Bayesian model averaging

Dr. Jarad Niemi

Iowa State University

September 7, 2017

Bayesian model averaging

Let $\{M_\gamma : \gamma \in \Gamma\}$ indicate a set of models for a particular data set y. If Δ is a quantity of interest, e.g. effect size, a future observable, or the utility of a course of action, then its posterior distribution is

$$p(\Delta|y) = \sum_{\gamma \in \Gamma} p(\Delta|M_\gamma, y) p(M_\gamma|y)$$

where

$$p(M_\gamma|y) = \frac{p(y|M_\gamma)p(M_\gamma)}{p(y)} = \frac{p(y|M_\gamma)p(M_\gamma)}{\sum_{\lambda \in \Gamma} p(y|M_\lambda)p(M_\lambda)}$$

is the posterior model probability and

$$p(y|M_\gamma) = \int p(y|\theta_\gamma, M_\gamma) p(\theta_\gamma|M_\gamma) d\theta_\gamma$$

is the marginal likelihood for model M_γ and θ_γ is the set of parameters in model M_γ.

Bayesian model averaged moments

Since $p(\Delta|y)$ is a discrete mixture, we may be interested in simplifying inference concerning Δ to a couple of moments. Let $\hat{\Delta}_\gamma = E[\Delta|y, M_\gamma]$. Then the expectation is

$$E[\Delta|y] = \sum_{\gamma \in \Gamma} \hat{\Delta}_\gamma p(M_\gamma|y)$$

and the variance is

$$Var[\Delta|y] = \left[\sum_{\gamma \in \Gamma} (Var[\Delta|y, M_\gamma) + \hat{\Delta}_\gamma^2) p(M_\gamma|y) \right] - E[\Delta|y]^2$$

The appealing aspect here is that the moments only depend on the moments from each individual model and the posterior model probability.

Difficulties with BMA

- Evaluating the summation can be difficult since $|\Gamma|$, the cardinality of Γ, might be huge.
- Calculating the marginal likelihood.
- Specifying the prior over models.
- Choosing the class of models to average over.

Reducing cardinality

If $|\Gamma|$ is small enough, we can enumerate all models and perform model averaging exactly. But if $|\Gamma|$ is too large, we will need some parsimony.

Rather than summing over Γ, we can only include those models whose posterior probability is sufficiently large

$$\mathcal{A} = \left\{ M_\gamma : \frac{\max_\lambda p(M_\lambda|y)}{p(M_\gamma|y)} = \frac{\max_\lambda p(y|M_\lambda)p(M_\lambda)}{p(y|M_\gamma)p(M_\gamma)} \leq C \right\}$$

relative to other models where C is chosen by the researcher. Also, appealing to Occam's razor, we should exclude complex models which receive less support than sub-models of that complex model, i.e.

$$\mathcal{B} = \left\{ M_\gamma : \forall M_\lambda \in \mathcal{A}, M_\lambda \subset M_\gamma, \frac{p(M_\lambda|y)}{p(M_\gamma|y)} < 1 \right\}$$

So, we typically sum over the smaller set of models $\Gamma' = \mathcal{A} \setminus \mathcal{B}$.

Searching through models

One approach is to search through models and keep a list of the best models. To speed up the search the following criteria can be used to decide what models should be kept in Γ':

- When comparing two nested models, if a simpler model is rejected, then all submodels of the simpler model are rejected.
- When comparing two non-nested models, we calculate the ratio of posterior model probabilities

$$\frac{p(M_\gamma|y)}{p(M_{\gamma'}|y)}$$

if this quantity is less than O_L, we reject M_γ and if it is greater than O_R we reject $M_{\gamma'}$.

Using MCMC to search through models

Construct a neighborhood around $M^{(i)}$ (the current model in the chain), call it $nbh(M^{(i)})$. Now propose a draw M^* from the following proposal distribution

$$q(M^*|M^{(i)}) = \begin{cases} 0 & \forall M^* \notin nbh(M^{(i)}) \\ \frac{1}{|nbh(M^{(i)})|} & \forall M^* \in nbh(M^{(i)}) \end{cases}$$

Set $M^{(i+1)} = M^*$ with probability $\min\{1, \rho(M^{(i)}, M^*)\}$ where

$$\rho(M^{(i)}, M^*) = \frac{p(M^*|y)}{p(M^{(i)}|y)} \frac{|nbh(M^{(i)})|}{|nbh(M^*)|}$$

and otherwise set $M^{(i+1)} = M^{(i)}$. This Markov chain converges to draws from $p(M_\gamma|y)$ and therefore can estimate posterior model probabilities.

Evaluating the marginal likelihoods

Recall that as the sample size n increases, the posterior converges to a normal distribution. Let

$$g(\theta) = \log(p(y|\theta, M)p(\theta|M)) = \log p(y|\theta, M) + \log p(\theta|M)$$

Let $\hat{\theta}_{MAP}$ be the MAP for θ in model M. Taking a Taylor series expansion of $g(\theta)$ around $\hat{\theta}_{MAP}$, we have

$$g(\theta) \approx g(\hat{\theta}_{MAP}) - \frac{1}{2}(\theta - \hat{\theta}_{MAP})A(\theta - \hat{\theta}_{MAP})^\top$$

where A is the negative Hession of $g(\theta)$ evaluated at $\hat{\theta}_{MAP}$. Combining this with the first equation and exponentiating, we have

$$p(y|\theta, M)p(\theta|M) \approx p(y|\hat{\theta}_{MAP}, M)p(\hat{\theta}_{MAP}) \exp\left(-\frac{1}{2}(\theta - \hat{\theta}_{MAP})A(\theta - \hat{\theta}_{MAP})^\top\right)$$

Hence, the approximation to $p(\theta|y, M) \propto p(y|\theta, M)p(\theta|M)$ is normal.

Evaluating the marginal likelihoods (cont.)

If we take the integral over θ of both sides and take the logarithm, we have

$$\log p(y|M) \approx \log p(y|\hat{\theta}_{MAP}, M) + \log p(\hat{\theta}_{MAP}|M) + \frac{p}{2}\log(2\pi) - \frac{1}{2}\log|A|$$

where p is the dimension of θ, i.e. the number of parameters. We call this approximation the Laplace approximation.

Another approximation that is more computationally efficient but less accurate is to only retain terms that increase with n:

- $\log p(y|\hat{\theta}, M)$ increases linearly with n
- $\log|A|$ increases as $p \log n$

As n gets large $\hat{\theta}_{MAP} \to \hat{\theta}_{MLE}$. Taking these two together we have

$$\log p(y|M) \approx \log p(y|\hat{\theta}_{MLE}, M) - \frac{p}{2}\log n$$

Multiplying by -2, we obtain Schwarz's Bayesian Information Criterion (BIC)

$$BIC = -2\log p(y|\hat{\theta}_{MLE}, M) + p\log n$$

Priors over models

For data-based comparisons of models, you can use Bayes Factors directly since

$$BF(M_\gamma : M_{\gamma'}) = \frac{p(y|M_\gamma)}{p(y|M_{\gamma'})} = \frac{\int p(y|\theta_\gamma)p(\theta_\gamma|M_\gamma)d\theta_\gamma}{\int p(y|\theta_{\gamma'})p(\theta_{\gamma'}|M_{\gamma'})d\theta_{\gamma'}}$$

where the last equality is a reminder that priors over parameters still matter.

For model averaging, you need to calculate posterior model probabilities which require specification of the prior probabability of each model. One possible prior for regression models is

$$p(M_\gamma) = \prod_{i=1}^{p} w_i^{1-\gamma_i}(1-w_i)^{\gamma_i}$$

Setting $w_i = 0.5$ corresponds to a uniform prior over the model space.

BMA output

The quantities of interest from BMA are typically

- Posterior model probabilities $p(M_\gamma|y)$
- Posterior inclusions probabilities (for regression)

$$p(\text{including explanatory variable } i|y) = \sum_{\gamma\in\Gamma} p(M_\gamma|y)\mathrm{I}(\gamma_i = 1)$$

  which provides an overall assessment of whether explanatory variable i is important or not.
- Posterior distributions, means, and variances for "parameters", e.g.

$$E(\theta_i|y) = \sum_{\gamma\in\Gamma} p(M_\gamma|y)E[\theta_{\gamma,i}|y]$$

  But does this make any sense? What happened to θ_γ?
- Predictions:

$$p(\tilde{y}|y) = \sum_{\gamma\in\Gamma} p(M_\gamma|y)p(\tilde{y}|M_\gamma, y)$$

R packages for BMA

There are two main packages for Bayesian model average in R

- BMA: glm model averaging using BIC
- BMS: lm model averaging using g-priors and (possibly) MCMC

Until recently there was another package

- BAS: lm model averaging with a variety of priors and (possibly) MCMC (additionally performed sampling without replacement)

BMA in R

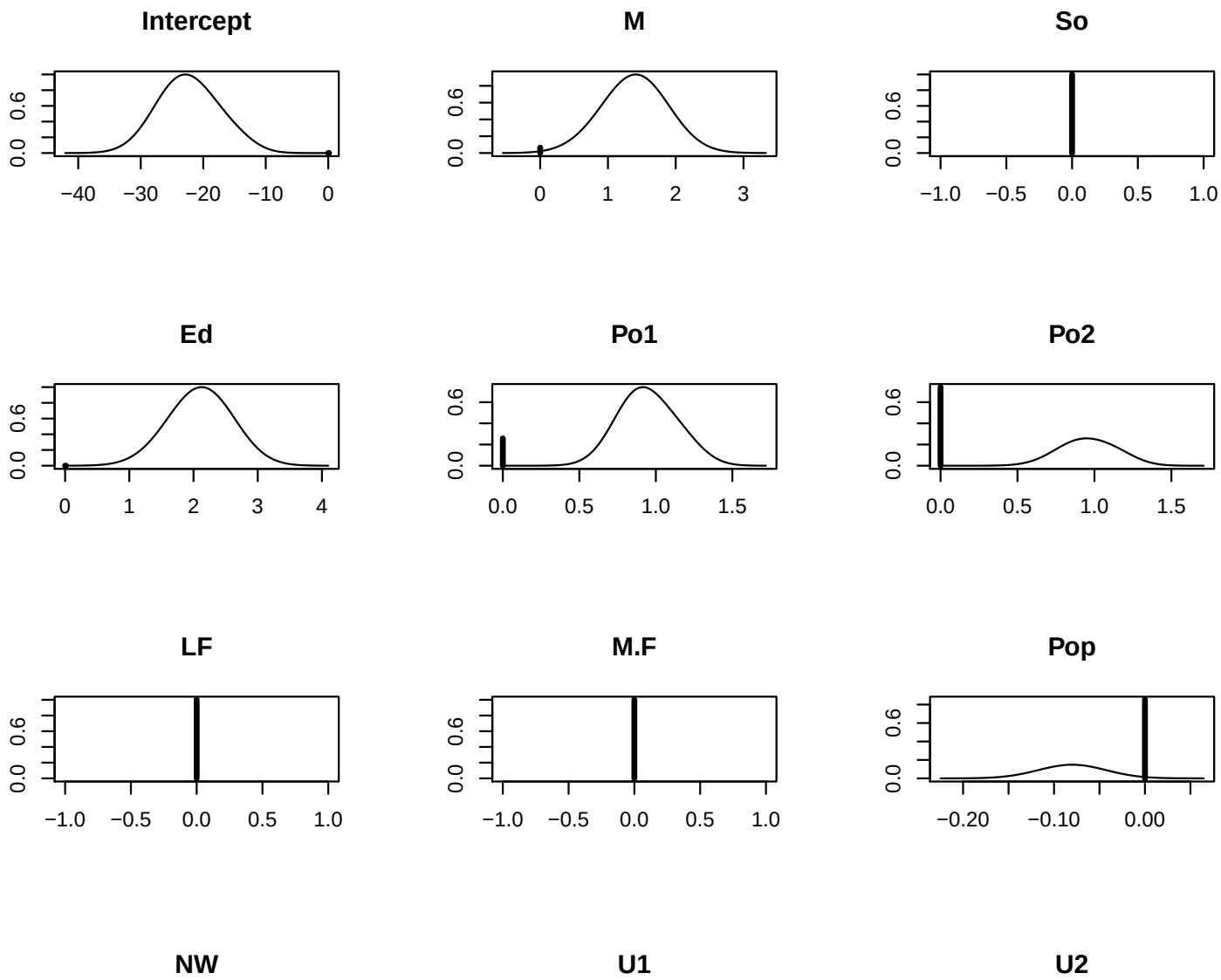
```
library(BMA)
UScrime <- MASS::UScrime

# Set up data
x = UScrime[,-16]
y = log(UScrime[,16])
x[,-2] = log(x[,-2])

# Run BMA using BIC
lma = bicreg(x, y,
             strict = TRUE, # remove submodels that are less likely
             OR = 20)       # maximum BF ratio
```

```
summary(lma)
```

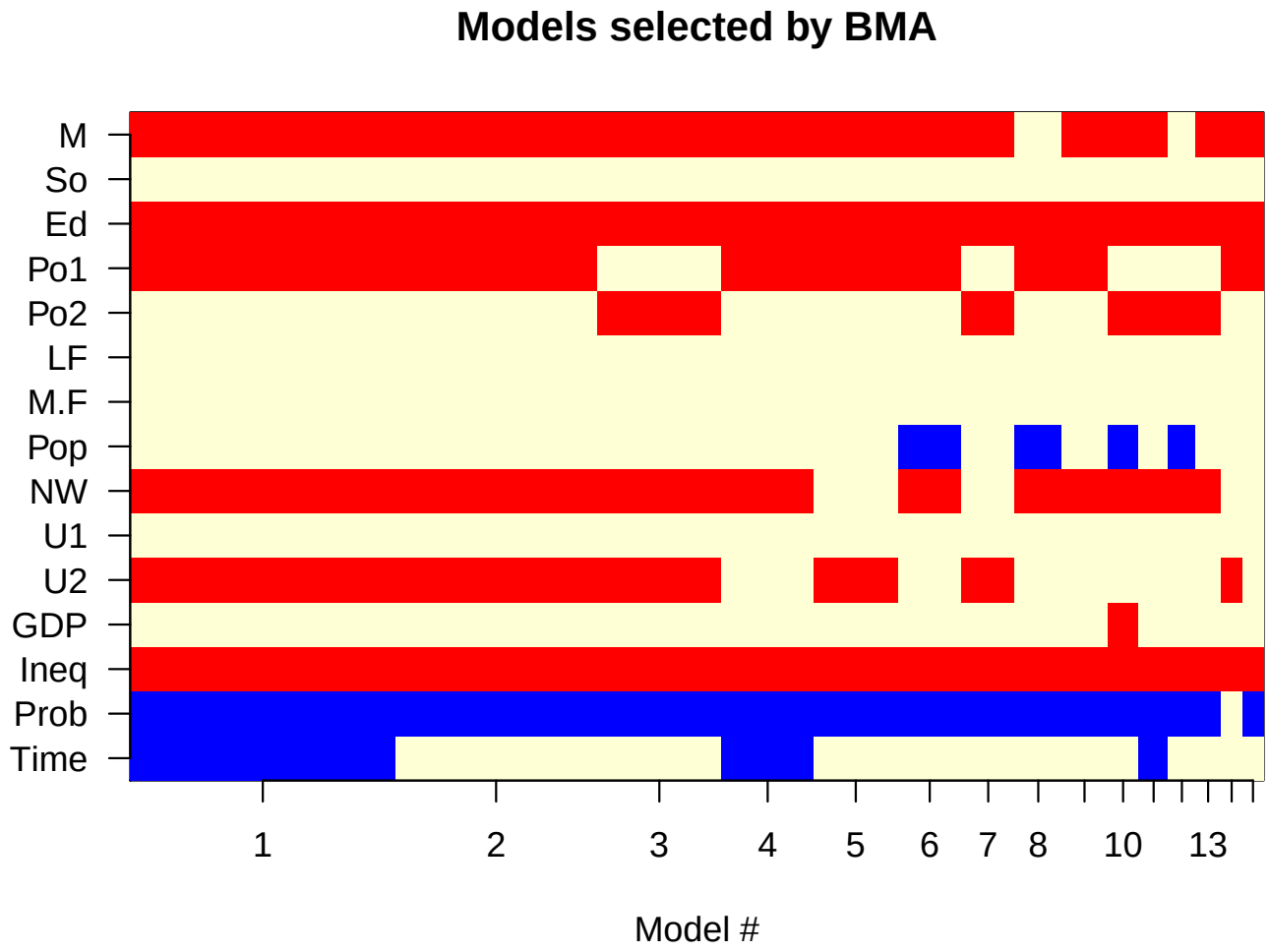
```
##
## Call:
## bicreg(x = x, y = y, strict = TRUE, OR = 20)
##
##
##   15  models were selected
##  Best  5  models (cumulative posterior probability =  0.6774 ):
##
##             p!=0    EV        SD       model 1    model 2    model 3    model 4    model 5
## Intercept  100.0  -22.23688  5.10541  -22.63715  -24.38362  -24.50477  -18.26547  -26.29204
## M           93.5    1.29995  0.59146    1.47803    1.51437    1.46061    1.12775    1.68239
## So           0.0    0.00000  0.00000       .          .          .          .          .
## Ed         100.0    2.10530  0.51224    2.22117    2.38935    2.39875    1.83590    2.08662
## Po1         74.2    0.70466  0.44869    0.85244    0.91047       .       0.89054    1.14936
## Po2         25.8    0.24704  0.43075       .          .       0.90689       .          .
## LF           0.0    0.00000  0.00000       .          .          .          .          .
## M.F          0.0    0.00000  0.00000       .          .          .          .          .
## Pop         14.8   -0.01189  0.03188       .          .          .          .          .
## NW          84.1    0.08479  0.05307    0.10888    0.08456    0.08534    0.11670       .
## U1           0.0    0.00000  0.00000       .          .          .          .          .
## U2          66.3    0.20895  0.18420    0.28874    0.32169    0.32977       .       0.33564
## GDP          2.8    0.02025  0.13628       .          .          .          .          .
## Ineq       100.0    1.33551  0.32647    1.23775    1.23088    1.29370    1.24269    1.57576
## Prob        98.1   -0.23801  0.10132   -0.31040   -0.19062   -0.20614   -0.33521   -0.16699
## Time        34.1   -0.10186  0.16713   -0.28659       .          .       -0.33177       .
##
## nVar                                       8          7          7          7          6
## r2                                       0.842      0.826      0.823      0.821      0.805
## BIC                                    -55.91243  -55.36499  -54.40788  -53.79094  -53.62430
## post prob                                0.234      0.178      0.110      0.081      0.074
```

Does this make any sense?

```
plot(lma)
```



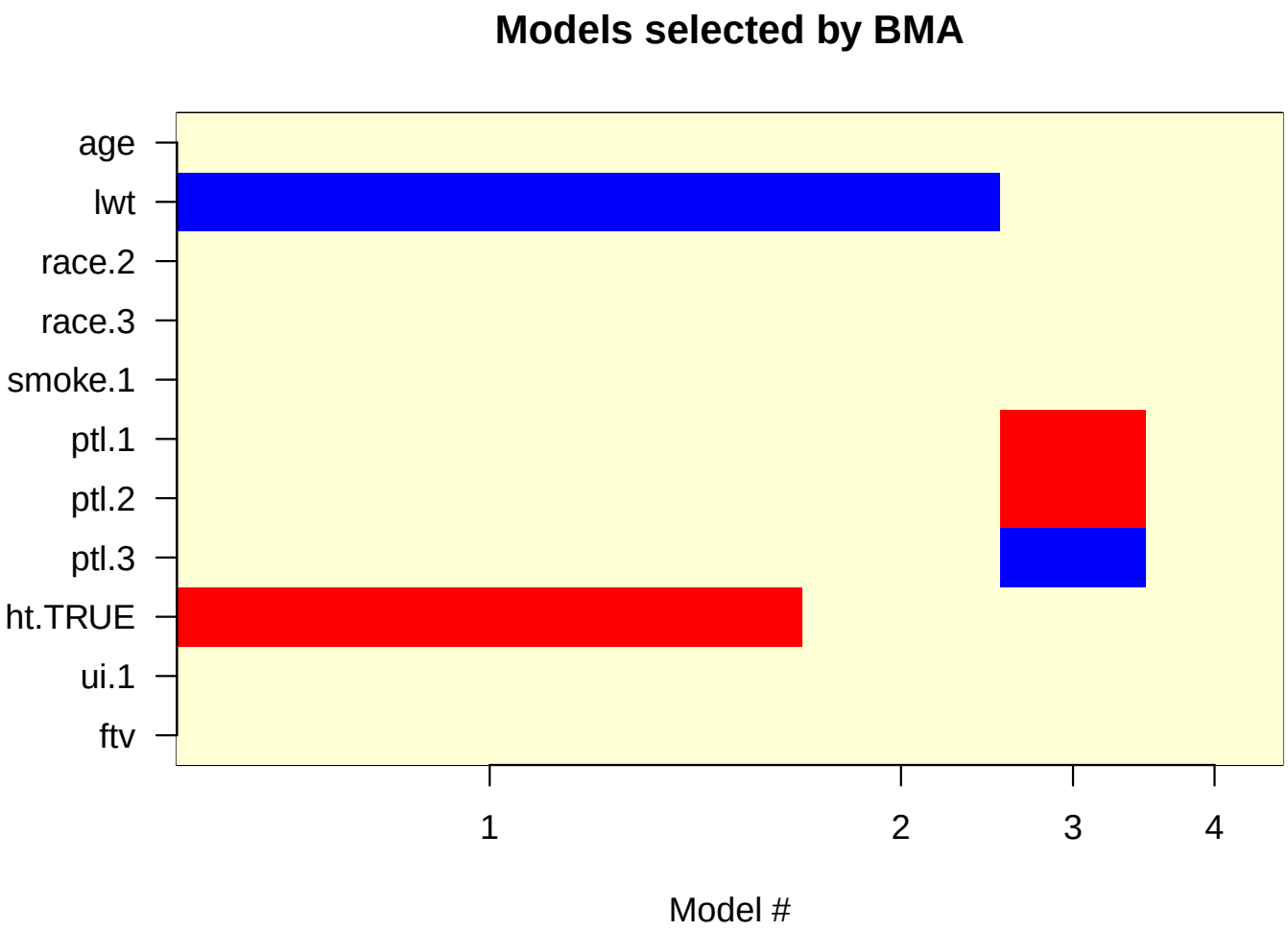
```
imageplot.bma(lma)
```



BMA for GLMs in R

```
# Set up data
y <- MASS::birthwt$low                    # 1 indicates birthweight < 2.5 kg

x <- MASS::birthwt %>%
  dplyr::select(-low,-bwt) %>%
  mutate(race  = factor(race),     # mother's race (1 = white, 2 = black, 3 = other)
         smoke = factor(smoke),    # smoking status during pregnancy.
         ptl   = factor(ptl),      # number of previous premature labours
         ht    = factor(ht>=1),    # history of hypertension
         ui    = factor(ui))       # presence of uterine irritability

# Use BIC-based BMA
lma <- bic.glm(x, y, strict = TRUE, OR = 20,
                     glm.family="binomial",
                     factor.type=TRUE) # remove all levels of a factor
```

```
##
## Call:
## bic.glm.data.frame(x = x, y = y, glm.family = "binomial", strict = TRUE,     OR = 20, factor.type = TRUE)
##
##
##   4  models were selected
##  Best  4  models (cumulative posterior probability =  1 ):
##
##             p!=0   EV       SD          model 1     model 2     model 3     model 4
## Intercept   100    0.76092  1.230e+00     1.45068     0.99831    -1.05711    -0.79000
## age          0.0   0.00000  0.000e+00        .           .           .           .
## lwt         74.4  -0.01306  9.639e-03    -0.01865    -0.01406        .           .
## race         0.0
##     .2            0.00000  0.000e+00        .           .           .           .
##     .3            0.00000  0.000e+00        .           .           .           .
## smoke        0.0
##      .1           0.00000  0.000e+00        .           .           .           .
## ptl         13.3
##    .1            0.23225  6.179e-01        .           .         1.75026       .
##    .2            0.08647  4.047e-01        .           .         0.65165       .
##    .3           -1.79255  3.216e+02        .           .       -13.50896       .
## ht          56.6
##   .TRUE          1.04938  1.060e+00     1.85551        .           .           .
## ui           0.0
##   .1            0.00000  0.000e+00        .           .           .           .
## ftv          0.0   0.00000  0.000e+00        .           .           .           .
##
## nVar                                         2           1           1           0
## BIC                                   -753.82285  -751.51602  -750.92334  -750.77644
## post prob                                0.566       0.178       0.133       0.123
```


Models selected by BMA
age
lwt
race.2
race.3
smoke.1
ptl.1
ptl.2
ptl.3
ht.TRUE
ui.1
ftv
1
2
3
4
Model #

Predictions

```
npkBMA = bicreg( x = npk[, c("block","N","P","K")], y = npk$yield)
summary(npkBMA)
```

```
##
## Call:
## bicreg(x = npk[, c("block", "N", "P", "K")], y = npk$yield)
##
##
##   31  models were selected
##  Best  5  models (cumulative posterior probability =  0.4714 ):
##
##             p!=0    EV      SD      model 1  model 2  model 3  model 4  model 5
## Intercept  100.0  53.5651  2.6232   55.125   50.742   54.371   56.333   55.717
## block2      47.0   2.0923  2.9592      .      5.892    2.262       .        .
## block3      88.6   5.9034  3.4860    4.833    9.217    5.588       .      4.833
## block4      67.0  -3.5910  3.3608   -5.817       .     -5.063   -7.025   -5.817
## block5      60.9  -3.1264  3.2565   -5.417       .     -4.663   -6.625   -5.417
## block6      34.4   1.2048  2.4195      .      4.792       .        .        .
## N1         100.0   5.6167  1.6870    5.617    5.617    5.617    5.617    5.617
## P1          15.6  -0.1845  0.7817      .        .        .        .      -1.183
## K1          91.4  -3.6402  1.9466   -3.983   -3.983   -3.983   -3.983   -3.983
##
## nVar                                    5        5        6        4        6
## r2                                    0.688    0.674    0.704    0.608    0.698
## BIC                                 -12.097  -11.034  -10.150   -9.792   -9.669
## post prob                             0.183    0.107    0.069    0.058    0.054
```

Predictions

```
p = predict( npkBMA, newdata = npk, quantiles = c(.5,.05,.95))
bind_cols(npk, as.data.frame(p$quantiles)) %>% head(20)
```

```
##    block N P K yield      0.5     0.05     0.95
## 1      1 0 1 1  49.5 49.74304 35.25090 64.22129
## 2      1 1 1 0  62.8 59.00702 47.68398 70.27740
## 3      1 0 0 0  46.8 53.57471 42.52489 64.57243
## 4      1 1 0 1  57.0 55.54665 45.20892 65.85694
## 5      2 1 0 0  59.8 61.28192 48.27810 74.24343
## 6      2 1 1 1  58.5 57.45323 46.73372 68.15231
## 7      2 0 0 1  55.5 52.02269 41.10688 62.90898
## 8      2 0 1 0  56.0 55.48394 45.50012 65.40843
## 9      3 0 1 0  62.8 57.00087 45.92406 68.01996
## 10     3 1 1 1  55.8 58.97088 46.29370 71.62528
## 11     3 1 0 0  69.5 62.79647 44.75176 80.81328
## 12     3 0 0 1  55.0 53.54233 42.10282 64.93923
## 13     4 1 0 0  62.0 58.34444 47.17278 69.50355
## 14     4 1 1 1  48.8 54.52196 41.61422 67.40531
## 15     4 0 0 1  45.5 49.08786 33.07165 65.08997
## 16     4 0 1 0  44.2 52.54414 39.94970 65.12154
## 17     5 1 1 0  52.0 58.31288 46.50321 70.11675
## 18     5 0 0 0  51.5 52.87987 41.26720 64.49108
## 19     5 1 0 1  49.8 54.85758 44.09082 65.61609
## 20     5 0 1 1  48.8 49.05549 35.62485 62.48318
```

MCMC for sampling θ and M

Suppose, you construct a Markov chain to sample jointly from $p(M_\gamma, \theta_\gamma|y)$. An issue here is that when you move $M_\gamma \rightarrow M_{\gamma'}$, there is a chance that you change the dimension of θ, i.e. $\sum_{i=1}^{p} \gamma_i \neq \sum_{i=1}^{p} \gamma_i'$. This can be done via Metropolis-Hastings where the change of dimension is taken into account and this approach is called reversible jump MCMC.

An alternative is to fully incorporate γ as a parameter in the model. For example,

$$\begin{aligned} y_{ij} &\overset{ind}{\sim} N(\gamma_i\theta_i, \sigma^2) \\ \theta_i &\overset{ind}{\sim} N(\mu, \tau) \\ \gamma_i &\overset{ind}{\sim} Ber(w_i) \end{aligned}$$

This is essentially a way to implement the point-mass prior.

MCMC for Model averaging for GLMs

We can implement a similar MCMC to perform model averaging in Bayesian GLMs. Let $\theta_i = E[y_i|\theta_i]$and ϕ as a dispersion parameter, then we can define a GLM as

$$\begin{aligned} y_i &\sim p(y_i|\theta_i, \psi) \\ \theta_i &= g^{-1}(X_i'\beta) \\ \beta_j &= \gamma_j \phi_j \\ \phi_j &\overset{ind}{\sim} N(\mu, \tau) \\ \gamma_j &\overset{ind}{\sim} Ber(w_j) \end{aligned}$$

For probit and ordinal regression, we can augment the model with parameters ζ_i, e.g. for probit regression

$$y_i = \mathrm{I}(\zeta_i > 0) \text{ and } \zeta_i \overset{ind}{\sim} N(\theta_i, \psi).$$

There is a similar augmentation for logistic regression, see the `BayesLogit` and references therein. For these models and linear regression, we can construct an MCMC entirely using Gibbs steps. For other models, e.g. Poisson regression, sampling ϕ_j results in a non-Gibbs step.

BMS

```
library(BMS) # Bayesian model sampling
data(datafls)
dim(datafls)
```

```
## [1] 72 42
```

```
bma1 = bms(datafls,
           burn=1000,
           iter=2000,
           g="EBL",           # Local empirical Bayes
           mprior="uniform", # model over priors (extremely flexible)
           user.int = interactive())
```

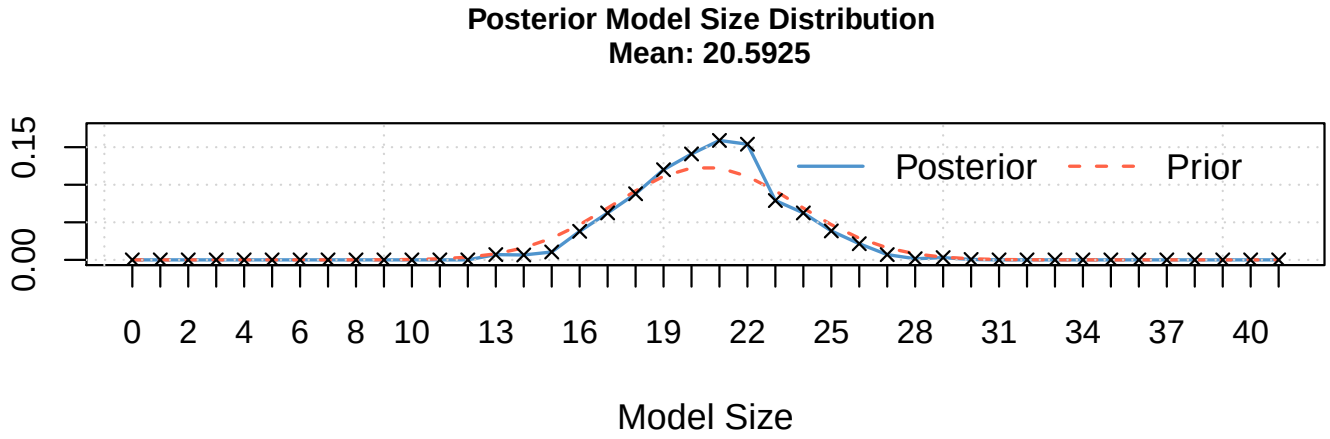
```
print(bma1)
```

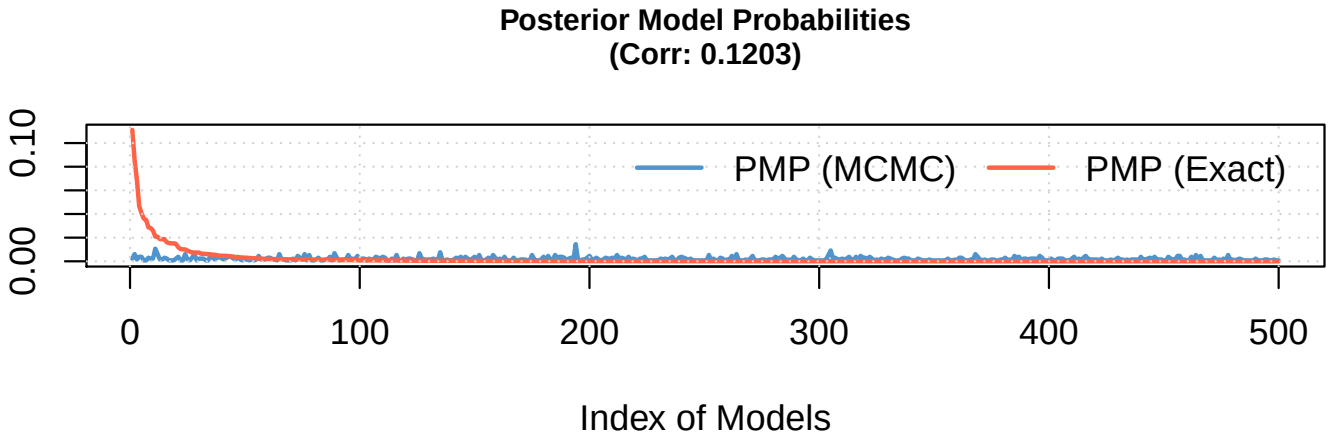
```
##                PIP     Post Mean      Post SD Cond.Pos.Sign Idx
## SubSahara   1.0000 -2.007590e-02 5.412787e-03    0.00000000   7
## LifeExp     1.0000  8.498348e-04 2.545923e-04    1.00000000  11
## GDP60       1.0000 -1.627958e-02 3.222063e-03    0.00000000  12
## Confucian   1.0000  5.694173e-02 1.335290e-02    1.00000000  19
## Mining      0.9400  3.516006e-02 1.755775e-02    1.00000000  13
## NequipInv   0.8870  4.717581e-02 2.659100e-02    1.00000000  39
## Hindu       0.8565 -5.838992e-02 3.780397e-02    0.00000000  21
## EcoOrg      0.8460  1.770278e-03 1.118150e-03    1.00000000  14
## EquipInv    0.8405  1.018319e-01 6.251474e-02    1.00000000  38
## LatAmerica  0.8270 -9.399873e-03 6.887528e-03    0.00000000   6
## LabForce    0.7190  1.811121e-07 1.524010e-07    0.96036161  29
## EthnoL      0.6855  5.864931e-03 5.869203e-03    1.00000000  20
## RuleofLaw   0.6450  7.517609e-03 7.157062e-03    1.00000000  26
## Protestants 0.6210 -7.180561e-03 7.059607e-03    0.00000000  25
## Spanish     0.5420  5.548817e-03 6.842165e-03    0.95571956   2
## HighEnroll  0.5065 -4.509488e-02 5.324049e-02    0.00000000  30
## BlMktPm     0.5055 -3.386223e-03 4.247989e-03    0.00000000  41
## PolRights   0.4925 -7.781644e-04 1.271603e-03    0.07005076  33
## WarDummy    0.4600 -1.625100e-03 2.456650e-03    0.00000000   5
## Brit        0.4525  1.102050e-03 3.443830e-03    0.69281768   4
## French      0.4410  3.377976e-03 5.020818e-03    0.93424036   3
## Age         0.4060 -1.589293e-05 2.679064e-05    0.00000000  16
## Popg        0.3725  3.141678e-02 1.283215e-01    0.82818792  27
## Buddha      0.3605  2.907252e-03 5.426750e-03    0.99861304  17
## CivlLib     0.3385 -4.999110e-04 1.342356e-03    0.14180207  34
## PrExports   0.3260 -1.707068e-03 4.309557e-03    0.11963190  24
## Catholic    0.3145 -1.687446e-03 3.760720e-03    0.06677266  18
## Foreign     0.3110  5.935380e-04 2.171490e-03    0.81189711  36
## PublEdupct  0.2915  4.197105e-02 9.976899e-02    0.95540309  31
## PrScEnroll  0.2730  2.909505e-03 6.842505e-03    0.95421245  10
## Abslat      0.2680 -1.067199e-05 7.014253e-05    0.34514925   1
## REEXDist    0.2635 -6.564200e-06 2.067941e-05    0.14231499  37
```

```
summary(bma1)
```

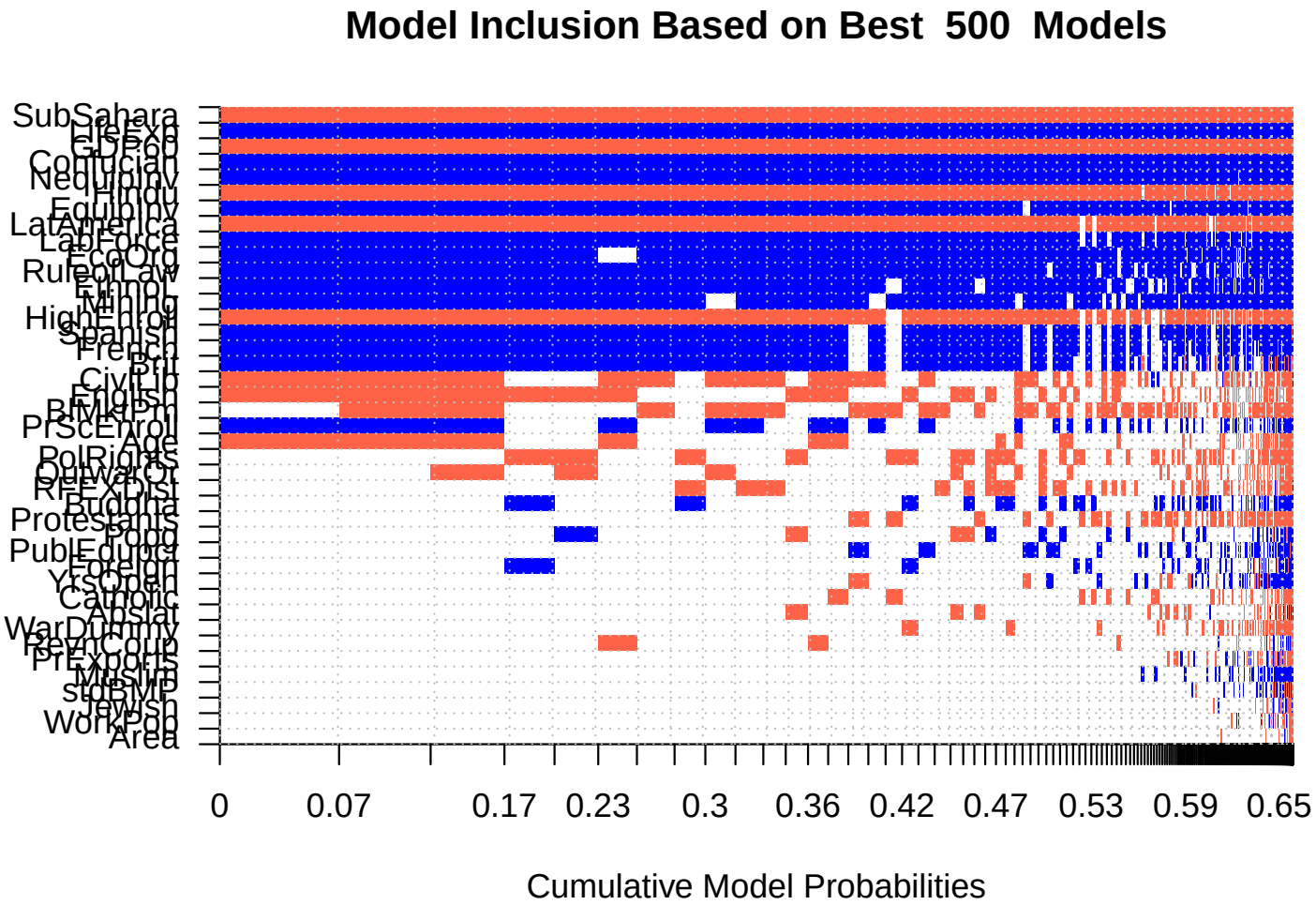
```
## Mean no. regressors              Draws             Burnins                Time  No. models visited
##           "20.5925"             "2000"              "1000"  "0.7215638 secs"              "1260"
##      Modelspace 2^K          % visited       % Topmodels            Corr PMP           No. Obs.
##           "2.2e+12"          "5.7e-08"                "66"            "0.1203"              "72"
##         Model Prior            g-Prior     Shrinkage-Stats
##    "uniform / 20.5"              "EBL"         "Av=0.9598"
```

```
plot(bma1)
```

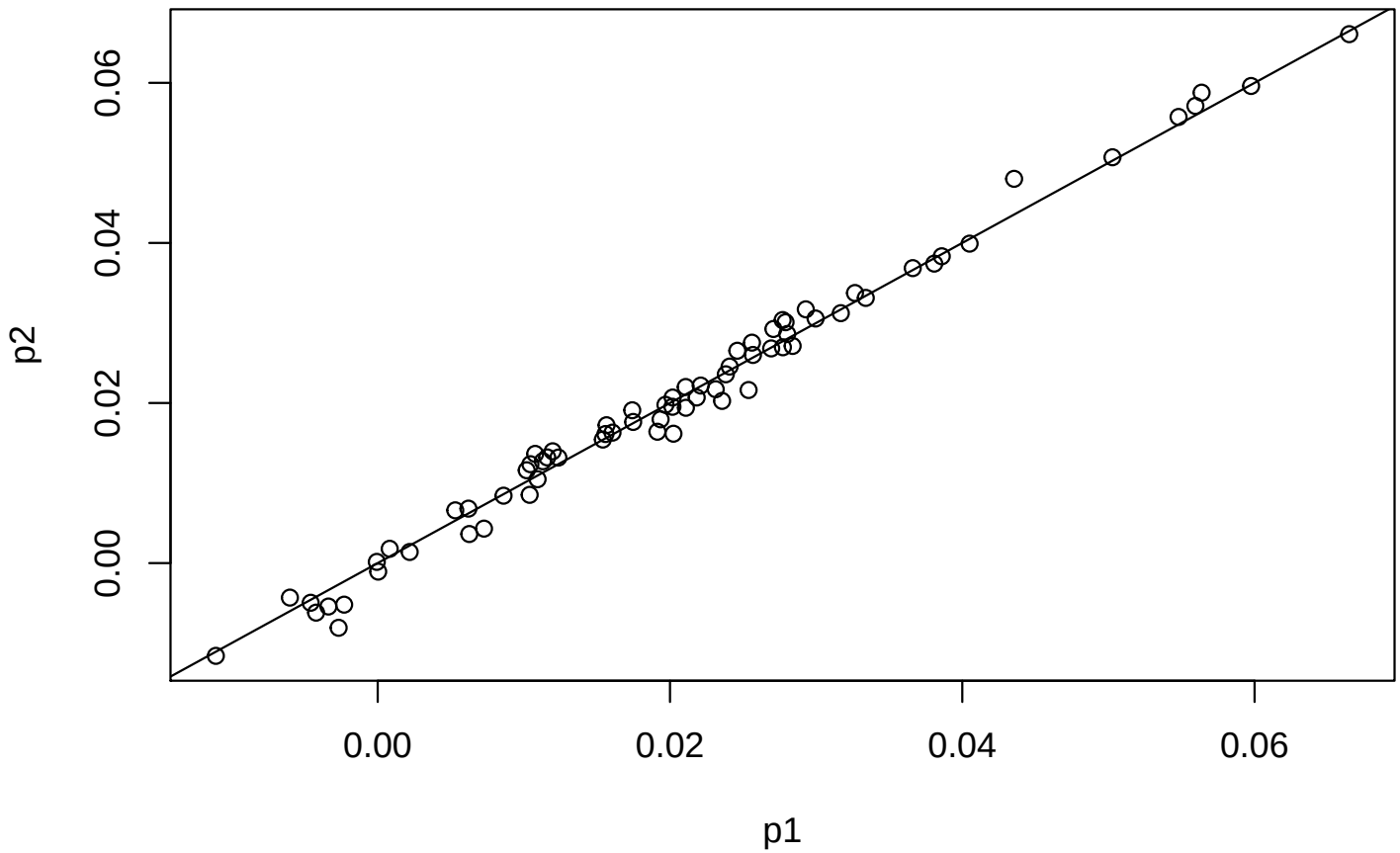

Posterior Model Size Distribution
Mean: 20.5925
0.00
0.15
Posterior
Prior
0 2 4 6 8 10 13 16 19 22 25 28 31 34 37 40
Model Size


Posterior Model Probabilities
(Corr: 0.1203)
0.00
0.10
PMP (MCMC)
PMP (Exact)
0
100
200
300
400
500
Index of Models

```
image(bma1)
```


Model Inclusion Based on Best 500 Models
SubSahara
Area
0
0.07
0.17
0.23
0.3
0.36
0.42
0.47
0.53
0.59
0.65
Cumulative Model Probabilities

```r
p1 = predict(bma1)              # fitted values based on MCMM frequencies
p2 = predict(bma1, exact=TRUE) # fitted values based on best models
plot(p1,p2); abline(0,1)
```



Model averaging of regression coefficients

```
set.seed(20171007)
n <- 100
x1 <- rnorm(n)
y <- x1 + rnorm(n)
x2 <- x1 + rnorm(n,0,.01)

library(BMA)
m <- bicreg(cbind(x1,x2), y)

summary(m)
```

```
##
## Call:
## bicreg(x = cbind(x1, x2), y = y)
##
##
##   3  models were selected
##  Best  3  models (cumulative posterior probability =  1 ):
##
##             p!=0    EV       SD       model 1    model 2    model 3
## Intercept  100.0   0.09687  0.08774    0.09686    0.09646    0.09980
## x1          51.6  -0.12069  3.07643       .       0.78988   -7.23838
## x2          54.9   0.91107  3.07714    0.79065       .       8.03022
##
## nVar                                      1          1          2
## r2                                      0.480      0.479      0.483
## BIC                                   -60.81432  -60.67277  -56.82626
## post prob                               0.484      0.451      0.066
```