mean | 0.6985 | 1.0060 | 0.6990 | 3.0045 | 0.6996 | 5.0059 |
| | MC s.d. | 0.0170 | 0.0601 | 0.0154 | 0.1612 | 0.0154 | 0.2616 |
| | s.d. MC | 0.0166 | 0.0586 | 0.0156 | 0.1588 | 0.0153 | 0.2595 |

*Table 2. Monte Carlo summary of CML estimates, NB–INAR(1) model.*

| | | $\alpha$ | $\gamma$ | $\beta$ | $\alpha$ | $\gamma$ | $\beta$ |
|---|---|---|---|---|---|---|---|
| True Parameter | | 0.7 | 5 | 2 | 0.7 | 10 | 2 |
| n=200 | MC mean | 0.6966 | 5.4446 | 2.1871 | 0.6978 | 10.4605 | 2.1027 |
| | MC s.d. | 0.0301 | 2.2843 | 0.9135 | 0.0322 | 3.2153 | 0.6625 |
| | s.d. MC | 0.0315 | 2.2848 | 0.9029 | 0.0309 | 3.8199 | 0.7644 |
| n=400 | MC mean | 0.6982 | 5.1203 | 2.0641 | 0.6999 | 10.2773 | 2.0509 |
| | MC s.d. | 0.0235 | 1.3345 | 0.5470 | 0.0231 | 2.5169 | 0.5039 |
| | s.d. MC | 0.0220 | 1.2826 | 0.5200 | 0.0224 | 2.3679 | 0.4718 |
| n=800 | MC mean | 0.6988 | 5.1733 | 2.0667 | 0.6979 | 10.0901 | 2.0185 |
| | MC s.d. | 0.0152 | 0.8362 | 0.3387 | 0.0168 | 1.9481 | 0.3928 |
| | s.d. MC | 0.0155 | 0.8863 | 0.3546 | 0.0166 | 1.7098 | 0.3399 |

Maximum of the log–likelihood function lies in a flat–top parabolic plateau that is sloping towards larger values of both $\gamma$ and $\beta$ parameters and, even though this valley may be easy to find, convergence to the global maximum

is difficult. For this reason the use of a derivative–free optimization algorithm such as the Nelder Mead has been adopted. When true value of $\gamma$ is low, the size of the plateau is modest and optimization methods are still capable to find the maximum with enough precision. But with high values of $\gamma$ the plateau enlarges drawing a flat area having elliptical contour.

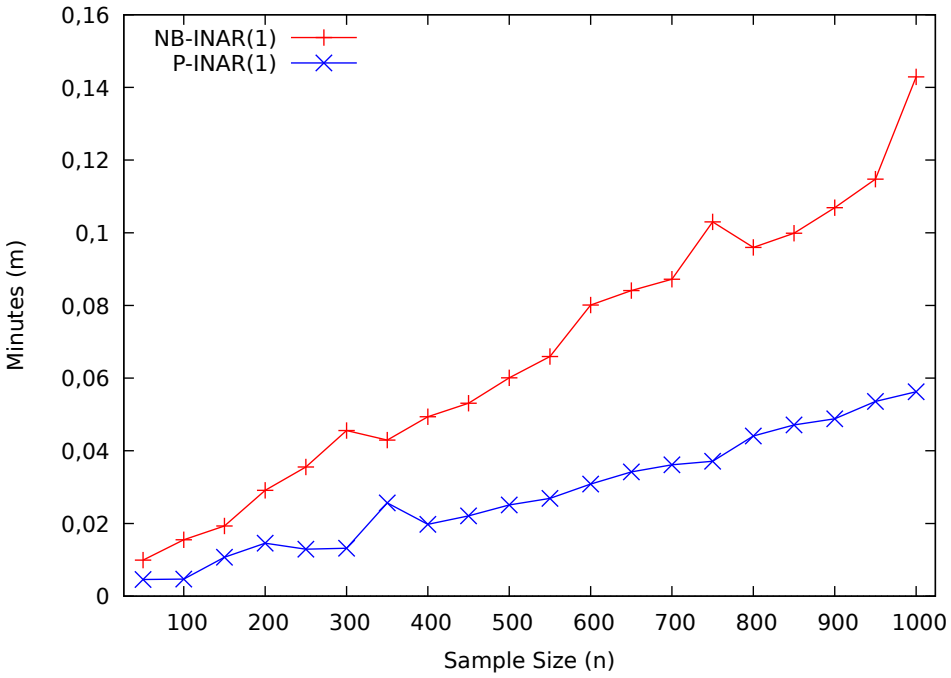



*Figure 1. Process time of CML estimators of Poisson and Negative Binomial INAR(1) processes over increasing sample size.*

## *5. Concluding Remarks*

In this paper a set of *Gretl* Econometric Software functions that perform time series estimation of one–lagged univariate INAR models with Poisson and Negative Binomial marginals has been introduced.

The NB–INAR(1) is a suitable process to model overdispersed and zero inflated data but, due to the presence of a flat plateau, optimization methods lack of precision in finding the maximum value when the true parameter $\gamma$ is

higher than 10. However, a large shape parameter is unrealistic in a context of low counts and could be seen as a signal of a degenerate solution.

When innovations are generated from a Poisson then the estimation algorithm is relatively fast since it consists on the convolution between a Binomial and a Poisson random variables. At the contrary, the higher execution time related with the computation of the NB–INAR(1) innovations probability mass function slows down the main algorithm.

*Acknowledgements:* Author would like to thank professors F. Di Iorio and R. Lucchetti for their the useful suggestions and for providing the hypergeometric function in Gretl.

## *References*

Alzaid A., Al-Osh M.A. (1988) First-Order Integer-Valued Autoregressive (INAR (1)) Process: Distributional and Regression Properties, *Statistica Neerlandica*, 42(1),53-61.

Du J.G., Li,Y. (1991) The integer valued autoregressive (INAR(p)) model, *Journal of Time Series Analysis*, 12(2),129-142.

Grunwald G.K., Hyndman R.J., Tedesco L., Tweedie R.L. (2000) Theory & methods: Non-gaussian conditional linear AR(1) models, *Australian & New Zealand Journal of Statistics*, 42(4), 479-495.

Jacobs P.A., Lewis P.A.W. (1983) Stationary discrete autoregressive-moving average time series generated by mixtures, *Journal of Time Series Analysis*, 4(1), 19-36.

Leonenko N., Savani V., Zhigljavsky A. (2007) Autoregressive negative binomial processes, *Ann de l'ISUP LI*, 51, 25-47.

McKenzie E. (1985) Some simple models for discrete variate time series, *JAWRA Journal of American Water Resources Associations*, 21(4), 645-650.

Pedel X., Karlis D. (2013) On composite likelihood estimation of a multivariate INAR(1) model, *Journal of Time Series Analysis*, 34(2), 206-220.

Pegram G.G.S. (1980) An autoregressive model for multilag markov chains, *Journal of Applied Probability*, 17(2), 350-362.

Steutel F.W., van Harn K. (1979) Discrete Analogues of Self-Decomposability and Stability, *Annals of Probability*, 7(5), 893-899.

Zeger S.L. (1988) A regression model for time series of counts, *Biometrika*, 75(4), 621-629.

# The "Calzolari" package (1.0 version)

Giulio Palomba*

*Abstract:* The `Calzolari.gfn` package for `Gretl` is designed to produce special time series. Specifically, given a set of $k$ explanatory variables $X_1, X_2, \ldots, X_k$ and $k$ coefficients associated to them, it is possible to create a dependent variable $y$ in order to obtain an estimation output where all coefficients are those previously assigned by the user. The explanatory variables can be lagged variables or deterministic time trends. The generation of the dependent is possible through an unusual application of indirect estimation calibration algorithm. The aim of the package is to generate as quickly as possible a large number of different data sets with different expected estimation results. This practice could be very useful especially for didactical purposes because it can provide to each student a different exam text.

*Keywords:* Round numbers, Indirect inference, Calibration.

## *1. General remarks*

The `Calzolari.gfn` package for `Gretl` could be a useful tool to create special data sets that can be used in econometric exams and sometimes in simulations. The main characteristic of these data sets is returning estimated parameters which all were previously decided by the user.

Nowadays, several examiners use of Monte Carlo methods to generate exam datasets in order to obtain an exam version customized for each student; thus, students are forced to solve individually the assigned exercises and, maybe, they do not copy from someone sitting nearby. Time series with a known *data generating process* (hereafter, DGP) sometimes could also be a useful tool in scientific activities, especially in simulations.

The package is named `Calzolari` because it is inspired by Calzolari (2017), an article where the author illustrates an indirect estimation calibration algorithm and provides some examples where the method is applied to several econometric models. Professor Giorgio Calzolari is the one who have introduced this type of data sets in teaching activity and Econometrics exams. He

*Università Politecnica delle Marche, g.palomba@univpm.it

proposes some calibration algorithms, typical of indirect estimation methods, that solve the issue of student's initial random errors and reduce evaluation bias. In practice, such procedure allows students to evaluate if their initial estimates were carried out correctly, since the obtained numbers should be round. Moreover, students can immediately correct their initial estimates when the results are not round numbers, thus reducing any possible evaluation bias.

The 1.0 version of the package is able to generate dependent variables for the following univariate time series models:

1. linear dynamic regression models (OLS, DL or ADL),
2. linear time series models (ARMA or ARMAX),
3. regression models with conditional heteroskedasticity (GARCH).

The `Calzolari.gfn` package could be used

- within a `.gdt` data file exploiting the variables included (and also the sample size),
- in a `Gretl` script, by simply using the standard `nulldata` and `setobs` commands.

This paper proceeds as follows: section 2 presents all the functions in the package, section 3 briefly discusses the drawbacks of the calibration procedure, while section 4 provides an example of a time series generation and finally section 5 concludes with further of improvement proposals.

## *2. The package*

The actual version of `Calzolari.gfn` package is composed by a public function and 11 private functions.

### *2.1. The public function*

The public function has the following inputs:

```
Calzolar(int Model[1:14:1] {list of the available models},
      list Xvars "List of regressors",
      matrix Beta "Target coefficients",
      scalar se[0.000001::1] "Std. Error of the regression",
      int decimals[0:6:3] "Number of decimal digits",
      int maxiter[0::0] "if 0: one-shot dataset production
          (not adjusted for decimals)")
```

The return type is a `series` that consists of the generated time series. This function has 6 arguments, namely

1. `int Model[1:14:1]` indicates the time series econometric model. Table 1 shows that each number from 1 to 14 corresponds to a different model. In several lines in the table the model changes according the presence of exogenous explanatory variables or lagged explanatory variables.

*Table 1. Econometric models*

| user's choice | basic model | exogenous variables | lagged variables |
|---|---|---|---|
| `Model=1` | OLS on a constant | OLS | DL($k$) |
| `Model=2` | AR(1) | ARX(1) | ADL($1, k$) |
| `Model=3` | AR(2) | ARX(2) | ADL($2, k$) |
| `Model=4` | AR(3) | ARX(3) | ADL($3, k$) |
| `Model=5` | MA(1) | ARMAX(0,1) | |
| `Model=6` | MA(2) | ARMAX(0,2) | |
| `Model=7` | MA(3) | ARMAX(0,3) | |
| `Model=8` | ARMA(1,1) | ARMAX(1,1) | |
| `Model=9` | ARMA(2,1) | ARMAX(2,1) | |
| `Model=10` | ARMA(1,2) | ARMAX(1,2) | |
| `Model=11` | ARMA(2,2) | ARMAX(2,2) | |
| `Model=12` | GARCH(1,1) | | |
| `Model=13` | AR(1)-GARCH(1,1) | | |
| `Model=14` | AR(2)-GARCH(1,1) | | |

Table 1 clearly shows that the maximum lag order is 3. This is valid also for ADL models because $k \in \{0, 1, 2, 3\}$. The number of lags in ARMA, ADL and GARCH models could be easily augmented in this

package but, from the technical point of view, this will increase the number of the available model specifications. In this context, the question that arises is if it is really useful. Hence, a superior limit of 3 lags can be adequate. This choice is motivated by the fact that, in practice, a model with 3 lags can be considered sufficiently general. Moreover, models with a superior number of lags often contain a great amount of parameters and this "lack of parsimony" makes them not so common (even rare) in practical situations. For the same reasons, a more stringent limit is adopted for the ARMA and GARCH data generation.
The generated models include the constant, while any other deterministic time trend (linear, quadratic, *etc.*) can be inserted as explanatory variable. The only exception is the case of GARCH(1,1) models where time trends lead to severe numerical problems due to "non horizontality" of the generated dependent variable. It is also possible to specify the conditional mean of GARCH(1,1) models with exogenous variables. In this case the stationarity of such variables is obviously required. Regressors in variance can not be used in the current version of `Calzolari.gfn` package.

2. `list Xvars` is the list of exogenous explanatory variables in the conditional mean equation of all models. As we claimed before, the constant is mandatory therefore it is excluded from this list. Dummy variables can be included, as well as lagged variables and time trends. For example, a linear trend can be inserted simply using the command `time`;

3. `matrix Beta` is the column vector that contains all the parameter values previously decided by the user. The first element is always the coefficient associated to the model constant. The order of the following parameters depends upon the choice made with regard to the econometric model. In particular, if the model includes $p$ lags of dependent variable (AR or ARMA models), the user should fix the values of the parameters from the second to the $(p+1)^{\text{th}}$ position. Similarly, if a moving average of order $q$ is present (MA or ARMA models), parameters from the $(2+p)^{\text{th}}$ to the $(2+p+q)^{\text{th}}$ position should be set by the user. Finally, all the remaing positions from the $(3+p+q)^{\text{th}}$

onwards, should fix all the coefficients associated to all the exogenous explanatory variables contained in the `Xvars` list;

4. `scalar se[0.000001::1]` corresponds to the desired standard error of the regression, must be positive, and the default value is 1. The estimated S.E. of the regression lies around the value of this scalar. The only exception is in the case of OLS model without any lagged variable, where the estimated S.E. of the regression correponds exactly to the value imposed by the user;

5. `int decimals[0:6:3]` is an integer that sets the number of decimal digits in the generated dependent variable. The minimum value is 0, while the maximum is 6. The default value is 3.

6. `int maxiter[0::0]` is an integer that sets the grid of increments during the calibration phase. The default value is 0, which implies that the dependent variable is created 'One-shot' without any calibration due to the imposition of a prefixed number of decimal digits. If `maxiter` is set to a positive number, the calibration is activated and the dependent variable is created with a number of decimal digits given by the argument `decimals`. The discussion of the calibration algrithm is presented further in section 3.

### *2.2. The private functions*

The private functions are

1. `scalar CheckXLags(strings vn, scalar *laggedseries)`.
   This function calculates the maximum number of lags of any explanatory variable. The only input is a a set of strings `vn`, while `*laggedseries` is an address that counts the number of exogenous lagged variables.
   It is worth noting that this function requires that all the lagged variables have to be labelled with the suffix "`_1`", "`_2`" or "`_3`", since 3 is the maximum admissible lag order in the `Calzolari.gfn`.

2. `bundle Bench(int EcModel, list Xvars, matrix Beta, scalar se, scalar Ndigits, int maxiter)`

This function sets the bundle and the inputs are exactly those of the public function `Calzolar()`. The instruction provided by its arguments are sufficient to set/extrapolate other important objects, as:

- the sample size,
- the number of all exogenous variables (including the constant),
- the possible presence of a linear time trend,
- the estimation method (OLS regression, conditional maximum likelihood or GARCH ML),
- the number of ARMA lags (if any),
- the type of data generation ('One shot' or calibrated).

Moreover, the bundle defines/initializes other objects used in the package.

3. `series Trunc(series Data, scalar Ndigits)`
   This function trasforms the data by truncating them into the number of decimals previously decided by the user. The arguments are the time series and the number of desired decimal digits, while the output is the truncated time series.

4. `scalar RMSE(matrix ref, matrix target)}`
   This is a very simple function that calculates the root mean squared error (herafter RMSE) between the two column vectors provided by arguments.

5. `matrix EstimateModel(series y, list Xvars, string method, scalar plag, scalar qlag, bool PrintResults)`
   This function carries out the model estimation. The return type is the estimated parameter vector, while the inputs are

   - the dependent variable (`y`) generated by the package,
   - the complete list of explanatory variables `Xvars` taken from the bundle,

- the estimation method taken from the bundle,
- the ARMA parameters (scalars `plag` and `qlag`) taken from the bundle,
- the boolean `PrintResult` (if 1, the estimation is printed, 0 otherwise).

6. `series GenrModelError(bundle *B)`
   This function generates the time series of model errors $\varepsilon_t \sim N(0, \sigma^2)$. The argument is the bundle from which it takes the standard error of the regression (parameter $\sigma$) and the number of decimal digits.

7. `series OneShotOLS(matrix par, list Xvars, bundle *B)`
   This function produces a 'One shot' dependent variable with an unlimited number of decimals. It is the time series that always returns the desired vector of parameters `par` and the standard error of the regression in the estimation output. The inputs are the parameter vector, all the explanatory variables and the bundle. This function is invoked when in the public function the input `EcModel` is set to 1.

8. `OneShotTS(matrix par, list Xvars, bundle *B)`
   This function is similar to `OneShotOLS`, but it is invoked when `EcModel` has an integer value between 2 and 11. Here the estimated standard error of the regression is approximately the one provided in the pubic function.

9. `OneShotGARCH(matrix par, list Xvars, bundle *B)`
   This function is similar to `OneShotTS`, but it is invoked only when `EcModel` has an integer value between 12 and 14.

10. `series AdjustData(list Xvars, bundle *B)`
    This function consists in the calibration mechanism that produces the generated dependent variable with the number of decimal digits imposed by the user. The inputs are the list of the exogenous variables and the bundle, while the return type is a time series adjusted with a limited number of decimals. The calibration is described in section 3.

11. `scalar Info(bundle *B)`
    This function is crucial because it publishes all the information about the choices adopted by the user and provides a preliminary check about of the feasibility of the data generation. In practice, it compares two numbers, namely $S_1$ and $S_2$. On the one hand, $S_1 = 1 + p + q + k + g$ is the total amount of parameters requested by each selected model in the public function, where $p$ and $q$ are the AR and MA lags, $k$ is the numerosity of all the exogenous explanatory variables (lagged, dummies and time trends included), and $g = 3$ is the number of parameters in the GARCH(1,1) equation. On the other hand $S_2$ is the number of rows in the parameter vector chosen by the user. The data generation proceeds only when the equality $S_1 = S_2$ holds. Otherwise a warning message about the discrepancy appears and the data generation does not start.
    The return type is a scalar that is 1 when the generation is made and zero otherwise. The input is only the bundle.

*3. Calibration*

Sometimes the estimated coefficients obtained from the data sets generated by the `Calzolari.gfn` package could be slightly different from the ones prefixed by the user. This inconvenience is avoided when the 'One-shot' generation is performed. In this case the dependent variable is generated with an unlimited number of decimal digits, which allows the estimated coefficient to converge to the values previously imposed by the user.

When the user aims to obtain a dependent variable with a prefixed number of decimal digits, the 'One-shot' estimation is modified/adjusted by a calibration mechanism via the private function `AdjustSeries()`. The entire mechanism of data generation and calibration is carefully described by Calzolari (2017). It suffices to mention that the calibration proceeds observation-by-observation, with a very simple grid search algorithm. Specifically:

(a) the procedure starts using the 'One shot' generated time series. Once the model estimation is performed, the "real" parameter vector $\beta$ is obtained;

(b) once the decimal digits in the dependent variable have been reduced to the prefixed number, the estimation returns $\tilde{\beta}_0 \neq \beta$;

(c) the integer `maxiter`$= m > 0$ in public function defines a sequence of positive integers ranging from $-m$ to $m$ (zero is excluded). If $n$ is the number of desired decimal digits, the grid of increments is given by the $2m \times 1$ vector

$$x_{m,n} = 10^{-n} \begin{bmatrix} -m & -m+1 & \dots & -1 & 1 & \dots & m-1 & m \end{bmatrix}. \quad (1)$$

The calibration algorithm consists in allpying the $2m$ increments in $x_{m,n}$ to all the $t = 1, 2, \dots, T$ observations in the truncated generated variable in order to calculate the RMSE($\tilde{\beta}_t, \beta$). This is a time consuming mechanism that updates the generated series when

$$\text{RMSE}(\tilde{\beta}_t, \beta) < \text{RMSE}(\tilde{\beta}_\tau, \beta),$$

with $0 < \tau < t$. Clearly, the search of the minimum RMSE throughout the sample is repeated until the equality

$$\text{RMSE}(\tilde{\beta}^*, \beta) = \text{RMSE}(\tilde{\beta}_1, \beta) = \text{RMSE}(\tilde{\beta}_T, \beta)$$

is obtained.

(d) The generated time series $y$ is the one corresponding to the minimum RMSE, and the final vector of parameters $\tilde{\beta}^*$ is immediately estimated and printed.

The procedure often can fail because $\tilde{\beta}^* \neq \beta$ happens. In this case, the generated variable should be discharged and all the generation and calibration must be repeated. The suggestion is to generate a large number of variables until the "right" data set is obtained (it is sufficient to insert the public function inside a `loop`).

The probability of failure is connected with the user preferences. In particular, the following choices could be of some help in reducing such probability:

1. augmenting the sample size,

2. augmenting the number of decimal digits,

3. augmenting the number of iterations (variable `maxiter`),

4. reducing the number of explanatory variables.

*4. An example*

Suppose your available data is

```
periodiciy: 12, maxobs: 836
observations range: 1948:01-2017:08

Listing 2 variables:
 0) const     1) indpro    2) unrate    3) infl
```

where `indpro` is the american monthly industrial production index, `unrate` is the unemployment rate and `infl` is the monthly inflation rate.

In order to generate the dependent variable, the following intructions are required:

```
list X = indpro infl
scalar stderr=1
scalar digits=4
```

The first instruction sets the list of explanatory variables, the second imposes a unitary value to the S.E. of regression and the third selects the number of decimal digits in the model error time series.

As already mentioned in section 2.2, the list could contain, among others, lagged variables with the proper suffix (*i.e.* `indpro_1`, `indpro_2`, `indpro_3`) or a time linear trend (by simply using the `Gretl` command `time`)

Once the primary instruction was provided, the user has two possible choices: 'One-shot' data generation or calibrated data generation.

*4.1. 'One-shot' data generation*

Suppose that the aim of the user is to produce a 'One-shot' generation of a dependent variable for a standard OLS model with exogenous regressors given by the list `X`. First of all the column vector of parameters must be set via the instruction

```
matrix Target={2;0.5;-0.25}
```

The first parameter is automatically assigned to the model constant, while the others are those related to the variables contained in `X`, therefore the estimates of the OLS model should be exactly

$$\hat{y}_t = 2 + 0.5\texttt{indpro}_t - 0.25\texttt{infl}_t.$$

The 'One-shot' data generation is performed via the instructions

```
maxiter=0
series y=Calzolar(1,X,Target,stderr,digits,maxiter)
```

The first assignement sets to zero the lenght of the calibration grid, so the calibration algorithm can not operate and the data generation is 'One-shot'. The second assignement invokes the public function. A time series `y` is generated considering the user preferences, namely model 1 (OLS), list `X` of regressors, vector `Target` of parameters, 4 decimal digits in error time series and 'One-shot' estimation (`maxiter=0`).

The procedure returns the output presented in Table 2 in Appendix.

Figure 4.1 shows that the data description `generated dependent variable (One shot)` appears in the `.gdt` file, associated to the variable `y` .

'One-shot' generation

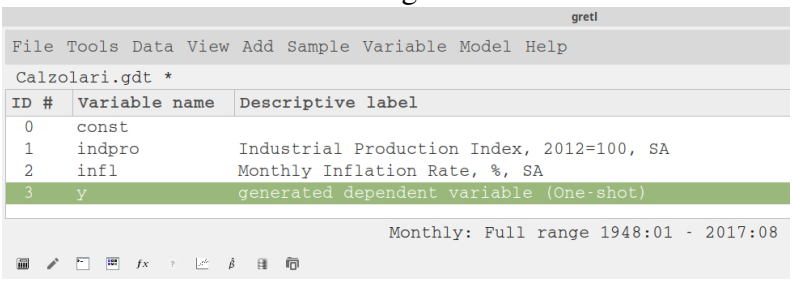



*Figure 1. Updated .gdt file*

Basically, the generation output is composed by three parts:

1. general information regarding details of the data generation,
2. the estimated model,

3. the time elapsed for data generation.

### *4.2. Calibrated data generation*

The calibrated data generation is used when the observations in the dependent variable must have a finite number of decimal digits. In order to produce an example with the same instructions assigned in section 4.1, it is sufficient to augment the value of `maxiter`. For instance, suppose that the following assignements are given.

```
maxiter=4
y=Calzolar(1,X,Target,stderr,digits,maxiter)
```

In this case the package performs the dependent variable generation with the same inputs used in section 4.1 (OLS model, `list X`, unit S.E. of the regression and $\beta = \begin{bmatrix} 2 & 0.5 & -0.25 \end{bmatrix}'$). The only difference consists of 4 decimal digits instead of an unlimited number. This choice triggers a calibration mechanism based on a grid search, where the grid is built according to equation (1). In this example `maxiter` is 4, hence the grid contains the sequence -0.0004, -0.0003, -0.0002, -0.0001, 0.0001, 0.0002, 0.0003 and 0.0004 (8 elements).

After several attempts, the output in Table 3, reported in Appendix, is obtained. The calibrated generation output is composed by four parts: the three parts already seen in Table 2 and one part where `Final estimation` appears, followed by

1. `iteration` is the number of times the grid search is repeated over the sample,

2. `sample observation` is the observation number whose increment determines a RMSE reduction,

3. `increment` is the value of the grid applied to the sample observation,

4. `RMSE` is the value of the Root Mean Squared Error.

After data generation, the variable `y` with data description `generated dependent variable with 4 decimal digits` appears in the `.gdt` file, as shown in Figure 4.2.

Calibrated generation

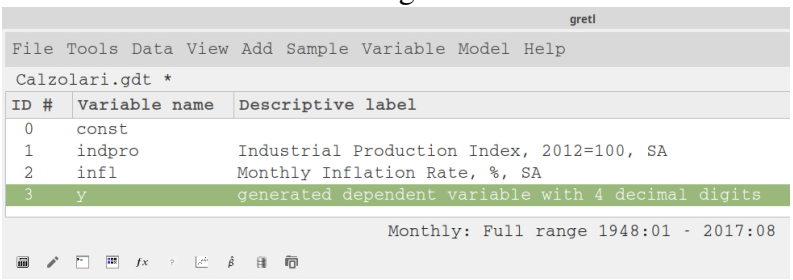



*Figure 2. Updated .gdt file*

As mentioned in section 3, the calibration mechanism is time consuming and often can fail because the final estimated coefficients are not exactly those imposed by the user. In this case the dependent variable is unaccurate, and the user has to discharge the generated variable and proceed with a new variable generation (a `loop` is suggested).

The probability of failure is connected with the user preferences. One can reduce such probability by

1. augmenting the sample size,
2. reducing the number of explanatory variables,
3. augmenting the variance (or S.E),
4. augmenting the number of decimal digits,
5. augmenting the number of increments in the grid.

## *5. Concluding remarks and further developments*

`Calzolari.gfn` package for `Gretl` allows to produce special data sets in order to obtain estimation outputs with previously decided coefficient values. This is possible for 14 time series models with two types of data generation, 'One-shot' and calibrated. In the next future, after fixing eventual bugs, the package could be enriched: for time series models, it is possible to remove the maximum lags constraint or to extend the choice to seasonal AR or MA.

Also, as suggested by Calzolari (2017), possible extensions can be

1. linear regression models for cross section data,
2. linear panel data model with random effects,
3. logit models.

Another improvement is the multivariate generalization of the package: this is already possible for simultaneous equations (estimation via OLS, 2SLS and 3SLS, see Calzolari, 2017), while for VAR models some promising preliminary results were obtained.

## *6. Appendix*

*Table 2. 'One-shot' generation output*

```
================================================================================
DEPENDENT VARIABLE DATA GENERATION
OLS model:  y=2+0.5*indpro-0.25*infl+e,
where y is the dependent variable and e is the error/innovation.
T=836 observations, S.E. of regression=1.
The model includes the constant.
The total amount of other exogenous explanatory variables is 2 (0 lagged).
Dependent variable is generated One Shot.

================================================================================

One Shot Model Estimation:

OLS, using observations 1948:01-2017:08 (T = 836)
Dependent variable: y

              coefficient   std. error   t-ratio      p-value
  -------------------------------------------------------------
  const        2.00000      0.0831144     24.06       1.49e-97 ***
  indpro       0.500000     0.00117064   427.1        0.0000   ***
  infl        -0.250000     2.58640       -0.09666    0.9230

Mean dependent var      31.11316   S.D. dependent var   14.83771
Sum squared resid       833.0000   S.E. of regression   1.000000
R-squared               0.995469   Adjusted R-squared   0.995458
F(3, 85)                91499.31   P-value(F)           0.000000
Log-likelihood         -1184.730   Akaike criterion     2375.460
Schwarz criterion       2389.646   Hannan-Quinn         2380.898
rho                     0.025007   Durbin-Watson        1.949693

Time elapsed: 0.00 seconds.
```

*Table 3. Calibrated data generation output (final attempt)*

```
================================================================================

DEPENDENT VARIABLE DATA GENERATION

OLS model:  y=2+0.5*indpro-0.25*infl+e,

where y is the dependent variable and e is the error/innovation.
T=836 observations, S.E. of regression=1.
The model includes only the constant.
The total amount of other exogenous explanatory variables is 2 (0 lagged).
Dependent variable is generated with 4 decimal digits
  using increments from -0.0004 to 0.0004.
================================================================================


Final Estimation:
iteration=1, sample observation=1, increment=-0.0001: RMSE=6.51336e-07
iteration=1, sample observation=1, increment=-0.0002: RMSE=1.1966e-07
iteration=1, sample observation=2, increment=0.0001: RMSE=2.35012e-08
iteration=1, sample observation=5, increment=0.0001: RMSE=1.49537e-08
iteration=1, sample observation=5, increment=0.0002: RMSE=7.60218e-09

     OLS, using observations 1948:01-2017:08 (T = 836)
     Dependent variable: y

               coefficient   std. error   t-ratio      p-value
  -------------------------------------------------------------
  const        2.00000      0.0831144      24.06       1.49e-97 ***
  indpro       0.500000     0.00117064    427.1        0.0000   ***
  infl        -0.250000     2.58640        -0.09666    0.9230

Mean dependent var      31.11316   S.D. dependent var   14.83771
Sum squared resid       833.0000   S.E. of regression   1.000000
R-squared               0.995469   Adjusted R-squared   0.995458
F(3, 85)                91499.31   P-value(F)           0.000000
Log-likelihood         -1184.730   Akaike criterion     2375.460
Schwarz criterion       2389.646   Hannan-Quinn         2380.898
rho                     0.065932   Durbin-Watson        1.865239

Time elapsed: 2.88 seconds.
```

## *References*

Calzolari G. (2017) Econometrics exams and round numbers: Use or misuse of indirect estimation methods?, *Communications in Statistics - Simulation and Computation*, doi: 10.1080/03610918.2017.1348514.

# Fixed effects in Early Warning Systems: A penalized Maximum Likelihood approach

Claudia Pigini*

*Abstract:* Binary panel logit models have proved to be simple and effective tools to build Early Warning Systems (EWS) for banking crises. But because crises are rare events, the estimation of EWS does not usually account for country fixed effects, so as to avoid losing all the information relative to countries that never experience a crisis. I propose a penalized maximum likelihood estimator for fixed-effects logit-based EWS where all the observations are retained. I show that including country effects, while preserving the entire sample, greatly improves the predictive power of EWS with respect to the pooled and standard fixed effects models. I also consider a dynamic formulation of EWS to separately predict the crisis *persistence* and *entry* rates. The bias corrected estimator as well as the Receiver Operating Characteristic (ROC) curve and the Area Under ROC (AUROC) are implemented in gretl.

*Keywords:* Banking crisis, Bias reduction, Separation problem.

## *1. Introduction*

Logit models have proved to be simple and effective tools to build Early Warning Systems (EWS) for banking crises. Their predictive power is employed to generate accurate out of sample warning signals and their specification as binary choice models offers a clear interpretation of the drivers of banking crises (Demirgüç-Kunt and Detragiache, 1998, Davis and Karim, 2008, Caggiano at al., 2016).

When logit-based EWS are built on panel data, permanent country unobserved heterogeneity could be accounted for and supposedly improve their predictive power. But because crises are rare events, the estimation of EWS does not usually account for country fixed effects, so as to avoid losing a sizable number of countries in the dataset. This is due to the separation problem, because of which the Maximum Likelihood (ML) estimator for the intercept

*Università Politecnica delle Marche, MoFiR and GLO, c.pigini@univpm.it

of a country that never experiences a crisis does not exist and thereby prevents it from contributing to the estimation. Unfortunately, in general, retaining the whole sample of subjects excluding the intercepts for those with all zeros (or ones) in the dependent variable will lead to a biased ML estimator (Heinze and Schemper, 2002).

In this paper, I propose estimating a fixed-effects logit-based EWS where all the observations are retained by a Penalized Maximum Likelihood (PML) approach. The bias reduction technique is after Firth (1993), who defined the ML estimator as the solution to a modified score function. This approach was adapted to the separation problem in the binary logit model by Heinze and Schemper (2002) and recently applied to the prediction of civil wars by Cook et al. (2018). Under the assumption that every country will experience a crisis as $T \to \infty$, the bias of the ML estimator is reduced from $O(T^{-1})$ to $O(T^{-2})$, where $T$ is the number of time occasions.

By using an unbalanced panel dataset of 129 countries from 1982 to 2017 and by defining crises events as in Laeven and Valencia (2018), I show that including country effects, while preserving the entire sample, greatly improves the predictive power of EWS with respect to the pooled and standard fixed effects models. I also consider a dynamic formulation of EWS by including the lagged dependent variable among the set of covariates, which has been shown to be highly relevant in crises predictions (Antunes et al., 2018). In this framework, it might be crucial to separately predict the crisis *persistence* and *entry* rates of countries, that is the conditional probability of a crisis occurring at time $t$ given the crisis status in $t-1$, rather than the marginal probability of the crisis occurrence.

The bias corrected estimator is implemented in `hansl`, whereas the Receiver Operating Characteristic (ROC) curves and the Area Under ROC (AUROC) to evaluate the model performance are estimated using the `gretl` function package `roc` by Summers (2017).

## 2. The Penalized Maximum Likelihood estimator

Consider a dataset of countries $i = 1, \dots, n$ observed for $t = 1, \dots, T$ years. The logit formulation

$$p_{it} = \frac{\exp\left[y_{it}(\boldsymbol{x}'_{i,t-1}\boldsymbol{\beta} + \alpha_i)\right]}{1 + \exp(\boldsymbol{x}'_{i,t-1}\boldsymbol{\beta} + \alpha_i)} \tag{1}$$

is the probability of country $i$ facing a banking crisis in year $t$ if the binary variable $y_{it}$ is equal to 1. Moreover, $\boldsymbol{x}_{i,t-1}$ is a set of covariates usually lagged by one or more periods, $\boldsymbol{\beta}$ is a vector of regression parameters and $\alpha_i$ is the country-specific intercept. If country $i$ never experiences a crisis, the ML estimator of $\alpha_i$ does not exist. Therefore, EWS are usually based on pooled models, as fixed-effects estimators would otherwise entail the loss of the observations for countries never facing a crisis.

Retaining the whole sample while excluding the $\alpha_i$ for countries with $y_{it} = 0$ for $t = 1, \dots, T$ will lead to a biased ML estimator. The PML estimator entails a reduction of the bias of the ML estimator for the parameters of interest from $O(T^{-1})$ to $O(T^{-2})$. Let $\boldsymbol{\eta}$ be the vector of country-specific intercepts excluding those relative to countries that never face a crisis and let $\boldsymbol{\theta} = (\boldsymbol{\beta}', \boldsymbol{\eta}')'$. The PML estimator of $\boldsymbol{\theta}$ is defined as the as the solution to the modified score equation

$$U^*(\boldsymbol{\theta}) \equiv U(\boldsymbol{\theta}) + \frac{1}{2}\mathrm{tr}\left[I(\boldsymbol{\theta})^{-1}\{\partial I(\boldsymbol{\theta})/\partial\boldsymbol{\theta}\}\right] = 0, \tag{2}$$

where $U(\boldsymbol{\theta})$ is the score for the log-likelihood based in (1) evaluated in $\boldsymbol{\theta}$ and $I(\boldsymbol{\theta})$ is the Information Matrix. The PML estimator $\hat{\boldsymbol{\theta}}$ can be easily obtained by Newton-Raphson. Standard errors can be obtained as the roots of the diagonal elements of the panel robust variance estimator.

## 3. Results

Elaborations are based on an unbalanced dataset of 129 countries followed for 36 years, from 1982 to 2017. The dependent variable is provided by Laeven and Valencia (2018), taking value 1 for the crisis occurrence and 0

otherwise. Their definition lead to 237 years of crisis, 64 of which are new episodes. The list of explanatory variables for the specification of the logit-based EWS (loosely based on the one adopted by Caggiano et al., 2016) as well as their source, mean and standard deviation in the sample are reported in Table 1.

*Table 1. Variable sources and descriptive statistics*

| | Source | Mean | SD |
|---|---|---|---|
| *Dependent variable* | | | |
| Crisis | Laeven and Valencia (2018) | 0.07 | 0.26 |
| *Explanatory variables* | | | |
| Real GDP growth(-1) | IFS | 3.87 | 4.23 |
| Log per capita GDP(-1) | WDI | 7.71 | 1.46 |
| Inflation(-1) | WDI | 5.94 | 386.8 |
| Real interest rate | IFS | 2.35 | 11.37 |
| M2 to foreign exchange reserves(-1) | WDI | 15.10 | 33.38 |
| Growth of real domestic credit(-1) | WDI | 16.08 | 34.29 |
| Growth of net foreign assets to GDP(-1) | WDI | 1.00 | 2.24 |

IFS: International Financial Statistics (International Monetary Fund). WDI: World Development Indicators (World Bank). Inflation is the growth rate of the GDP deflator.

The performance of the PML estimator for the fixed-effects logit model here proposed is compared with the performance of the ML estimator of the pooled and standard fixed-effects logit models. It is worth recalling that the latter entails a substantial observation loss. In particular the dataset shrinks from 3045 to 1757 observations, because countries that never experience a crisis episode are dropped form the sample.

Figure 1 depicts the ROC curves corresponding to the three models. The ROC curve plots the sensitivity, that is the true positive rate, against the false positive rate, also known as 1-specificity, for various thresholds (here 100) at which ones are predicted. A point on the curve indicates the percentage of non-crisis observations (on the $x$ axis) that will be classified incorrectly in order to have the corresponding percentage on the $y$ axis of crisis correctly predicted. Therefore, the Area Under the ROC curve (AUROC) is a measure for the model performance.

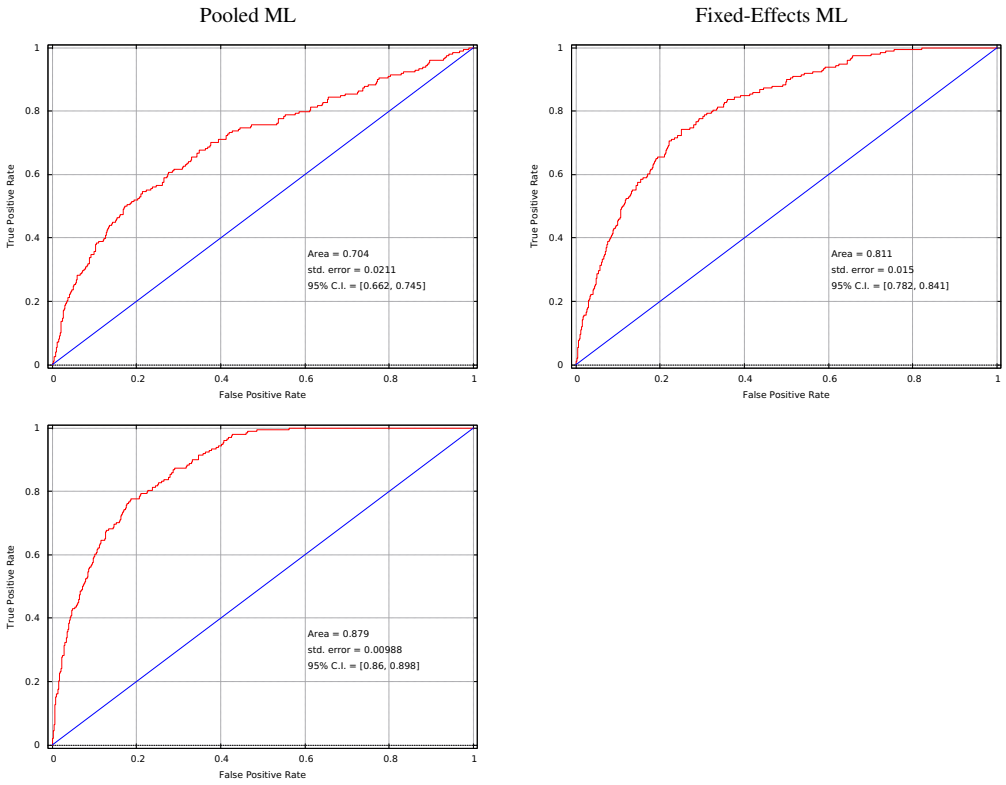



*Figure 1. Logit model: ROC curves for for the ML estimation of the pooled and fixed effects models and PML estimation of the fixed-effects model*

The values of the corresponding AUROC for the three models are reported at the top of Table 2. Results suggest that including fixed-effects substantially increases the model performance. It is worth noting that the AUROC is even higher for the biased reduced version of the fixed-effects estimator, which is based on the entire sample. This finding is also confirmed by the value of the McFadden $R^2$.

Table 3 reports the AUROC and McFadden $R^2$ for the dynamic logit model, where the lagged dependent variable is included in the set of covariates. Here I focus on the crisis *entry* and *persistence* rates, rather than on the marginal probability of the crisis occurrence. The *entry* rate is defined as the probabil-

ity of a crisis occurring at time $t$ conditional on there being no crisis at time $t-1$. Instead, the *persistence* rate is the probability of a crisis at time $t$ conditional on being already in a state of crisis at time $t-1$. It emerges that the proposed PML estimator of the fixed-effects logit outperforms both the ML estimators of the pooled and standard fixed-effects models, as testified by the values of the AUROC for the *entry* and *persistence* rates, and also picked up by the McFadden $R^2$.

*Table 2. Logit model: results for the ML estimation of the pooled and fixed effects models and PML estimation of the fixed-effects model*

| | ML Pooled | ML Fixed effects | PML Fixed effects |
|---|---|---|---|
| # Obs. | 3045 | 1757 | 3045 |
| McFadden $R^2$ | 0.067 | 0.189 | 0.280 |
| AUROC | 0.704 | 0.811 | 0.879 |
| (95% CI) | (0.662, 0.745) | (0.782,0.841) | (0.860,0.898) |
| Max correctly predicted | 0.935 | 0.891 | 0.938 |
| Real GDP growth(-1) | -0.130*** | -0.107*** | -0.111*** |
| | [0.023] | [0.030] | [0.028] |
| Log per capita GDP(-1) | -0.040 | -0.625** | 0.100 |
| | [0.080] | [0.262] | [0.246] |
| Inflation(-1) | 0.000 | 0.000 | 0.000 |
| | [0.000] | [0.001] | [0.001] |
| Real interest rate | 0.012* | 0.010 | 0.013 |
| | [0.007] | [0.012] | [0.010] |
| M2 to foreign | 0.007*** | 0.021*** | 0.026*** |
| exchange reserves(-1) | [0.003] | [0.006] | [0.005] |
| Growth of real | 0.010* | 0.006 | 0.011 |
| domestic credit(-1) | [0.006] | [0.011] | [0.010] |
| Growth of net foreign | -1.127* | -0.557 | -1.167 |
| assets to GDP(-1) | [0.598] | [1.069] | [1.006] |

The model specification includes the explanatory variables listed in Table 1. *** p-value $< 0.01$; ** p-value $< 0.05$; * p-value $< 0.10$. Panel robust standard errors are in square brackets.

*Table 3. Dynamic logit model: results for the ML estimation of the pooled and fixed effects models and PML estimation of the fixed-effects model*

| | ML Pooled | ML Fixed effects | PML Fixed effects |
|---|---|---|---|
| # Obs. | 3045 | 1757 | 3045 |
| McFadden $R^2$ | 0.449 | 0.456 | 0.533 |
| AUROC *entry* | 0.618 | 0.720 | 0.825 |
| (95% CI) | (0.538, 0.698) | (0.653,0.787) | (0.782,0.867) |
| Max correctly predicted | 0.984 | 0.973 | 0.984 |
| AUROC *persistence* | 0.710 | 0.792 | 0.854 |
| (95% CI) | (0.667, 0.754) | (0.761,0.824) | (0.833,0.875) |
| Max correctly predicted | 0.950 | 0.914 | 0.950 |
| $\text{Crisis}_{t-1}$ | 4.449*** | 3.690*** | 3.512*** |
| | [0.170] | [0.179] | [0.156] |
| Real GDP growth(-1) | -0.042* | -0.020 | -0.024 |
| | [0.024] | [0.027] | [0.025] |
| Log per capita GDP(-1) | 0.032 | -0.069 | 0.369*** |
| | [0.056] | [0.218] | [0.130] |
| Inflation(-1) | 0.000 | 0.000 | 0.000 |
| | [0.000] | [0.000] | [0.000] |
| Real interest rate | 0.011** | 0.016* | 0.016*** |
| | [0.005] | [0.008] | [0.006] |
| M2 to foreign | 0.007*** | 0.023*** | 0.025*** |
| exchange reserves(-1) | [0.002] | [0.006] | [0.004] |
| Growth of real | 0.009* | 0.009 | 0.011** |
| domestic credit(-1) | [0.006] | [0.007] | [0.006] |
| Growth of net foreign | -0.493 | -0.366 | -0.687 |
| assets to GDP(-1) | [0.557] | [0.086] | [0.068] |

*** p-value $< 0.01$; ** p-value $< 0.05$; * p-value $< 0.10$. Panel robust standard errors are in square brackets. The model specification includes the explanatory variables listed in Table 1 and the lag of the dependent variable.

## *References*

Antunes A., Bonfim D., Monteiro N., Rodrigues P.M. (2018) Forecasting banking crises with dynamic panel probit models, *International Journal of Forecasting*, 34(2), 249-275.

Caggiano G., Calice P., Leonida L., Kapetanios G. (2016) Comparing logit-based early warning systems: Does the duration of systemic banking crises matter?, *Journal of Empirical Finance*, 37, 104-116.

Cook S.J., Hays, J.C., Franzese R.J. (2018) Fixed effects in rare events data: a penalized maximum likelihood solution, *Political Science Research and Methods*, 1-14.

Davis E.P., Karim D. (2008) Comparing early warning systems for banking crises, *Journal of Financial Stability*, 4(2), 89-120.

Demirgüç-Kunt A., Detragiache E. (1998) The determinants of banking crises in developing and developed countries, *IMF Staff Papers*, 45(1), 81-109.

Firth D. (1993) Bias reduction of maximum likelihood estimates, *Biometrika*, 80(1), 27-38.

Heinze G., Schemper M. (2002) A solution to the problem of separation in logistic regression, *Statistics in medicine*, 21(16), 2409-2419.

Laeven L., Valencia F. (2018) Systemic banking crises revisited, IMF Working Papers 18/206, International Monetary Fund, Washington (DC).

Summers P.M. (2017) `roc`: Computation and analysis of receiver operating characteristic curves (ROC) for binary choice models, gretl *function package version 1.02*.

# On (bootstrapped) cointegration tests in partial systems

Sven Schreiber*

*Abstract:* As applied cointegration analysis faces the challenge that (a) potentially relevant variables are unobservable and (b) it is uncertain which covariates are relevant, partial systems are often used and potential (stationary) covariates are ignored. Recently it has been argued that a nominally significant cointegration outcome using the bootstrapped rank test Cavaliere *et al.* (2012) in a bivariate setting might be due to test size distortions when a larger data-generating process (DGP) with covariates is assumed. This study reviews the issue systematically and generally finds noticeable but only mild size distortions, even when the specified DGP includes a large borderline-stationary root. The previously found drastic test size problems in an application of a long-run Phillips curve (inflation and unemployment in the euro area) appear to hinge on the particular construction of a time series for the output gap as a covariate. We conclude that the problems of the bootstrapped rank test are not severe and that it is still to be recommended for applied research.

*Keywords:* Cointegration rank test, Partial systems, Empirical size

## *1. Introduction*

The cointegration rank test conducted in a multivariate system ("Johansen procedure") is a widespread and popular tool for applied time series analysis. It has long been known that asymptotic inference with that test suffers from substantial size distortions in small samples typical of macroeconomic datasets. Johansen himself developed a finite-sample Bartlett correction for the trace test statistic (Johansen, 2002), and later on bootstrap techniques were proposed (Cavaliere *et al.* 2012, Cavaliere *et al.* 2015). This could be considered as the state of the art.

Recently, however, by conducting an extensive array of simulations Benati (2015) arrived at the interesting result that even the bootstrapped version of

*Macroeconomic Policy Institute Duesseldorf (IMK at Hans Boeckler Foundation), and Free University Berlin, mail@sven.schreiber.name

the rank test could still be subject to considerable size distortions[1]. In one of the many simulations in his paper he essentially analyzed the performance of the bootstrapped rank test in a partial system, i.e. in a situation where the VAR used for the test is lower-dimensional than the DGP, even when only stationary covariates are omitted, not variables in the cointegration relationships themselves. Let inflation be denoted as $\pi_t$ and unemployment as $u_t$, while the short- and long-term interest rates $s_t$ and $l_t$ are transformed a priori to the stationary term spread $(l - s)_t$ together with the differenced short rate $\Delta s_t$ and the output gap $y_t$: Then the analysis concerns $x_{2,t} = (\pi_t, u_t)'$ with $N = 2$ versus $x_{5,t} = (\pi_t, u_t, l_t - s_t, \Delta s_t, y_t)'$ with $N = 5$. For the bivariate system he reports in his Table 2 a p-value of 0.049 for the bootstrapped test of a cointegrating rank $r = 0$ versus $r = 1$. This finding would usually suggest to reject non-cointegration of euro-area inflation and unemployment at the 5% level of significance. By simulation under the null hypothesis he then found a considerable size distortion of the bootstrapped test based on $x_{2,t}$ when the DGP was assumed to contain $x_{5,t}$ and dismissed the nominal findings of cointegration as a "statistical fluke".

Because the reliability of the cointegration test is crucial for many applied research areas, simulations using the actual data are also supplemented here with some simulations of artificial data[2]. Our main finding is that generally the bootstrapped rank test does not over-reject to any alarming extent. This is true for example in simulations of the full 5-dimensional system with $x_{5,t}$ when the output gap $y_t$ is measured as a a standard HP-filter cycle of real

[1] Benati's paper was not meant as an econometrics methods study but investigated the existence of long-run Phillips curve relationships in various economies (synthetical euro area, UK, USA, Canada, and Australia). In this context the term "long-run Phillips curve" refers to a connection between $\pi$, the growth rate of the price level (not wage inflation), and $u$, the level of the unemployment rate; see section 4.2 for plots of the euro area data. We focus here on the results for the euro area and follow the choice of Benati's synthetical sample that actually predates the introduction of the euro (quarterly data 1970-1998).

[2] The original application also considered cointegration ranks $r > 1$ including interest rate levels, and checked CPI inflation as a variant. The datasets are not strictly identical, but we obtain qualitatively the same results, see the appendix (1). For the bootstrap procedures we use the johansensmall.gfn function package (version $\geq$2.6) by Sven Schreiber and Andreas Noack Jensen for the open-source gretl program and freely available online from within gretl. Similar code for Matlab is for example available on De Angelis' homepage https://sites.google.com/view/luca-de-angelis/research.

output. In the literature the bootstrapped rank test was found to have somewhat inflated test sizes when there is a large (stationary) root in the null model (Cavaliere *et al.* 2015), but this effect appears to be limited in the given partial system setting. We can qualitatively replicate the over-rejection of Benati (2015) only with a particular output gap measure that was formerly distributed with the ECB's area-wide model dataset (AWM), some properties of which we will discuss below. Hence overall we conclude that the problems of the bootstrapped rank test are not severe and that it is still to be recommended for applied research.

## *2. Theoretical considerations*

Before turning to the simulations and replications, we briefly revisit the relevant theoretical background for cointegration in potentially partial systems. First of all, note that the meaning of a "partial" system is different from the one used in Harbo *et al.* (1998) and related works. There the considered systems are specified conditional on contemporaneous values of some of the $I(1)$ variables that are part of the cointegrating relations. In contrast, we use the term "partial" to refer to a model that completely disregards some stationary variables of the underlying full system. If the full system vector $x_{N,t}$ is $N$-dimensional and suitably ordered, we define a partial system as modelling the subvector $x_{M,t} = Fx_{N,t}$, where $F = [I_M : 0]$, $M < N$. Sometimes the process representing $x_{M,t}$ is called a subprocess or marginal process; this subprocess is assumed to contain all $I(1)$ components of $x_{N,t}$, such that for the remainder process it holds that $[0 : I_{N-M}]x_{N,t} \sim I(0)$.
The standard starting point that we will adopt is that the data of the full system $x_{N,t}$ are generated by a finite-order VAR. It is well known that in general the subprocess $x_{M,t}$ will then not possess a finite-order VAR representation but instead some VARMA form, which in turn entails an infinite-order VAR model. Before addressing any bootstrap techniques, an important question thus concerns the cointegration analysis of inifite-order VARs.
In this context, one important insight which can be attributed to Saikkonen and Luukkonen (1997) and Lüetkepohl and Saikkonen (1999) is that the application of the standard Johansen rank test in $VAR(\infty)$ systems is asymp-

totically valid. Of course, for the asymptotics to work the chosen lag order must not grow too fast relative to the sample size, but this restriction is either irrelevant for practical applications in given samples or is easy to implement in an automated fashion.

Therefore, given that (1) the partial system $x_{M,t}$ has a $VAR(\infty)$ representation, that (2) the cointegration rank test using a finite lag order is asymptotically still justified, and that (3) the mentioned bootstrap approaches to the rank test are also known to be asymptotically justified, by implication the bootstrapped rank test could in principle be expected to be valid for partial systems, too.

However, approximating a $VAR(\infty)$ with a $VAR(p)$ obviously leaves some autocorrelation in the residuals "by construction". This is not the situation for which the *iid*-residual bootstrap is designed and hence it is not obvious whether it continues to be valid. In such a situation, the residual-based block bootstrap might be promising; see Jentsch *et al.* (2015), who deal with the VECM coefficients for a given cointegration rank, however. Also, as mentioned by Kilian and Lütkepohl (2017), p.348, "no formal results ... about the validity of conducting inference about structural impulse responses in cointegrated VAR models based on the residual-based block bootstrap" exist. While our topic here is not structural impulse responses, a similar gap seems to apply to rank testing, especially in the $VAR(\infty)$ context of a partial system.

Until the statistical theory is completely settled, we must turn primarily to simulation studies. Also, it appears essential to obtain a good approximation to the $VAR(\infty)$ in the first place, such that the difference becomes negligible. Intuitively, if the residuals of a $VAR(p)$ fitted to the partial system are close to being white noise, then there is hope that a standard *iid*-residual bootstrap will work as usual. Building on this insight, we will therefore choose the VAR lag order for the partial systems endogenously based on diagnostic autocorrelation testing as part of the simulation algorithm.

## *3. Bootstrap test specifications*

Throughout this note we focus on the popular case of an unrestricted constant, which was formally justified in Cavaliere *et al.* (2015). For lag length selection in the test VARs we deliberately choose not to use information criteria. The reason is that the non-autocorrelation of residuals is essential for the validity of the standard *iid*-residual based bootstrap, and some of the lag order suggestions by information criteria led to substantial remaining residual autocorrelation. Thus we specify lag orders based on passing a diagnostic autocorrelation test instead.

We focus on the case where the permanent effects on inflation of many shocks are unrestricted (allowed but not forced to be permanent) because it leaves the reduced-form coefficients of the VAR unchanged, allowing the standard application of the Johansen rank test.

The original simulation study used a five-dimensional DGP including inflation and unemployment that imposed absence of cointegration, and then applied the bootstrapped rank test of the null hypothesis $r = 0$ vs. $r \geq 1$ to the bivariate sub-system of simulated inflation and unemployment (in levels) in each simulation draw. Table 3 in Benati (2015) shows that the bootstrap procedure rejected the null hypothesis of no cointegration at a nominal 5% significance in 18.3% of the simulation draws.

Thus he concluded that the bootstrap test grossly exceeded its nominal significance level, and that therefore the original test rejection with a p-value of just under 5% might be "a fluke".

The original study's suggested simulation design is absolutely reasonable. However, this test approach is not the only possible one, at least two different test variants come to mind when further variables are suspected to be relevant for the system dynamics. To systematically address these issues, we enumerate the following three possibilities of cointegration testing with stationary co-variates in small samples:

1. (Bivariate, Benati's method) The null model is given by an unrestricted autoregression for the vector $x'_{0,t} = (\Delta u_t, \Delta \pi_t, y_t, \Delta s_t, l_t - s_t)$, where $y_t$ is the output gap, and $l_t - s_t$ is the term spread between longer-term and short-term interest rates. To ensure a common lag length in

levels, the $K-$th lag coefficients for the differences of unemployment and inflation are set to zero for the simulation DGP:

$$x_{0,t} = c + \sum_{i=1}^{K-1} A_i x_{0,t-i} + (0_{5,2}|\tilde{A}_K) x_{0,t-K} + \epsilon_t,$$

where $\tilde{A}_K$ is an unrestricted $5 \times 3$ matrix for the $K$-th coefficients of the three stationary co-variates. Use this model to generate pseudo data, then run the Cavaliere *et al.* 2015 bootstrapped cointegration test with an unrestricted constant on each simulated draw of the bivariate data $x^{*'}_{2,t} = (u^*_t, \pi^*_t)$ with a lag order $K$[3].

2. (Swensen, unmodelled covariates method) Another bootstrap possibility in the presence of stationary covariates is given by Swensen (2011). The null model is again set up and simulated as in 1, and the bootstrap test is also applied to the bivariate vector $x^{*'}_{2,t} = (u^*_t, \pi^*_t)$. However, the test system is augmented with lags of the co-variates $x^{*'}_{3,t} = (y^*_t, \Delta s^*_t, (l_t - s_t)^*)$, i.e. $x^*_{3,t-1}...x^*_{3,t-K}$ are added as unrestricted regressors[4].

3. (Full system method) If the researcher suspects that there are some important covariates which are known to be $I(0)$, it seems natural to simply include them in the test system. Thus the null model and the bootstrap framework is again given as in method 1, but here the vector to be tested is $x^{*'}_{5,t} = (u^*_t, \pi^*_t, y^*_t, \Delta s^*_t, (l_t - s_t)^*)$, and since the co-variates add three stationary directions to the system already under the null, the relevant hypothesis to test cointegration between unemployment and inflation is $r = 3$ vs. $r = 4$ (again with $K$ lags).

[3] It is not obvious from Benati's description how exactly he handles the lag structure in his simulation, i.e. whether or not he chooses a different lag length for the bivariate subsystem. We determine the lag length in each rank test based on autocorrelation diagnostics.

[4] We do not include contemporaneous values of the covariates as this would obviously violate the necessary assumption of uncorrelatedness. These pseudo covariates are re-generated in each simulation run, but are then held fixed for the inner bootstrap. This corresponds to the test variant described in remark 6 in Swensen (2011). His remark 3 also applies in our implementation, as we use the restricted non-cointegrated model in the bootstrap algorithm.

## 4. Simulation results

In order to have full control and to avoid any unknown properties of actual data we start with the following artificial setup, where the role of unemployment and inflation is taken by $v_t$ and $w_t$.

### 4.1. Size simulations with artificial data

Consider the vector $x_3 = (v, w, z)'$ where the first two components $(v_t,\ w_t)$ are $I(1)$ while the last one $(z_t)$ is a stationary co-variate. Due to the presence of $z_t$ the formal cointegration rank (dimension of the stationary directions) of the full system is one, even though the $I(1)$ variables are not cointegrated. The VECM representation is given by $\Delta x_{3,t} = \alpha\beta' x_{3,t-1} + \Gamma_1 \Delta x_{3,t-1} + c + \epsilon_t$ with a diagonal covariance matrix and the trivial cointegration vector $\beta = (0,\ 0,\ 1)'$. The loading coefficients are $\alpha = (0.1,\ 0.3,\ a_z)'$, the unrestricted constant term is arbitrarily[5] set to $c = (0.9,\ -0.5,\ 0.3)'$ and the short-run dynamics are specified as:

$$\Gamma_1 = \begin{bmatrix} 0.4 & 0.3 & 0.1 \\ 0 & 0.5 & 0.1 \\ 0 & 0 & 0 \end{bmatrix}.$$

The covariate here is specified as an exogenous AR(1) process. Because of the insight from Cavaliere *et al.* (2015) that a large stationary roots in the system can affect the empirical size of the boostrapped rank test, we analyze the cases $a_z = -0.5$ (small root) and $a_z = -0.08$ (large root). As usual, the corresponding levels form VAR with two unit roots is $x_{3,t} = B_1 x_{3,t-1} + B_2 x_{3,t-2} + \epsilon_t$, where $B_1 = \alpha\beta' + I_3 + \Gamma_1$ and $B_2 = -\Gamma_1$. With $a_z = -0.5$ the roots of the system are: $1,\ 1,\ 0.5,\ 0.5,\ 0.4,\ 0$, while with $a_z = -0.08$ they are: $1,\ 1,\ 0.92,\ 0.5,\ 0.4,\ 0$. In the latter case obviously the largest stationary root is quite close to the unit circle and implies considerable persistence.

Running the test size simulations with the bootstrapped test variants described in Section 3, and using these two DGP variants, we obtain the results in ta-

[5] Since the rank test with an unrestricted constant term is not similar and depends on the presence of the drift term, it cannot be omitted.

*Table 1. Bootstrapped test size simulation, artificial DGPs*

| (simulated rejection frequencies under H0, resampling as-if-iid) | small root (0.5) | large root (0.92) |
|---|---|---|
| Bivariate, $r_0 = 0$ | 0.070 | 0.083 |
| Swensen 2 + 1 covar., $r_0 = 0$ | 0.071 | 0.065 |
| Full 3-dim, $r_0 = 1$ | 0.052 | 0.050 |

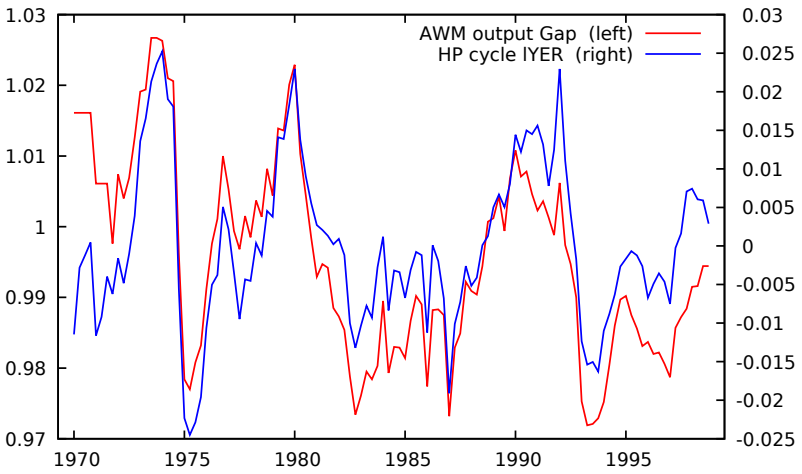
**Notes:** Nominal 0.05 significance level; 5000 replications; sample size $T = 100$.

ble 1. First of all, despite the small sample length of $T = 100$ the test size distortions are relatively mild. In the full-system approach we even do not observe any impact of the larger stationary root on the rejection frequency. In the bivariate partial-system setup (first row in the table) there is an increase from an effective size of 7.1% to a size of 8.3% in the presence of additional high persistence, i.e. by roughly one percentage point.
Nevertheless, while these results are far from the previously reported distortions with apparent test sizes $> 15\%$ (at nominal $5\%$), given a borderline rejecting test result in actual data (for a chosen nominal significance level) it may of course make a difference for the decision whether the effective level of the test is $\alpha$ or $1.5\alpha$.

### *4.2. Simulated empirical size*

We now turn to the actual data analysis. The underlying system in these subsections 4.2 through 4.4 is a 5-dimensional VAR using the cycle component of a standard Hodrick-Prescott (HP) filter applied to real GDP as the relevant measure of the output gap $y_t$ (see Figure 1). The two $I(1)$ series are reported in Figure 2, and the interest rate data as further stationary covariates in Figure 3.

We simulate the effective size (rejection probability under the null) of the bootstrapped cointegration test in the three different test strategies. Following Benati's approach we take the parameters of a non-cointegrated 5-dimensional VAR fitted to the data as the posited DGP; to this end the two $I(1)$ variables are differenced and the stationary variables are left as is. We



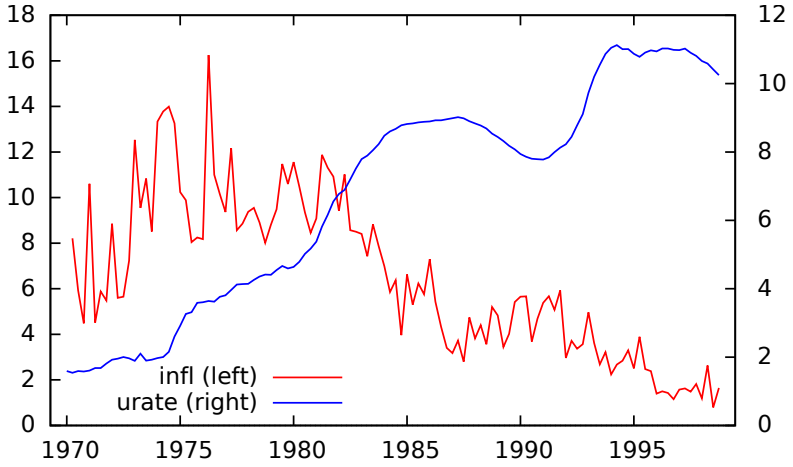

**Notes:** HP cycle is the result of a standard Hodrick-Prescott filter on log real GDP. (AWM is the output gap series from an earlier vintage of the ECB's area-wide model database, displayed for comparison. See also appendix 1.)

*Figure 1. Output gaps*

use 4 lags to obtain the parameters under the null, as this satisfies both the AC and ARCH residual tests[6]. For fitting the model to the simulated data in each draw we do not impose the original lag length but the algorithm chooses the lag order endogenously based on diagnostic residual testing. As explained above, this is important to obtain empirical residuals as close to white noise as possible.

Table 2 reports the size simulation results. For the rightmost column "wild", the rank test is based on a wild bootstrap scheme from the cited literature to account for potential heteroskedasticity. The takeaway from that simulation is that again there are only mild size distortions. The empirical sizes of the bivariate partial-system test and of Swensen's approach are roughly equal, and the full-system approach is mildly conservative which implies that its size is

[6] Having approximately white noise innovations is preferable because we use resampling for the simulation. If we drew the simulation innovations from a parametric model instead the lag length would of course be less important. In any case there are no qualitative differences whether one bootstrap variant or the other is used.




**Notes:** Data from the ECB's AWM, $400 \times \Delta \log(YED)$ and $100 \times URX$.

*Figure 2. Inflation and unemployment rates*

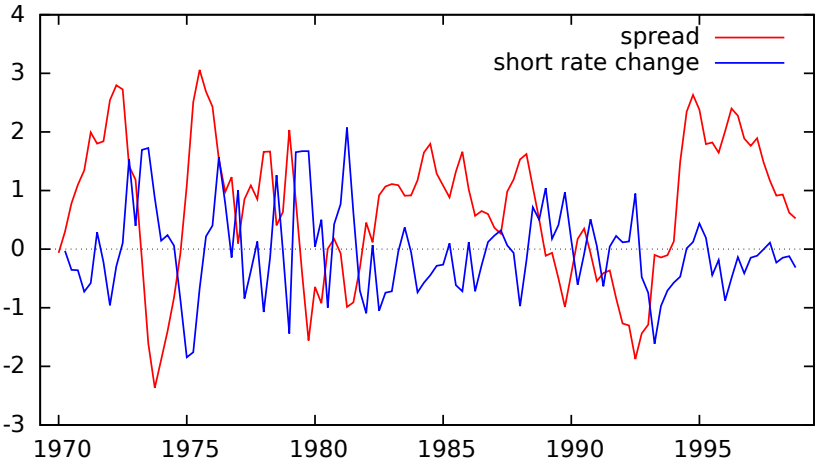
only about half of the sizes of the other aproaches (for a nominal 0.05 level).

### *4.3. Test results*

Although the main motivation for this study is the behavior of the bootstrap rank test in partial systems in general, it is also interesting to replicate the test outcome from the original application. Given that we do not have the strictly identical dataset and vintages we do not expect identical results anyway, but the primary difference concerns the lag length specification: Because of the importance of non-correlated residuals we base our lag choice on diagnostic tests instead of information criteria.

Our test results on the actual data are reported in Table 3; the bivariate results in the first row are qualitatively similar to Benati (2015), namely that the null of no cointegration is nominally rejected at the 5% level but not at the 1% level[7]. The p-values are actually a little lower than in the original study, such

[7] As a memo item, note that the standard bivariate rank test without a bootstrap and using asymptotic critical values here has a p-value ten times lower at 0.001.




**Notes:** Data from the ECB's AWM, $LTN - STN$ and $\Delta STN$.

*Figure 3. Interest rates*

*Table 2. Test size simulations*

| (simulated rejection frequencies under H0) | as-if-iid | wild |
|---|---|---|
| Bivariate, $r_0 = 0$ | 0.069 | 0.083 |
| Swensen 2 + 3 covariates, $r_0 = 0$ | 0.079 | 0.077 |
| Full 5-dim, $r_0 = 3$ | 0.033 | 0.040 |

**Notes:** Simulation of the size of the bootstrapped rank test. Nominal 5%; 2000 simulation replications; the bootstrap test in each simulation draw uses 1000 replications. The time series length is $T = 109$.

*Table 3. Bootstrapped cointegration rank tests (inflation / unemployment)*

| (bootstrapped p-values) | iid | wild |
|---|---|---|
| Bivariate | 0.011 | 0.027 |
| Swensen 2 + 3 covar., $r_0 = 0$ | 0.182 | 0.213 |
| Full 5-dim, $r_0 = 3$ | 0.159 | 0.185 |

**Notes:** 4999 replications; lags are chosen based on diagnostic tests: bivariate – 7 lags, Swensen's approach – 7 lags, full system – 4 lags. The respective sample size $T$ is 113 minus the lag order.

that even after considering the noticeable size distortions from Table 2 the result would seem significant at the 5% level. However, the test conclusion is still not very clearcut.

While the residuals are free from autocorrelation in the bivariate specification with seven lags, there are always remaining ARCH effects, so the wild bootstrap variant (right column) may be preferred for the bivariate case. The other approaches were not considered in the original application.

Swensen's approach, where the bivariate system is augmented with the stationary covariates, is also subject to ARCH-type residuals, again suggesting the use of the wild bootstrap. Here the bootstrapped p-value is far above conventional critical levels (0.213), suggesting non-rejection of no cointegration. Finally, the full-system setup with four lags is well behaved, so the *iid* bootstrap is the method of choice, but it shares with Swensen's setup the non-rejection result (p-value 0.159).

### *4.4. Power assessment*

The test results in Table 3 represent a dilemma. Given that in Table 2 we found that the size distortions of the bootstrapped rank test variants are not dramatic, we do not prefer one approach in Table 3 over any other based on the size assessment – that is, if we share the prior belief about the relevant covariates; otherwise the bivariate test would be preferred. But obviously the test outcomes are very different, so a test decision is difficult.

Therefore we turn to an assessment of the empirical power of the three test

*Table 4. Test power simulations*

| (simulated rejection frequencies) | iid | wild |
|---|---|---|
| Bivariate, $r_0 = 0$ | 0.810 | 0.798 |
| Swensen 2 + 3 covariates, $r_0 = 0$ | 0.139 | 0.128 |
| Full 5-dim, $r_0 = 3$ | 0.224 | 0.235 |

**Notes:** Simulation of the power of the bootstrapped rank test for the fixed alternative given by the cointegrated system (cointegration between unemployment and inflation plus the three stationary covariates) estimated from actual data. Nominal 5%; 2000 simulation replications; the bootstrap test in each simulation draw uses 1000 replications. The time series length is $T = 109$.

approaches. To this end we run a similar simulation as before in Section 4.2, but using as the DGP a system under the alternative hypothesis, with cointegration: the parameters are taken from the estimated error correction system (VECM) of the actual data under an assumed rank of 4, including the cointegration coefficients $\beta$. Three of the four columns of $\beta$ are trivial unit vectors picking the stationary covariates, which technically increases the cointegration rank. The only "actual" cointegration relationship is still the one between unemployment and inflation. Then we simulate artificial data many times with resampled innovation processes, and each time we run the bootstrapped cointegration rank test on the artificial data.

The results of that simulation exercise are reported in Table 4. There is a surprisingly large gap between the power of around 80% in the bivariate case and the power below 25% or even 15% in the full-system and Swensen approaches. This means that the latter two approaches would quite rarely result in rejection of the null hypothesis even if it were false. Against this background it appears that the bivariate setup is the most reliable, combining only mild size distortions with large power advantages. Given the present covariates, the most natural test conclusion would therefore seem to be that euro area unemployment and inflation are cointegrated at a significance level of 5%, but not at the 1% level.

## *5. Conclusions*

The issue of how cointegration rank tests behave when they are applied in partial systems is important, because in practice (a) either potentially relevant variables are unobservable, or (b) it is fundamentally uncertain which covariates might be relevant. This study has partly confirmed the worrying insight that rejection results in partial systems may sometimes be misleading. However, the good news is that the amount of the size distortion appears far smaller than previously suggested in the literature.

The conjecture (inspired from Cavaliere *et al.* 2015) that the size distortion may be due to additional large (stationary) roots in the DGP in the background was only partly reflected in simulations with artificial data, and the effect did not appear large. For the original application of a euro-area long-run Phillips curve we were only able to replicate dramatic size distortions by simulations when the special AWM gap variable from Figure 1 was used as a covariate. (Various vintages of that series were formerly published as part of the area-wide model dataset of the ECB, see also the appendix). That time series possesses a mean in the second subsample which is lower by about 72% of the series' standard deviation; thus it may not really be stationary, which is unusual for such a gap concept. We also suspect that this output gap measure was constructed depending on the in-sample development of inflation, and that this causes the decline in the medium to long run. Hence it induces a larger root in the fitted model that was then used as the DGP in the simulations. Nevertheless, the quantitatively dramatic consequences of basing the simulations on this particular co-variate remains surprising.

In contrast, the test size distortions are very limited with a standard HP filter gap in the background, even though its univariate autoregressive root is also quite large (0.85). Therefore, the econometric evidence for cointegration in this sample between unemployment and inflation remains intact, unless one is completely convinced a priori that the true output gap was given by the AWM measure. We also showed that using full-system methods instead does not pay off, suffering from a severe lack of power.

Finally, it should be acknowledged that this study has addressed a very specific methodological aspect of Benati (2015), which also includes an impres-

sive amount of other empirical and theoretical work. It is not the purpose of this note to question the broad conclusions of his work, summarized as "uncertainty ... is ... substantial" (p. 27). We fully agree. Nevertheless, we regard it as important to clarify for applied economists that conducting cointegration tests in small samples with a bootstrap remains a justified practice and that its results cannot be easily discarded as "statistical flukes".

*1. Supplementary results with the AWM gap*

The euro-area output gap measure in Benati (2015) is not a standard HP-filtered cycle but was based on a certain vintage "from the ECB's database" (quote from the online appendix to Benati, 2015). The precise calculation method of that series is unknown.

As a proxy we used the output gap series that we obtained from an earlier vintage of the ECB's area-wide model (AWM) database. In Figure 1 this proxy and the HP gap were compared. At business-cycle frequencies the two series are highly correlated, as should be expected. However, while the HP cycle measure fluctuates around a constant mean (by construction), the AWM gap is more persistent in the longer run, starting with a sequence of higher-than-average values and finishing the sample with many lower-than-average values. Its AR(1) root is 0.90, opposed to the slightly lower root of the HP cycle of 0.85. Given the limited effects of a large stationary root (see Section 4.1) we do not expect this property alone to have a large impact.

In the test size simulations analogous to Section 4.2, using this described AWM gap instead then requires 7 lags under the null to obtain innovations close to white noise. We observe in Table 5 that again the full-system approach is somewhat conservative, Swensen's approach is mildly oversized, but that now the bivariate partial-system test approach is dramatically oversized with an empirical size over 30% for a nominal 5%. This is even more drastic than Benati's original finding (based on a different lag length and probably slightly different data). Together with the actual test outcomes in Section 4.3 this represents a qualitatively successful replication of the original results.

It could be seen in Figure 1 that the initial values of the earlier AWM output gap are artificially extended and perhaps not very intuitive. As a robustness

*Table 5. Test size simulations under 5-dim DGP with YGA*

| (simulated rejection frequencies under H0) | resampling as-if-iid | wild |
|---|---|---|
| Bivariate, $r_0 = 0$ | 0.349 | 0.327 |
| Swensen 2 + 3 covar., $r_0 = 0$ | 0.067 | 0.086 |
| Full 5-dim, $r_0 = 3$ | 0.023 | 0.023 |

**Notes:** nominal level 0.05; 2000 replications, 7 lags in DGP, sample 1970Q2-1998Q4 (including initial values).

*Table 6. Robustness: Test size simulations under 5-dim DGP with shorter YGA*

| (simulated rejection frequencies under H0) | resampling as-if-iid |
|---|---|
| Bivariate, $r_0 = 0$ | 0.230 |
| Swensen 2 + 3 covar., $r_0 = 0$ | 0.089 |
| Full 5-dim, $r_0 = 3$ | 0.035 |

**Notes:** nominal level 0.05; 2500 replications, 4 lags in DGP, sample 1971Q4-1998Q2 (including initial values).

analysis we therefore took a more recent vintage of the AWM database where the output gap variable (YGA) only ranges from 1971Q4 to 1998Q2, see Figure 4. (Note that in more recent vintages of the AWM dataset the constructed YGA variable does not appear anymore.) In this shorter sample without the starting episode 4 lags are sufficient, and the corresponding simulation results are given in Table 6.
It can be seen that the results correspond qualitatively to the ones in Table 5. The bivariate partial-system test results of the actual data are of course unaffected by any variations of the covariates in the background simulations and are therefore not repeated.

For completeness we also report the bootstrapped actual test results with the older AWM gap in Table 7. (Again, the bivariate test by definition does not depend on the output gap variable and was already shown in Table 3.) For the Swensen approach there are always remaining ARCH effects, thus the

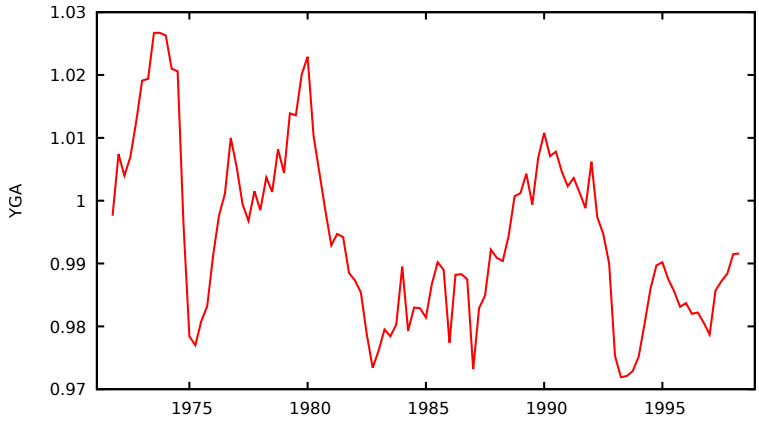



*Figure 4. Shorter AWM output gap (YGA) range*

*Table 7. Test results with actual data (AWM gap)*

| (bootstrapped p-values) | iid | wild |
|---|---|---|
| Swensen 2 + 3 covar., $r_0 = 0$ | 0.007 | 0.011 |
| Full 5-dim, $r_0 = 3$ | 0.366 | 0.335 |

**Notes:** 2000 replications; lag choices: Swensen – 5 lags, Full-system – 7 lags.

wild bootstrap results may be preferred, with a p-value of 0.011 suggesting rejection of no cointegration at the 5% significance level. Given the only mild size distortions of the Swensen approach this appears to be a valid result. The full-system approach here implies well-behaved residuals, so the preferred variant is the iid bootstrap, yielding a p-value of 0.366, not providing evidence in favor of cointegration. This may be accurate or could also be due to the lack of power demonstrated before.

*Acknowledgements:* I am grateful for valuable feedback from Robert Taylor, Anders Rahbek, Giuseppe Cavaliere, and Jack Lucchetti. They are not responsible for any errors that may remain.

## *References*

Benati L. (2015) The long-run Phillips curve: A structural VAR investigation, *Journal of Monetary Economics*, 76, 15-28.

Cavaliere G., Rahbek A., Taylor A.M.R. (2012) Bootstrap Determination of the Co-integration Rank in Vector Autoregressive Models, *Econometrica*, 80(4), 1721-1740.

Cavaliere G., Rahbek A., Taylor A.M.R. (2015) Bootstrap Determination of the Co-integration Rank in VAR Models with Unrestricted Deterministic Components, *Journal of Time Series Analysis*, 36, 272-289.

Harbo I., Johansen S., Nielsen B., Rahbek A. (1998) Asymptotic Inference on Cointegrating Rank in Partial Systems, *Journal of Business & Economic Statistics*, 16(4), 388-399.

Jentsch C., Politis D.N., Paparoditis E. (2015) Block Bootstrap Theory for Multivariate Integrated and Cointegrated Processes, *Journal of Time Series Analysis*, 36, 416-441.

Johansen S. (2002) A Small Sample Correction for the Test of Cointegrating Rank in the Vector Autoregressive Model, *Econometrica*, 70(5), 1929-1961.

Kilian L., Lütkepohl H. (2017) *Structural Vector Autoregressive Analysis*, Themes in Modern Econometrics, Cambridge University Press, Cambridge UK.

Lütkepohl H., Saikkonen P. (1999) Order selection in testing for the cointegrating rank of a VAR process, in Engle R.F., White H. (eds) *Cointegration, Causality, and Forecasting - A Festschrift in Honour of Clive W.J. Granger*, 168-199, Oxford University Press, Oxford UK.

Saikkonen P., Luukkonen R. (1997) Testing cointegration in infinite order vector autoregressive processes, *Journal of Econometrics*, 81, 93-129.

Swensen A.R. (2011) A bootstrap algorithm for testing cointegration rank in VAR models in the presence of stationary variables, *Journal of Econometrics*, 165, 152-162.

# CUB for gretl

Rosaria Simone*, Francesca Di Iorio**, Riccardo Lucchetti***

*Abstract:*

CUB models are a class of mixture distributions for analyzing ordinal responses in the form of ratings or marginal ranking that designs the decision process as the combination of two main components: a personal *feeling* and an inherent *uncertainty*. This methodology is receiving an increasing interest due to successful applications to the analysis of evaluations, opinions and perceptions, in various fields of research, and thanks to advantageous graphical interpretation of results. Currently the estimation of CUB models can be run by the R library 'CUB' ver. 1.1.3. In this paper a gretl package for CUB models is presented, all the estimation examples and graphs proposed in the R vignette are replicated.

*Keywords:* CUB models, ML estimation, Graph tools

## *1. Introduction*

Ordinal responses in the form of ratings arise frequently in applications where human preferences, judgments and perceptions play a key role. A well-known example are the studies on customers/users' satisfaction where it is common to collect raters' evaluation on a hedonic scale, along with a set of categorical and/or quantitative covariates that characterize the respondent. In this framework the main goal is modeling the relationships between the ordinal variable and the covariates to identify different response patterns. The analysis of ordinal data is the subject of a wide literature, see among others Agresti (2010) and Tutz (2012). An alternative approach is based on CUB models (D' Elia and Piccolo 2005, Piccolo and D'Elia 2008, Piccolo and Simone 2019), a family of discrete mixtures that, in the last years, has received an increasing attention due to successful applications to various research fields. The acronym CUB stands for *C*ombination of *U*niform and

*Università di Napoli Federico II, fdiiorio@unina.it

** Università di Napoli Federico II, rosaria.simone@unina.it

*** Università Politecnica delle Marche, r.lucchetti@univpm.it

(shifted) *B*inomial random variables in the mixture that defines the model. The idea behind these models is that discrete choices arise from a psychological mechanism that involves two components: a personal *feeling* and an inherent *uncertainty* surrounding the decision process. Feeling is mainly related to the subjects' motivations and it can be adequately represented by the shifted Binomial random variable; the discrete Uniform random variable, instead, is the most appropriate distribution to deal with the intrinsic indecision of the discrete choice process. In particular, the shifted Binomial random variable represents a discrete version of the continuous latent trait allowing the unobserved perception be mapped to an ordinal evaluation; the Uniform distribution corresponds to the maximum entropy on a finite support. One of the strengths of a CUB model is that it is identified by two parameters only, then an effective graphical representation of the model can be obtained as a point in the parameter space, allowing comparisons among items or repeated measurements over time.

A simple CUB model, without covariates, can be employed *per se* to estimate the expected distribution given a sample of observed ordinal values, but the usefulness and relevance are strengthened if subjects' or objects covariates are associated to *feeling* and/or *uncertainty* parameters, in the same spirit of Generalized Linear Models (McCullagh, 1980). This technique leads to CUB regression models, in which a relevant role is played by the selection of the covariates for *uncertainty* and/or *feeling* that mostly explain the response and improve the fitting.

The CUB model can be formally described as follows.

Let $R_i$ the response of the $i$-th subject to a given item of a questionnaire collected on a $m$-point scale, the probability of $R_i = r$ is assumed as:

$$Pr(R_i = r|\pi_i, \xi_i) = \pi_i \binom{m-1}{r-1} \xi_i^{m-r}(1-\xi_i)^{r-1} + (1-\pi_i)\frac{1}{m}, \quad (1)$$

where $r = 1, \ldots, m$, with $m > 3$ for identifiability purposes, and the parameters $\pi_i$ and $\xi_i$ are called uncertainty and feeling parameters, respectively.

As to the preference part of the model, the larger $1 - \xi_i$, the more the distribution is concentrated along the highest scores. Thus, if the scale is oriented

as the greater the score the more positive the evaluation, a high level of $1-\xi_i$ indicates a positive tendency in the data w.r.t. the topic under investigation. For instance, $1-\xi_i$ is a measure of agreement if the respondent is asked to express his/her accordance to a given statement, or an indicator of satisfaction if he/she is asked to assess the quality of a service/product. Owing this versatile interpretation, $\xi_i$ is referred to as *feeling* parameter.
Left panel of Figure 2.4 shows some CUB probability distributions for several values of $\pi$ and $\xi$ in the parameter space $\Omega(\boldsymbol{\theta}) = (0,1] \times [0,1]$. The peculiar parametrization of CUB models allows a deeper descriptions than those obtained by standard indicators: for instance, Models F and I share the same mean value, although they have quite different distributions. In addition, as mentioned above, since each CUB distribution is identified by two parameters, it can be represented as a point in the parameter space as in right panel of Figure 2.4, offering an intuitive tool to compare several models (for instance, each model in the figure could correspond to an item of a questionnaire). A richer CUB model can be obtained including explanatory variables so that feeling and/or uncertainty directly depend on respondents' profiles assuming a *logit* link between parameters and covariates:

$$logit(\pi_i) = \beta_0 + \beta_1\, y_{1i} + \beta_2 y_{2i} + \cdots + \beta_p y_{pi}, \tag{2}$$

$$logit(\xi_i) = \gamma_0 + \gamma_1\, w_{1i} + \gamma_2 w_{2i} + \cdots + \gamma_q w_{qi}, \qquad i = 1, \ldots, n. \tag{3}$$

The resulting model with $p$ covariates $\boldsymbol{y}_1 = (y_{1i}, \ldots, y_{1n})', \ldots, \boldsymbol{y}_p = (y_{pi}, \ldots, y_{pn})'$ considered to explain uncertainty and $q$ covariates $\boldsymbol{w}_1 = (w_{1i}, \ldots, w_{1n})', \ldots, \boldsymbol{w}_q = (w_{qi}, \ldots, w_{qn})'$, (with $\boldsymbol{y}_i \in Y$, $\boldsymbol{w}_i \in W$ include the selected covariates for the $i-th$ subject), is referred to as a CUB $(p,q)$ model, with parameters $(\boldsymbol{\beta}', \boldsymbol{\gamma}')'$, $\boldsymbol{\beta}' = (\beta_0, \ldots, \beta_p)'$, $\boldsymbol{\gamma}' = (\gamma_0, \ldots, \gamma_q)$. With this notation, the baseline definition of CUB models with no covariate neither for feeling nor for uncertainty is indicated with CUB $(0,0)$ model (in this case, $\pi_i = \pi$ and $\xi_i = \xi$ are constant among subjects). Note that the choice of the logit link is rather arbitrary, and it is mainly preferred since it grants easiness of interpretation and implementation and no restraint on the nature of explanatory variables is required. If $D$ is dichotomous variable considered as a covariate for $\pi_i$ and $\xi_i$, then a CUB $(1,1)$ model

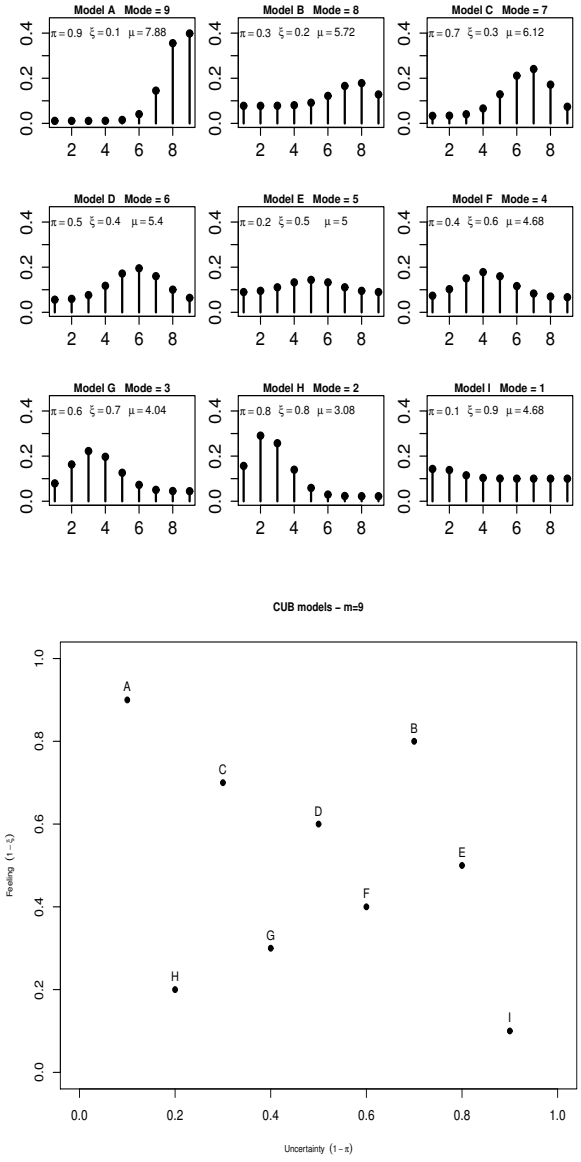



*Figure 1. Selected* CUB *probability distributions and scatterplot in the parameter space. Source: Fig. 4.1 and 4.2 in* CUB *R package vignette*

fitted to the data will identify two sub-samples, each of which characterized by feeling and uncertainty obtained via (2)-(3). Similarly, a CUB $(1, 0)$ model sets only a dichotomous variable to explain uncertainty: if significant, two different uncertainty values and a common estimate for the feeling parameter will describe the data. Similar arguments apply for CUB $(0, 1)$ models. In each of these cases, in the end the sample will be split in two separate subsets of observations.

The CUB model can be inflated to take into account the presence of a shelter category (*shelter effect*, see Iannario 2012). A shelter category $s \in 1, ..., m$ is the modality that receives an upward bias of preference with respect to the expected response. Testing the presence and measuring the shelter effect can be done extended the CUB model with the introduction of a degenerate distribution $D_r^{(s)} = I(R = s)$, whose probability mass is concentrated at $r = s$. Thus, the model assumes the expression:

$$Pr(R_i = r|\pi_{1i}, \pi_{2i}, \xi_i) = \pi_{1i}\binom{m-1}{r-1}\xi_i^{m-r}(1-\xi_i)^{r-1}+ \\ (1-\pi_{2i})\frac{1}{m} + (1-\pi_{1i}-\pi_{2i})D_r^{(s)}, \quad r = 1, \ldots, m, \tag{4}$$

or the equivalent specification:

$$Pr(R_i = r|\pi_i^*, \delta_i, \xi_i) = \delta_i D_r^{(s)} + (1-\delta_i)\left[\pi_i^*\binom{m-1}{r-1}\xi_i^{m-r}(1-\xi_i)^{r-1}\right. \\ \left. +(1-\pi_i^*)\frac{1}{m}\right] +, \quad r = 1, \ldots, m, \tag{5}$$

where $\delta_i = (1 - \pi_{1i} - \pi_{2i})$ and $\pi_i^* = \frac{\pi_{1i}}{\pi_{1i}+\pi_{2i}}$. With this re-parametrization the shelter effect can be quantified by the parameter $\delta_i$ and the associated modification of the uncertainty component is evaluated by comparing the $\pi_i$ parameter in model (1) with $\pi_{1i}$ in model (5). Estimation of CUB models relies

on Likelihood methods, where the log-likelihood is:

$$\mathcal{L}_n(\boldsymbol{\theta}) = \sum_{i=1}^{n} log\left[Pr(R_i = r|\boldsymbol{y}_i, \boldsymbol{w}_i, \boldsymbol{\theta})\right] \tag{6}$$

with $\boldsymbol{\theta}$ is the vector of estimable parameters. Fit improvements yielded by CUB regression models can be tested via the standard Likelihood Ratio Test (LRT): indeed, a CUB $(0, 0)$ is nested into any of the extensions CUB $(p, q)$.
Likelihood estimation methods and the implementation of the Expectation-Maximization (EM) algorithm for mixtures by McLachlan and Krishnan (1997) for CUB models are currently implemented within the CUB model R library ver. 1.1.3 (Iannario et.al, 2018).
In this paper, the gretl translation of the R package is presented. The current version of the gretl package is based on ML estimation via **mle** gretl command.
The implementation of the EM estimation algorithm and two generalization of the CUB models, the case of the Beta-binomial and the degenerate Binomial distributions implemented in the R package, are subject of ongoing work.
All the estimation examples and graphs proposed in the R vignette are replicated and some exercises based on the satisfaction survey from the NTTS 2019 conference are presented.

## *2. The gretl package*

To describe the gretl package we follows the examples reported in the vignette to the R package 'CUB'. These examples are base on the dataset `univer`, for which detailed descriptions and further references can be found at `http://www.labstat.it/home/research/resources/cub-data-sets-2`.
The dataset `univer` collects data from a sample survey on students' evaluation of the Orientation services that has been administered across all the Faculties of University of Naples Federico II in 2002 and consists of 2179 observations. Participants were asked to express their ratings on a 7 point Likert-type scale (1 ="very unsatisfied", 7 = "extremely satisfied") on the following items:

- informat: Level of satisfaction about the acquired information

- willingn: Level of satisfaction about the willingness of the staff
- officeho: Level of satisfaction about the opening hours
- compete: Level of satisfaction about the competence of the staff
- global: Level of global satisfaction

There are also subjects covariates: `gender`, equal to $0$ for men and to $1$ for women, **age** in years, and `freqserv`, indicating the usage frequency of the service with levels $0$ and $1$ for non-regular and regular users.

The ML estimation in the CUB gretl package is based on the **mle** command with default options: `bfgs` algorithm and **OP** covariance matrix. To fit CUB models in gretl, the ordinal data and the covariates must be arranged as *series*.

### *2.1. Estimation of a* CUB $(0, 0)$ *model*

As above described, the baseline CUB model is the CUB $(0, 0)$ distribution. The simplest call to run a CUB $(0, 0)$ on the ordinal variable `officeho` of the dataset `univer` is:

```
# include functions
include cub_functions.inp
include cub_graph.inp

# initial settings
scalar m = 7  # number of possible ranking or rating
scalar shelter = 0   # shelter if >0 with shelter integer in [1,m]
scalar verb=0  # if 1 verbose

# ------  CUB(p,q)

# declare covariates , constant is compulsory
list Y = const
list W = const

# call estimation procedure
```

```
mod00 = CUB(officeho, Y, W, m, shelter,verb)

# print results
printout(mod00)
```

As for the CUB $(0,0)$ $\pi$ and $\xi$ lie in the parameter space $\Omega(\boldsymbol{\theta}) = (0,1] \times [0,1]$, they are logit-transformed during the estimation procedure. Then the displayed output, obtained using the function **printout**, shows the MLE estimation results and the reverse-transformation in terms of the original parameters $\pi$ and $\xi$. Usual statistical indicator as Log-likelihood, AIC and BIC are reported.

```
Function evaluations: 17
Evaluations of gradient: 17

ML, using observations 1-2179
loglik = log(prob_cub(y, m, theta, Y, W, infl))
Standard errors based on Outer Products matrix

            estimate    std. error      z       p-value
  ---------------------------------------------------------
  theta[1]    0.755766   0.0877592      8.612   7.19e-018 ***
  theta[2]   -1.40396    0.0328171    ?42.78    0.0000    ***

Log-likelihood      ?3759.917   Akaike criterion     7523.834
Schwarz criterion    7535.208   Hannan-Quinn         7527.992

Dep.var = officeho

            coefficient   std. error     z      p-value
  ---------------------------------------------------------
  pai          0.680434     0.0190827    35.66   1.83e-278 ***
  csi          0.197189     0.00519512   37.96   0.0000    ***

max LLn = -3759.92
AIC = 7523.83
BIC = 7535.21
```

The estimated coefficients are the same reported in pag.14-15 in the R vignette; all the procedure output are stored in a *bundle*, named **mod** in this example, that includes, among others, the estimated coefficients and the estimated individual probabilities. The available elements in the output bundle are listed in the appendix.

### *2.2. Estimation of a* CUB $(p, q)$ *model*

Even for the CUB $(p, q)$ example we follows R vignette. To explain the feeling component in a model for the variable `officeho`, we include the dichotomous covariate `freqserv`. The call becomes:

```
list Y = const
list W = const  freqserv
mod01 = CUB(officeho, Y, W, m, shelter,verb)
# print results
printout(mod01)
```

The results, again very close to that reported at pag. 16 in the R vignette, are:

```
Function evaluations: 17
Evaluations of gradient: 17

ML, using observations 1-2179
loglik = log(prob_cub(y, m, theta, Y, W, infl))
Standard errors based on Outer Products matrix

Dep.var = officeho

             coefficient   std. error      z       p-value
  ---------------------------------------------------------
  beta         0.787360    0.0873000      9.019   1.90e-019 ***
  const       -1.15167     0.0384859    -29.92    9.48e-197 ***
  freqserv    -0.810594    0.0723345    -11.21    3.80e-029 ***

max LLn = -3704.36
AIC = 7414.71
BIC = 7431.77
```

The likelihood ratio test can be used to compare the two models, as well as the comparison of the BIC index: in that case, BIC(mod00)= 7535.21 is reduced to to BIC(mod01)= 7431.77 giving a clear support to the inclusion of the covariate `freqserv` in the model feeling.
A continuous covariate and a dichotomous one can be jointly considered. Let `Lage` be the deviation from the mean of the logarithmic transform of covariate `age`, and the dummies `gender` and `freqserv` can be considered to explain both feeling and uncertainty for a CUB $(2, 1)$ model. The model can be run with the following call:

```
# center countinous variables
lage = log(age)
lage -= mean(lage)

list Y = const  lage gender
list W = const  lage freqserv

modpq = CUB(officeho, Y, W, m, shelter,verb)
printout(modpq)
```

and the results (see R vignette pag. 19) are

```
Function evaluations: 48
Evaluations of gradient: 48

ML, using observations 1-2179
loglik = log(prob_cub(y, m, theta, Y, W, infl))
Standard errors based on Outer Products matrix

 Dep.var = officeho

             coefficient   std. error      z       p-value
  ---------------------------------------------------------
  const        0.564891    0.115899       4.874   1.09e-06  ***
  lage         1.23720     0.635596       1.947   0.0516    *
  gender       0.495619    0.169527       2.924   0.0035    ***
  const       -1.14668     0.0385482    -29.75    1.92e-194 ***
```

```
  lage         -0.590458    0.229282      -2.575   0.0100    **
  freqserv     -0.822841    0.0725023    -11.35    7.49e-030 ***

max LLn = -3693.89
AIC = 7399.78
BIC = 7433.89
```

### *2.3. Estimation of a* CUB $(p, q)$ *model with shelter*

The use of shelter can be described by the adding a shelter in category 5 to the previous call for a CUB $(0, 0)$ model:

```
scalar shelter = 5
list Y = const
list W = const
modshe = CUB(officeho, Y, W, m, shelter,verb)
printout(modshe)

               stima     errore std.      z       p-value
  ---------------------------------------------------------
  theta[1]    0.380070    0.0931511      4.080   4.50e-05  ***
  theta[2]   -1.72252     0.0614531    -28.03    7.03e-173 ***
  theta[3]   -2.21316     0.154588     -14.32    1.73e-046 ***


Dep.var = officeho

             coefficiente   errore std.     z       p-value
  ----------------------------------------------------------
  pai          0.593890      0.0224666     26.43    5.52e-154 ***
  csi          0.151547      0.00790165    19.18    5.53e-082 ***
  delta        0.0985751     0.0137364      7.176   7.17e-013 ***

max LLn = -3741.66 AIC = 7489.33 BIC = 7506.39
```

The estimated shelter coefficient $\hat{\delta} = 0.098$ is statistically different from zero,

showing a better fit with respect to the CUB $(0, 0)$ model without shelter (the BIC decreases to 7506.39 while without shelter it was 7535.21).

Shelter effect can be tested also in presence of covariates, as the following example shows:

```
list Y = const
list W = const   lage
scalar shelter = 5
mod = CUB(officeho, Y, W, m, shelter,verb)
printout(mod)
```

```
Dep.var = officeho
             coefficient   std. error      z       p-value
  ---------------------------------------------------------
  beta         0.388420     0.0930838      4.173   3.01e-05  ***
  const       -1.71704      0.0609261    -28.18    9.64e-175 ***
  lage        -0.658582     0.318984      -2.065   0.0390    **
  delta        0.0982375    0.0136916      7.175   7.23e-013 ***

max LLn = -3739.86 AIC = 7487.72  BIC = 7510.47
```

*2.4. Multicub function*

As in the R package, we can visualize simultaneously the ordinal variables included in the dataset `univer` by means of the function `multicub` that fits a CUB $(0, 0)$ model to every ordinal variable in a given list. With this function, the estimated parameters $(\hat{\pi}, \hat{\xi})$ for each model are considered as the coordinates in the parameter space corresponding to the obtained ML uncertainty $(1 - \hat{\pi})$ and feeling $(1 - \hat{\xi})$ estimates. The CUB parameters are $\pi$ and $\xi$ but their interpretation depends on the orientation of the measurement scale. If the evaluation is positive in the direction of the scale, then the actual measure of feeling is $(1 - \xi)$. The call to run the `multicub` function on the ordinal variables included within the dataset `univer` is:

```
# ------  multiCUB

scalar m = 7  # number of possible ranking or rating
```

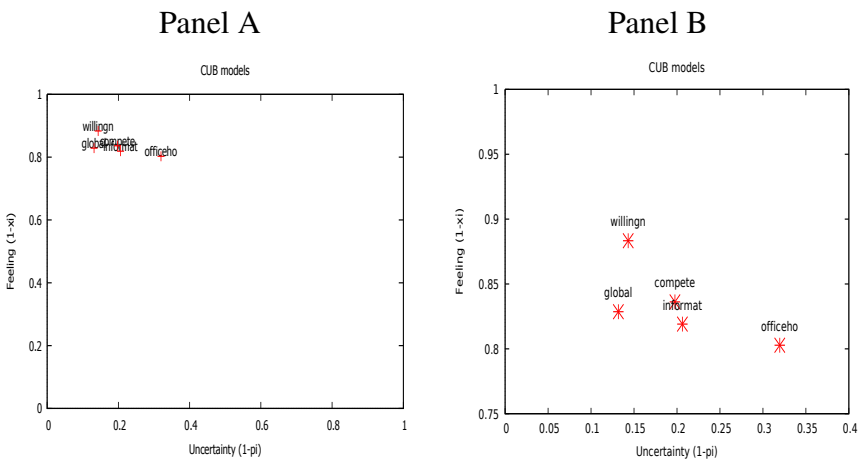



*Figure 2.* `plot_multicub` *function output: scatterplot in the parameter space, default (Panel A) and customized (Panel B) settings*

```
# list of the ordinal variables to be compared by CUB(0,0)

list My= officeho global informat willingn compete

mshelter = {0, 0, 0, 0,0}   # set shelter  foreach i in My

coord=multicub( My,  m, mshelter,verb)

plot_multicub( My, coord,)
```

The results are the estimated parameters $(\hat{\pi}, \hat{\xi})$ for each model, collected in the matrix `coord`, and the associated scatterplot that can be invoked using the function `plot_multicub`. By default the plotting area is the whole parameter space $\Omega(\boldsymbol{\theta}) = (0,1] \times [0,1]$, but the graph can be modified according the usual gretl procedure. See for example the left panel (default graph settings) and right panel (customized settings) of Figure 2.4. The Figure reproduce Figure 4.2 in the CUB R vignette reported in Figure above.

*3. Graphs*

As mentioned in the previous sections, graphical tools are one of the strengths of CUB models.

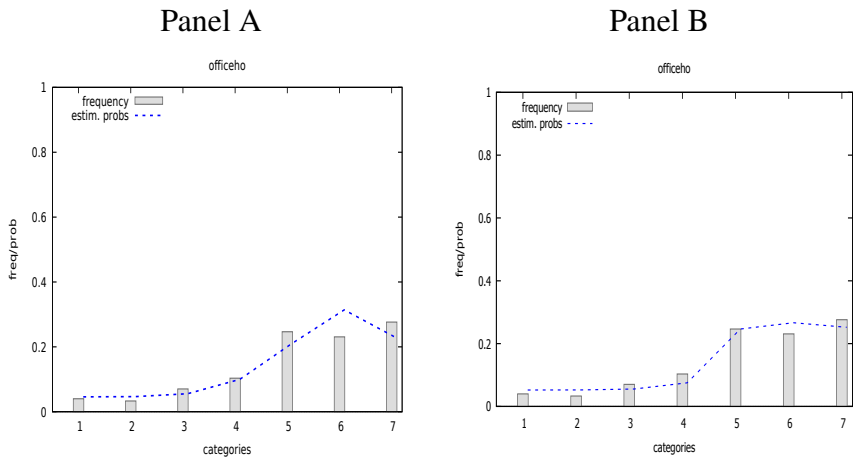



*Figure 3. Estimated probability distributions from* `plot_distr` *for* CUB $(0,0)$ *models without (Panel A) and with shelter (Panel B)*

The gretl package, as the original R library, produces as default some useful graphs. One of that is the above mentioned scatterplot `plot_multicub` that shows the estimated CUB $(0,0)$ coefficients for a list of ordinal variables in the parameter space. The graph can be customized as usual using the gretl or Gnuplot facilities.
After a single CUB $(0,0)$ estimation, the call to `plot_distr` produces a plot comparing the estimated distribution and the observed frequencies. For the case discussed in section 2.1, the estimation procedure returns the bundle **mod00**, then the call

```
plot_distr(mod00)
```

produces such graph for CUB $(0,0)$ without shelter specification in Panel A and the call `plot_distr(modshe)` gives the Panel B of Figure 3, respectively.

The call `plot_distr_exod` produces a plot comparing the two fitted probability distributions conditional on the values a the dichotomous covariate. Still considering the example in section 2.1, we can obtain the fitted probability distributions for `officeho` conditional to the variable `freqserv` using the call:

```
list Y = freqserv
list W = freqserv
scalar shelter = 0
mod11 = CUB(officeho, Y, W, m, shelter,verb)
plot_distr_exod(mod11, freqserv)
```

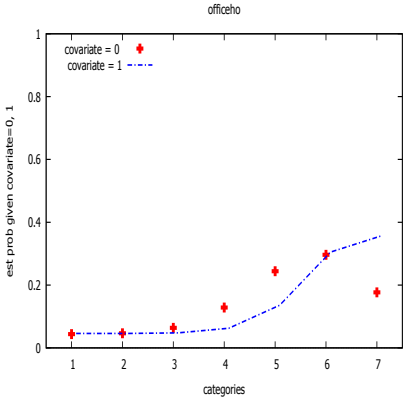



*Figure 4. Estimated probability distributions for* `officeho` *conditional to* `freqserv`*, obtained via* `plot_distr_exod` *function output*

The resulting plot is shown in Figure 3.

When a CUB $(p, q)$ is estimated, the procedure return the estimated probabilities for each individual, then the function `plot_distr` returns for each category the mean of the estimated individual probabilities given the individual observed covariates. Please note that in that case the sum of the mean of the estimated probabilities is not necessarily equal to 1. For this reason the observed frequency is not over-imposed in the graph. In the example illustrated in the section 2.2 the results of the function

```
plot_distr(modpq)
```

are in Figure 3.

*4. A simple speed comparison*

A simple speed comparison is performed in the following way: a set of 5000 samples with n=500 observations drawn from a CUB $(0, 0)$ with parameters $\pi = 0.7$ and $\xi = 0.8$ is generated using the R procedure `simcub`. On that dataset the speed performance of the CUB $(0, 0)$ ML estimation based on bfgs algorithm in the gretl package and the ML estimation based EM algorithm in the original R package are compared. The current version of the gretl package results 10 times slower then the R pakage: 357 seconds using commandline program `gretlcli` vs 39 seconds using

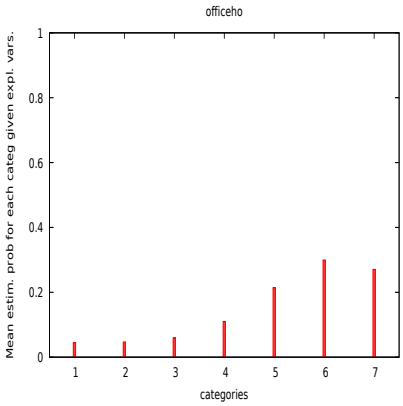



*Figure 5. Distribution of the means of the estimated probabilities given covariates, for each category, obtained from* `plot_distr` *for* CUB $(p, q)$

the `R studio` interface on a Windows 10 pc x64 based processor, Intel i5, 2.60Ghz, 8 Gb RAM. Then an code optimization are in order.

*5. Appendix:* CUB *functions input output description*

```
---
CUB
---
usage: mod = CUB(ordinal, Y, W, m, shelter,verb)

Main function, performs the main computations and stores results
into the returned bundle

inputs:
ordinal:   series of rating or ranking on m categories
Y          list of series: covariate for Uncertainty parameter pai
W          list of series: covariate for Feeling parameter csi
m          number of categories for series ordinal
shelter    scalar in [1,m] default 0, no shelter
verb       scalar, default 0: verbose off

output: Bundle containing:
```

```
bundle mod:
  y          series, depended variable
  phat       series individual estimated probabilities
  lnl        scalar max value of loglik
  shelter    scalar
  simple     scalar if 1 CUB(0,0)
  freq       matrix: m x 1  observed frequency for ordinal
  raw_coeff  matrix  p+q+2 or p+q+3 (if shelter) x 1, estimated
                   parameter before transformation
  T          scalar, total number of observation
  bic        scalar, information criteria
  depvarname  string name of the depended variable y
  fhat       matrix: m x 1  estimated frequencies
  vcv        matrix: coefficient estimated covariance matrix
  aic        scalar, information criteria
  stderr     matrix: coefficient standard error
  m          scalar, number of categories
  ncoeff     scalar, number of coefficients
  coeff      matrix: ncoeff x 1 estimated parameter
                   after transformation

----------------
printout
--------------------
usage:  printout(mod)

print the estimated results after transforming
estimated row coefficients

inputs:
mod    bundle, is the return of the CUB function


--------------------
multicub
```

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

usage: multicub( my,  m, mshelter,verb)

estimate CUB(0,0) models, also with shelter, on a list
of dependent variables

inputs:

my        list, list of series rating or ranking on m categories
m         scalar, number of categories
mshelter  matrix 1 x nelem(my) shelter for each element of my
verb      scalar, verbose control, default 0

output

matrix   nelem(my) x 2, estimated pai and csi for each
         element in my

--------
plot_distr
--------

usage: plot_distr(mod)

Plot the observed and estimated frequencies

inputs:
mod    bundle, is the return of the CUB function


--------
plot_distr_exod
--------

usage: plot_distr_exod(mod, dummyexo)
```

```
Plot the estimated probabilities for dummy covariates

inputs:
mod       bundle, is the return of the CUB function
dummyexo  series, dummy covariates


-------------------------------
plot_multicub
-------------------------------
usage : plot_multicub( my, coord,dest)

plot the results of multicub function. pai and csi
for each model as coordinates

inputs:
 my        list, list of series rating or ranking
           on m categories
 coord     matrix, output of multicub function
 dest      string, filename for storing the graph,
           default null=display
```

*References*


Agresti A. (2010) *Analysis of Ordinal Categorical Data*, $2^{nd}$ edition, J.Wiley & Sons, Hoboken

D'Elia A., Piccolo D. (2005) A mixture model for preference data analysis, *Computational Statistics & Data Analysis*, 49, 917-934.

Iannario M. (2012) Modelling shelter choices in a class of mixture models for ordinal responses, *Statistical Methods and Applications*, 21, 1-22.

Iannario M., Piccolo D., Simone R. (2018) The R package CUB: a class of mixture models for ordinal rating data *R package version 1.1.3*, `http://CRAN.R-project.org/package=CUB`

Piccolo D., D'Elia A. (2008) A new approach for modelling consumers' preferences, *Food Quality & Preference*, 19, 47-259.

Piccolo D., Simone R. (2019) The class of CUB models: statistical foundations, inferential issues and empirical evidence. *Statistical Method and Applications*, https://doi.org/10.1007/s10260-019-00461-1 (with discussions)

McCullagh, P. (1980) Regression models for ordinal data, *Journal of the Royal Statistical Society, Series B*, 42, 109-142.

McLachlan G.J., Krishnan T. (1997) *The EM Algorithm and Extensions*, John Wiley & Sons, New York.

Tutz G. (2012) *Regression for Categorical Data*, Cambridge Series in Statistical and Probabilistic Mathematics.

Scientific Sponsorship


Financial Support


Università degli Studi di Napoli Federico II
Scuola delle Scienze Umane e Sociali

This book collects the papers presented at the 6th Gretl Conference, which took place at the Dipartimento di Scienze Politiche, University of Naples, on 13th-14th of June, 2019.
The papers (which had been selected through a refereeing process) contain theoretical topics on computational problems as well as empirical applications using the program, and cover a wide range of topics in statistics and applied econometrics, among which Generalized Dynamic Factor Models, Propensity Score Matching, Bayesian Model Averaging, Spatial Models, Cointegration and Boostrap Techniques.

Francesca Di Iorio is associate professor in Economic Statistics at the Department of Political Sciences, University of Naples Federico II. She has published several articles in these fields including non-stationary panels, bootstrap inference, and computational solution for indirect inference, time series analisys and regressions trees.

Riccardo Lucchetti is full professor of Econometrics at the Università Politecnica delle Marche (formerly known as Università di Ancona) and has partecipated in the development team of the open source package Gretl for over ten years. He has published several articles on computational econometrics, time series analysis and migration economics.

ISBN: 978-88-6887-057-7
DOI: 10.6093/978-88-6887-057-7
Online ISSN: 2532-4608