



Munich Personal RePEc Archive

Bayesian Model Averaging and Jointness Measures for gretl

Blazejowski, Marcin and Kwiatkowski, Jacek

Wyższa Szkoła Bankowa w Toruniu, Uniwersytet Mikołaja
Kopernika w Toruniu

10 February 2013

Online at <https://mpra.ub.uni-muenchen.de/61431/>
MPRA Paper No. 61431, posted 05 Dec 2015 14:17 UTC

Bayesian Model Averaging and Jointness Measures for gretl

Marcin Błażejowski

Toruń School of Banking
Department of Quantitative Methods
87-100 Toruń, Poland
marcin.blazejowski@wsb.torun.pl

Jacek Kwiatkowski

Nicolaus Copernicus University in Toruń
Faculty of Economics and Management
87-100 Toruń, Poland
jkwiat@umk.pl

January 18, 2015

Abstract

This paper presents a software package that implements Bayesian model averaging for Gnu Regression, Econometrics and Time-series Library - `gretl`. The Bayesian model averaging is a model-building strategy that takes account of model uncertainty in conclusions about estimated parameters. It is an efficient tool for discovering the most probable models and obtaining estimates of their posterior characteristics. In recent years we have observed an increasing number of software packages devoted to Bayesian model averaging for different statistical and econometric software. In this paper, we propose the **BMA** package for `gretl`, which is an increasingly popular free, open-source software for econometric analysis with an easy-to-use graphical user interface. We introduce the **BMA** package for the linear regression models with jointness measures proposed by Ley and Steel (2007) and Doppelhofer and Weeks (2009).

Keywords: Bayesian model averaging, jointness measures, `gretl`, Hansl.

JEL: C11, C51, C87, O47.

1 Introduction

We know, from the elementary statistical theory that linear regression attempts to model the relationship between two or more variables by fitting a simple linear equation to observed data. In a classical approach, we usually rely on the Ordinary Least Squares (OLS) or the Maximum Likelihood (ML) estimates and the popular model selection criteria, i.e.: Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), to find the “best” model. The main problem arises when we have to select a “good” subset of variables from a large set of regressors. When the number of possible exogenous variables is K , the number of possible linear models is 2^K . If we have, for example, $K = 30$ possible regressors, the number of possible linear combination equals 1073741824. This means that it is very difficult, if not impossible, to find the estimates for all combinations. Moreover, Raftery, Madigan, and Hoeting (1997) show that standard variable selection procedures lead to different estimates and conflicting conclusions about the main questions of interest.

Bayesian model averaging is a useful alternative to other variable selection procedures, because it incorporates model uncertainty into conclusions about the estimated parameters.

The Bayesian model averaging is a standard Bayesian solution to model uncertainty, where the inference on parameters is based on a weighted average over all possible models under consideration, rather than on one single regression model. These weights are Bayesian posterior probabilities of the individual models.

There is a recent and growing literature on Bayesian model averaging. Examples of applications of Bayesian model averaging can be found in a number of works (see, for example, Hoeting, Madigan, Raftery, and Volinsky (1999), Steel (2011) for a recent overview). Our software package for parameter estimation and model comparison of linear regression models is based on Fernández, Ley, and Steel (2001a,b) and Koop (2003). We use the Markov chain Monte Carlo Model Composition (MC³) sampling algorithm developed by Madigan, York, and Allard (1995) to select a representative subset of models.

Doppelhofer and Weeks (2005, 2009) define jointness measure of dependence among explanatory variables that appear in the linear regression models. We use this measure to identify whether different sets of two variables are substitutes, complements or neither. A similar jointness measure was also proposed by Ley and Steel (2007).

In this paper, we propose Bayesian model averaging package for `gretl`, named **BMA**. We can list several reasons why, in our opinion, it is important to address this topic. The `gretl` is increasingly popular free, open-source software for econometric analysis, both for students and academics. It is written in pure C “by econometricians for econometricians” with the international community gathered around two men: Allin Cottrell and Riccardo “Jack” Lucchetti. Since 2009 the community have been meeting every two years at the `gretl` conference. The next conference will be held in Berlin on June 12th/13th, 2015. `gretl` is licensed under the GPL3 license¹ and has been reviewed several times in various international journals. The best known reviews were written by Baiocchi and Distaso (2003), Yalta and Yalta (2007) and Rosenblad (2008). Some brief information about `gretl` was also submitted in Lucchetti (2011) and Adkins (2011). Unlike most other statistical software `gretl` has an easy-to-use graphical user interface (GUI). Our software package is, therefore, a free and easy tool for Bayesian model averaging.

The rest of the paper is organized as follows: Section 2 briefly outlines the Bayesian model averaging for linear regression models with MC³ sampling algorithm and jointness measures. Section 3 provides an overview of `gretl` packages for Bayesian model averaging. Section 4 includes an overview of Bayesian model averaging software. Section 5 presents empirical illustration. The final section concludes.

2 Bayesian inference in normal linear regression models

In this section, we briefly introduce the main features of Bayesian inference in linear regression models. We present Bayesian estimation and prediction with normal-gamma natural conjugate prior and many explanatory variables, as well as model selection and Bayesian model averaging techniques. Finally, in this section we present the basics of Markov Chain Monte Carlo Model Composition sampling algorithm and jointness measures.

¹More details are available at link <http://www.gnu.org/copyleft/gpl.html>.

2.1 Bayesian estimation, model selection and prediction in normal linear regression models

Let us consider the normal linear regression models which differ in their explanatory variables (see Koop (2003) for further details). Suppose that we have K potential explanatory variables, which means there are 2^K possible models and let M_r for $r = 1, \dots, 2^K$ denote 2^K different models under consideration. Suppose also that y_i and x_i denote the observed data on the dependent and explanatory variables for $i = 1, \dots, N$. The observations are placed in $(N \times 1)$ vector y and $(N \times k_r)$ matrix X_r containing the set of regressors included in model M_r ². Thus, we can write our model as

$$y = \alpha \iota_N + X_r \beta_r + \epsilon$$

where ι_N is a $(N \times 1)$ vector of ones, β_r is a $(k_r \times 1)$ vector of unknown parameters, ϵ is a $(N \times 1)$ vector of errors which are assumed to be normally distributed, $\epsilon \sim N(0_N, h^{-1} I_N)$ and h is error precision, which is defined as $h = \frac{1}{\sigma^2}$. Following Koop (2003), the prior for β_r is normally distributed

$$\beta_r \mid h, M_r \sim N \left(0_{k_r}, h^{-1} \left[g X_r^\top X_r \right]^{-1} \right)$$

while we use noninformative prior for intercept and precision

$$p(\alpha) \propto 1, p(h) \propto \frac{1}{h}$$

where $N(\mu, \Sigma)$ denotes a normal distribution with mean μ and variance Σ . The factor of proportionality g is so-called Zellner (1986) g-prior. This prior is a convenient way to specify the prior variance matrix, because it reduces the number of prior variance parameters and considerably simplifies posterior computations. The `gretl` package offers the five most popular alternative Zellner's g-priors (see Fernández *et al.* (2001a) and Moral-Benito (2010))

- Unit Information Prior (g-UIP), recommended by Kass and Wasserman (1995)

$$g = \frac{1}{N}. \tag{1}$$

- Risk Inflation Criterion (g-RIC), proposed by Foster and George (1994)

$$g = \frac{1}{K^2}. \tag{2}$$

- Benchmark Prior, recommended by Fernández *et al.* (2001a)

$$g = \begin{cases} \frac{1}{K^2} & \text{for } N \leq K^2; \\ \frac{1}{N} & \text{for } N > K^2 \end{cases}. \tag{3}$$

- g-HQ prior which mimics the Hannan and Quinn criterion, see Fernández *et al.* (2001a)

$$g = \frac{1}{(\ln N)^3}. \tag{4}$$

²We subtract mean from all regressors as in Fernández *et al.* (2001a).

- Root of g-UIP, see Fernández *et al.* (2001a)

$$g = \sqrt{\frac{1}{N}}. \quad (5)$$

By Bayes rule, the mean of the posterior distribution of slope parameters β_r , conditional with respect to model M_r , can be written as

$$\mathbb{E}(\beta_r | y, M_r) = \left[(1 + g) X_r^\top X_r \right]^{-1} X_r^\top y \quad (6)$$

It is easy to see that if $g \approx 0$ the mean of the posterior distribution (see Equation 6) equals to the OLS estimates. The posterior variance of β_r , conditional with respect to model M_r , is given by

$$\text{VAR}(\beta_r | y, M_r) = \frac{N s_r^2}{N - 2} \left[(1 + g) X_r^\top X_r \right]^{-1} \quad (7)$$

where

$$s_r^2 = \frac{\frac{1}{1+g} y^\top P_{W_r} y + \frac{g}{1+g} (y - \bar{y} \iota_N)^\top (y - \bar{y} \iota_N)}{N}$$

and $P_{W_r} = I_N - W_r (W_r^\top W_r)^{-1} W_r^\top$ for $W_r = (\iota_N, X_r)$.

Marginal data density, conditional with respect to model M_r , may be written as

$$p(y | M_r) \propto \left(\frac{g}{1 + g} \right)^{\frac{k_r}{2}} \left[\frac{1}{1 + g} y^\top P_{W_r} y + \frac{g}{1 + g} (y - \bar{y} \iota_N)^\top (y - \bar{y} \iota_N) \right]^{-\frac{N-1}{2}} \quad (8)$$

In the Bayesian approach to comparing models, it is considered useful to employ probabilities to represent the degree of belief associated with the alternative models. For the normal linear regression models we can easily test two mutually exclusive (non-nested) and jointly exhaustive models with different subset of variables. Using Bayes's theorem, the posterior odds ratio for a model M_l against model M_n is given by

$$\frac{\mathbb{P}(M_l | y)}{\mathbb{P}(M_n | y)} = \frac{\mathbb{P}(M_l) p(y | M_l)}{\mathbb{P}(M_n) p(y | M_n)} \quad (9)$$

where $\frac{\mathbb{P}(M_l)}{\mathbb{P}(M_n)}$ is the prior odds ratio and $\frac{p(y|M_l)}{p(y|M_n)}$ is the Bayes factor. If the ratio $\frac{\mathbb{P}(M_l|y)}{\mathbb{P}(M_n|y)}$ is larger than 1, we can say that the data supports model M_l over model M_n . In our package, we use two popular model priors

- binomial prior

$$\mathbb{P}(M_r) = \theta^{k_r} (1 - \theta)^{K - k_r} \text{ for } r = 1, \dots, 2^K \quad (10)$$

Note that for $\theta = 0.5$ we have a uniform prior on the model space, i.e., $\mathbb{P}(M_r) = 2^{-K}$.

- binomial-beta prior (see Gelman, Carlin, Stern, and Rubin (1997))

$$\mathbb{P}(\Xi = k_r) = \frac{\Gamma(K + 1)}{\Gamma(k_r + 1) \Gamma(K - k_r + 1)} \cdot \frac{\Gamma(a + k_r) \Gamma(K + b - k_r)}{\Gamma(a + b + k_r)} \cdot \frac{\Gamma(a + b)}{\Gamma(a) \Gamma(b)} \quad (11)$$

where Ξ denotes model size.

In our package, we only need to specify the prior expected model size $\mathbf{E}(\Xi) \in (0, K]$. Note that in case of binomial distribution we have $\mathbf{E}(\Xi) = K\theta$ and the choice of $\mathbf{E}(\Xi)$ automatically produces a value for the prior inclusion probability θ . If we have binomial-beta distribution, the average model size will satisfy $\mathbf{E}(\Xi) = \frac{a}{a+b}K$. Here, we follow Ley and Steel (2009) and fix $a = 1$ hence obtaining the value of the second hyperparameter $b = \frac{K - \mathbf{E}(\Xi)}{\mathbf{E}(\Xi)}$.

One of the most widely accepted norms of scientific investigation is Occam's razor. Simply speaking, Occam's razor states that the simplest hypothesis is also the most probable (see for example Madigan and Raftery (1994)). In **BMA** package we penalize large models through the prior model probabilities (Osiewalski and Steel (1993)). Note that one can make $\mathbf{P}(M_r)$ a decreasing function of k_r simply by defining the prior expected model size $\mathbf{E}(\Xi)$. The smaller the prior expected model size $\mathbf{E}(\Xi)$ the less probable large models are.

It is easy to show that the posterior probability of model M_l is given by

$$\mathbf{P}(M_l | y) = \frac{\mathbf{P}(M_l)p(y | M_l)}{\sum_r^{2^K} \mathbf{P}(M_r)p(y | M_r)}$$

The posterior density of vector β is the average of the posterior densities $p(\beta_r | y, M_r)$ conditional on the models

$$p(\beta | y) = \sum_{r=1}^{2^K} \mathbf{P}(M_r | y)p(\beta_r | y, M_r)$$

Once the model posterior probabilities have been calculated, we can also easily evaluate the mean and variance of the posterior distribution of slope parameters (see Leamer (1978))

$$\mathbf{E}(\beta | y) = \sum_{r=1}^{2^K} \mathbf{P}(M_r | y)\mathbf{E}(\beta_r | y, M_r) \quad (12)$$

and

$$\begin{aligned} \text{VAR}(\beta | y) &= \sum_{r=1}^{2^K} \mathbf{P}(M_r | y)\text{VAR}(\beta_r | y, M_r) + \\ &+ \sum_{r=1}^{2^K} \mathbf{P}(M_r | y) (\mathbf{E}(\beta_r | y, M_r) - \mathbf{E}(\beta | y))^2 \end{aligned} \quad (13)$$

In a similar manner, we can find other characteristics of the posterior distribution (see for example (Koop, 2003, p. 266)). We might be also interested in the estimates of posterior inclusion probability $\mathbf{P}(i | y)$ (PIP) i.e., the probability that, conditional on the data, but unconditional with respect to the model space, the variable x_i is relevant in explaining the dependent variable y (see Leamer (1978); Mitchell and Beauchamp (1988); Doppelhofer and Weeks (2009)). The posterior inclusion probability is calculated as the sum of the posterior model probabilities for all of the models including variable x_i .

In Bayesian model averaging approach, predictions of the dependent variable are made by taking a weighted average of the individual model forecast. As described in Fernández *et al.*

(2001b), prediction of an unobserved data point y_f is based on calculating

$$p(y_f | y) = \sum_{r=1}^{2^K} f_S(y_f | N-1, \bar{y} + \frac{1}{g+1} x_{f,r}^\top \beta_r^*, \frac{N-1}{Ns_r^2} \{1 + \frac{1}{N} + \frac{1}{g+1} x_{f,r}^\top (X_j^\top X_j)^{-1} x_{f,r}\}^{-1}) P(M_r | y) \quad (14)$$

where $f_S(y_f | v, b, a)$ denotes the univariate Student-t density with v degrees of freedom, mean b and precision a (with variance $v/\{a(v-2)\}$). In addition, $\beta_r^* = [X_r^\top X_r]^{-1} X_r^\top y$ is Ordinary Least Squares estimator for β_r and $x_{f,r}$ groups the r elements of x_f corresponding to the regressors in model M_r .

2.2 MC³ sampling algorithm

Our MC³ sampling algorithm is based on the Metropolis-Hastings algorithm, and was originally developed by Madigan *et al.* (1995). It simulates a chain of models $M^{(s)}$ for $s = 1, \dots, N$ to find the equilibrium distribution $P(M_r | y)$ of the posterior model probabilities. We do it as follows. We set a candidate model from the set of models, including the previously accepted model $M^{(s-1)}$, all models which delete one independent variable from $M^{(s-1)}$ and all models which add one independent variable to $M^{(s-1)}$. The chain is then constructed by drawing a candidate model M' and the acceptance probability has the form

$$\alpha(M^{(s-1)}, M') = \min \left\{ \frac{P(M')p(y | M')}{P(M^{(s-1)})p(y | M^{(s-1)})}, 1 \right\}$$

In order to assess the stability and the convergence of the chain, we look at the Pearson's correlation between the analytical and MC³ posterior model probabilities. The convergence is achieved if the correlation is above 0.99 (see Fernández *et al.* (2001b) and Koop (2003)). Note that we measure correlation between the analytical and MC³ posterior model probabilities *only* for the top ranked models. If the number of top ranked models is very small, it may lead to high value of Pearson's correlation even when the convergence has not been achieved.

2.3 Jointness measures

The main implementations of model averaging are concerned with selection of variables when model uncertainty is present. Another relevant issue which arises in this framework is to identify whether different sets of two variables x_i and x_j are substitutes, complements or neither over the model space. For that reason, Ley and Steel (2007) and Doppelhofer and Weeks (2009) define the ex-post jointness measures of the dependence between different sets of the explanatory variables. According to Ley and Steel (2007), the logarithm of the jointness statistic has the form

$$J_{LS} = \ln \left[\frac{P(i \cap j | y)}{P(i | y) + P(j | y) - 2P(i \cap j | y)} \right] \quad (15)$$

where $P(i \cap j | y)$ represents the sum of the posterior probabilities of those models that contain both variables x_i and x_j , $P(i | y)$ and $P(j | y)$ are the posterior inclusion probabilities of x_i and x_j .

J_{LS} can be interpreted as the posterior odds ratio of the models including both i and j vs the models that include them only individually (see Ley and Steel (2007)).

An alternative jointness measure was proposed by Doppelhofer and Weeks (2009). It can be written as follows

$$J_{DW} = \ln \left[\frac{P(i \cap j | y)P(\tilde{i} \cap \tilde{j} | y)}{P(i \cap \tilde{j} | y)P(\tilde{i} \cap j | y)} \right] \quad (16)$$

where $P(\tilde{i} \cap \tilde{j} | y)$ denotes the sum of the posterior probabilities of the regression models in which neither x_i and x_j are included, $P(i \cap \tilde{j} | y)$ corresponds to sum of the posterior probabilities of all the models in which x_i is included and x_j is excluded. The last probability $P(\tilde{i} \cap j | y)$ is defined accordingly.

J_{DW} corresponds to the posterior odds of including i , given that j is included, divided by the posterior odds of including i given that j is not included (see Doppelhofer and Weeks (2009)). According to Doppelhofer and Weeks (2009), we use the following classification of jointness among the variables

Evidence	Jointness statistics
strong substitutes	$J_{LS}, J_{DW} \leq -2$
significant substitutes	$-2 < J_{LS}, J_{DW} \leq -1$
not significantly related	$-1 < J_{LS}, J_{DW} < 1$
significant complements	$1 \geq J_{LS}, J_{DW} < 2$
strong complements	$J_{LS}, J_{DW} \geq 2$

3 Implementation in gretl

In this section, we describe the code as well as the use of the `gretl` package for Bayesian model averaging, together with accompanying jointness statistics. First, we will characterize our code and the use of the graphical interface, then we will present how to use **BMA** script. Finally, we will show the outputs that are returned.

3.1 Hansl programming language

“Hansl (the name expands, in recursive fashion, to “Hansl’s a neat scripting language”) is `gretl`’s scripting language.” (Cottrell and Lucchetti, 2014, p. 1). Hansl’s syntax is very similar to C language, including passing pointers to functions. What is very useful for the end user is that Hansl provides a friendly mechanism for building GUI interfaces to its functions/packages. Such packages consist on (at least) one “public” function and zero or more “private” helper functions (see Cottrell and Lucchetti (2013b)). This distinction gives programmers flexibility in writing the packages for `gretl` and allows them to split the code into small pieces (functions) responsible for logically separated computations.

The **BMA** package consists of 1 public and 17 private functions, but only 16 of them are used regularly. The name of each function starts with “BMA_” prefix.

3.2 The core of BMA code

3.2.1 The main function

The core package function which runs and controls the main loop is

```
function matrix BMA_main (
  list big_list "List of all variables for BMA (Y must be the first one)",
  int acc_type[1:2:1] "Model prior" {"Binomial", "Binomial-beta"},
  scalar av_model_size[0::1] "Prior average model size",
  scalar alpha[0:1:0.6] "Significance level for the initial model",
  int l_rank[2::10] "Number of the top ranked models",
  int g_type[1:5:1] "g-prior type" {"Benchmark prior",
    "Unit Information Prior (g-UIP)", "Risk Inflation Criterion (g-RIC)",
    "Hannan and Quinn HQC", "Root of g-UIP"},
  int do_joint[0:2:0] "Jointness analysis" {"None",
    "Ley-Steel Measure", "Doppelhofer-Weeks Measure"},
  int h_predict[0::] "Number of out-of-sample forecasts",
  int Nrep[1000000] "Total number of replications",
  int burn[0:99:10] "Percentage of burn-in draws",
  int verbosity[1:2:1] "Verbosity")
```

The words in quotation marks are the labels for GUI interface shown in Figure 2.

The `big_list` is a `gretl`'s object "named list" which is just a set of $K + 1$ variables (defined by names or dataset ID). What is very important is the fact that the first member of the `big_list` is treated as y variable and the rest of the members are treated as K explanatory variables. Furthermore, the `big_list` cannot contain a `const` (`gretl`'s internal and automatically generated constant term).

The `acc_type[1:2:1]` is an integer indicating model prior type (the default is binomial, see Equations 10, 11).

The `scalar av_model_size[0::1]` is a scalar with prior average model size. Note, if $av_model_size = \frac{K}{2}$ and model prior is set to binomial, we get uniform prior on the model space. The default value is 1.

The `scalar alpha[0:1:0.6]` is the significance level in OLS estimation. An independent variable enters the initial model if its p-value is less than the significance level (see Koop (2003)). The default value is $\alpha = 0.6$, but setting $\alpha = 1$ results in a model consisting of 0 to K randomly chosen explanatory variables.

The `int l_rank[2::10]` is the number of the top ranked models. The default value is 10.

The `int g_type[1:5:1]` indicates a type of g-prior to be used (the default value is 1: "Benchmark prior"), see Equations 1, 2, 3, 4 and 5.

The `int do_joint[0:2:0]` indicates whether we do jointness analysis and if so, which measure to use. The default value is 0: "None", see 2.3.

The `h_predict[0::]` is the number of out-of-sample forecasts.

The `int Nrep[1000000]` is the total number of replications in Monte Carlo simulation. The default value is 1000000.

The `int burn[0:99:10]` is the percentage of burn-in draws ranging from 0% to 99%. The default value is 10%.

The `int verbosity[1:2:1]` indicates the level of verbosity of the **BMA** package when results are printed. The default value is 1, i.e., silent mode.

3.2.2 The main loop

The main loop of the **BMA** package is split into four parts:

1. Setting up the MC³ sampling algorithm.
2. Starting model.
3. Markov Chain Monte Carlo simulation.
4. Results printing.

In the first part of the main loop (*Setting up the MC³ sampling algorithm*), we set up internal variables and also check correctness of the parameters passed to the package. We use two private functions here

1. `function string BMA_parse (list big_list, const scalar *k,
const scalar *av_model_size, const scalar *l_rank);`
2. `function void BMA_scaling_factors (matrix *factors,
const scalar *k, scalar *y_sq, const int g_type, const matrix *Y);`

where arguments indicated by the * modifier are pointers, see Cottrell and Lucchetti (2013b) for explanation. If there is no error, we run the function `BMA_scaling_factors` which calls the function `scalar BMA_gprior (const scalar *k, int type)` to compute g-prior according to Equations 1, 2, 3, 4, and 5. and sets up some scalars needed for further computations.

In the second part of the main loop (*Starting model*), we construct the initial model for MC³ sampling and set up some additional internal variables. Here we use five private functions

1. `function list BMA_initial_model (const series *Y, list X,
const scalar *alpha, scalar *k, matrix var_order);`
2. `function void BMA_new_X_matrix (const matrix *big_mat_dem,
const matrix *Ones, const scalar *k, matrix *X_new_num,
matrix *X_new, list X_list, const scalar *k_new,
const matrix *var_numbers2, const matrix *big_mat_dem_predict[null],
matrix *X_new_predict[null], const scalar *h_predict);`
3. `function void BMA_matrix_precompute (const matrix *Y, matrix *X,
const scalar *k, const matrix *Ones, const matrix *factors,
matrix *XtY, matrix *XtXinv, scalar *yMy,
matrix *ZtZinv[null], const scalar h_predict);`
4. `function void BMA_ols (const matrix *factors, matrix *XtY,
matrix *XtXinv, scalar *yMy, matrix *bhat, matrix *bvar);`
5. `function matrix BMA_model_structure (matrix *X_new_num,
const scalar *k, matrix *models_rank[null],
const scalar *l_models_rank[null], bool start_model[0]);`

The function `BMA_initial_model` returns `X_old_1` – the list of explanatory variables in the initial model according to the scalar `*alpha`. Next, the function `BMA_new_X_matrix` constructs `X_new` – the matrix of demeaned explanatory variables based on the `X_old_1`.

Furthermore, the X_{new} matrix is taken by the function `BMA_matrix_precompute` for the linear algebra computations necessary to compute Equations 6, 7, and 8. Now we run the following code snippet to compute Equation 8

```
lprob_old = scaling_f[5] * (k_new + 1) - scaling_f[6] * log(scaling_f[3] * yMy
+ scaling_f[4])
```

Following, we run the function `BMA_ols` to compute Equations 6 and 7. Finally, we call the function `BMA_model_structure`, which returns the $1 \times K$ row vector with 1 if certain explanatory variable was in the initial model and 0 elsewhere.

In the third part of the main loop (*Markov Chain Monte Carlo simulation*), we discard the first `Nburn = round(burn / 100 * Nrep)` draws as the burn-in replications and then we simulate a chain of models. The most important code snippets are

1. Drawing a candidate model.

```
potential_var = randint(0, k)
...
if (potential_var > 0)
  if (mod_struct[potential_var] == 1)
    X_new_l = X_old_l - var_numbers[potential_var + 1]
  else
    X_new_l = X_old_l + var_numbers[potential_var + 1]
  endif
...

```

2. Making a decision if to accept the candidate model.

```
if (log(randgen1(u, 0, 1)) < BMA_accept_prob(acc_type, &lprob_new,
&lprob_old, &k_new, &k_old, &k, &a, &b, &c))
...
endif;
```

3. Constructing/modifying the analytical and numerical model rankings.

```
function void BMA_build_rank (matrix *mod_rank,
matrix *mod_rank_prob, matrix *mod_nume_prob,
const matrix *mod_struct, const scalar *l_rank,
const scalar *lprob_old);
```

4. Computing Bayesian model averaging stuff.

```
mod_size += k_new
var_prob += mod_struct
loop for i = 1..k_new --quiet
  bhat_avg[X_new_num[i] - 1] += bhat[i + 1]
  bvar_avg[X_new_num[i] - 1] += (bvar[i + 1] + bhat[i + 1]^2)
endloop;
```

5. Out-of-sample forecasting (if needed).

```

if (h_predict > 0 && k_new > 0)
  Yhat = X_new_predict * bhat[2:k_new + 1] + y_mean
  Yhat_avg += Yhat
  precision = ((nobs - 1) / dj) * ((1 + 1 / nobs + scaling_f[3]
    * diag(X_new_predict * ZtZinv * X_new_predict')).^-1)
  Yhat_var_avg += ((nobs - 1) / (precision * (nobs - 3)) + Yhat.^2)
endif;

```

6. Analysing jointness (if needed).

```

function void BMA_jointness_matrix (const matrix *mod_struct,
  const scalar *k, matrix *jointness_m);

```

At 1. we draw the number of a variable ranging from 0 to K using the `gretl`'s build-in function `randint()` which uses the SIMD-oriented fast Mersenne twister (SFMT) random number generator (see Cottrell and Lucchetti (2013a); Yalta and Schreiber (2012))³. If the drawn variable was in the last model, this variable was then removed from it, otherwise it was added to the last model.

At 2. we take a decision whether to accept the new draw (model) or not. We call the `BMA_accept_prob` function, which implements posterior odds ratio (see Equation 9).

At 3. we call the `BMA_build_rank` function, which is responsible for creating analytical, as well as numerical rankings.

At 4. we do some counting needed for the essential Bayesian model averaging computations formulated in Equations 12 and 13, that is the mean and variance of the posterior distribution of slope parameters, as well as average model size and posterior inclusion probability.

At 5. if out-of-sample forecasts were chosen we compute mean and variance of predictions according to Equation 14.

Finally at 6., if jointness analysis was chosen, we call the `BMA_jointness_matrix` function, which counts each coexistence (jointness) of every pair of explanatory variables in the given draw.

In the last part of the main loop (*Results printing*), we finally call the `BMA_print_results` function in order to print the MC³ sampling results. A detailed description of the structure of the results printed here will be depicted in Section 3.3.

3.2.3 The matrix returned by BMA package

The **BMA** package can optionally return a matrix containing substantial results obtained in the analysis. The structure of that matrix is shown on Figure 1. The result matrix has K rows, one for each explanatory variable. The first five columns are: the posterior inclusion probabilities, the posterior mean and standard deviation of each coefficient (Mean and Std.Dev.) and the posterior mean and standard deviation of each coefficient conditional on the variable being included in the model (Cond.Mean and Cond.Std.Dev.), see Outputs

³The code of **BMA** package contains private function `ran2` which implements so-called “ran2” random number generator (RNG) by L’Ecuyer with Bays-Durham shuffle and added safeguards (see Press, Flannery, Teukolsky, and Vetterling (1988)). We implemented this RNG for convenience in replication of the earlier published results, see Fernández *et al.* (2001b) and Ley and Steel (2007). In the **BMA**'s main loop code there are guidelines how to switch from SFMT to `ran2` RNG. Note: our `ran2` function is much slower than `gretl`'s internal RNG.

on page 15 for details. The next K columns appear only if any of the jointness analysis was selected and contain values of one of the jointness measures: Ley-Steel or Doppelhofer-Weeks.

	PIP	Mean	Std.Dev.	Cond.Mean	Cond.Std.Dev.	x_1	x_2	x_3	...	x_K
x_1										
x_2										
x_3										
\vdots										
x_K										

Figure 1: Structure of the matrix returned by the **BMA** package.

3.3 Usage of the BMA package

3.3.1 The GUI way

Once you start `gretl`, you must open a data file and then you can load the relevant **BMA** function package from the `gretl` server (In the main window, go to `File > Function files > On server heading`). By choosing it, you will open a window similar to the one shown in Figure 2.

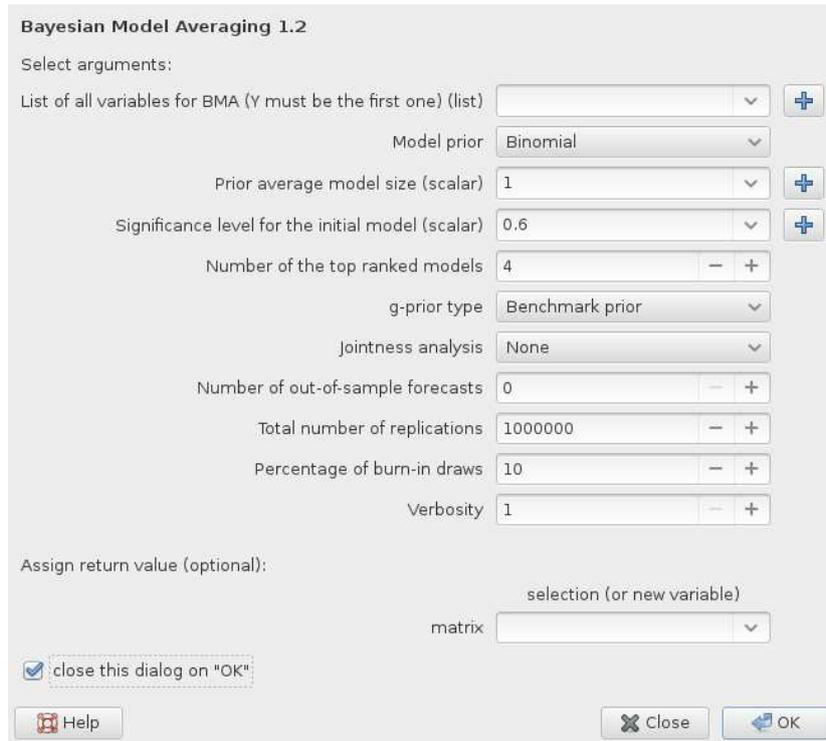


Figure 2: Main window for **BMA**.

According to Figure 2, we can specify the following entries in the GUI **BMA** window

- **List of all variables for BMA (Y must be the first one)** - Loading variables from the database, which must have been opened previously. The dependent variable must

be the first one on the list of the variables currently available. Notice that by default we assume that you want to estimate an intercept; therefore, a constant is implicitly included to the list of the variables.

- **Model prior** - Indicates the choice of model prior. One can employ the binomial model prior or the binomial-beta model prior. Note that the uniform model prior is a special case of binomial model prior therefore, in fact, our package allows three types of priors.
- **Prior average model size** - Specifies the prior expected model size $E(\Xi)$. The expected model size may range from 0 to K . The default value equals 1. It is the smallest reasonable prior expected model size. It means that we penalize large models and we assign high probability for small ones following the Occam's razor philosophy. Note that for the binomial model prior and $E(\Xi) = 0.5K$ one can define the uniform prior on the model space. For example, for $K = 10$ regressors we can define the uniform prior with $E(\Xi) = 5$. For $E(\Xi) < 5$ we assign high probability to small models. The smaller the prior expected model size $E(\Xi)$, the less probable larger models are.
- **Significance level for the initial model** - Defines the significance level which was used to build the initial model. An explanatory variable enters the initial model if its p-value is less than the significance level. If the significance level equals 1, the initial model will be randomly chosen (with equal probability) from all available models. Note that if all available explanatory variables enter the initial model, you will get the following gretl error message "No independent variables were omitted".
- **Number of the top ranked models** - Specifies the number of the best models for which detailed information is stored.
- **g-prior type** - Here one can choose between four Zellner's g-prior for the regression coefficients. Choices include: Benchmark prior, Unit Information Prior, Risk Inflation Criterion, Hannan and Quinn prior, Root of g-UIP.
- **Jointness analysis** - If 'None' (the default), the jointness analysis is omitted. On the other hand, one can choose jointness measure of Ley and Steel (2007) or Doppelhofer and Weeks (2009).
- **Number of out-of-sample forecasts** - Defines the total number of out-of-sample forecasts of the dependent variable.
- **Total number of replications** - Defines the total number of iteration draws to be sampled.
- **Percentage of burn-in draws** - Provides a number of burn-in replications, calculated as the percentage of the total number of iteration draws.
- **Verbosity** - An integer ranging from 1 to 2: the default is 1, which allows to see the basic Bayesian model averaging results. If verbosity equals 2, a more detailed description of the analysis is provided (initial model, speed of convergence, estimation results for top ranked models).
- **matrix** - The output can be saved under a specified name to the current session.

3.3.2 Script

The **BMA** package can also be used inside **Hansl** scripts. The very minimal code could be as follows

```
open greene9_1.gdt
include BMA.gfn
```

```
list greene = dataset
```

```
BMA_main(greene, 1, 1.5, 0.6, 4, 1, 0, 0, 1000000, 10, 1)
```

The above example consists of three blocks.

The first block is just the opening of the so-called `greene9_1` dataset, which is bundled in every standard `gretl` installation. This dataset contains the cross-sectional data on the manufacturing of transportation equipment presented in Greene (1999)⁴ and it consists of the following variables: `valadd` – output, `capital` – capital input, `labor` – labor input and `nfir` – number of firms.

The second block is the definition of the `green` list which contains all variables available in the `greene9_1` dataset. The first variable – `valadd` – will be the dependent variable.

The third block contains the definition for Bayesian model averaging analysis: binomial model prior, prior average model size set to 1.5 (which means that we have uniform model prior), significance level for the initial model set to 0.6, 4 top ranked models, Benchmark `g`-prior, without jointness analysis, without out-of-sample predictions, 100000 replications with 10% burn-in draws and basic output (verbosity set to 1).

Suppose we want to run the same analysis but following the Occam's razor philosophy. The simplest way to do it, is to set the prior average model size to 1. The code would be as follows

```
BMA_main(greene, 1, 1, 0.6, 4, 1, 0, 0, 1000000, 10, 1)
```

In this case, we penalize large models assigning the high probability to the small ones. Apart from the binomial model prior, we can use the binomial-beta prior. To do it, we have to set model prior to 2. The code would be as follows

```
BMA_main(greene, 2, 1, 0.6, 4, 1, 0, 0, 1000000, 10, 1)
```

Suppose we want to set `g`-prior to Risk Inflation Criterion, uniform model prior, do jointness analysis with Ley-Steel Measure, compute forecasts for 3 observations and print additional information (verbosity set to 2). The code would be as follows

```
BMA_main(greene, 1, 1.5, 0.6, 4, 3, 1, 3, 1000000, 10, 2)
```

Finally, if we want to save results of the above Bayesian model averaging analysis in the matrix named `results_mat`, the code should be as follows

```
results_mat = BMA_main(greene, 1, 1.5, 0.6, 4, 3, 1, 3, 1000000, 10, 2)
```

⁴This dataset is also available at link <http://people.stern.nyu.edu/wgreene/Text/Edition7/TableF7-2.txt>.

3.3.3 Outputs

If you select the appropriate entries in the GUI **BMA** window, our package returns: PIP, Mean, Std.Dev., Cond.Mean, Cond.Std.Dev. What is more, the package also returns the predictive results of the dependent variable (Mean and Std.Dev.).

Let us consider the data used in Fernández *et al.* (2001b) (FLS hereafter). This data comprises the information about 72 countries and 41 potential growth determinants for the period 1960 to 1992⁵. For a given sample size, the initial estimation period is $t = 1, \dots, 69$ and the out-of-sample evaluation period is from 70 to 72.

For example, for the FLS data, the following estimates should appear

```

Posterior moments (unconditional and conditional on inclusion):
      PIP      Mean      Std.Dev.      Cond.Mean      Cond.Std.Dev
GDPsh560  0.998744  -1.445131  0.305403  -1.446948  0.301263
Confuncious  0.887263  0.423122  0.190773  0.476884  0.124018
Life_Exp  0.933948  0.990182  0.401268  1.060211  0.313302
Equip_Inv  0.905398  0.552386  0.250695  0.610103  0.184937
SubSahara  0.651029  -0.427443  0.368546  -0.656566  0.241246
Muslim  0.348160  0.123571  0.192175  0.354925  0.154797
Rule_of_Law  0.408207  0.200133  0.268593  0.490273  0.185691
Yrs_Open  0.560306  0.318524  0.318723  0.568484  0.197999
      ...      ...      ...      ...      ...
-----

```

Posterior probability of models:

```

Model 1: 0.006757
Model 2: 0.005386
Model 3: 0.003674
Model 4: 0.003350
Model 5: 0.003137
      ...      ...

```

Total probability of the models in ranking (numerical): 0.057048

Correlation coefficient between the analytical
and numerical probabilities of the above models: 0.969230

Predictive results:

```

      Growth      Mean      Std.Dev.
Obs. 70  2.714600  1.552617  1.280030
Obs. 71  2.067600  2.013866  0.940486
Obs. 72  1.889700  2.459841  0.980175

```

The **BMA** estimate function accepts a scalar, which sets the verbosity of the output. Its default value is 1, which causes the estimation output to be printed out. Value 2 forces **BMA** function to print out all the details of the estimation. You can print out the above-mentioned

⁵The dataset is publicly available on the Journal of Applied Econometrics online data archive. The reported chain (1000000 replications with 10% burn-in draws) took about 17 minutes of CPU time on a PC with AMD Phenom II X6 1100T CPU, 6.0 Gb of RAM running under Debian GNU/Linux. We used `gretl` 1.9.92cvs compiled by GCC 4.9.1. The seed for RNG was set to 1000000. The dependent variable is the growth rate from 1960-1996.

results and additionally the following information: the total CPU time, type of model prior, prior average model size, significance level for the initial model, type of g-prior, total number of iterations and finally number of burn-in draws. Moreover, the **BMA** estimate function produces the estimation results for the initial and top ranking models. The jointness analysis is inactive by default. If is active you will get: posterior joint probability of explanatory variables, jointness statistic Equation 15 or 16 and classification of jointness measures. The jointness analysis for the previous example should look like this

Posterior joint probability of variables:

	GDPsh560	Confuncious	Life_Exp	Equip_Inv	...
GDPsh560	0.000000	0.886008	0.933928	0.904142	...
Confuncious	0.000000	0.000000	0.824880	0.813726	...
Life_Exp	0.000000	0.000000	0.000000	0.841522	...
Equip_Inv	0.000000	0.000000	0.000000	0.000000	...
...

Jointness statistics (Ley-Steel Measure):

	GDPsh560	Confuncious	Life_Exp	Equip_Inv	...
GDPsh560	0.000000	2.050596	2.667528	2.244121	...
Confuncious	0.000000	0.000000	1.570940	1.594406	...
Life_Exp	0.000000	0.000000	0.000000	1.683428	...
Equip_Inv	0.000000	0.000000	0.000000	0.000000	...
...

Strong substitutes:

Brit_Col,Jewish	-4.475934
Outwar_Or,Rev_Coup	-4.453896
...	...

Significant substitutes:

SubSahara,Catholic	-1.995271
Life_Exp,War_Dummy	-1.986929
...	...

Significant complements:

Confuncious,Life_Exp	1.570940
Confuncious,Equip_Inv	1.594406
...	...

Strong complements:

GDPsh560,Confuncious	2.050596
GDPsh560,Equip_Inv	2.244121
...	...

4 Comparison to other BMA software

Bayesian model averaging software has been developed by many researchers⁶. The most popular in the literature are R packages. A recent and comprehensive overview of Bayesian model

⁶The useful information about Bayesian model averaging software are available at link <http://bms.zeugner.eu/resources/>.

averaging software in R is given in Amini and Parmeter (2011). They outline the following BMA packages: **BMS** by Feldkircher and Zeugner (2013), **BAS** by Clyde, Littman, and Ghosh (2012) and **BMA** by Raftery, Hoeting, Volinsky, Painter, and Yeung (2014). According to their research the **BMS** package provides the most accurate results and it is also the most flexible with numerous options. Table 1 presents the main features of the abovementioned packages. furthermore we compare our **BMA** package to **FLSBMA** program developed in Fortran 77 by Fernández *et al.* (2001b).

Main features	R-BMS	R-BMA	R-BAS	FLSBMA	gretl-BMA
Model Sampling	MC ³	BIC	BAS, AMCMC, BAS+MCMC	MC ³	MC ³
Model Priors	5	1	3	2	3
Zellner’s g-Priors	6	1	9	9	5
Jointness Analysis				✓	✓
Jointness Measures				1	2
Out-of-Sample Forecast	✓		✓	✓	✓
No. of Burn-in Draws	✓		✓	✓	✓
No. of Draws	✓	✓	✓	✓	✓
No. of Top Ranked Models	✓			✓	✓
Graphical User Interface					✓

Table 1: Main features of the selected Bayesian model averaging software. Model Sampling – type of model sampling algorithms, Model Priors – number of available model priors, Zellner’s g-Priors – number of available Zellner’s factors of proportionality, Jointness Analysis – information about which package provides jointness analysis, Jointness Measures – number of available jointness measures, Out-of-Sample Forecast – information about which package can be used for out-of-sample forecasting, Number of Burn-in Draws, Number of Draws and Number of Top Ranked Models – number of user-defined parameters in the sampling algorithm, Graphical User Interface – presence of GUI in the package.

The most flexible package according to the model sampling category is **R-BAS** which contains three sampling algorithms: Bayesian Adaptive Sampling (BAS), Adaptive MCMC (AMCMC) and Bayesian Adaptive Sampling with MCMC for the initial replications (BAS+MCMC). All packages offer different numbers of model priors. The most sparse is **R-BMA** with only fixed model prior, while the most comprehensive is **R-BMS** which contains the following model priors: fixed, uniform, binomial, binomial-beta and custom (user-controlled prior probabilities). What is more, the number of g-priors may differ depending on the certain package starting from one for **R-BMA**, to nine for **R-BAS** and **FLSBMA**.

The **gretl-BMA** and **FLSBMA** are the only packages which offer Bayesian model averaging together with accompanying jointness statistics. **FLSBMA** contains one jointness measure (Ley-Steel) while the **gretl-BMA** contains two measures (Ley-Steel and Doppelhofer-Weeks). The estimation of out-of-sample forecasts is possible in almost all packages excluding **R-BMA**, where this option is not available. Three out of five packages enable full control of the replication algorithm. In the **R-BMS**, **FLSBMA** and the **gretl-BMA**, the user can set the number of burn-in draws, replication draws and detailed information about top ranked models. Unlike the other packages the **gretl-BMA** offers an easy-to-use GUI interface for Bayesian model averaging approach.

In order to compare runtime of the above mentioned packages we used the FLS dataset with the following options: benchmark g-prior, 1000000 burn-in draws, 2000000 iterations. Model prior was set to uniform. We considered two scenarios for our experiment: with or without jointness analysis.

Runtime	R-BMS	R-BMA	R-BAS	FLSBMA	gretl-BMA
without jointness analysis	249	318	145	669	2535
with jointness analysis				710	4800

Table 2: Runtime in seconds of Bayesian model averaging software for FLS data.

According to the results presented in Table 2, the fastest Bayesian model averaging software is the **R-BAS**, while the slowest one is **gretl-BMA**. We are aware of this disadvantage of our package, but it strongly depends on the **gretl** development process, especially on the improvements of the **Hansl** interpreter. It is possible that in the near future **gretl** will be much faster than the current 1.9.92cvs version (see Conclusions at the end of the paper for more details).

5 Empirical illustration

In this section we examine the ability of our package in replicating the results published by Fernández *et al.* (2001b) and Ley and Steel (2007). We use the same original dataset to attempt to replicate their results. In our empirical illustration, we discard the first 1 million models and draw samples from the model space 2 million times. We specify the following entries in the GUI **BMA** window: model prior = ‘binomial’, prior average model size = ‘20.5’ (we set the models priors to the uniform distribution), number of the top ranked models = ‘20’, g-prior type = ‘Benchmark prior’, total number of replications = ‘3000000’, percentage of burn-in draws = ‘33’. Table 3 and 4 present the estimation results⁷. They report the posterior means, standard errors of regressors and PIPs calculated from the **R-BMS**, **R-BMA**, **R-BAS** and **FLSBMA** packages. These benchmarking results allow us to compare and analyse the performance of our package. Bold font indicates identical results.

⁷The reported chain took about 42 minutes of CPU time on a PC with AMD Phenom II X6 1100T CPU, 6.0 Gb of RAM running under Debian GNU/Linux. We used **gretl** 1.9.92cvs compiled by GCC 4.9.1. The seed for RNG was set to 1000000.

Regressors	R-BMS		R-BMA		R-BAS		FLSBMA		gretl-BMA	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
log GDP in 1960	-1.425	0.277	-1.590	0.197	-1.464	0.242	-1.418	0.269	-1.428	0.275
Fraction Confucian	0.494	0.128	0.637	0.089	0.577	0.103	0.490	0.117	0.494	0.126
Life expectancy	0.965	0.391	1.018	0.215	1.007	0.241	0.957	0.371	0.967	0.386
Equipment investment	0.550	0.236	0.507	0.122	0.451	0.162	0.558	0.222	0.550	0.234
Sub-Saharan dummy	-0.475	0.347	-0.923	0.173	-0.850	0.210	-0.477	0.334	-0.474	0.346
Fraction Muslim	0.257	0.229	0.138	0.155	0.139	0.196	0.257	0.219	0.261	0.229
Population growth	0.005	0.048	-0.005	0.042	0.006	0.059	0.003	0.035	0.005	0.048
...					...					
British colony dummy	-0.003	0.031	0.352	0.139	0.148	0.191	-0.002	0.022	-0.003	0.031
Outward orientation	-0.004	0.029	-0.158	0.100	-0.070	0.100	-0.002	0.021	-0.004	0.030
Fraction Jewish	-0.002	0.027	n/a	n/a	0.000	0.021	-0.001	0.020	-0.002	0.027
Revolutions and coups	0.000	0.023	n/a	n/a	0.000	0.022	0.000	0.017	0.000	0.023
Public education share	0.001	0.024	0.018	0.060	0.061	0.109	0.000	0.017	0.001	0.023
Area (scale effect)	-0.001	0.021	n/a	n/a	-0.002	0.030	-0.001	0.014	-0.001	0.021

Table 3: The posterior estimates of regression coefficients based on Bayesian model averaging software for FLS data. The dependent variable is the growth rate from 1960-1996 across 72 countries. All the regressors have been centered to have zero mean. Abbreviation “n/a” means that **R-BMA** dropped variable in pre-selection procedure. Bold font denotes identical results.

As it is apparent in Table 3 and 4, the only minor differences in posterior results are found between **gretl-BMA** and the results published in Ley and Steel (2007). All the PIPs

and estimated posterior means or standard deviations are reasonably close to all cases and the same variables are identified to be relevant. Note that the `gretl` package results are almost identical to the results produced by the `R-BMS` package.

Regressors	R-BMS	R-BMA	R-BAS	FLSBMA	gretl-BMA
log GDP in 1960	1.00	1.00	1.00	1.00	1.00
Fraction Confucian	0.99	1.00	1.00	1.00	0.99
Life expectancy	0.93	1.00	1.00	0.95	0.94
Equipment investment	0.92	1.00	0.98	0.94	0.92
Sub-Saharan dummy	0.74	1.00	1.00	0.76	0.74
Fraction Muslim	0.64	0.56	0.43	0.66	0.65
Population growth	0.04	0.05	0.08	0.02	0.04
...			...		
British colony dummy	0.04	0.97	0.46	0.02	0.04
Outward orientation	0.04	0.85	0.42	0.02	0.04
Fraction Jewish	0.03	n/a	0.05	0.02	0.04
Revolutions and coups	0.03	n/a	0.05	0.02	0.03
Public education share	0.03	0.13	0.32	0.02	0.03
Area (scale effect)	0.03	n/a	0.06	0.02	0.03

Table 4: Posterior inclusion probabilities based on Bayesian model averaging software for FLS data. Abbreviation “n/a” means that `R-BMA` dropped variable in pre-selection procedure. Bold font denotes identical results.

In our opinion, the main reason for the very similar results produced by the `R-BMS` and `gretl-BMA` is implementation of the same RNG algorithm (the SIMD-oriented Fast Mersenne Twister), while `FLSBMA` package uses an older RNG algorithm i.e., “ran2” developed by L’Ecuyer. The `R-BMA` and `R-BAS` packages are not capable to reproduce our results (see Amini and Parmeter (2011) for additional information).

6 Conclusions

This paper has outlined the new software package that implements Bayesian model averaging and jointness measures for `gretl`. Bayesian model averaging is a straightforward and natural extension of standard Bayesian analysis and it is a useful and popular alternative to other variable selection procedures, especially for a large set of regressors. Here we used `gretl`, which is a free, open-source software for econometric analysis with an easy-to-use GUI. Our goal was to familiarize potential users with the features and the different options that our package has to offer. We described how our package implements the Bayesian model averaging, as well as the outputs that are returned.

The recent trends in Bayesian model averaging still concern applications in various areas of empirical research, i.e., economic growth (see Cuaresma, Doppelhofer, and Feldkircher (2014), Gazda and Puziak (2012)), stock returns (see Beckmann and Schüssler (2014)), geography (see Baran (2014)) and many more. One can also encounter new papers about theory of Bayesian model averaging, for example: two-stage Bayesian model averaging (see Lenkoski, Eicher, and Raftery (2014)) or comparison of Bayesian model averaging to Weighted Average Least Squares method (see De Luca and Magnus (2011)).

The future development of the **BMA** package could be focused on two main aspects: speed improvements and features extension. The first one is strongly correlated with the **gretl** development process, especially with work on improvements in speed of the **Hansl** interpreter. It is possible that the interpreter will be rewritten in the near future to achieve speed boost with backward compatibility. The features extension would concern two aspects: implementing new g-priors and model priors. In the literature we can find many types of g-priors (see Fernández *et al.* (2001a)). Moreover new model priors could be also considered. For instance, Sala-I-Martin, Doppelhofer, and Miller (2004) proposed so-called fixed model prior. These implementation surely would improve flexibility of the **BMA** package.

References

- Adkins LC (2011). “Using **gretl** for Monte Carlo Experiments.” *Journal of Applied Econometrics*, **26**(5), 880–885. ISSN 1099-1255. doi:10.1002/jae.1228. URL <http://dx.doi.org/10.1002/jae.1228>. 2
- Amini S, Parmeter CF (2011). “Bayesian Model Averaging in R.” *Journal of Economic and Social Measurement*, **36**(4), 253–287. doi:10.3233/JEM-2011-0350. URL <http://iospress.metapress.com/content/W15378385573L08X>. 17, 20
- Baiocchi G, Distaso W (2003). “GRETLM: Econometric Software for the GNU Generation.” *Journal of Applied Econometrics*, **18**(1), 105–110. ISSN 1099-1255. doi:10.1002/jae.704. URL <http://dx.doi.org/10.1002/jae.704>. 2
- Baran S (2014). “Probabilistic Wind Speed Forecasting Using Bayesian Model Averaging with Truncated Normal Components.” *Computational Statistics & Data Analysis*, **75**(C), 227–238. doi:10.1016/j.csda.2014.02.013. URL <http://ideas.repec.org/a/eee/csdana/v75y2014icp227-238.html>. 20
- Beckmann J, Schüssler R (2014). “Forecasting Equity Premia Using Bayesian Dynamic Model Averaging.” *CQE Working Papers 2914*, Center for Quantitative Economics (CQE), University of Muenster. URL <http://ideas.repec.org/p/cqe/wpaper/2914.html>. 20
- Clyde M, Littman M, Ghosh J (2012). *Bayesian Model Averaging Using Bayesian Adaptive Sampling*. URL <http://cran.r-project.org/package=BAS>. 17
- Cottrell A, Lucchetti R (2013a). *gretl Command Reference – Gnu Regression, Econometrics and Time-Series Library*. URL <http://ricardo.ecn.wfu.edu/pub/gretl/manual/PDF/gretl-ref.pdf>. 11
- Cottrell A, Lucchetti R (2013b). *gretl User’s Guide – Gnu Regression, Econometrics and Time-Series Library*. URL <http://ricardo.ecn.wfu.edu/pub/gretl/manual/PDF/gretl-guide.pdf>. 7, 9
- Cottrell A, Lucchetti R (2014). *A Hansl Primer*. URL <http://ricardo.ecn.wfu.edu/pub/gretl/manual/PDF/hansl-primer-a4.pdf>. 7
- Cuaresma JC, Doppelhofer G, Feldkircher M (2014). “The Determinants of Economic Growth in European Regions.” *Regional Studies*, **48**(1), 44–67. doi:10.1080/00343404.2012.678824. URL <http://ideas.repec.org/a/taf/regstd/v48y2014i1p44-67.html>. 20

- De Luca G, Magnus J (2011). “Bayesian Model Averaging and Weighted Average Least Squares: Equivariance, Stability, and Numerical Issues.” *Discussion Paper 2011-082*, Tilburg University, Center for Economic Research. URL <http://ideas.repec.org/p/dgr/kubcen/2011082.html>. 20
- Doppelhofer G, Weeks M (2005). “Jointness of Growth Determinants.” *Cambridge Working Papers in Economics 0542*, Faculty of Economics, University of Cambridge. URL <http://ideas.repec.org/p/cam/camdae/0542.html>. 2
- Doppelhofer G, Weeks M (2009). “Jointness of Growth Determinants.” *Journal of Applied Econometrics*, **24**(2), 209–244. ISSN 1099-1255. doi:10.1002/jae.1046. URL <http://dx.doi.org/10.1002/jae.1046>. 1, 2, 5, 6, 7, 13
- Feldkircher M, Zeugner S (2013). *Bayesian Model Averaging with BMS*. URL <http://cran.r-project.org/package=BMS>. 17
- Fernández C, Ley E, Steel MFJ (2001a). “Benchmark Priors for Bayesian Model Averaging.” *Journal of Econometrics*, **100**(2), 381–427. ISSN 0304-4076. doi:10.1016/S0304-4076(00)00076-2. URL <http://www.sciencedirect.com/science/article/pii/S0304407600000762>. 2, 3, 4, 21
- Fernández C, Ley E, Steel MFJ (2001b). “Model Uncertainty in Cross-Country Growth Regressions.” *Journal of Applied Econometrics*, **16**(5), 563–576. ISSN 1099-1255. doi:10.1002/jae.623. URL <http://dx.doi.org/10.1002/jae.623>. 2, 5, 6, 11, 15, 17, 18
- Foster DP, George EI (1994). “The Risk Inflation Criterion for Multiple Regression.” *The Annals of Statistics*, **22**(4), 1947–1975. doi:10.1214/aos/1176325766. URL <http://dx.doi.org/10.1214/aos/1176325766>. 3
- Gazda J, Puziak M (2012). “The Sources of Economic Growth in the Regions.” In M Kocińska (ed.), *Stages of the Convergence in the Developed European Economies*, pp. 107–125. Poznań University of Economics Press, Poznań, Poland. ISBN 978-83-7417-711-5. 20
- Gelman A, Carlin J, Stern H, Rubin D (1997). *Bayesian Data Analysis*. Chapman and Hall, London, UK. 4
- Greene WH (1999). *Econometric Analysis*. 4th edition. Prentice Hall, Upper Saddle River, NJ, USA. 14
- Hoeting JA, Madigan D, Raftery AE, Volinsky CT (1999). “Bayesian Model Averaging: A Tutorial.” *Statistical Science*, **14**(4), 382–417. doi:10.1214/ss/1009212519. URL <http://dx.doi.org/10.1214/ss/1009212519>. 2
- Kass RE, Wasserman L (1995). “A Reference Bayesian Test for Nested Hypotheses and its Relationship to the Schwarz Criterion.” *Journal of the American Statistical Association*, **90**(431), 928–934. doi:10.1080/01621459.1995.10476592. URL <http://www.tandfonline.com/doi/abs/10.1080/01621459.1995.10476592>. 3
- Koop G (2003). *Bayesian Econometrics*. John Wiley & Sons, New Jersey, USA. ISBN 978-04-7084-567-7. 2, 3, 5, 6, 8

- Leamer E (1978). *Specification Searches*. John Wiley & Sons, New Jersey, USA. ISBN 0471015202. 5
- Lenkoski A, Eicher TS, Raftery AE (2014). “Two-Stage Bayesian Model Averaging in Endogenous Variable Models.” *Econometric Reviews*, **33**(1-4), 122–151. doi:10.1080/07474938.2013.807150. URL <http://ideas.repec.org/a/taf/emetr/v33y2014i1-4p122-151.html>. 20
- Ley E, Steel MFJ (2007). “Jointness in Bayesian Variable Selection with Applications to Growth Regression.” *Journal of Macroeconomics*, **29**(3), 476–493. ISSN 0164-0704. doi:10.1016/j.jmacro.2006.12.002. URL <http://www.sciencedirect.com/science/article/pii/S0164070407000560>. 1, 2, 6, 7, 11, 13, 18, 19
- Ley E, Steel MFJ (2009). “On the Effect of Prior Assumptions in Bayesian Model Averaging with Applications to Growth Regression.” *Journal of Applied Econometrics*, **24**(4), 651–674. ISSN 1099-1255. doi:10.1002/jae.1057. URL <http://ideas.repec.org/a/jae/japmet/v24y2009i4p651-674.html>. 5
- Lucchetti R (2011). “State Space Methods in gretl.” *Journal of Statistical Software*, **41**(11), 1–22. ISSN 1548-7660. URL <http://www.jstatsoft.org/v41/i11>. 2
- Madigan D, Raftery AE (1994). “Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occam’s Window.” *Journal of the American Statistical Association*, **89**(428), 1535–1546. doi:10.1080/01621459.1994.10476894. URL <http://www.tandfonline.com/doi/abs/10.1080/01621459.1994.10476894>. 5
- Madigan D, York J, Allard D (1995). “Bayesian Graphical Models for Discrete Data.” *The International Statistical Review*, **63**(2), 215–232. URL <http://www.jstor.org/stable/1403615>. 2, 6
- Mitchell TJ, Beauchamp JJ (1988). “Bayesian Variable Selection in Linear Regression.” *Journal of the American Statistical Association*, **83**(404), 1023–1032. doi:10.1080/01621459.1988.10478694. URL <http://www.tandfonline.com/doi/abs/10.1080/01621459.1988.10478694>. 5
- Moral-Benito E (2010). “Model Averaging in Economics.” *Working Papers wp2010_1008*, CEMFI. URL http://ideas.repec.org/p/cmfi/wpaper/wp2010_1008.html. 3
- Osiewalski J, Steel MFJ (1993). “Una Perspectiva Bayesiana en Selección de Modelos; eng. A Bayesian Perspective on Model Selection.” *Cuadernos Económicos de ICE*, **55**, 327–351. ISSN 0210-2633. <http://www.cyfronet.krakow.pl/~eeosiewa/A%20Bayesian%20Perspective%20on%20Model%20Selection.pdf>, URL <http://dialnet.unirioja.es/servlet/articulo?codigo=44762>. 5
- Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1988). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, USA. ISBN 0-521-35465-X. 11
- Raftery A, Hoeting J, Volinsky C, Painter I, Yeung KY (2014). *Bayesian Model Averaging – Package BMA*. URL <http://cran.r-project.org/package=BMA>. 17

- Raftery AE, Madigan D, Hoeting JA (1997). “Bayesian Model Averaging for Linear Regression Models.” *Journal of the American Statistical Association*, **92**(437), 179–191. doi:10.1080/01621459.1997.10473615. URL <http://www.tandfonline.com/doi/abs/10.1080/01621459.1997.10473615>. 1
- Rosenblad A (2008). “gretl 1.7.3.” *Journal of Statistical Software, Software Reviews*, **25**(1), 1–14. ISSN 1548-7660. URL <http://www.jstatsoft.org/v25/s01>. 2
- Sala-I-Martin X, Doppelhofer G, Miller RI (2004). “Determinants of Long-Term Growth: A Bayesian Averaging of Classical Estimates (BACE) Approach.” *The American Economic Review*, **94**(4), 813–835. doi:10.1257/0002828042002570. URL <http://ideas.repec.org/a/aea/aecrev/v94y2004i4p813-835.html>. 21
- Steel MFJ (2011). “Bayesian Model Averaging and Forecasting.” *Bulletin of E.U. and U.S. Inflation and Macroeconomic Analysis*, **200**, 30–41. URL <http://e-revistas.uc3m.es/index.php/BIMA/article/view/1240>. 2
- Yalta AT, Schreiber S (2012). “Random Number Generation in gretl.” *Journal of Statistical Software, Code Snippets*, **50**(1), 1–13. ISSN 1548-7660. URL <http://www.jstatsoft.org/v50/c01>. 11
- Yalta AT, Yalta AY (2007). “GRETTL 1.6.0 and Its Numerical Accuracy.” *Journal of Applied Econometrics*, **22**(4), 849–854. ISSN 1099-1255. doi:10.1002/jae.946. URL <http://dx.doi.org/10.1002/jae.946>. 2
- Zellner A (1986). “On Assessing Prior Distributions and Bayesian Regression Analysis with g-Prior Distributions.” In P Goel, A Zellner (eds.), *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti*. Elsevier, Amsterdam, Holland. ISBN 978-04-4487-712-3. 3