Chapter 20: Bayesian model averaging[1]

The bayesian model averaging is a bayesian method developed in the 90's to take into account the uncertainty associated to the presence of a variable in the true model when the econometrician has at her disposal a set of variables potentially important to explain an endogenous variable. Instead of estimating a lone model, the method estimates many models built from various subset of the initial set of variables and attribute them posterior probabilities. Part 1 presents the theoretical background, part 2 how to estimate in practice by this method and part 3 presents the precise description of the functions and their specifications

1. Some theoretical background

Be $y=(y_{1,}\dots,y_T)$ the quantity of interest $X=\{X_{1,}\dots,X_n\}$ and a set of *n* potential regressors, where $X_i=(X_{i,1},\dots,X_{i,T})$. We consider a representation in the linear framework, where $X^*=\{X_1^*,\dots,X_q^*\}$ is a $(q,1)$ subset of $\{X_{1,}\dots,X_n\}$ such as

$$y_t=X_t^*\beta+\varepsilon_t$$

Let us define a given model M_i and M the finite set of model M_γ. In theory M includes 2^k elements. Here, $\gamma\in\{0,1\}^n$ is a summarized representation of the regressors entering the model. For all $j=1,\dots,n$, $\gamma_j=1$ (resp. $\gamma_j=0$) means that X_j does belong (resp. not) to the model. Said differently, M_γ is a short representation of the following model:

$$y/\beta_\gamma,\sigma^2\sim(X_\gamma\beta_\gamma,\sigma^2 I) \qquad (1)$$

with $\theta=(\beta_\gamma,\sigma^2)$ the unknown vector of parameters.

Given the set $M=\{M_{1,}\dots,M_{2^n}\}$ of possible models, we can associate $p(M_\gamma)$ and $p(\theta_\gamma/M_\gamma)$ the prior probabilities for a given model and distributions of the related parameters. Let $L(\theta_\gamma/M_\gamma)$ be the likelihood associated to the model (1) then, the posterior probabilities of the model are given by:

$$p(M_\gamma/y)=\frac{m(y/M_{\gamma_0})\,p(M_{\gamma_0})}{\sum_\gamma m(y/M_\gamma)\,p(M_\gamma)}=\left[\sum_\gamma\frac{p(M_\gamma)m(y/M_\gamma)}{p(M_{\gamma_0})m(y/M_{\gamma_0})}\right]^{-1}=\left[\sum_\gamma\frac{p(M_\gamma)}{p(M_{\gamma_0})}B_{\gamma_0,\gamma}\right]^{-1}$$

where $m(y/M_\gamma)$ is the marginal likelihood, conditional to the model M_γ,

$$m(y/M_\gamma)=\int_\theta L(y/\theta_\gamma,M_\gamma)\,p(y/\theta_\gamma,M_\gamma)\,d\theta_\gamma$$

and $B_{\gamma_0,\gamma}$ is the Bayes factor. It remains to specify the prior model probabilities $p(M_\gamma)$. As in Fernandez *et alii* (2001) we use an uniform prior on model space: $p(M_\gamma)=2^{-k}$.

The posterior probability that the variable *i* belongs to the "true" model is:

[1]This chapter owes for the most part to Emmanuel Michaux's work. Thanks Emmanuel!

$$p(x_i/y)=\sum_{\gamma} \boldsymbol{I}(x_j \in M_{\gamma})\, p(M_{\gamma}/y)$$

where $\boldsymbol{I}(x_j \in M_{\gamma})$ is one if x_j is included in model γ or zero otherwise. If we come back to our initial objective, the minimum square error forecast of our quantity of interest (here y_t taking into account bayesian uncertainty is given by the following average[2]:

$$E(y_{t+h}/y)=\sum_{\gamma} E(y_{t+h}/y, M_{\gamma})\, p(M_{\gamma}/y)$$

In Grocer, we stick to the the Fernandez *et alii* (2001)[3] approach using g-priors and considering only models with a constant. We use non-informative priors for the ordinary regression parameters through g-prior:

$$\beta_{\gamma}/\sigma^{2,} M_{\gamma} \sim N(0, \sigma^2 (g X_{\gamma}' X_{\gamma})^{-1})$$

with *g* accounting for the share of information[4], which is "a priori"' available in the studied sample[5]. As in Raftery *et alii* (1997)[6], Fernandez *et alii* (2001) consider the same prior for the constant vector of the matrix β_{γ}. The mean is defined as follows: $\mu=(m_1,0,\ldots,0)$, with a non informative law $p(\mu)\propto 1$. We follow the common g-prior specifications: $g_{\gamma}=g=q^{-2}$ as $n \leq q^2$.

Contrary to Raftery *et alii* (1997), where *prior* laws are conjugate and *hyperparameters* have considerable influence on the *posterior* distributions and the Bayes factor, *prior distribution* of σ^2 is non informative. We may derive in this framework the posterior distribution:

$$p(y/\mu_{\gamma}, g_{\gamma}, M_{\gamma}, \gamma) \propto \left(\frac{g_{\gamma}}{1+g_{\gamma}}\right)^{\frac{g_{\gamma}}{2}} \left[\frac{1}{1+g_{\gamma}} y' M_X y + \frac{g_{\gamma}}{1+g_{\gamma}} (y-\bar{y}\mathbf{1})'(y-\bar{y}\mathbf{1})\right]^{\frac{n-1}{2}}$$

where $M_X = I - X(X'X)^{-1}X'$.

We propose two algorithms to approximate the distribution of M_{γ} : the MC[3] as proposed by Madigan and York (1995)[7] and Green (1995)[8] MCMC algorithm with reversible jump. The MC[3] works as follow:

- define an arbitrary initial model M_{γ_0} ;
- define the neighbourhood (nbd) of a model as the set of models with either one variable more or one variable fewer than M_{γ} ;

---

[2] The coefficient β^j_{BMA} related to the regressor *j* may be extracted in the same way: $\beta^j_{BMA}=E(\beta^j_{\gamma}/y)=\sum_{\gamma} \hat{\beta}^j_{\gamma}\, p(M_{\gamma}/y)$.

[3] Fernandez, C., E. Ley and F. J. Steel} (2001), *Benchmark Priors for Bayesian Model Averaging*, Journal of Econometrics, 100, 381-427.

[4] In this formulation of the prior X_{γ} is supposed to be strictly exogenenous and residuals are assumed to be iid. g-pior does not allow for lagged endogenous variable, so is not compatible with GLS like variance-covariance matrix. This can be seen as a straightforward limit of the method when used for forcasting time series.

[5] For instance, if g=0.1$, 10% of information of the sample is taken into account.

[6] Raftery, A. E. , D. Madigan and J. A. Hoeting (1997), *Bayesian Model Averaging for Linear Regression Models*, Journal of the American Statistical Association, 92, 179-191.

[7] Madigan, D. and York, J. (1995), *Bayesian Graphical Models for Discrete Data*, Inter-national Statistical Review, 63, 215-232.

[8] Green, J. P. (1995), *Reversible Jump Markov Chain Monte Carlo Computation Analysis and Bayesian Model Determination,* Biometrika, Vol. 83, n°4, 711-713.

- define a transition matrix q by setting $q(M_\gamma \to M_{\gamma'}) = q(M_\gamma, M_{\gamma'}) = 0$ for all $M_\gamma \notin nbd(M_\gamma)$ and $q(M_\gamma \to M_{\gamma'})$ non zero for all $M_\gamma \in nbd(M_\gamma)$;
- If the chain is currently in sate M_γ proceed by drawing $M_{\gamma'}$ from $q(M_\gamma \to M_{\gamma'})$, $M_{\gamma'}$ is accepted with probability:
-

$$\alpha = min\left(1, \frac{p(M_{\gamma'}/y)\, p(M_{\gamma'})\, q(M_{\gamma'}, M_\gamma)}{p(M_\gamma/y)\, p(M_\gamma)\, q(M_\gamma, M_{\gamma'})}\right) = min\left(1, \frac{p(M_{\gamma'})}{p(M_\gamma)}\right)$$

(this simplification holds because we use equal *a priori* probability for each model).

- set $M_\gamma = M_{\gamma'}$ if the move is accepted;

The MCM with reversible jump can be sketched[9] as follows:

- define an arbitrary initial model M_{γ_0} ;
- propose a jump from $M_{\gamma'}$ to $M_{\gamma'}$ with probability $j(M_\gamma, M_{\gamma'})$;
- generate a vector i u_γ (which can have a different dimension than θ_{M_γ}) from a specified proposal density $q(u_\gamma / \theta_{M_\gamma}, M_\gamma, M_{\gamma'})$;
- set $(\theta_{M_{\gamma'}}, u_{\gamma'}) = g_{M_\gamma, M_{\gamma'}}(\theta_{M_\gamma}, u_\gamma)$ where $g_{M_\gamma, M_{\gamma'}}(.,.)$ is a specified invertible function. Hence $dim(\theta_{M_\gamma}) + dim(u_\gamma) = dim(\theta_{M_{\gamma'}})$.
- Accept the proposal move with probability

$$\alpha = min\left(1, \frac{p(M_{\gamma'}/y)\, p(M_{\gamma'})\, j(M_{\gamma'}, M_\gamma)}{p(M_\gamma/y)\, p(M_\gamma)\, j(M_\gamma, M_{\gamma'})}\right)$$

Two types of model modifications are considered:

- Draw a variable at random and drop if it is in the model or add it to the model. The step is attempted with a probability p_A ;
- Swap randomly a selected variable in the model and a randomly selected variable outside the model. This step is attempted with a probability $1 - p_A$;

For this two moves $j(M_\gamma, M_{\gamma'}) = j(M_{\gamma'}, M_\gamma)$ so the acceptance ratio simplifies to:

$$\alpha = min\left(1, \frac{p(M_{\gamma'}/y)\, p(M_{\gamma'})\, j(M_{\gamma'}, M_\gamma)}{p(M_\gamma/y)\, p(M_\gamma)\, j(M_\gamma, M_{\gamma'})}\right) = min\left(1, \frac{p(M_{\gamma'})}{p(M_\gamma)}\right)$$

2. Performing a bayesian model averaging estimation

Performing a bayesian averaging estimation involves a call to the function bma_g with the following arguments:

- first the endogenous variable, in the form of a vector, a time series, eventually between quotes;
- second, the number of draws to carry out.

[9] see Eklund, J. & S. Karlsson (2005), *Forecast Combination and Model Averaging Using Predictive Measures,* Sveriges Riksbank Working Paper Series, n°191.

Then you have to enter in any order the following arguments:

- a time series, a real vector, a real matrix, a string vector or a list of such objects, all pertaining to one or several exogenous variables;
- 'nvmax=xx' where xx is the maximum number of exogenous variables allowed in each model (optional: if nvmax is not given, then no limit is imposed)
- 'burnin=xx' number of burn-in MCMC simulation, that is the number of simulations that are discarded at the start of the process in order to prevent the starting conditions to distort the results (optional: if burnin is not given, then all simulations are kept);
- 'g =xx' where xx is the value of g-prior (optional: default = 1/max(n,k^2))
- 'mcmc = "mc3" or "jump"' type of MCMC algorithm (MC3 or reversible jump) must be between quotes;
- the string 'noprint' if the user doesn't want to print the results of the regression

As an example, take the modelisation of the crime rate in the US states that have been used extensively in the literature and whose data are gathered in the database crime.dat in your library SCI/macros/grocer/db.

```
-->load('SCI/macros/grocer/db/crime.dat')
```

Estimating this model with the MC3 algorithm, 30000 draws of which 10000 are discarded is done as follows:

```
-->rbma=         bma_g('log(crime)',30000,'log(m)','so','log(ed)','log(po1)','log(po2)',
'log(lf)','log(mf)','log(pop)','log(nw)','log(u1)','log(u2)','log(gdp)','log(ineq)',
'log(prob)', 'log(time)','burnin=10000','mcmc=''mc3''');
```

With the following result:

```
MCMC sampling, be patient ...



        Bayesian model averaging estimates

dependent variable: log(crime)
MCMC            = mc3
# of obs        = 47
# of var        = 16
g-prior         = 0.0044444
# of draws      = 30000
# of burn-in    = 10000
# of models     = 2953
max # of var.   = 15
time (sec.)     = 22

standard error of the regression: 0.2124261
sum of squared residuals: 1.3988704

        Model averaging information

models    model 1   model 2  model 3   model 4   model 5   model 6   model 7
log(m)    1         1        1         1         1         1         0
so        0         0        0         0         0         0         0
log(ed)   1         1        1         1         1         1         1
log(po1)  1         1        0         1         1         0         1
log(po2)  0         0        1         0         0         1         0
```

log(lf)	0	0	0	0	0	0	0
log(mf)	0	0	0	0	0	0	0
log(pop)	0	0	0	0	0	0	1
log(nw)	0	1	0	0	0	1	1
log(u1)	0	0	0	0	0	0	0
log(u2)	1	1	1	1	0	1	0
log(gdp)	0	0	0	0	0	0	0
log(ineq)	1	1	1	1	1	1	1
log(prob)	1	1	1	0	1	1	1
log(time)	0	0	0	0	0	0	0
prob. (%)	3.6645623	3.524985	2.3658671	2.3424289	2.3169626	2.2316282	2.0769821
visits	90	65	47	121	142	45	25

models	model 8	model 9	model 10	model 11	model 12	model 13	model 14
log(m)	1	1	1	1	1	0	1
so	0	0	0	0	0	0	0
log(ed)	1	1	1	1	1	1	1
log(po1)	1	1	1	1	0	0	0
log(po2)	0	0	0	0	1	1	1
log(lf)	0	0	0	0	0	0	0
log(mf)	0	0	0	0	0	0	0
log(pop)	0	0	0	0	0	1	0
log(nw)	1	1	0	1	0	1	1
log(u1)	0	0	0	0	0	0	0
log(u2)	0	1	0	0	0	0	0
log(gdp)	0	0	0	0	0	0	0
log(ineq)	1	1	1	1	1	1	1
log(prob)	1	1	0	1	1	1	1
log(time)	0	1	0	1	0	0	0
prob. (%)	2.043623	1.905743	1.8068672	1.6623433	1.3208134	1.2193039	1.1718728
visits	90	44	201	49	56	17	50

models	model 15	model 16	model 17
log(m)	1	0	1
so	0	0	0
log(ed)	1	1	1
log(po1)	1	1	1
log(po2)	0	0	0
log(lf)	0	0	0
log(mf)	0	0	0
log(pop)	1	0	1
log(nw)	1	1	1
log(u1)	0	0	0
log(u2)	0	0	1
log(gdp)	0	0	0
log(ineq)	1	1	1
log(prob)	1	1	1
log(time)	0	0	0
prob. (%)	1.1695046	1.0447255	1.0229731
visits	34	74	45

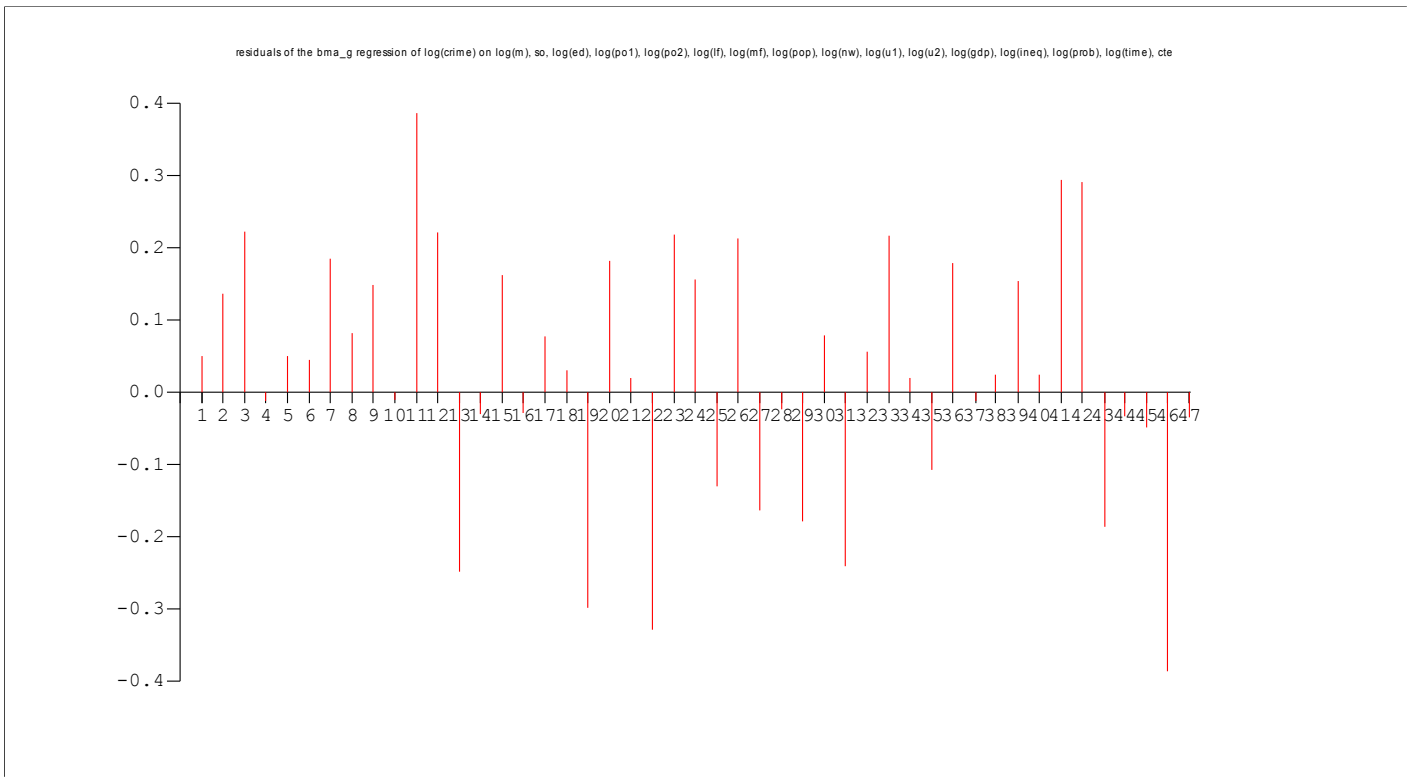
Posterior Estimates

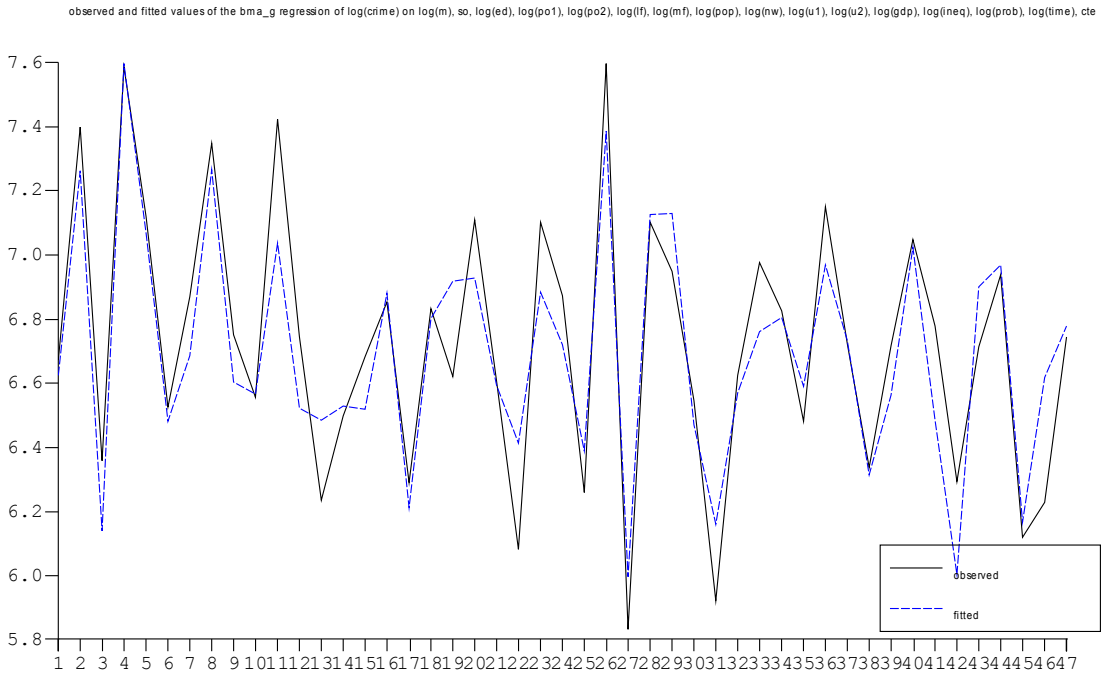
variable	coeff.	prob. (%)
log(m)	1.0690828	76.235402
so	0.0215394	13.581143
log(ed)	1.8147624	95.388281
log(po1)	0.6933074	66.506064
log(po2)	0.3543414	37.405194
log(lf)	0.0192253	7.1828681
log(mf)	0.0685876	8.3501908

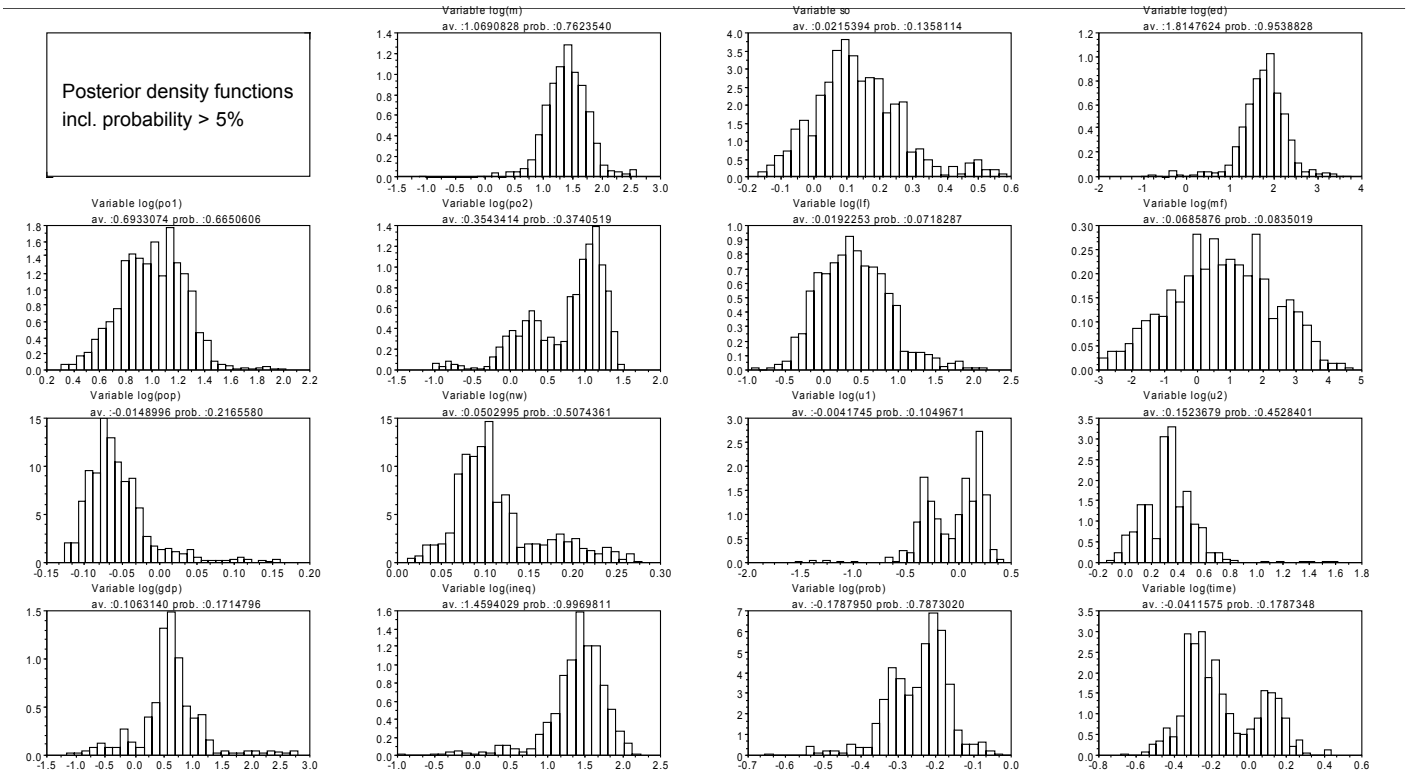
```
log(pop)      -0.0148996 21.655796
log(nw)       0.0502995  50.74361
log(u1)       -0.0041745 10.49671
log(u2)       0.1523679  45.284013
log(gdp)      0.1063140  17.14796
log(ineq)     1.4594029  99.698106
log(prob)     -0.1787950 78.730202
log(time)     -0.0411575 17.873478

                         *
                    *     *
```

Note that the visited models are provided in columns: first exogenous variables are accorded 0 values if they do not belong to the model and 1 if they belong to; then the posterior probability of the model and lastly the # of times the model has been visited. The results are provided by the function prtbma_g.

Three graphs are provided. The first 2 graphs are the traditional graphs, one of the residuals, the other one of the observed and fitted values, but applied in this case to the model where each coefficient is averaged over all simulations (results provided by function pltuniv). The third one provides the posterior probability distribution of each exogenous variable (result provided by function pltdensbma_g).




observed and fitted values of the bma_g regression of log(crime) on log(m), so, log(ed), log(po1), log(po2), log(lf), log(mf), log(pop), log(nw), log(u1), log(u2), log(gdp), log(ineq), log(prob), log(time), cte
observed
fitted


Posterior density functions
incl. probability > 5%
Variable log(m)
av. :1.0690828 prob. :0.7623540
Variable so
av. :0.0215394 prob. :0.1358114
Variable log(ed)
av. :1.8147624 prob. :0.9538828
Variable log(po1)
av. :0.6933074 prob. :0.6650606
Variable log(po2)
av. :0.3543414 prob. :0.3740519
Variable log(lf)
av. :0.0192253 prob. :0.0718287
Variable log(mf)
av. :0.0685876 prob. :0.0835019
Variable log(pop)
av. :-0.0148996 prob. :0.2165580
Variable log(nw)
av. :0.0502995 prob. :0.5074361
Variable log(u1)
av. :-0.0041745 prob. :0.1049671
Variable log(u2)
av. :0.1523679 prob. :0.4528401
Variable log(gdp)
av. :0.1063140 prob. :0.1714796
Variable log(ineq)
av. :1.4594029 prob. :0.9969811
Variable log(prob)
av. :-0.1787950 prob. :0.7873020
Variable log(time)
av. :-0.0411575 prob. :0.1787348

3. The bma estimation functions and their specifications.

bma_g_____Bayesian model averaging estimation

CALLING SEQUENCE
rbma_g = bma_g(namey0,ndraw,arg1,...,argn)

PARAMETERS

INPUT:
* namey0 = dependent variable vector
* ndraw = # of draws to carry out
* arg1,..., argn = a string which can be
  - a time series
  - a real (nxk) vector
  - a string equal to the name of a time series or a (nxp) real vector between quotes
  - 'burnin=xx' : # of burn-in MCMC simulation
  - 'g =XX' : value of g-prior (default = 1/max(n,k^2))
  - 'mcmc = "mc3" or "jump"' : type of MCMC algorithm (MC3 or reversible jump) must be in quote (default: MC3)
  - 'nvmax = xx' : max # of variables allowed in each models
  - the string 'noprint' if the user doesn't want to print the results of the regression

---

OUTPUT:
a tlist result with:
- rbma_g('meth') = 'bma g-prior'
- rbma_g('nmod') = # of models visited during sampling
- rbma_g('beta') = bhat averaged over all models
- rbma_g('mprob') = posterior prob of each model
- rbma_g('vprob') = posterior prob of each variable
- rbma_g('model') = indicator variables for each model (nmod x k)
- rbma_g('yhat') = yhat averaged over all models
- rbma_g('resid') = residuals based on yhat averaged over models
- rbma_g('sige') = averaged over all models
- rbma_g('nobs') = nobs
- rbma_g('nvar') = # of exogenous
- rbma_g('y') = y data vector
- rbma_g('x') = y data vector
- rbma_g('visit') = visits to each model during sampling (nmod x 1)
- rbma_g('time') = time taken for MCMC sampling
- rbma_g('ndraw') = # of MCMC sampling draws
- rbma_g('burnin') = # of burn-in MCMC simulation
- rbma_g('gprior') = value of g-prior
- rbma_g('namey') = name of the y variable
- rbma_g('namex') = name of the x variables
- rbma_g('bounds') = if there is a time series in the regression, the bounds of the regression
- rbma_g('mcmc') = type of MCMC algorithm

DESCRIPTION
Provides Bayesian model averaging estimates of Fernandez, Levy and Steel. This is the high level version which allows various data types (time series, real vectors and matrices, eventually between quotes). Some options (such as the maximum number of variables that is allowed for any model, the g-prior or the algorithm) are not necessarily to be entered and take in that case default values.

Example
load('SCI/macros/grocer/db/crime.dat')
rbma= bma_g('log(crime)',30000,'log(m)','so','log(ed)','log(po1)','log(po2)','log(lf)','log(mf)','log(pop)', 'log(nw)','log(u1)','log(u2)','log(gdp)','log(ineq)','log(prob)','log(time)','burnin=10000','mcmc="mc3'");

Example taken from function bma_d. The bma method is applied to Rafferty crime data. The number of draws is set to 30000. The results obtained from the first 10000 are discarded. The algorithm used is the MCM one.

bma_g1_____Bayesian model averaging estimation

CALLING SEQUENCE
rbma_g = bma_g1(y,x,ndraw,burnin,mcmc,g,nvmax)

PARAMETERS

INPUT:
* y = dependent variable vector (nobs x 1)
* x = explanatory variables (nobs x k)
* ndraw = # of draws to carry out
* burnin = # of burn-in MCMC simulation
* g = value of g-prior (default = 1/max(nobs,k^2))
* nvmax = max number of variable allowed in each model
* mcmc = name of the MCMC algorithm (jump_g or mc3_g)

OUTPUT:
a tlist result with:
- rbma_g('meth') = 'bma g-prior'
- rbma_g('nmod') = # of models visited during sampling
- rbma_g('beta') = bhat averaged over all models
- rbma_g('mprob') = posterior prob of each model
- rbma_g('vprob') = posterior prob of each variable
- rbma_g('model') = indicator variables for each model (nmod x k)
- rbma_g('yhat') = yhat averaged over all models
- rbma_g('resid') = residuals based on yhat averaged over models
- rbma_g('sige') = averaged over all models
- rbma_g('nobs') = nobs
- rbma_g('nvar') = # of exogenous
- rbma_g('y') = y data vector
- rbma_g('x') = y data vector
- rbma_g('visit') = visits to each model during sampling (nmod x 1)
- rbma_g('time') = time taken for MCMC sampling
- rbma_g('ndraw') = # of MCMC sampling draws

- rbma_g('burnin') = # of burn-in MCMC simulation
- rbma_g('gprior') = value of g-prior

DESCRIPTION
Provides Bayesian model averaging estimates of Fernandez, Levy and Steel. This is the low level function that works only for an endogenous in vector form, exogenous variables in a matrix form and where all options must given.

Example
1)rbma_g = bma_g1(y,x,ndraw,burnin,tmcmc,g,nvmax)
2) load('SCI/macros/grocer/db/crime.dat');
[y,namey,x]=explouniv('log(crime)',['log(m)','so','log(ed)','log(po1)','log(po2)','log(lf)','log(mf)','log(pop)','log(nw)','log(u1)','log(u2)','log(gdp)','log(ineq)','log(prob)','log(time)'],[]);
r=bma_g1(y,x,30000,10000,mc3_g);

Example 1 is taken from function bma_g. Example 2 provides the same estimation as in bma_d, but using the function bma_g1 instead of bma_g. Function explouniv is used to recover the y and x matrices corresponding to the data. Note that the names of the variables are nevertheless lost in the operation and that the function prtbma_g and prtdensbma cannot be used to print the results.