Parallelized BMA: the `ParMA` package

R. Lucchetti, L. Pedini

Dipartimento di Scienze Economiche e Sociali (DiSES)
Univpm - Facoltà di Economia G.Fuà

Gretl Conference 2021

The package

Why a `ParMA` package?

Providing a tool for dealing with BMA in Generalized Linear Models (GLMs)

What `ParMA` does:

- Reversible Jump MCMC sampler (Green, 1995, 2003; Fouskakis et al., 2009; Lamnisos et al., 2009)
- Models: linear, binary (probit/logit/cloglog), count (poisson);
- Flexible prior choices;
- Parallelization using MPI;
- Diagnostic and graphical tools;

Public functions

- `bma_glm()` $\rightarrow$ the main function
- `bma_printout()` $\rightarrow$ for printing the results;
- `marginal_graph()` $\rightarrow$ marginal distribution plots;
- `mcmc_checks()` $\rightarrow$ diagnostic tests.

RJMCMC in brief

Traditional MH:

$$\beta_{i+1} \rightarrow \begin{cases} \beta^* & \text{with prob. } \alpha \\ \beta_i & \text{with prob. } 1-\alpha \end{cases}$$

β_i and β^* belong to the same space (e.g. $\Re^k$).

RJMCMC in brief

Traditional MH:

$$\beta_{i+1} \to \begin{cases} \beta^* & \text{with prob. } \alpha \\ \beta_i & \text{with prob. } 1-\alpha \end{cases}$$

β_i and β^* belong to the same space (e.g. $\Re^k$).

Classical BMA with MH:

$$M_{i+i} \to \begin{cases} M^* & \text{with prob. } \alpha \\ M_i & \text{with prob. } 1-\alpha \end{cases}$$

M_i and M^* belong to the same space (a k-dimensional lattice on $\{0,1\}$).

Reversible Jump MCMC put together β and M into θ:

$$\theta_i = \begin{bmatrix} M_i \\ \beta_i \end{bmatrix}$$

where M_i is a point in the lattice, and $\beta_i \in \Re^{k_i}$, where $k_i = \sum_j (M_i)_j$.

$$\theta_{i+1} \rightarrow \begin{cases} \theta^* & \text{with prob. } \alpha \\ \theta_i & \text{with prob. } 1-\alpha \end{cases}$$

θ_i and θ^* may not belong to the same space $\rightarrow$ an artificial variable u^* is introduced to match dimension!

In particular:

- $M^* \leftarrow q(M|M_i)$;
- $\beta^* \leftarrow$ a differentiable function $g(\beta_i)$;

▸ More

Parallelization

Parallelization is easy to be implemented in simple MC, but in MCMC?

In `ParMA` we exploit “vertical” parallel chains:

- if convergence is quick a long single chain may be splitted in several ones with time benefits;
- better exploration of the parametrical space.

To do so we exploit the MPI architecture in `gretl`.

Diagnostic checks

- Brooks and Gelman (1998) univariate and multivariate statistics $\rightarrow$ default when MPI on;
- Geweke (1992) convergence test $\rightarrow$ `mcmc_checks`;
- Heidelberger and Welch (1983) test $\rightarrow$ `mcmc_checks`;
- Effective Sample Size (Vats et al., 2019) $\rightarrow$ `mcmc_checks`;
- Convergence plot $\rightarrow$ `mcmc_checks`.

Poisson regression

Data from Cameron and Trivedi (2013) example on doctor visits.

Dependent variable: DVISIT

Regressors:

- SEX
- AGE
- AGESQ
- INCOME
- LEVYPLUS, dummy for private health insurance;
- FREEPOOR, dummy for government insurance (income);
- FREEREPA, dummy for government insurance (age);
- ILLNESS, number of illnesses;
- ACTDAYS, numner of reduced activity days;
- HSCORE, health questionnaire score;
- CHCOND1, dummy for chronic cond.;
- CHCOND2, dummy for chronic cond. and reduced mob.

Standard estimation result

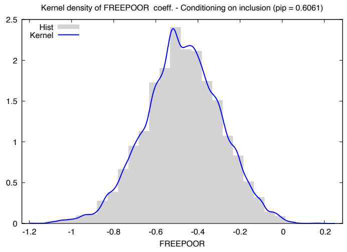
Model: Poisson, using observations 1–5190
Dependent variable: DVISITS
Standard errors based on Hessian

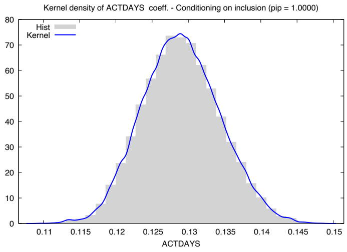
	Coefficient	Std. Error	z	p-value
const	−2.22385	0.189816	−11.72	0.0000
SEX	0.156882	0.0561368	2.795	0.0052
AGE	1.05630	1.00078	1.055	0.2912
AGESQ	−0.848704	1.07778	−0.7875	0.4310
INCOME	−0.205321	0.0883793	−2.323	0.0202
LEVYPLUS	0.123185	0.0716398	1.720	0.0855
FREEPOOR	−0.440061	0.179811	−2.447	0.0144
FREEREPA	0.0797984	0.0920603	0.8668	0.3860
ILLNESS	0.186948	0.0182805	10.23	0.0000
ACTDAYS	0.126846	0.00503397	25.20	0.0000
HSCORE	0.0300810	0.0100994	2.979	0.0029
CHCOND1	0.114085	0.0666396	1.712	0.0869
CHCOND2	0.141158	0.0831451	1.698	0.0896

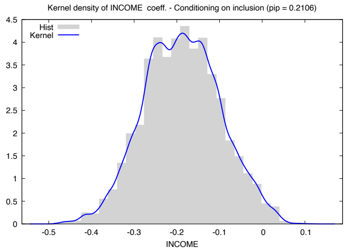
BMA in action

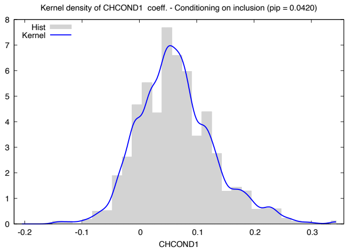
```
------------------------------------------------------
Type of specification: Poisson model
Model Prior: P(M) ~ Uniform
Model dynamics: MCMCMC - add/delete (1)var
Resampling allowed: No
MPI - threads: 4
Number of iterations/burn-in/thinning: 100000/10000/0
------------------------------------
Overall sampling statistics
              mean  std.err.       pip   c. mean     c. se    BGstat
   const  -1.47367   0.03055   1.00000  -1.47367   0.03055   1.00109
     SEX   0.18726   0.07293   0.93662   0.19993   0.05609   1.00042
     AGE   0.33237   0.32932   0.61384   0.54146   0.25191   1.00871
   AGESQ   0.16917   0.33283   0.35460   0.47708   0.40682   1.00825
  INCOME  -0.03945   0.08671   0.21060  -0.18732   0.08945   1.00016
LEVYPLUS   0.00910   0.03841   0.08289   0.10973   0.08222   1.00173
FREEPOOR  -0.28690   0.27082   0.60611  -0.47335   0.18099   1.00053
FREEREPA   0.00653   0.04080   0.04653   0.14035   0.13039   1.00103
 ILLNESS   0.20208   0.01898   1.00000   0.20208   0.01898   1.00151
 ACTDAYS   0.12901   0.00534   1.00000   0.12901   0.00534   1.00035
  HSCORE   0.02538   0.01607   0.78297   0.03242   0.01008   1.00151
 CHCOND1   0.00270   0.01889   0.04203   0.06428   0.06729   1.00025
 CHCOND2   0.00594   0.03089   0.05533   0.10736   0.07972   1.00128
------------------------------------
Gelman & Brooks multivariate R: 1.016
------------------------------------
Best specifications (Posterior > 0.10):
1) Model 00000c5c: P(M|y) = 0.18184
   Covariates: const SEX AGE FREEPOOR ILLNESS ACTDAYS HSCORE
2) Model 00000c1c: P(M|y) = 0.13176
   Covariates: const SEX AGE ILLNESS ACTDAYS HSCORE
```

Marginal plots

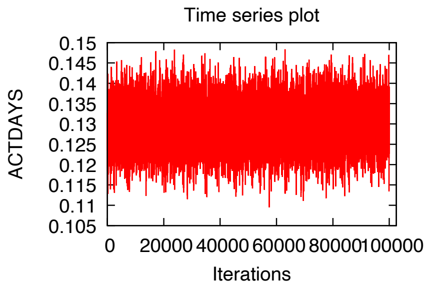


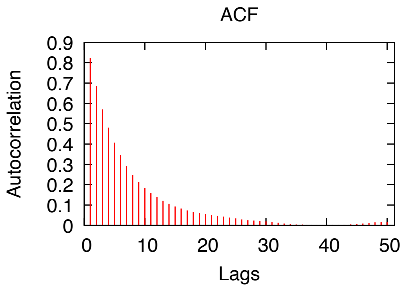


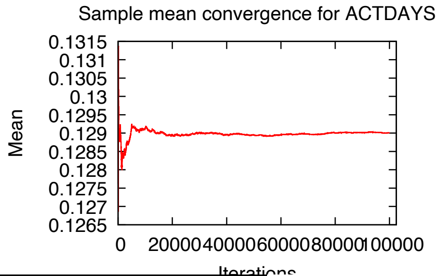




Diagnostic checks

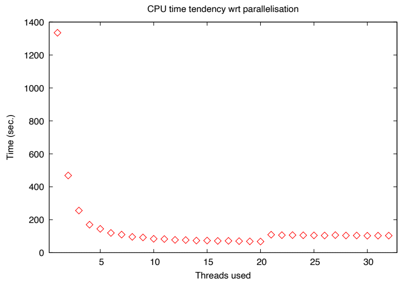


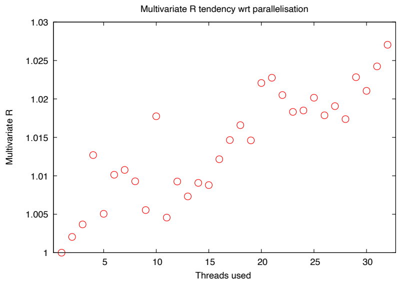




Parallelization contribution

Compare the performance and convergence on 1,...,32 threads with 200000 iterations (20000 burn-in)





Conclusion

- `ParMA` provides a RJMCMC sampler which can deal effectiviely with GLMs;
- Several diagnostic tools are provided;
- MPI leads to huge computational gain in CPU time.

References I

Brooks, S. P. and A. Gelman (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics 7*(4), 434–455.

Cameron, C. A. and P. K. Trivedi (2013). *Regression Analysis of Count Data* (2nd ed.), Volume 53. Cambridge: Cambridge University Press.

Fouskakis, D., I. Ntzoufras, and D. Draper (2009). Bayesian variable selection using cost-adjusted bic, with application to cost-effective measurement of quality of health care. *The Annals of Applied Statistics 3*(2), 663–690.

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In *Bayesian Statistics (J.M. Bernardo, J.O. Berger, A.P. Dawid and A.F.M. Smith eds.)*, Volume 4, pp. 169–193. Clarendon Press, Oxford, UK.

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

Green, P. J. (2003). Trans-dimensional markov chain monte carlo. In *Highly Structured Stochastic System*, pp. 179–198. Oxford University Press.

References II

Heidelberger, P. and P. D. Welch (1983). Simulation run length control in the presence of an initial transient. *Operations Research 31*(6), 1109–1144.

Lamnisos, D., J. E. Griffin, and M. F. Steel (2009). Transdimensional sampling algorithms for bayesian variable selection in classification problems with many more variables than observations. *Journal of Computational and Graphical Statistics 18*(3), 592–612.

Lamnisos, D., J. E. Griffin, and M. F. Steel (2013). Adaptive monte carlo for bayesian variable selection in regression models. *Journal of Computational and Graphical Statistics 22*(3), 729–748.

Vats, D., J. M. Flegal, and G. L. Jones (2019). Multivariate output analysis for markov chain monte carlo. *Biometrika 106*(2), 321–337.

RJMCMC: a graphic representation

Consider a three regressor example and assume to be in $\theta_i = [M_i = (1,0,1); \beta_i = \{\beta_{1i}, \beta_{3i}\}]$:

$$\begin{tikzcd}
 & & \begin{array}{c} M_j = (1,1,1); \\ \{\beta_{1j}, \beta_{2j}, \beta_{3j}\} = g(\beta_{1i}, u_i, \beta_{3i}) \end{array} \\
\begin{array}{c} M_i = (1,0,1); \\ \{\beta_{1i}, \beta_{3i}\} \end{array} \arrow[r, "\text{prop.}", "\text{kernel}"'] & q(M_j|M_i) \arrow[ur, "1/3"] \arrow[r, "1/3"'] \arrow[dr, "1/3"] & \begin{array}{c} M_j = (1,0,0); \\ \{\beta_{1j}, (u_j)\} = g(\beta_{1i}, \beta_{3i}) \end{array} \\
 & & \begin{array}{c} M_j = (0,0,1); \\ \{(u_j), \beta_{3j}\} = g(\beta_{1i}, \beta_{3i}) \end{array}
\end{tikzcd}$$

Technicalities

Assume to be in M_i, β_i with $E(\beta_i|M_i, y) = \mu_i$ and
$V(\beta_i|M_i, y) = V_i = B_i B_i^T$
$(\beta_i, M_i) \rightarrow (\beta_j, M_j) = g(\beta_i, M_i)$ could be

$$\beta_j = g(\beta_i, M_i, u_i) = \mu_j + B_j \beta_{std}$$

with β_{std} defined as:

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

where k_i denotes the number of variables in M_i; u is a vector of standard normal random numbers with dimension $k_j - k_i$, R a random permutation matrix and finally the operator $[...]^{k_j}$ indicates the first k_j elements of the vector.

The probability of accepting the move α is :

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

and:

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

with $f()$ as the density function;

$$P(\beta_i, M_i|y) \propto p(y|M_i, \beta_i)P(\beta|M_i)P(M_i)$$

where

- $p(y|M_i, \beta_i)$: likelihood of model M_i
- $P(\beta|M_i)$: prior β_i conditioned on M_i
- $P(M_i)$: model prior.

In brief:

Following Lamnisos et al. (2013), the procedure can be summarised as follow:

1. Set the initial β_i related to the model M_i, in general the full specification;
2. Propose a new model M_j from a transitional kernel $q(M_j|M_i)$ and compute its β_j;
3. Accept the move with probability α, otherwise stay in (β_i, M_i);
4. (Within model movement);
5. Repeat from 2, till convergence.

Prior choices

- $M_i \rightarrow$ each entry has an independent Bernoulli with parameters up to the user (by default 0.5);

- $\beta_i \rightarrow$ Normal distribution with
  - prior mean μ_i up to the user choice (by default 0)
  - prior covariance matrix V_i, which can be defined as a ridge prior, a Zellner-g type or even a totally customizable one.
  - the intercept has a diffuse prior in case of linear model or a N(0, 100) in case of other GLMs. (Lamnisos et al., 2009)

back