Bayesian model averaging

Dr. Jarad Niemi

Iowa State University

September 4, 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

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

and

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

where 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

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

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

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

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

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 j 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)

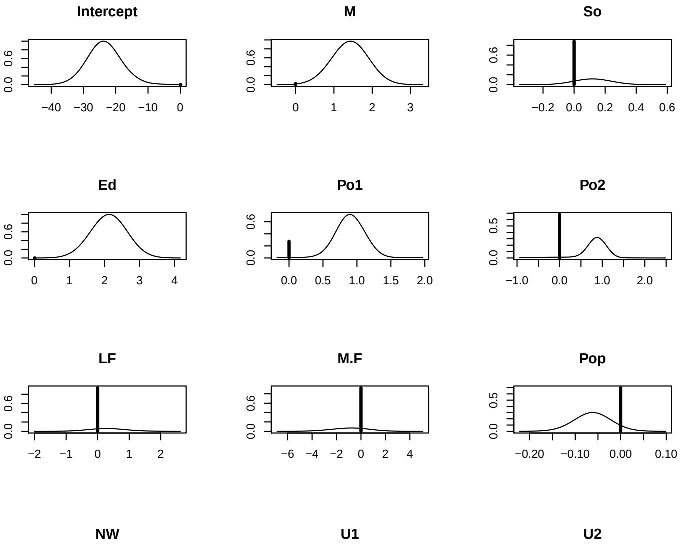
BMA in R

```
library(BMA)
library(MASS); data(UScrime)
x = UScrime[,-16]
y = log(UScrime[,16])
x[,-2] = log(x[,-2])
lma = bicreg(x, y,
             strict = FALSE, # include models the more probability submodel?
             OR = 20)        # cutoff for including models
```

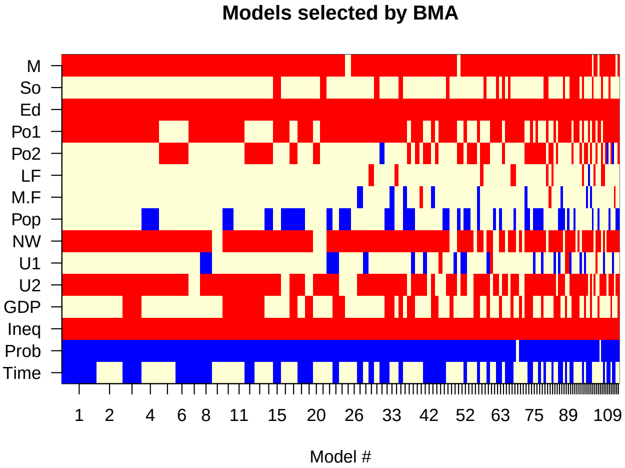
```
summary(lma)

##
## Call:
## bicreg(x = x, y = y, strict = FALSE, OR = 20)
##
##
##   115  models were selected
##  Best  5  models (cumulative posterior probability =  0.2039 ):
##
##            p!=0    EV        SD        model 1    model 2    model 3    model 4    model 5
## Intercept  100.0  -23.45301  5.58897  -22.63715  -24.38362  -25.94554  -22.80644  -24.50477
## M           97.3    1.38103  0.53531    1.47803    1.51437    1.60455    1.26830    1.46061
## So          11.7    0.01398  0.05640       .          .          .          .          .
## Ed         100.0    2.12101  0.52527    2.22117    2.38935    1.99973    2.17788    2.39875
## Po1         72.2    0.64849  0.46544    0.85244    0.91047    0.73577    0.98597       .
## Po2         32.0    0.24735  0.43829       .          .          .          .        0.90689
## LF           6.0    0.01834  0.16242       .          .          .          .          .
## M.F          7.0   -0.06285  0.46566       .          .          .          .          .
## Pop         30.1   -0.01862  0.03626       .          .          .       -0.05685       .
## NW          88.0    0.08894  0.05089    0.10888    0.08456    0.11191    0.09745    0.08534
## U1          15.1   -0.03282  0.14586       .          .          .          .          .
## U2          80.7    0.26761  0.19882    0.28874    0.32169    0.27422    0.28054    0.32977
## GDP         31.9    0.18726  0.34986       .          .        0.54105       .          .
## Ineq       100.0    1.38180  0.33460    1.23775    1.23088    1.41942    1.32157    1.29370
## Prob        99.2   -0.24962  0.09999   -0.31040   -0.19062   -0.29989   -0.21636   -0.20614
## Time        43.7   -0.12463  0.17627   -0.28659       .       -0.29682       .          .
##
## nVar                                      8          7          9          8          7
## r2                                      0.842      0.826      0.851      0.838      0.823
## BIC                                   -55.91243  -55.36499  -54.69225  -54.60434  -54.40788
## post prob                               0.062      0.047      0.034      0.032      0.029
```

Does this make any sense?

```
plot(lma)
```

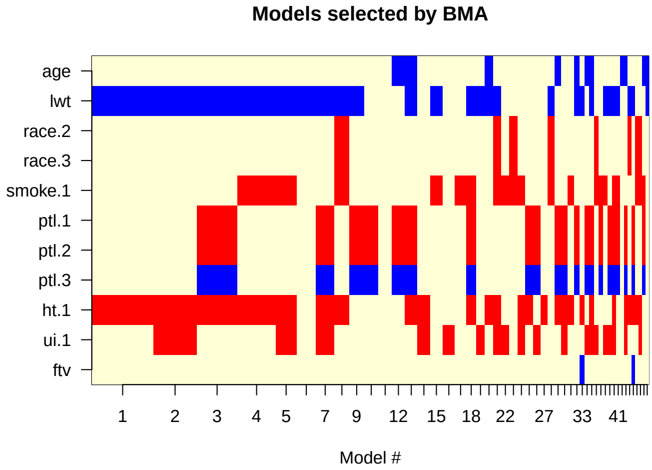


```r
imageplot.bma(lma)
```



BMA in R

```r
data(birthwt)
y<- birthwt$lo # 1 indicates low birthweight
x<- data.frame(birthwt[,-1])
x$race<- as.factor(x$race)
x$ht<- (x$ht>=1)+0
x<- x[,-9]
x$smoke <- as.factor(x$smoke)
x$ptl<- as.factor(x$ptl)
x$ht <- as.factor(x$ht) # history of hypertension
x$ui <- as.factor(x$ui)

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


Models selected by BMA
age
lwt
race.2
race.3
smoke.1
ptl.1
ptl.2
ptl.3
ht.1
ui.1
ftv
1
2
3
4
5
7
9
12
15
18
22
27
33
41
Model #

Predictions

```
npkBMA = bicreg( x = npk[, c("block","N","K")], y=npk$yield)
p = predict( npkBMA, newdata = npk)
head(p$mean)
```

```
##        1        2        3        4        5        6
## 49.84128 59.03477 53.41810 55.45794 61.11086 57.53403
```

```
head(p$sd)
```

```
##        1        2        3        4        5        6
## 8.339862 8.031185 6.983813 6.715649 8.676357 6.384029
```

```
head(p$quantiles)
```

```
##        0.1      0.5      0.9
## 1 38.86070 49.84251 60.82005
## 2 48.43992 59.04048 69.62193
## 3 44.12041 53.42561 62.70556
## 4 46.48933 55.45993 64.42350
## 5 49.85010 61.11715 72.36342
## 6 49.18471 57.53947 65.87621
```

BMS

```
library(BMS)
data(datafls)
dim(datafls)
```

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

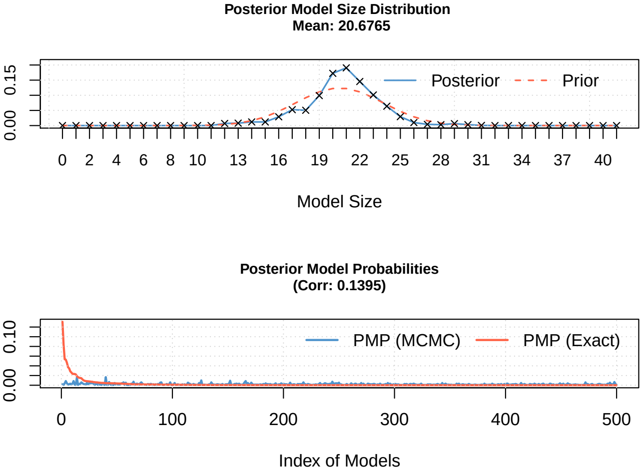
```
bma1 = bms(datafls,
           burn=1000,
           iter=2000,
           g="EBL",          # Local empirical Bayes
           mprior="uniform", # model over priors (extremely flexible)
           user.int = interactive())
```

```
print(bma1)
```

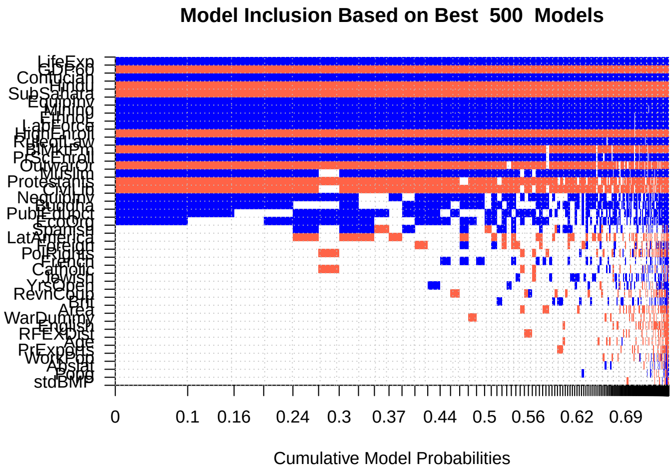
```
##                PIP     Post Mean      Post SD Cond.Pos.Sign Idx
## LifeExp     1.0000  8.547023e-04 2.557000e-04    1.00000000  11
## GDP60       1.0000 -1.590056e-02 3.075448e-03    0.00000000  12
## Confucian   1.0000  6.651576e-02 1.421762e-02    1.00000000  19
## Hindu       0.9465 -6.104396e-02 3.286575e-02    0.00264131  21
## SubSahara   0.9405 -1.529358e-02 6.480905e-03    0.00000000   7
## EquipInv    0.8820  1.228659e-01 6.128307e-02    1.00000000  38
## Mining      0.8415  3.387366e-02 1.940601e-02    1.00000000  13
## BlMktPm     0.8160 -6.648214e-03 4.477110e-03    0.00000000  41
## EthnoL      0.7940  1.022020e-02 7.067890e-03    1.00000000  20
## RuleofLaw   0.7840  9.561194e-03 6.881912e-03    1.00000000  26
## LabForce    0.7795  2.125252e-07 1.407974e-07    0.99871713  29
## HighEnroll  0.7750 -6.866858e-02 5.135660e-02    0.00129032  30
## Muslim      0.7640  9.790710e-03 7.254388e-03    1.00000000  23
## Protestants 0.6925 -6.233197e-03 5.927200e-03    0.00000000  25
## PrScEnroll  0.6905  1.372672e-02 1.200714e-02    0.98841419  10
## NequipInv   0.6555  2.944638e-02 2.950377e-02    1.00000000  39
## EcoOrg      0.5275  1.014045e-03 1.215330e-03    1.00000000  14
## CivlLib     0.5225 -1.124123e-03 1.542876e-03    0.03157895  34
## PublEdupct  0.4465  7.885392e-02 1.186366e-01    0.99552072  31
## OutwarOr    0.4430 -1.386622e-03 2.076290e-03    0.00564334   8
## PolRights   0.4085 -4.332674e-04 1.127484e-03    0.12974296  33
## Buddha      0.4075  3.919231e-03 5.983174e-03    1.00000000  17
## LatAmerica  0.3555 -3.263911e-03 5.650404e-03    0.00562588   6
## Foreign     0.3465 -1.127672e-04 1.984180e-03    0.43722944  36
## English     0.3330 -2.190998e-03 4.224402e-03    0.00000000  35
## Jewish      0.2850 -5.066562e-04 6.305646e-03    0.49824561  22
## RevnCoup    0.2815  7.681598e-05 2.658143e-03    0.48134991  32
## YrsOpen     0.2785  1.319433e-03 3.843790e-03    0.83842011  15
## Spanish     0.2665  1.606546e-03 3.938406e-03    0.93621013   2
## French      0.2530  1.118173e-03 2.815441e-03    0.96640316   3
## WorkPop     0.2445 -7.270927e-04 4.118719e-03    0.25971370  28
## Area        0.2410 -6.734922e-08 3.387692e-07    0.18257261   9
```

```
summary(bma1)

## Mean no. regressors               Draws             Burnins                Time  No. models visited
##           "20.6765"              "2000"              "1000"   "0.7011709 secs"              "1190"
##      Modelspace 2^K           % visited         % Topmodels            Corr PMP            No. Obs.
##           "2.2e+12"           "5.4e-08"                "75"            "0.1395"                "72"
##         Model Prior             g-Prior     Shrinkage-Stats
##    "uniform / 20.5"               "EBL"         "Av=0.9607"
```

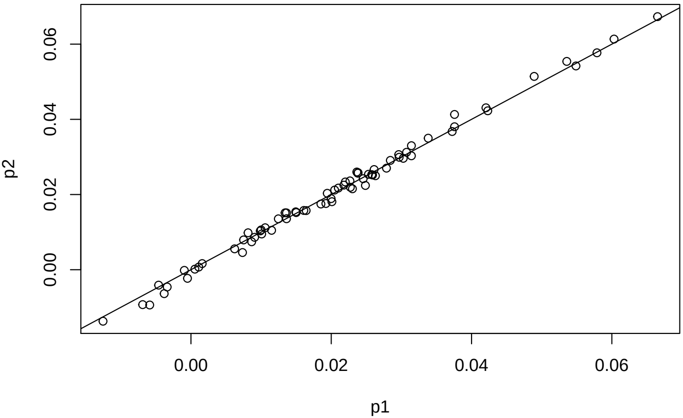
```
plot(bma1)
```


Posterior Model Size Distribution
Mean: 20.6765
Posterior
Prior
0.00
0.15
0 2 4 6 8 10 13 16 19 22 25 28 31 34 37 40
Model Size
Posterior Model Probabilities
(Corr: 0.1395)
PMP (MCMC)
PMP (Exact)
0.00
0.10
0 100 200 300 400 500
Index of Models

```r
image(bma1)
```


Model Inclusion Based on Best 500 Models
LifeExp
GDP60
Confucian
Hindu
SubSahara
Equipment
Mining
EthnoL
LabForce
HighEnroll
RuleofLaw
BlMktPm
PrScEnroll
OutwarOr
Muslim
Protestants
CivLib
NequipInv
Buddha
PublEdupct
EcoOrg
Spanish
LatAmerica
Foreign
PolRights
French
Catholic
Jewish
YrsOpen
RevnCoup
Brit
Area
WarDummy
English
RFEXDist
Age
PrExports
WorkPop
Abslat
Popg
stdBMP
0
0.1
0.16
0.24
0.3
0.37
0.44
0.5
0.56
0.62
0.69
Cumulative Model Probabilities

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



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 $M_\gamma \to M_{\gamma'}$, 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,

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

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

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

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

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

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.