Bayesian Model Averaging and Jointness Measures for gretl

Blazejowski, Marcin and Kwiatkowski, Jacek

Wyższa Szkoła Bankowa w Toruniu, Uniwersytet Mikolaja Kopernika w Toruniu

10 February 2013

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

February 10, 2013

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 (**BMA**) is a model-building strategy that takes account of model uncertainty into 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 package devoted to **BMA** for different statistical and econometric software. In this paper, we propose **BMA** package for gretl, which is more and more popular free, open-source software for econometric analysis with easy-to-use GUI. We introduce **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 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. AIC and 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. It 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 main questions of interest.

Bayesian model averaging is a useful alternative to other variable selection procedures, because it incorporates model uncertainty into conclusions about estimated parameters. The **BMA** 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 **BMA** can be found in a number of works (see, for example, Hoeting, Madigan, Raftery, and Volinsky (1999) and 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 MC^3 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 linear regression models. We use that measure to identify whether different sets of two variables are substitutes, complements or neither. Similar jointness measure was also proposed by Ley and Steel (2007).

In this paper, we propose **BMA** package for gretl. We can list several reasons why, in our opinion, it is important to address this topic. The gretl is more and more popular free, open-source software for econometric analysis, both for students and academics. Unlike most other statistical software it has easy to use GUI interface. Our software package is, therefore, a free and easy tool for Bayesian model averaging.[1]

The rest of the paper is organized as follows: Section 2 briefly outlines the Bayesian model averaging for linear regression models with MC^3 sampling algorithm and jointness measures. Section 3 provides an overview of gretl packages for **BMA**. Section 4 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 in linear regression models 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 (MC^3) sampling algorithm and jointness measures.

2.1 Bayesian estimation and model selection in Normal linear regression models

Consider the Normal linear regression models which differ in their explanatory variables[2]. Suppose that we have K potential explanatory variables, which means there are 2^K possible models and let M_r for $r = 1, \ldots, 2^K$ 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 $i = 1, .., N$. 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[3]. Thus, we can write our model as

$$y = \alpha \iota_N + X_r \beta_r + \epsilon \tag{1}$$

[1] A recent overview of **BMA** software in R is given in Amini and Parmeter (2011). Another useful informations about **BMA** software are available on website: `http://www2.research.att.com/~volinsky/bma.html`

[2] See Koop (2003) for further details.

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

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

$$\beta_r \mid h, M_r \sim N\left(0_{k_r}, h^{-1}\left[gX_r'X_r\right]^{-1}\right) \tag{2}$$

while we use noninformative prior for intercept and precision

$$p(\alpha) \propto 1, p(h) \propto \frac{1}{h} \tag{3}$$

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 four 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)

$$g = \frac{1}{N} \tag{4}$$

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

$$g = \frac{1}{K^2} \tag{5}$$

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

$$g = \begin{cases} \frac{1}{K^2} & \text{for } N \leq K^2; \\ \frac{1}{N} & \text{for } N > K^2 \end{cases} \tag{6}$$

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

$$g = \frac{1}{(\ln N)^3} \tag{7}$$

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

$$E(\beta_r \mid y, M_r) = \left[(1+g)\, X_r'X_r\right]^{-1} X_r'y \tag{8}$$

It is easy to see that if $g \approx 0$ the mean of the posterior distribution (8) equals to the OLS estimates. The posterior variance of β_r, conditional with respect to model M_r, is given by

$$Var\left(\beta_r \mid y, M_r\right) = \frac{N s_r^2}{N-2}\left[(1+g)\, X_r'X_r\right]^{-1} \tag{9}$$

where

$$s_r^2 = \frac{\frac{1}{1+g} y' P_{W_r} y + \frac{g}{1+g}\left(y - \bar{y}\iota_N\right)'\left(y - \bar{y}\iota_N\right)}{N} \tag{10}$$

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

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

$$p(y \mid M_r) \propto \left(\frac{g}{1+g}\right)^{\frac{k_r}{2}} \left[\frac{1}{1+g} y' P_{W_r} y + \frac{g}{1+g} \left(y - \bar{y}\iota_N\right)' \left(y - \bar{y}\iota_N\right)\right]^{-\frac{N-1}{2}} \tag{11}$$

In the Bayesian approach to comparing models, it is considered useful to employ probabilities to represent the degree of belief associated with 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 for a model M_l against model M_n is given by

$$\frac{p(M_l \mid y)}{p(M_n \mid y)} = \frac{p(M_l)}{p(M_m)} \frac{p(y \mid M_l)}{p(y \mid M_n)} \tag{12}$$

where $\frac{p(M_l)}{p(M_n)}$ is the prior odds ratio and $\frac{p(y|M_l)}{p(y|M_n)}$ is the Bayes factor. If the ratio (12) is larger than one, we can say that the data supports model M_l over model M_m. In our package, we use two popular model priors

- Binomial prior, i.e. $p(M_r) = \theta^{k_r}(1-\theta)^{K-k_r}$ for $r = 1, \ldots, 2^K$. Note that for $\theta = 0.5$ we have Uniform prior on the model space, i.e. $p(M_r) = 2^{-K}$
- Binomial-Beta prior i.e. (see Gelman, Carlin, Stern, and Rubin (1997))

$$p(\Xi = k_r) = \frac{\Gamma(K+1)}{\Gamma(k_r+1)\Gamma(K-k_r+1)} \cdot \frac{\Gamma(a+k_r)\Gamma(K+b-k_r)}{\Gamma(a+b+k_r)} \cdot \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}$$

  where Ξ denotes model size.

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

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

$$p(M_l \mid y) = \frac{p(M_l)p(y \mid M_l)}{\sum_r^{2^K} p(M_r)p(y \mid M_r)} \tag{13}$$

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

$$p(\beta \mid y) = \sum_{r=1}^{2^K} p(M_r \mid y)p(\beta_r \mid y, M_r) \tag{14}$$

Once the model posterior probabilities have been calculated, we can also easily evaluate the mean and variance of the posterior distribution of slope parameters[4]

$$E(\beta \mid y) = \sum_{r=1}^{2^K} p(M_r \mid y)E(\beta_r \mid y, M_r) \tag{15}$$

[4]See Leamer (1978).

and

$$\begin{aligned} Var(\beta \mid y) &= \sum_{r=1}^{2^K} p(M_r \mid y) Var(\beta_r \mid y, M_r) + \\ &+ \sum_{r=1}^{2^K} p(M_r \mid y) \left(E(\beta_r \mid y, M_r) - E(\beta \mid y)\right)^2 \end{aligned} \tag{16}$$

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 $p(i \mid y)$ (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.

2.2 MC^3 sampling algorithm

Our MC^3 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 $s = 1, \ldots, N$ to find the equilibrium distribution $p(M_r \mid y)$ 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

$$\alpha\left(M^{(s-1)}, M'\right) = min\left\{\frac{p(M')p(y \mid M')}{p(M^{(s-1)})p(y \mid M^{(s-1)})}, 1\right\} \tag{17}$$

In order to assess the stability and convergence of the chain, we look at the Pearson's correlation between the analytical and MC^3 posterior model probabilities. 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 MC^3 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 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 ex-post jointness measures of dependence between different sets of explanatory variables. According to Ley and Steel (2007), the logarithm of the jointness statistic has the form

$$J_{LS} = \ln\left[\frac{p(i \cap j \mid y)}{p(i \mid y) + p(j \mid y) - 2p(i \cap j \mid y)}\right] \tag{18}$$

where $p(i \cap j \mid y)$ represents the sum of the posterior probabilities of those models that contain both variables x_i and x_j, $p(i \mid y)$ and $p(j \mid y)$ 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

$$J_{DW} = \ln\left[\frac{p(i \cap j \mid y)p(\tilde{i} \cap \tilde{j} \mid y)}{p(i \cap \tilde{j} \mid y)p(\tilde{i} \cap j \mid y)}\right] \tag{19}$$

where $p(\tilde{i} \cap \tilde{j} \mid y)$ denotes the sum of the posterior probabilities of the regression models in which neither x_i and x_j are included, $p(i \cap \tilde{j} \mid y)$ 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 $p(\tilde{i} \cap j \mid y)$ 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 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 document the code as well as the use of the gretl package for Bayesian model averaging, together with accompanying jointness statistics. First, we will describe our code and the use of the graphical interface, then we will present how to use our **BMA** script. At the end we will present 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, 2013a, p. 1). Hansl's syntax is very similar to C language including passing pointers to functions. What is very useful for end user is that Hansl provides a nice mechanism for building GUI interfaces to functions/packages. Such packages consist on (at least) one "public" function and zero or more "private" helper functions (see Cottrell and Lucchetti (2013c)). This distinction gives programmers flexibility in writing the packages for gretl and allows to split the code into small pieces (functions) responsible for logically separated computations.

The **BMA** package consist on 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] "Prior" {"Binomial", "Binomial-Beta"},
  scalar av_model_size[0::] "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[1:4:1] "g-prior type" {"Benchmark prior",
    "Unit Information Prior (g-UIP)",
    "Risk Inflation Criterion (g-RIC)", "Hannan and Quinn HQC"},
  int do_joint[0:2:0] "Jointness analysis" {"None",
    "Ley-Steel Measure", "Doppelhofer-Weeks Measure"},
  int Nrep[1000000] "Total number of replications",
  int burn[0:99:10] "Percentage of burn-in draws",
  int verbosity[1:2:1] "Verbosity")
```

Text in quotation marks are labels for GUI interface shown in Figure 2.

The `big_list` is a gretl's object "named list" which is just set of $K+1$ variables (defined by names or dataset ID). What is very important: the first member of the `big_list` is treated as `y` variable and 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 prior type (the default is Binomial, see page 4).

The `scalar av_model_size[0::]` is a scalar with prior average model size. Note: if $av\_model\_size = \frac{K}{2}$ and prior is set to Binomial, we get Uniform prior on the model space.

The `scalar alpha[0:1:0.6]` is the significance level in OLS estimation. A independent variable enters the initial model if its p-value is less than the significance level (see 2.2). The default value is $\alpha = 0.6$, but setting $\alpha = 1$ results in model consisting on 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. See page 5.

The `int g_type[1:4:1]` indicates type of g-prior to be used (the default value is 1: "Benchmark prior"). See page 3.

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 page 5.

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: silent mode.

3.2.2 The main loop

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

1. Setting up the MC^3 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 (2013c) 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 formulas (4)-(7) 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 MC^3 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)`
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)`
4. `function void BMA_ols (const matrix *Y, matrix *X,`
   `const matrix *factors, matrix *XtY, matrix *XtXinv,`
   `scalar *yMy, matrix *bhat, matrix *bvar)`
5. `function matrix BMA_model_structure (const matrix *var_numbers,`
   `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_l` – 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_l`.

Next, the X_new matrix is taken by the function BMA_matrix_precompute for linear algebra computations necessary to compute formulas (8)-(11). Now we run the following code snippet to compute formula (11)

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

Next, we run the function BMA_ols to compute formulas (8)-(10). 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 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. Taking 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))
  ...
endif
```

3. Construction/modification of 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. 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. Jointness analysis (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 by the gretl's build-in function `randint()` which uses the SIMD-oriented Fast Mersenne Twister (SFMT) RNG (see Cottrell and Lucchetti (2013b); Yalta and Schreiber (2012))[5]. If the dawned 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 priors described on page 4.

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 essential **BMA** computations formulated in (15)-(16), that is the mean and variance of the posterior distribution of slope parameters, as well as average model size and posterior inclusion probability (PIP).

Finally at 5., 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 MC^3 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: PIP, Mean, Std.Dev., Cond.Mean, Cond.Std.Dev, see Outputs on page 13 for details. The next K columns appears only if any of jointness analysis was selected and contain values of one of the bi-jointness measures: Ley-Steel or Doppelhofer-Weeks.

$$
\begin{array}{cccccc|ccccc}
 & \text{PIP} & \text{Mean} & \text{Std.Dev.} & \text{Cond.Mean} & \text{Cond.Std.Dev} & x_1 & x_2 & x_3 & \dots & x_K \\
x_1 & & & & & & & & & & \\
x_2 & & & & & & & & & & \\
x_3 & & & & & & & \multicolumn{3}{c}{J_{LS} \text{ or } J_{DW}} & \\
\vdots & & & & & & & & & & \\
x_K & & & & & & & & & &
\end{array}
$$

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

[5]The code of **BMA** package contains private function `ran2` which implements so-called "ran2" 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, i.e. 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.

3.3 Usage of the BMA package

3.3.1 The GUI way

Once you start the 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.

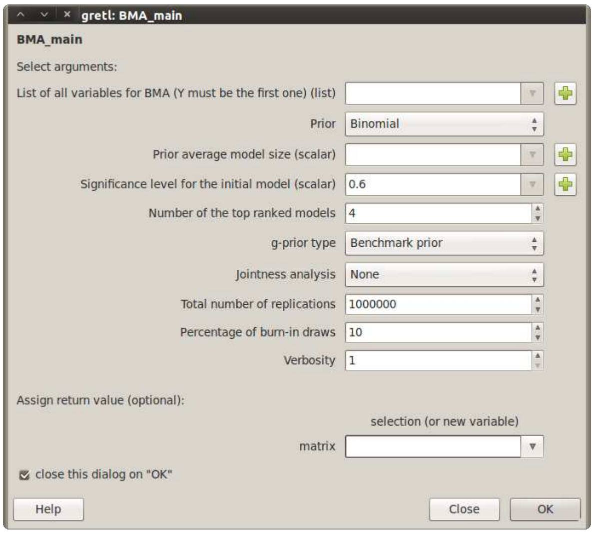

Figure 2: Main window for **BMA**

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.

- **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 $E(\Xi)$. Note that for the Binomial model prior and $E(\Xi) = 0.5K$ one can define Uniform prior on the model space.

- **Significance level for the initial model** - Defines the significance level which was used to build the initial model. A explanatory variable enters the initial model if its p-value is less than the significance level. If significance level equals 1 the initial model will be randomly chosen (with equal probability) from all available models. Note that if all available explanatory variable enters the initial model you will get the following gretl's error messages "No independent variables were omitted".

- **Number of the top ranked models** - Specifies the number of the best models for which 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 (g-UIP), Risk Inflation Criterion (g-RIC) and Hannan and Quinn prior (g-HQP).
- **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).
- **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 analysis is provided (initial model, speed of convergence, estimation results for top ranked models).
- **matrix** - You can save the output under a specified name to the current session.

3.3.2 Script

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

```
open greene9_1.gdt
include BMA.gfn

list green = dataset

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

The above example consists of three blocks. The first block is just opening of the so called greene9_1 dataset, which is bundled in every standard gretl installation. This dataset contains cross-sectional data on manufacturing of transportation equipment presented as Table 9.1 in Greene (1999).

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 of **BMA** analysis: Binomial prior, prior average model size set to 1.5, significance level for the initial model set to 0.6, 4 top ranked models, Benchmark g-prior, without jointness analysis, 100000 replications with 10% burn in draws and basic output (verbosity set to 1).

Suppose we want to set g-prior to Risk Inflation Criterion (g-RIC), do jointness analysis with Ley-Steel Measure and print additional information in results (verbosity set to 2). The code would be as follows:

```
BMA_main(arg1, 1, 1.5, 0.6, 4, 3, 1, 1000000, 10, 2)
```

Finally, if we want to save results of the above **BMA** analysis in the matrix `results_mat`, the code should be as follows:

```
results_mat = BMA_main(arg1, 1, 1.5, 0.6, 4, 3, 1, 1000000, 10, 2)
```

3.3.3 Outputs

If you select the appropriate entries in the GUI **BMA** window our package returns the posterior inclusion probabilities (PIP), 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).
Let us consider the data used in Fernández *et al.* (2001b) (FLS hereafter). These data comprises information about 72 countries and 41 potential growth determinants for the period 1960 to 1992[6]. 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	0.999327	-1.620005	0.306288	-1.621096	0.303491
Confuncious	0.991093	5.673435	1.431048	5.724422	1.332078
Life_Exp	0.942527	0.084238	0.033110	0.089374	0.026533
Equip_Inv	0.914864	15.124520	6.762346	16.531993	5.168802
SubSahara	0.761057	-1.207077	0.844624	-1.586053	0.579907
Muslim	0.643448	0.886294	0.782963	1.377413	0.525597
Rule_of_Law	0.541924	0.794867	0.827829	1.466751	0.528284
Yrs_Open	0.473959	0.650766	0.770991	1.373043	0.512306
...					

Posterior probability of models:
Model 1: 0.007459
Model 2: 0.005290
Model 3: 0.002812
Model 4: 0.002947
Model 5: 0.002500
...
Total probability of the models in ranking (numerical): 0.030975
Correlation coefficient between the analytical
and numerical probabilities of the above models: 0.997971

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. The value 2 forces **BMA** function to print out all the details of estimation. You can print out the above-mentioned results and additional the following informations: 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 it is, not you will get: posterior joint probability of explanatory variables, jointness statistic (18) or (19) 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	0.987472	0.930539	0.924864	...
Confuncious	0	0	0.922697	0.915026	...
Life_Exp	0	0	0	0.858191	...
Equip Inv	0	0	0	0	...
...	...	...	...	...	...

[6]The dataset is publicly available on the Journal of Applied Econometrics online data archive.

```
Jointness statistics (Ley-Steel Measure):
              GDPsh560  Confuncious  Life_Exp  Equip Inv  ...
GDPsh560             0     4.367153  2.606921   2.510349  ...
Confuncious          0            0  2.527561   2.386118  ...
Life_Exp             0            0         0   1.812569  ...
Equip Inv            0            0         0          0  ...
...                ...          ...       ...        ...  ...
Strong substitutes:
Rev_Coup, Area           -4.453350
Publ_Edu_pct, Area       -4.432202
...
Significant substitutes:
Mining, Hindu            -1.999312
SubSahara, Catholic      -1.991931
...
Significant complements:
Life_Exp, Equip_Inv       1.812569
GDPsh560, SubSahara       1.033650
...
Strong complements:
GDPsh560, Confuncious     4.367153
GDPsh560, Life_Exp        2.606921
...
```

4 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: 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 1 present the estimation results. This table also reports the posterior means and standard errors of regressors calculated from the **BMS** Zeugner (2012) package and the results published in Ley and Steel (2007). These benchmarking results allow us to compare and analyse the performance of our package.

As is apparent from Table 1, the gretl package is reasonably successful at matching the reported results in Ley and Steel (2007). All the PIPs and estimated posterior means or standard deviations are reasonably close for 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 **BMS** package. The only minor differences in posterior results are found between them and the results published in Ley and Steel (2007).

[6] The reported chain took about 120 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 1.9.11.cvs compliled by GCC 4.7.2. The seed for RNG was set to 1000000.

Table 1: Performance of gretl **BMA** package for FLS data

	BMS			gretl			Ley and Steel (2007)		
Regressors	PIP	Mean	SD	PIP	Mean	SD	PIP	Mean	SD
log GDP in 1960	1.00	-1.4247	0.277	1.00	-1.4275	0.275	1.00	-1.4180	0.269
Fraction Confucian	0.99	0.4936	0.128	0.99	0.4942	0.126	1.00	0.4900	0.117
Life expectancy	0.93	0.9647	0.391	0.94	0.9672	0.386	0.95	0.9574	0.371
Equipment investment	0.92	0.5497	0.236	0.92	0.5499	0.234	0.94	0.5575	0.222
Sub-Saharan dummy	0.74	-0.4748	0.347	0.74	-0.4741	0.346	0.76	-0.4772	0.334
Fraction Muslim	0.64	0.2572	0.229	0.65	0.2607	0.229	0.66	0.2565	0.219
Rule of law	0.50	0.2469	0.279	0.50	0.2463	0.279	0.52	0.2594	0.280
Number of years open economy	0.50	0.2540	0.283	0.51	0.2558	0.282	0.50	0.2556	0.283
Degree of capitalism	0.46	0.1533	0.183	0.46	0.1518	0.182	0.47	0.1577	0.184
Fraction Protestant	0.45	-0.1434	0.178	0.45	-0.1429	0.178	0.46	-0.1441	0.176
Fraction GDP in mining	0.47	0.1493	0.181	0.44	0.1366	0.174	0.44	0.1384	0.176
Non-Equipment investment	0.43	0.1361	0.174	0.47	0.1489	0.181	0.43	0.1346	0.173
Latin American dummy	0.21	-0.0815	0.188	0.22	-0.0822	0.190	0.19	-0.0729	0.175
Primary school enrollment. 1960	0.20	0.1026	0.233	0.21	0.1038	0.234	0.18	0.0941	0.224
Fraction Buddhist	0.21	0.0497	0.112	0.20	0.0481	0.110	0.17	0.0394	0.100
Black market premium	0.18	-0.0405	0.098	0.18	-0.0415	0.099	0.16	-0.0355	0.092
Fraction Catholic	0.13	-0.0111	0.124	0.13	-0.0098	0.125	0.11	-0.0123	0.113
Civil liberties	0.13	-0.0487	0.151	0.13	-0.0496	0.151	0.10	-0.0388	0.134
Fraction Hindu	0.13	-0.0356	0.122	0.13	-0.0360	0.124	0.10	-0.0247	0.094
Primary exports. 1970	0.10	-0.0286	0.104	0.10	-0.0287	0.104	0.07	-0.0209	0.089
Political rights	0.10	-0.0282	0.108	0.10	-0.0283	0.107	0.07	-0.0205	0.090
Exchange rate distortions	0.08	-0.0164	0.070	0.08	-0.0164	0.069	0.06	-0.0134	0.063
Age	0.08	-0.0147	0.058	0.09	-0.0153	0.060	0.06	-0.0098	0.048
War dummy	0.08	-0.0146	0.062	0.08	-0.0146	0.062	0.05	-0.0097	0.051
Fraction of Pop. Speaking English	0.07	-0.0107	0.048	0.07	-0.0106	0.048	0.05	-0.0071	0.039
Fraction speaking foreign language	0.07	0.0120	0.060	0.07	0.0120	0.059	0.05	0.0089	0.051
Size labor force	0.08	0.0194	0.101	0.08	0.0197	0.103	0.05	0.0099	0.069
Ethnolinguistic fractionalization	0.06	0.0106	0.058	0.06	0.0096	0.066	0.04	0.0059	0.042
Spanish Colony dummy	0.05	0.0087	0.063	0.06	0.0106	0.058	0.03	0.0058	0.050
S.D. of black-market premium	0.05	-0.0062	0.038	0.05	-0.0062	0.038	0.03	-0.0041	0.031
French Colony dummy	0.05	0.0067	0.040	0.05	0.0070	0.041	0.03	0.0042	0.031
Absolute latitude	0.04	0.0013	0.055	0.04	0.0011	0.054	0.02	0.0005	0.040
Ratio workers to population	0.04	-0.0057	0.044	0.04	-0.0050	0.041	0.02	-0.0030	0.031
Higher education enrollment	0.05	-0.0085	0.058	0.05	-0.0089	0.060	0.02	-0.0041	0.039
Population growth	0.04	0.0053	0.048	0.04	0.0054	0.048	0.02	0.0032	0.035
British colony dummy	0.04	-0.0033	0.031	0.04	-0.0028	0.031	0.02	-0.0019	0.022
Outward orientation	0.04	-0.0035	0.029	0.04	-0.0037	0.030	0.02	-0.0018	0.021
Fraction Jewish	0.03	-0.0024	0.027	0.04	-0.0024	0.027	0.02	-0.0014	0.020
Revolutions and coups	0.03	0.0000	0.023	0.03	-0.0001	0.023	0.02	0.0000	0.017
Public education share	0.03	0.0008	0.024	0.03	0.0008	0.023	0.02	0.0004	0.017
Area (scale effect)	0.03	-0.0009	0.021	0.03	-0.0009	0.021	0.02	-0.0006	0.014

Note: The dependent variable is the growth rate from 1960-1996 across 72 countries. All the regressors have been standardized to have zero mean and unit variance.

5 Conclusions

This paper has outlined the new software package that implements Bayesian model averaging analysis 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 free, open-source software for econometric analysis with 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 a **BMA** analysis, as well as the outputs that are returned.

References

Amini S, Parmeter CF (2011). "Bayesian Model Averaging in R." *Journal of Economic and Social Measurement*, **36**(4), 253–287. 2

Cottrell A, Lucchetti R (2013a). *A Guide to Hansl*. 6

Cottrell A, Lucchetti R (2013b). *gretl Command Reference – Gnu Regression, Econometrics* *and Time-Series Library*. URL http://ricardo.ecn.wfu.edu/pub/gretl/manual/PDF/gretl-ref.pdf. 10

Cottrell A, Lucchetti R (2013c). *gretl User's Guide – Gnu Regression, Econometrics and Time-Series Library*. URL http://ricardo.ecn.wfu.edu/pub/gretl/manual/PDF/gretl-guide.pdf. 6, 8

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, 12

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

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, 10, 13, 14

Foster DP, George EI (1994). "The Risk Inflation Criterion for Multiple Regression." *The Annals of Statistics*, **22**(4), 1947–1975. 3

Gelman A, Carlin J, Stern H, Rubin D (1997). *Bayesian Data Analysis*. Chapman and Hall, London. 4

Greene WH (1999). *Econometric Analysis*. 4 edition. Prentice Hall, Upper Saddle River, NJ. 12

Hoeting JA, Madigan D, Raftery AE, Volinsky CT (1999). "Bayesian Model Averaging: A Tutorial." *Statistical Science*, **14**(4), 382–417. 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. http://www.tandfonline.com/doi/pdf/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 and Sons. 2, 3, 5

Leamer E (1978). *Specification Searches*. John Wiley and Sons, New York. 4, 5

Ley E, Steel MF (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, 5, 6, 10, 12, 14, 15

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. URL `http://ideas.repec.org/a/jae/japmet/v24y2009i4p651-674.html`. 4

Madigan D, York J, Allard D (1995). "Bayesian Graphical Models for Discrete Data." *International Statistical Review*, **63**(2), 215–232. 2, 5

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. `http://www.tandfonline.com/doi/pdf/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/cmf/wpaper/wp2010_1008.html`. 3

Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1988). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA. ISBN 0-521-35465-X. 10

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. `http://www.tandfonline.com/doi/pdf/10.1080/01621459.1997.10473615`, URL `http://www.tandfonline.com/doi/abs/10.1080/01621459.1997.10473615`. 1

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`. 10

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. 3

Zeugner S (2012). *Bayesian Model Averaging with* **BMS**. URL `http://cran.r-project.org/web/packages/BMS/vignettes/bms.pdf`. 14