Accounting for Model Uncertainty via Trans-dimensional Genetic Algorithms

RICARDO S. EHLERS
Federal University of Paraná
MARCO A. R. FERREIRA
Federal University of Rio de Janeiro

Abstract

We develop for regression models trans-dimensional genetic algorithms for the exploration of large model spaces. Our algorithms can be used in two different ways. The first possibility is to search the best model according to some criteria such as AIC or BIC. The second possibility is to use our algorithms to explore the model space, search for the most probable models and estimate their posterior probabilities. This is accomplished by the use of genetic operators embedded in a reversible jump Markov chain Monte Carlo algorithm in the model space with several chains. As these chains run simultaneously and learn from each other via the genetic operators, our algorithm efficiently explores the large model space and easily escapes local maxima regions common in the presence of highly correlated regressors. We illustrate the power of our trans-dimensional genetic algorithms with applications to two real data sets.

Key Words: Model selection, Genetic algorithms, Markov chain Monte Carlo, reversible jump MCMC.

1 Introduction

The usual approach for model selection is to use information criteria such as the AIC (Akaike 1974) and/or BIC (Schwartz 1978) to discriminate between competing models. These criteria may be used to compare ARIMA time series models of different orders, polynomial models, for variable selection in (generalized) linear models, etc. In practical terms, the use of these criteria for model comparison involves the model fit and computation of the criterion for each competing model. However, in realistically complex cases the number of competing models may be

quite large, thus evaluation of each competing model would be too costly and, as a result, numerical techniques are necessary to efficiently explore model space. Moreover, as a result of the presence of highly correlated regressors or ambiguities in the model definition, the function to be optimized may present many local maxima regions. Here, we develop trans-dimensional genetic algorithms that use genetic operators embedded in a reversible jump Markov chain Monte Carlo algorithm with several chains. As these chains run simultaneously and learn from each other via the genetic operators, our algorithm efficiently explores the large model space and easily escapes local maxima regions.

In this paper, we distinguish between the steps of model identification and parameter estimation given the model. We assume that the estimation step may be performed without too much computational effort, conditional on the model, using standard statistical methods (and software). So, the main purpose here is to propose an effective and semi-automatic method for the identification task. This is accomplished by adapting the genetic algorithm to propose transdimensional jumps in several MCMC chains which are run simultaneously and let the chains learn from each other via genetic operators. The genetic algorithm is a highly effective technique for maximizing irregular functions defined over high-dimensional spaces (Holland 1975). Chatterjee et al. (1996) provide a good overview of statistical applications of genetic algorithms.

Assume that the possible models can be enumerated as $M_1, M_2, \ldots, M_c$ and indexed by a model indicator k. Associated with each model there is a likelihood function $p(\boldsymbol{y}|\boldsymbol{\theta}_k, k)$ depending upon an unknown parameter vector $\boldsymbol{\theta}_k \in \Theta_k$ of length n_k which may vary from model to model. We distinguish between the steps of model identification and parameter estimation given the model. We assume that for a given model M_k the estimation of $\boldsymbol{\theta}_k$ may be performed without too much computational effort using standard statistical methods (and software). Moreover, we assume that a measure of model performance can be computed fairly fast. For example, the measure of performance may be the logarithm of the model posterior probability, the Akaike Information Criterion (AIC) of the Bayesian Information Criterion (BIC). In particular, AIC and BIC for a given model take into account the maximum likelihood function for that model and a penalizing term for model complexity in terms of number of parameters, and are given by

$$\begin{aligned} AIC(\hat{\boldsymbol{\theta}}_k, k) &= -2\log p(\boldsymbol{y}|\hat{\boldsymbol{\theta}}_k, k) + 2n_k \\ BIC(\hat{\boldsymbol{\theta}}_k, k) &= -2\log p(\boldsymbol{y}|\hat{\boldsymbol{\theta}}_k, k) + n_k \log T \end{aligned}$$

and its minimum value will select the "best" model over all candidate models. Model selection using these or any other information criterion may become an issue when the number of competing models is high. Thus, the main purpose here is to develop an effective and semi-automatic method for the identification task.

This is accomplished by adapting a genetic algorithm to propose trans-dimensional jumps in several MCMC chains which are run simultaneously and let the chains learn from each other via genetic operators.

Genetic algorithms are highly effective for maximizing irregular functions defined over high-dimensional spaces (Holland 1975). An introduction to some basic genetic algorithms statistical applications can be found in Chatterjee et al. (1996). Genetic algorithms have operators that borrow their interpretation from biological evolution of species: selection, crossover and mutation. In the case of model space exploration, we start the genetic algorithm with a population of models that is a small subset of the competing models. In each iteration of the algorithm this population evolves according to the genetic operators. In the selection step, models with high performance are more likely to remain in the next population. In the crossover step, pairs of models are combined and generate offspring models that are more likely to be accepted in the population if they have high performance. In the mutation step, each individual model suffers a perturbation and the resulting mutant is accepted or not in the population depending on its performance.

This genetic algorithm is a Markov chain and we compute the acceptance probabilities associated with each genetic operator in such a way to guarantee that the chain will converge to a distribution defined in terms of the measure of performance. In particular, if the measure of performance of a model is proportional to its posterior probability, then the chain will converge to the posterior distribution on the model space. As a result, this genetic algorithm will provide estimates of the most probable set of models and their respective posterior probabilities.

Assuming that the prior information is equivalent to the information contained in one observation, the BIC provides an approximation (with error of order $O(n^{-0.5})$) to the log of the Bayes factor for nested hypotheses (Kass and Wasserman 1995). Thus, in that case the posterior distribution on the model space may be well approximated by the Boltzman distribution

$$P(M_k|\boldsymbol{y}) \propto \exp\{-BIC(\hat{\boldsymbol{\theta}}_k, k)/2\}. \tag{1}$$

We illustrate the application of our genetic algorithm with the BIC as the measure of performance of the models. As a result, the chain of population models converges in distribution to the Boltzman distribution in Equation (1). Nevertheless, given specific priors for each competing model and provided the predictive distribution for each model can be easily computed, we build our algorithm such that the chain of population models is guaranteed to converge to the appropriate posterior distribution on the model space.

This paper is organized as follows. Section 2 provides an introduction to fixed dimension genetic algorithms. Section 3 describes variable dimension genetic algorithms. Section 4 details our trans-dimensional genetic algorithm for model space

exploration. Section 5 illustrates our algorithm with two real data applications. Section 6 concludes with a brief discussion.

2 Fixed Dimension Genetic Algorithm

Suppose we wish to maximize a function $g(\boldsymbol{\theta})$ (typically called fitness function) where $\boldsymbol{\theta}$ is a L-dimensional vector $(\theta_1, \ldots, \theta_L)$. We need to create a population of M solutions $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M$. Each cycle of the algorithm is comprised of the steps selection, crossover and mutation which we now describe.

2.1 Selection Step

In this step the population is altered by allowing better solutions to remain with high probability while poorer solutions are potentially removed from the population. So, the selection step proceeds by drawing M elements $\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_M^*$ with replacement from the current population with probability w_i proportional to their fitness $g(\boldsymbol{\theta}_i)$ and then solutions with a higher fitness have a larger probability of being selected to form the next generation. Of course if g is not non-negative we can use say $w_i \propto \exp(g(\boldsymbol{\theta}_i))$.

2.2 Crossover Step

Now given this new population, we select a pair, say $(\boldsymbol{\theta}_i^*, \boldsymbol{\theta}_j^*,)$, according to some selection procedure and trade their values with a fixed probability p_c. A pair can be selected either uniformly from the current population or according to weights $w_i \propto \exp(g(\boldsymbol{\theta}_i))$. Given a pair, one possible scheme is the so-called one-point crossover where we choose $k \in \{1, \ldots, L-1\}$ and swap elements $k, \ldots, L$, i.e.

$$(\theta_{i,1}^*, \ldots, \theta_{i,k}^*, \theta_{i,k+1}^*, \ldots, \theta_{i,L}^*) \rightarrow (\theta_{i,1}^*, \ldots, \theta_{i,k}^*, \theta_{j,k+1}', \ldots, \theta_{j,L}')$$

$$(\theta_{j,1}^*, \ldots, \theta_{j,k}^*, \theta_{j,k+1}^*, \ldots, \theta_{j,L}^*) \rightarrow (\theta_{j,1}^*, \ldots, \theta_{j,k}^*, \theta_{i,k+1}', \ldots, \theta_{i,L}')$$

where each swap is accepted with probability p_c.

2.3 Mutation Step

Given this modified population, perturb each element with a small probability p_m by adding a random noise. The MCMC equivalent to this step is to update the parameters according to a Metropolis-Hastings or Gibbs sampler scheme.

3 Variable Dimension Genetic Algorithm

Define a population $(k_1, \boldsymbol{\theta}_1), \ldots, (k_M, \boldsymbol{\theta}_M)$ of model indicators and associated parameter vectors. Here the lenghts L_i of the elements may vary from one model to the next.

For each element compute $w_i \propto g(k_i, \boldsymbol{\theta}_i)$ and draw elements with replacement from this population with probability w_i (perhaps using the elitist strategy)

$$(k_1^*, \boldsymbol{\theta}_1^*), \ldots, (k_M^*, \boldsymbol{\theta}_M^*).$$

In the crossover step, randomly choose a pair $(k_i^*, \boldsymbol{\theta}_i^*), (k_j^*, \boldsymbol{\theta}_j^*)$ and trade values on the portion of the vectors that are of the same length.

The mutation step involves trans-dimensional moves and we use a reversible jump MCMC scheme (Green 1995). In this case the stationary distribution is $g(k, \boldsymbol{\theta})$ and, as is usual in model selection problems, we restrict attention to certain jump proposals. For the ith component in the population suppose that we propose a move of type r from $\phi_i = (k_i, \boldsymbol{\theta}_i)$ to $\phi_i' = (k_i', \boldsymbol{\theta}_i')$ with probability $P_r(\phi_i)$ where typically $L_i \neq L_{i'}$. This can be done by generating a random vector $\boldsymbol{u}$ from a proposal distribution q and choosing the proposed state as $f(\boldsymbol{\theta}_i, \boldsymbol{u})$. This proposal would then be accepted with probability $\min(1, A)$ where

$$A = \frac{g(k_i', \boldsymbol{\theta}_i) P_{r'}(\phi') q(\boldsymbol{u}')}{g(k_i, \boldsymbol{\theta}_i) P_r(\phi) q(\boldsymbol{u})} \left| \frac{\partial f(\boldsymbol{\theta}_i, \boldsymbol{u})}{\partial(\boldsymbol{\theta}_i, \boldsymbol{u})} \right| \tag{2}$$

where $P_r(\phi)$ denotes the probability of proposing a move type r when in state ϕ and r' is the move in the opposite direction.

4 Variable Selection Problems

This is probably one of the most common settings for application of MCMC-based model determination where we decide which parameters should be included in a model. In a regression problem, for a set of p candidate variates $X_1, \ldots, X_p$ and allowing for any subset of variates to appear in the model, there are 2^p possible candidate models as each covariate can either be included or not. Typically, there is substantial uncertainty about which subset of covariates should be included in the model for any given data set.

We can uniquely describe each candidate model as $M_i = (x_{i1}, \ldots, x_{ip})$ where $x_{ij} = 1$ if X_j is included in model M_i and $x_{ij} = 0$ otherwise. In a context where we have a large number of candidate models, which is the situation of interest here, it is more useful to derive an equivalent scalar model indicator k. As suggested in Brooks et al. (2002), this can be easily accomplished by treating the vector M_i as

a number expressed in base 2 and converting to the decimal equivalent by setting

$$k = 1 + \sum_{j=1}^{p} x_{ij} 2^{j-1} \ \in \ \{1, \ldots, 2^p\}.$$

We can easily map back to M_i by iteratively solving the equation $1+\sum_{j=1}^{p} x_{ij}2^{j-1} = k$ for $x_{i1}, \ldots, x_{ip}$.

In terms of the genetic algorithm we can now create a population of models $\boldsymbol{x} = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$ and propose between-model moves in each cycle using the genetic operators. In this paper we use the BIC to distinguish between competing models so that the fitness function (which is the target distribution here) is $b(\boldsymbol{\theta}_k, k) \propto \exp(-BIC(\boldsymbol{\theta}_k, k)/2)$.

Our motivation to use BIC to compare models comes from its derivation as a large sample approximation to twice the Bayes factor Schwartz (1978).

It is easy to see that $b(\boldsymbol{\theta}_k, k)$ defines a discrete probability distribution over the set of BIC values for each candidate model and the highest probability is placed at the "best" model. The parameters $\boldsymbol{\theta}_k$ within each model are estimated via maximum likelihood using any statistical software (we use the package R). In practice, however, computing the relevant BIC values for each possible model can be very time consuming when the number of competing models is high. For example, with 20 candidate covariates we would need to compute over one million BIC values. If the data consists of a large number of observations then clearly any time saving approach would be an attractive proposition.

The Boltzman distribution for each individual $\boldsymbol{x}_i$ can be defined as

$$b(\boldsymbol{x}_i) \propto \exp\{-BIC(\boldsymbol{x}_i)/2\}$$

As usual in the Evolutionary Monte Carlo literature we augment the state of the Markov chain to the population $\boldsymbol{x}$ and the Boltzman distribution of the population is

$$b(\boldsymbol{x}) = \prod_{i=1} b(\boldsymbol{x}_i) \propto \exp\left\{-\sum_{i=1} BIC(\boldsymbol{x}_i)/2\right\}.$$

The reversible jump MCMC algorithm is then used to obtain a sample from this target distribution. After a burn-in period, the frequency with which each model has been visited by the Markov chains estimates the probability of the associated BIC value. The model with highest probability is then an estimate of the "best" model. Also, a sensible non-Bayesian model averaging procedure can be done based on these estimated probabilities. If Δ is the quantity of interest that retains its interpretation in all models, such as a future observation, then

$$p(\Delta|D) = \sum_{i=1}^{2^p} p(\Delta|D, M_i) p(M_i|D).$$

Of course, the idea is that the chain will not cover the entire model space as this would require evaluating the BIC at each possible model. Rather, the objective in this context is to explore the model space and capture the models with higher probabilities. Thus, in any practical application of our algorithm the above sum would include a much smaller number of terms based upon model probabilities. We can go further and use the Occam's window method of Madigan and Raftery (1994) thus picking a small subset of parsimonious data-supported models.

In this particular implementation of the reversible jump MCMC, suppose that we propose a move from model k with parameters $\boldsymbol{\theta}_k$ to model k' with parameters $\boldsymbol{\theta}_{k'}$ with the whole set of new parameters estimated by maximum likelihood directly in the k'-dimensional space. In terms of dimension matching, this is equivalent to defining a k'-dimensional vector $\boldsymbol{u} = (u_1, \ldots, u_{k'})$ with the maximum likelihood estimates and then setting the change of variables as $\boldsymbol{\theta}_{k'} = \boldsymbol{u}$ and $\boldsymbol{u}' = \boldsymbol{\theta}$, i.e.

$$(\theta'_1, \ldots, \theta'_{k'}, u'_1, \ldots, u'_k) = (u_1, \ldots, u_{k'}, \theta_1, \ldots, \theta_k)$$

which has unity Jacobian. So, given the current population $\boldsymbol{x}$ we propose a new population $\boldsymbol{x}'$ via genetic operators and accept with probability

$$\min\left(1, \frac{\exp\{-BIC(\boldsymbol{x}')/2\}}{\exp\{-BIC(\boldsymbol{x})/2\}} \frac{P(\boldsymbol{x}', \boldsymbol{x})}{P(\boldsymbol{x}, \boldsymbol{x}')}\right)$$

where $P(\boldsymbol{x}, \boldsymbol{x}')$ is the probability of proposing a jump from population $\boldsymbol{x}$ to $\boldsymbol{x}'$.

In what follows we describe the way in which we apply genetic operators to propose trans-dimensional moves in this context.

4.1 Crossover Move

Given the current population of models $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$ we randomly select $[M/2]$ pairs of individuals without replacement and propose a new population as follows. For a particular pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$,

1. select those elements with different values $K = \{k : x_{ik} \neq x_{jk}\}$
2. randomly choose $k \in K$
3. set $x'_{ik} = x_{jk}$ and $x'_{jk} = x_{ik}$
4. Accept this new population with probability $\min(1, A)$ where

$$A = \frac{\exp(-BIC(\boldsymbol{x}'_i)/2 - BIC(\boldsymbol{x}'_j)/2)}{\exp(-BIC(\boldsymbol{x}_i)/2 - BIC(\boldsymbol{x}_j)/2)} \frac{P(\boldsymbol{x}', \boldsymbol{x})}{P(\boldsymbol{x}, \boldsymbol{x}')}$$

5. repeat the above steps for all $[M/2]$.

where $P(\boldsymbol{x}, \boldsymbol{x}')$ is the probability of proposing a move from population $\boldsymbol{x}$ to population $\boldsymbol{x}'$. We note that is this one-point crossover proposal the crosspoint k is chosen uniformly so $P(\boldsymbol{x}, \boldsymbol{x}')$=$P(\boldsymbol{x}', \boldsymbol{x})$.

This updating scheme is repeated for all $[M/2]$ pairs of individuals selected without replacement from the population.

4.2 Mutation Move

A mutation step is proposed for each individual in the current population by either including a new regressor with probability w, or deleting an existing one with probability $1 - w$. Suppose that we are now updating individual $\boldsymbol{x}_i$ and define the set $J = \{j : x_{ij} = 0\}$ of regressors not in the current model. If an inclusion is proposed we randomly choose $j \in J$ and set $x'_{ij} = 1$. This move is accepted with probability $\min(1, A)$ where

$$A = \frac{\exp(-BIC(\boldsymbol{x}'_i)/2)}{\exp(-BIC(\boldsymbol{x}_i)/2)} \frac{(1-w)\ |J|}{w\ (|\overline{J}| + 1)}$$

with $\overline{J} = \{j : x_{ij} = 1\}$, i.e. the set of regressors included in the current model, and $|J|$ denotes the cardinality of J. Likewise, if a deletion is proposed we remove a randomly chosen regressor by drawing $j \in \overline{J}$ and setting $x'_{ij} = 0$. This move is accepted with probability $\min(1, A^{-1})$. Of course an inclusion or deletion is proposed with probability one if the cardinality of the set J is either 0 or p. This updating scheme is repeated for all individuals in the population.

5 Real Data Examples

In this section we illustrate how our algorithm takes account of model uncertainty about the variables to be included in the model.

5.1 Logistic Regression

We now illustrate how our algorithm works in the context of logistic regression using a data set of risk factors associated with low infant birth weight. The data appeared in Hosmer and Lemeshow (1989) and is available in the MASS library of R. The dependent variable measures whether the weights of 189 babies was low at birth and there are 8 potential covariates two of which are categorical (race and number of previous premature labors, ptl). Table 1 shows a summary output based on 5000 iterations of our algorithm (after a 5000 burn-in) for 10 chains in the population. The columns show model probabilities, model indicator and inclusion indicator (0 or 1) for each covariate. The last row shows the probabilities that

each covariate is included in the model. Simulations took less than 2 minutes and visited a bit more than half the number of candidate models.

Table 1: Low infant birth weight. Model posterior probabilities for the 10 most visited models and covariates inclusion probabilities.

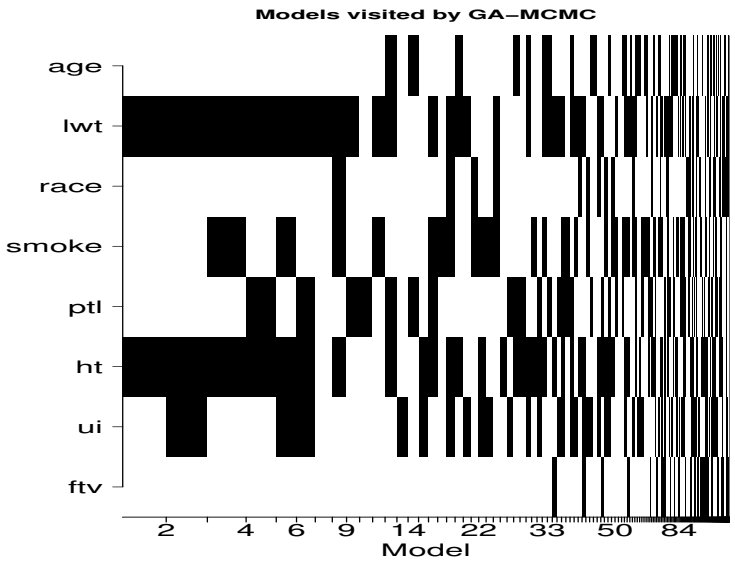
Model (k)	Covariates indicator								Model posterior
	age	lwt	race	smoke	ptl	ht	ui	ftv	probability
35	0	1	0	0	0	1	0	0	0.0962
99	0	1	0	0	0	1	1	0	0.0673
51	0	1	0	0	1	1	0	0	0.0600
43	0	1	0	1	0	1	0	0	0.0599
107	0	1	0	1	0	1	1	0	0.0333
3	0	1	0	0	0	0	0	0	0.0294
115	0	1	0	0	1	1	1	0	0.0287
17	0	0	0	0	1	0	0	0	0.0239
19	0	1	0	0	1	0	0	0	0.0202
47	0	1	1	1	0	1	0	0	0.0202
Covariates inclusion probability	0.190	0.696	0.140	0.381	0.349	0.659	0.376	0.081	–

A visual summary of the output is shown in Figure 1 where each row corresponds to a covariate and each column corresponds to a model. The corresponding rectangle is black if the covariate is included in the model and white otherwise. The basic idea of this plot was first proposed by Clyde (1999) and we follow Raftery, Painter, and Volinsky (2005) making the width of each column proportional to the model probability. Variables "lwt" (mother's weight in pounds at last menstrual period), "ht" (history of hypertension), "ui" (presence of uterine irritability), "smoke" (smoking status during pregnancy) and "ptl" (number of previous premature labors) have moderate to high probabilities of being in the model while "age" (mother's age in years), "race" (mother's race) and"ftv" (number of physician visits during the first trimester) have lower probabilities.

5.2 Censored Survival Models

We illustrate the use of our algorithm in the context of Cox proportional hazards using the well known data set of Fleming and Harrington (1991) concerning primary biliary cirrhosis, a rare autoimmune liver disease. We use the information on 312 randomized patients with the disease and the dependent variable is survival time, from which 187 are censored. There are 15 potential predictors and most

Figure 1: Image plot for the low infant birth weight data set.



variables have some missing data. This data set is available in the survival library of R.

When censoring is present there are as many terms in the partial likelihood (Cox, 1972) as there are events, d ($< n$). Volinsky and Raftery (2000) found that using d instead of n in the penalty term of the BIC results in an improved criterion and still satisfies the asymptotic properties shown by Kass and Wasserman (1995). This is also the approach we adopt here.

We ran 10000 iterations of our algorithm, discarding the first 5000 as burn-in, with 10 chains in the population. Simulations took around 3 minutes and 2047 models out of 32768 candidates were visited. A summary of the output is shown in Table 2 for the 10 most visited models. The first row shows model probabilities while the last column shows the probabilities that each covariate is included in the model. A visual summary of the output is shown in Figure 2.

It is clear that the covariates "age" (patient's age in years), "alb" (serum albumin), "bili" (serum bilirunbin), "edtrt" (no edema, untreated or successfully treated, unsuccessfully treated edema), "protime" (standardized blood clotting time) and "copper" (urine copper) all have high probabilities of being in the model. Two other variables, "sgot" (liver enzyme) and "stage" (histologic stage of disease) do not appear in the best model but there is a great deal of uncertainty about whether they should be included.

Table 2: Estimates of model probabilities for the 10 most visited models using the biliary cirrhosis data.

Probs	0.081	0.037	0.036	0.029	0.026	0.025	0.021	0.018	0.017	0.017	Prob.inc
age	1	1	1	1	1	1	1	1	1	1	0.999
alb	1	1	1	1	1	1	1	1	1	1	0.997
alkphos	0	0	0	0	0	0	0	1	0	0	0.038
ascites	0	0	0	0	0	0	0	0	0	0	0.012
bili	1	1	1	1	1	1	1	1	1	1	1.000
edtrt	1	1	1	0	1	1	1	1	1	1	0.916
hepmeg	0	0	0	0	0	0	0	0	0	0	0.012
platelet	0	0	0	0	0	0	0	0	0	0	0.014
protime	1	1	0	1	1	0	1	1	1	1	0.848
sex	0	0	0	0	0	0	0	0	0	0	0.020
sgot	0	0	0	0	0	0	1	0	1	1	0.101
spiders	0	0	0	0	0	0	0	0	0	0	0.007
stage	0	1	1	0	0	0	0	0	1	0	0.258
trt	0	0	0	0	0	0	0	0	0	0	0.011
copper	1	1	1	1	0	1	1	1	0	0	0.863

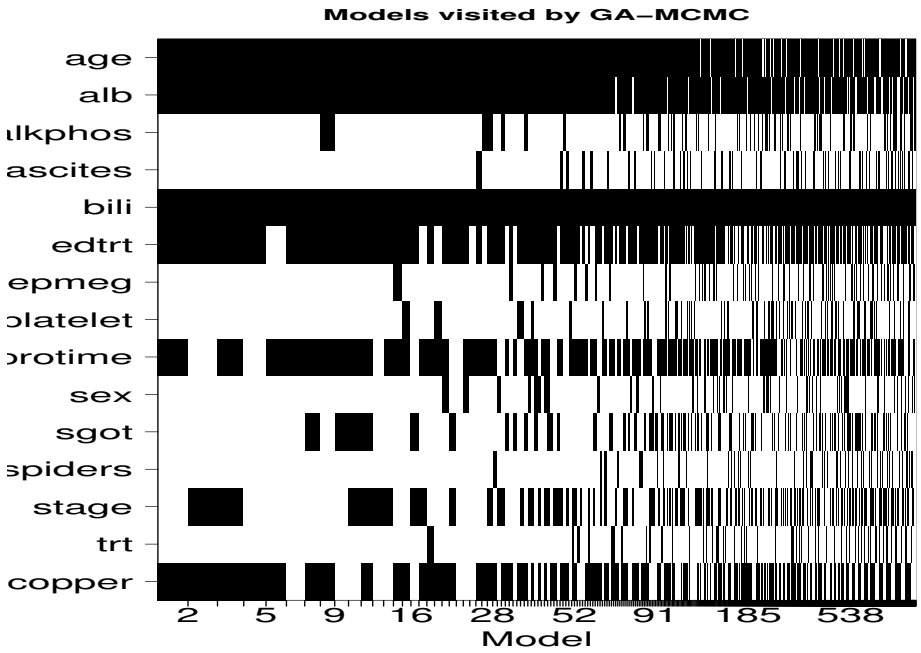
6 Discussion

In this paper we develop a new genetic algorithm for the exploration of high dimensional model spaces. We use genetic operators and reversible jump MCMC in such a way to guarantee that the parallel chains converge to the models posterior distribution. The main purpose was to propose an effective and semi-automatic method for the identification of a subset of models that concentrates high posterior probability.

We distinguish between the steps of model identification and parameter estimation given the model by assuming that the estimation step may be performed without too much computational effort, conditional on the model, using standard statistical methods and software. This was accomplished by adapting the genetic algorithm to propose jumps in several MCMC chains which are run simultaneously and letting the chains learn from each other via genetic operators.

Two real data examples were included to show the potentialities of our algorithm in practice. In both examples increasing the number of iterations and/or population size led to similar results. We have applied the algorithm to other data sets (both real and simulated) and obtained quite good results in terms of selecting the best models. Also, the R functions can be easily adapted to include other classes of models.

Figure 2: Image plot for the biliary cirrhosis data.



References

Akaike, H. (1974). A new look at the statistical identification model. *IEEE Transactions on Automatic Control 19*, 716–723.

Brooks, S. P., P. Giudici, and A. Philippe (2002). Nonparametric convergence assessment for MCMC model selection. *Journal of Computational and Graphical Statistics 12*, 1–22.

Chatterjee, S., M. Laudato, and L. Lynch (1996). Genetic algorithms and their statistical applications: an introduction. *Computational Statistics and Data Analysis 22*, 633–651.

Clyde, M. A. (1999). Bayesian model averaging and model search strategies. In J. M. Bernardo, A. F. M. Smith, A. P. Dawid, and J. O. Berger (Eds.), *Bayesian Statistics 6*. Oxford University Press.

Fleming, T. and D. Harrington (1991). *Counting Processes and Survival Analysis*. Wiley, New York.

Green, P. J. (1995). Reversible jump MCMC computation and Bayesian model determination. *Biometrika 82*, 711–732.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Ann Arbor.

Hosmer, D. and S. Lemeshow (1989). *Applied Logistic Regression*. New York: Wiley.

Kass, R. E. and L. Wasserman (1995). A reference Bayesian test for nested hypotheses with large samples. *Journal of the American Statistical Association* (90), 928–934.

Madigan, D. and A. E. Raftery (1994). Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association 89*, 1535–1546.

Raftery, A. E., I. S. Painter, and C. T. Volinsky (2005). BMA: An R package for Bayesian model averaging. *R News 5*(2), 2–8.

Schwartz, G. (1978). Estimating the dimension of a model. *Annals of Statistics 6*, 461–464.

Volinsky, C. T. and A. E. Raftery (2000). Bayesian information criterion for censored survival models. *Biometrics 56*, 256–262.