Multiclass Sparse Bayesian Regression for fMRI-Based Prediction

Vincent Michel, Evelyn Eger, Christine Keribin, Bertrand Thirion

Multi-Class Sparse Bayesian Regression for *fMRI*-based prediction

Vincent Michel[1,2,5], Evelyn Eger[3,5], Christine Keribin[2,4], and Bertrand Thirion[1,5]

[1] Parietal team,INRIA Saclay-Île-de-France, Saclay, France ,
[2] Université Paris-Sud 11, Orsay, France
[3] INSERM U562, Gif/Yvette, France
[4] Select team, INRIA Saclay-Île-de-France, France
[5] CEA, DSV, I2BM, Neurospin, Gif/Yvette, France

Abstract. *Inverse inference* has recently become a popular approach for the analysis of neuroimaging data, that quantifies the amount of information contained in brain images on perceptual, cognitive and behavioral parameters. As it outlines regions of the brain that convey information for accurately predicting the parameter of interest, it is also used to understand how the corresponding information is encoded in the brain. However, it relies on a prediction function that is plagued by the curse of dimensionality, as there are far more features (voxels) than samples (images). Dimension reduction is thus a mandatory step to extract relevant information from the whole set of features. Among different approaches, regularized regression/classification perform jointly the selection of relevant features and the learning of the associated parameters. Unlike classical alternatives, Bayesian regularization further adapts the amount of regularization to the available data. We introduce in this paper an new model, *Multi-Class Sparse Bayesian Regression* (*MCBR*), that is a generalization of classical Bayesian approaches. MCBR consists in grouping the features into several classes, and then to regularize each class differently in order to apply an adaptive and efficient regularization. We detail this framework and validate our algorithm on simulated and real neuroimaging data sets, showing that it performs better than reference methods while yielding interpretable clusters of features.

1 Introduction

In the context of neuroimaging, machine learning approaches have been used so far to address diagnostic problems, where patients were classified into different groups based on anatomical or functional data. By contrast, in cognitive studies, the standard framework for functional or anatomical brain mapping was based on mass univariate inference procedures [1]. Recently, a new way of analyzing functional neuroimaging data has emerged [2,3], that consists in assessing how well behavioral information or cognitive states can be predicted from brain activation images such as those obtained with functional Magnetic Resonance Imaging (fMRI). This approach opens new ways to understanding the mental representation of various perceptual and cognitive parameters, which can be regarded as the study of the corresponding *neural code*, albeit at a relatively low spatial resolution. The accuracy of the prediction of the behavioral or

cognitive target variable, as well as the spatial layout of predictive regions, can provide valuable information about functional brain organization; in short, it helps to *decode* the brain system [4].

Many different pattern recognition and machine leaning methods have been used to extract information from brain images and compare it to the corresponding target. Among them, LDA [3,5], SVM [6,7,8,9], or regularized prediction [10,11] are particularly used. The major bottleneck in this kind of analytical framework is that there are far more features than samples, so that the problem is plagued by the curse of dimensionality, that leads to overfitting. Dimension reduction can be used to extract relevant information from the data. The standard approach in functional neuroimaging is feature selection (*e.g. Anova*) [3,6,12,11]. However, this amounts to performing feature selection and parameter estimation separately, which is not optimal; a popular combined selection/estimation scheme, such as *Recursive Feature Elimination* [13] relies on a specific heuristic, that does not guarantee the optimality of the solution, and is particularly costly. By contrast, there is great interest in sparsity inducing regularizations, that optimize both simultaneously.

In this paper, we assume that the code under investigation is about some scalar parameter that characterizes the stimuli, such as a scale/shape parameters, but possibly also position, speed (assuming a 1-D space) or cardinality. We focus thus on regression problems, and defer the generalization to classification to future work. Let us introduce the following predictive linear model:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + b \tag{1}$$

where $\mathbf{y}$ represents the behavioral variable and $(\mathbf{w}, b)$ are the parameters to be estimated on a training set. A vector $\mathbf{w} \in \mathbb{R}^p$ can be seen as an image; p is the number of features (or voxels) and $b \in \mathbb{R}$ is called the *intercept*. The matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the design matrix. Each row is a p-dimensional sample, *i.e.*, an activation map related to the observation. With $n \ll p$, the estimation of $\mathbf{w}$ is ill-posed.

To cope with the high dimensionality of the data, one can penalize the estimation of $\mathbf{w}$, *e.g.* based on the ℓ_2 norm of the weights. Classical regularization schemes have been used in functional neuroimaging, such as Ridge regression [14], Lasso [15] or elastic net regression [16]. However, these approaches require the amount of penalization to be fixed beforehand, and possibly optimized by cross-validation. To deal with the choice of the amount of penalization, one can use Bayesian regression techniques, that include the estimation of regularization parameters in the whole estimation procedure. Standard Bayesian regularization schemes are based on the fact that a penalization by weighted ℓ_2 norm is equivalent to setting Gaussian priors on the weights :

$$\mathbf{w} \sim \mathcal{N}(0, A^{-1}),\ A = \text{diag}(\alpha_1, ..., \alpha_p) \text{ and } \forall i \in [1, \ldots, p]\,, \alpha_i \in \mathbb{R}^+ \tag{2}$$

The model in Eq. 2 defines two classical Bayesian regression schemes. The first one is *Bayesian Ridge Regression* (*BRR*) [17], that corresponds to the particular case $\alpha_1 = ... = \alpha_m$. By regularizing all the features identically, BRR is not well suited when only few features are relevant. The second classical scheme is *Automatic Relevance Determination* (*ARD*) [18], that corresponds to the case $\alpha_i \neq \alpha_j$ if $i \neq j$. The regularization performed by ARD is very adaptive, as all the weights are regularized differently.

However, by regularizing separately each feature, ARD is prone to underfitting when the model contains too many regressors [19], and also suffers from convergence issues [20].

These classical Bayesian regularizations schemes have been used in *fMRI* inverse inference studies [10,21,14]. However, these studies only used sparsity as built-in feature selection, and do not consider neuroscientific assumptions for improving the regularization (*i.e.* within the design of the matrix A). Indeed, due to the intrinsic smoothness of functional neuroimaging data [**?**], predictive information is rather encoded in different groups of features sharing similar information. A potentially more adapted approach is the Bayesian regression scheme presented in [22], that regularizes patterns of voxels differently. However, this approach relies on ad hoc voxel selection steps, so that there is no proof that the solution is correct.

In this paper, we detail a model for Bayesian regression in which the features are grouped into Q different classes that are subject to different regularization penalties. The estimation of the penalty is performed in each class separately, leading to a stable and adaptive regularization. The construction of the group of features, and the estimation of the predictive function, are performed jointly. This approach, called *Multi-Class Sparse Bayesian Regression (MCBR)*, is thus an intermediate solution between BRR and ARD. It requires less parameters to estimate than ARD, and is far more adaptive than BRR. Another asset of the proposed approach in fMRI inverse inference, is that it creates a clustering of the features, and thus yields useful maps for brain mapping. After introducing our model and giving some details on the parameter estimation algorithms (Variational Bayes or Gibbs sampling procedures), we show that the proposed algorithm yields better accuracy than reference methods, while providing more interpretable models.

2 Multi-Class Sparse Bayesian Regression

We first detail the notations of the problem and describe the priors and parameters of the model. Then, we detail the two different algorithms used for model inference.

2.1 Model and priors

We recall the linear model for regression:

$$\mathbf{y} = f(\mathbf{X}, \mathbf{w}, b) = \mathbf{X}\,\mathbf{w} + b \ , \tag{3}$$

We denote $\mathbf{y} \in \mathbb{R}^n$ the targets to be predicted, and $\mathbf{X} \in \mathbb{R}^{n \times p}$ the set of activation images related to the presentation of different stimuli. The integer p is the number of voxels and n the number of samples (images). Typically, $p \sim 10^3$ to 10^5 (for a whole volume), while $n \sim 10$ to 10^2.

Priors on the noise We use classical priors for regression, and we model the noise on $\mathbf{y}$ as an *i.i.d.* Gaussian variable:

$$\epsilon \sim \mathcal{N}(0, \alpha^{-1}\mathbf{I_n}) \tag{4}$$

$$\alpha \sim \Gamma(\alpha; \alpha_1, \alpha_2) \tag{5}$$

where α is the precision parameter, and Γ stands for the *gamma density* with two hyper-parameters α_1, α_2:

$$\Gamma(x; \alpha_1, \alpha_2) = \alpha_2^{\alpha_1} x^{\alpha_1 - 1} \frac{\exp^{-x\alpha_2}}{\Gamma(\alpha_1)} \tag{6}$$

Priors on the class assignment In order to combine the sparsity of *ARD* with the stability of *BRR*, we introduce an intermediate representation, in which each feature j belongs to one class among Q indexed by a discrete variable z_j ($\mathbf{z} = \{z_1, \ldots, z_p\}$). All the features within a class $q \in \{1, .., Q\}$ share the same precision parameter λ_q, and we use the following prior on $\mathbf{z}$:

$$\mathbf{z} \sim \prod_{j=1}^{p} \prod_{q=1}^{Q} \pi_q^{\delta_{jq}} \tag{7}$$

where δ is *Kronecker's* δ, defined as:

$$\begin{cases} \delta_{jq} = 0 \text{ if } z_j \neq q \\ \delta_{jq} = 1 \text{ if } z_j = q \end{cases} \tag{8}$$

We finally introduce an additional Dirichlet prior on π:

$$\pi \sim Dir(\eta) \tag{9}$$

with an hyper-parameter η. By updating at each step the probability π_k of each class, it is possible to prune classes. This model has no spatial constraints, and thus is not spatially regularized.

Priors on the weights As in ARD, we make use of an independent Gaussian prior for the weights:

$$\mathbf{w} \sim \mathcal{N}(0, \mathbf{A}^{-1}) \text{ with } diag\,(\mathbf{A}) = \{\lambda_{z_1}, ..., \lambda_{z_p}\} \tag{10}$$

where λ_{z_j} is the precision parameter of the j^{th} feature, with $z_j \in \{1, \ldots, Q\}$. We introduce the following prior on λ_q:

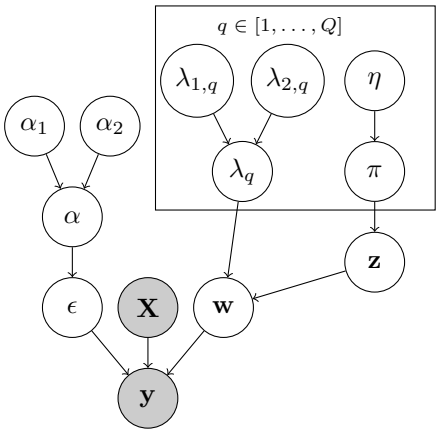
$$\lambda_q \sim \Gamma(\lambda_q; \lambda_{1,q}, \lambda_{2,q}) \tag{11}$$

with hyper-parameters $\lambda_{1,q}, \lambda_{2,q}$. The complete generative model is summarized in Fig.1.

Link with other *Bayesian regularization* schemes The link between the proposed MCBR model and the other regularization methods Bayesian Ridge Regression and Automatic Relevance Determination, is obvious:

- with $Q = 1$, *i.e.* $\lambda_{z_1} = \ldots = \lambda_{z_p}$, we retrieve the BRR model.
- with $Q = p$, *i.e.* $\lambda_{z_i} \neq \lambda_{z_j}$ if $i \neq j$, and assigning each feature to a singleton class (*i.e.* $z_j = j$), we retrieve the ARD model.

Moreover, the proposed approach is related to the one developed in [23]. In this paper, the authors proposed for the distribution of the weights of the features, a binary mixture of Gaussians with small and large precisions. This model is used for variable selection, and estimated by *Gibbs sampling*. Our work can be viewed as a generalization of this model to a number of classes $Q \geq 2$.



$$
\begin{aligned}
\mathbf{y} &= \mathbf{X}\,\mathbf{w} + \epsilon + b \\
\epsilon &\sim \mathcal{N}(0, \alpha^{-\mathbf{1}}\mathbf{I_n}) \\
\alpha &\sim \Gamma(\alpha; \alpha_1, \alpha_2) \\
\mathbf{z} &\sim \prod_{j=1}^{p}\prod_{q=1}^{Q} \pi_q^{\delta_{jq}} \\
\pi &\sim Dir(\eta) \\
\mathbf{w} &\sim \mathcal{N}(0, \mathbf{A}^{-1}) \\
&\text{with} \quad diag\,(\mathbf{A}) = \{\lambda_{z_1}, ..., \lambda_{z_p}\} \\
\lambda_q &\sim \Gamma(\lambda_q; \lambda_{1,q}, \lambda_{2,q})
\end{aligned}
$$

Fig. 1. Graphical model of *Multi-Class Sparse Bayesian Regression – MCBR*.

2.2 Model inference

For models with latent variables, such as MCBR, some singularities can exist. For instance in a mixture of components, a singularity is a component with one single sample and thus zero variance. In such cases, maximizing the *log likelihood* yields flawed solutions, and one can use the posterior distribution of the latent variables $p(\mathbf{z}|\mathbf{X}, \mathbf{y})$ for this maximization. However, the posterior distribution of the latent variables given the data has not a closed-form expression, and some specific estimation methods, such as *Variational Bayes* or *Gibbs Sampling*, have to be used.

We propose two different algorithms for inferring the parameters of the MCBR model. We first estimate the model by Variational Bayes, the resulting algorithm is thus called *VB-MCBR*. We also detail an algorithm, called *Gibbs-MCBR*, based on a Gibbs Sampling procedure.

Estimation by *Variational Bayes – VB-MCBR* The *Variational Bayes* (or *VB*) approach provides an approximation $q(\Theta)$ of $p(\Theta|\mathbf{y})$, where $q(\Theta)$ is taken in a given family of distributions, and $\Theta = [\mathbf{w}, \lambda, \alpha, \mathbf{z}, \pi]$. Additionally, the Variational Bayes approach often uses the following *mean field approximation*, that allows the factorization between the approximate distribution of the latent variables and the approximate distributions of the parameters:

$$q(\Theta) = q(\mathbf{w})q(\lambda)q(\alpha)q(\mathbf{z})q(\pi) \tag{12}$$

We introduce the *Kullback-Leibler* divergence $\mathcal{D}(q(\Theta))$ that measures the similarity between the true posterior $p(\Theta|\mathbf{y})$ and the variational approximation $q(\Theta)$. One can decompose the *marginal log-likelihood* $\log p(\mathbf{y})$ as:

$$\log p(\mathbf{y}|\Theta) = \mathcal{F}(q(\Theta)) + \mathcal{D}(q(\Theta)) \tag{13}$$

with:

$$\mathcal{F}(q(\Theta)) = \int d\Theta q(\Theta) \log \frac{p(\mathbf{y}, \Theta)}{q(\Theta)} \tag{14}$$

and:

$$\mathcal{D}(q(\Theta)) = \int d\Theta q(\Theta) \log \frac{q(\Theta)}{p(\Theta|\mathbf{y})} \tag{15}$$

where $\mathcal{F}(q(\Theta))$ is called *free energy*, and can be seen as measure of the quality of the model. As $\mathcal{D}(q(\Theta)) \geq 0$, the free energy is a lower bound on $\log p(\mathbf{y})$ with equality iff $q(\Theta) = p(\Theta|\mathbf{y})$. So, inferring the density $q(\Theta)$ of the parameters corresponds to maximizing $\mathcal{F}$, on all the free distribution $q(\Theta)$. In practice, the *VB* approach consists in maximizing the free energy $\mathcal{F}$ iteratively with respect to the approximate distribution $q(\mathbf{z})$ of the latent variables, and with respect to the approximate distributions of the parameters of the model $q(\mathbf{w})$, $q(\lambda)$, $q(\alpha)$ and $q(\pi)$.

The variational distributions and the pseudo-code of the VB-MCBR algorithm are provided in appendix A. This algorithm maximizes the free energy $\mathcal{F}$. In practice, iterations are performed until convergence to a local maximum of $\mathcal{F}$. With an ARD prior (*i.e.* $Q = p$ and fixing $z_j = j$), we retrieve the same formulas than the ones found for *Variational ARD* [18].

Estimation by *Gibbs Sampling – Gibbs-MCBR* We develop here an estimation of the model MCBR using Gibbs Sampling [24] . The resulting algorithm is called *Gibbs-MCBR*; the pseudo-code of the algorithm and the candidate distributions are provided in appendix B.

Initialization and priors on the model parameters Our model needs few hyper-parameters; we choose here to use slightly informative and class-specific hyper-parameters in order to reflect a wide range of possible behaviors for the weights distribution. This choice of priors is equivalent to setting heavy-tailed centered *Student* distributions with variance at different scales as priors on the weights parameters. We set $Q = 9$, with weakly informative priors $\lambda_{1,q} = 10^{q-4}, q \in [1, .., Q]$ and $\lambda_{2,q} = 10^{-2}\ , q \in [1, .., Q]$. Moreover, we set $\alpha_1 = \alpha_2 = 1$. Starting with a given number of classes and letting the model automatically prune the classes, can be seen as a means to avoid costly model selection procedures. The choice of class-specific priors is also useful to avoid label switching issues and thus speeds up convergence. Crucially, the priors used here can be used in any regression problem, provided that the target data is approximately scaled to the range of values used in our experiments. In that sense, the present choice of priors can be considered as *universal*. We also randomly initialize $q(\mathbf{z})$ for VB-MCBR (or $\mathbf{z}$ for Gibbs-MCBR).

2.3 Validation and model evaluation

Performance evaluation Our method is evaluated with a cross-validation procedure that splits the available data into training and validation sets. In the following, $(\mathbf{X}^l, \mathbf{y}^l)$ are a learning set, $(\mathbf{X}^t, \mathbf{y}^t)$ a test set and $\hat{\mathbf{y}}^t = F(\mathbf{X}^t\hat{\mathbf{w}})$ refers to the predicted target,

where $\hat{\mathbf{w}}$ is estimated from the training set. The performance of the different models is evaluated using ζ, the ratio of explained variance:

$$\zeta(\mathbf{y}^t, \hat{\mathbf{y}}^t) = \frac{\text{var}(\mathbf{y}^t) - \text{var}\,(\mathbf{y}^t - \hat{\mathbf{y}}^t)}{\text{var}(\mathbf{y}^t)}$$

This is the amount of variability in the response that can be explained by the model (perfect prediction yields $\zeta = 1$, while $\zeta < 0$ if prediction is worse than chance).

Competing methods In our experiments, the proposed algorithms are compared to different state of the art regularization methods:

- *Elastic net* regression [25], that requires setting two parameters λ_1 and λ_2. In our analyzes, a cross-validation procedure within the training set is used to optimize these parameters. Here, we use $\lambda_1 \in \{0.2\tilde{\lambda}, 0.1\tilde{\lambda}, 0.05\tilde{\lambda}, 0.01\tilde{\lambda}\}$, where $\tilde{\lambda} = \|\mathbf{X}^T\mathbf{y}\|_\infty$, and $\lambda_2 \in \{0.1, 0.5, 1., 10., 100.\}$. Note that λ_1 and λ_2 parametrize heterogeneous norms.
- *Support Vector Regression* (*SVR*) with a linear kernel [26], which is the reference method in neuroimaging. The C parameter is optimized by cross-validation in the range of 10^{-3} to 10^1 in multiplicative steps of 10.
- *Bayesian Ridge Regression (BRR)*, that is equivalent to MCBR with $Q = 1$ and $\lambda_1 = \lambda_2 = \alpha_1 = \alpha_2 = 10^{-6}$, *i.e.* weakly informative priors.
- *Automatic Relevance Determination (ARD)*, that is equivalent to MCBR with $Q = p$ and $\lambda_1 = \lambda_2 = \alpha_1 = \alpha_2 = 10^{-6}$, *i.e.* weakly informative priors.

All these methods are used after an *Anova*-based feature selection as this maximizes their performance. Indeed, irrelevant features and redundant information can decrease the accuracy of a predictor [27]. The optimal number of voxels is selected within the range $\{50, 100, 250, 500\}$, through a nested cross-validation within the training set. We do not select directly a threshold on p-value or cluster size, but rather a pre-defined number of features. The estimation of the parameters of the learning function is also performed using a nested cross-validation within the training set, to ensure a correct validation and an unbiased comparison of the methods. All methods are developed in *C* and used in *Python*. The implementation of elastic net is based on *coordinate descent* [28], while SVR is based on LibSVM [29]. Methods are used from *Python* via the *Scikit-learn* open source package [30].

For VB-MCBR and Gibbs-MCBR, in order to avoid a costly *internal cross-validation*, we select 500 voxels, and this selection is performed on the training set. The number of iterations used is fixed to 5000 (*burn in* of 4000 iterations) for Gibbs-MCBR and 500 for VB-MCBR. We set $Q = 9$.

3 Experiments and results

3.1 Experiments on simulated data

We now evaluate and illustrate MCBR on two different sets of simulated data.

Details on simulated regression data We first test MCBR on a simulated data set, designed for the study of ill-posed regression problem, *i.e.* $n \ll p$. Data are simulated as follows:

$$\mathbf{X} \sim \mathcal{N}(0,1) \text{ with } \epsilon \sim \mathcal{N}(0,1)$$
$$\mathbf{y} = 2\left(\mathbf{X_1} + \mathbf{X_2} - \mathbf{X_3} - \mathbf{X_4}\right) + 0.5\left(\mathbf{X_5} + \mathbf{X_6} - \mathbf{X_7} - \mathbf{X_8}\right) + \epsilon$$

We have $p = 200$ features, $n^l = 50$ images for the training set and $n^t = 50$ images for the test set. We compare MCBR to the reference methods, but we do not use feature selection, as the number of features is not very high.

Results on simulated regression data We average the results of 15 different trials, and the average explained variance is shown Tab. 1. Gibbs-MCBR outperforms the other approaches, yielding higher prediction accuracy than the reference elastic net and ARD methods. The prediction accuracy is also more stable than the other methods. VB-MCBR falls into local maximum of $\mathcal{F}$ and does not yield an accurate prediction.

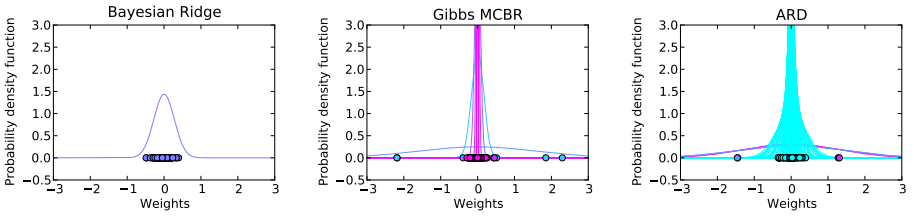
In Fig. 2, we represent the probability density function of the distributions of the weights obtained with BRR (a), Gibbs-MCBR (b) and ARD (c). With BRR, the weights are grouped in a mono-modal density. ARD is far more adaptive and sets lots of weights to zero. The Gibbs-MCBR algorithm creates a multi-modal distribution, lots of weights being highly regularized (pink distributions), and the informative features are allowed to have higher weights (blue distributions).

With MCBR, the weights are clustered into different groups, depending on their predictive power, which is interesting in application such as fMRI inverse inference, as it can yields more interpretable models. Indeed, the class where the features with higher weights ($\{\mathbf{X_1},\ \mathbf{X_2},\ \mathbf{X_3},\ \mathbf{X_4}\}$) belong to, is small (average size of 6 features) but has a high *purity* (percentage of relevant features in the class) of 74%.

Comparison VB-MCBR and Gibbs-MCBR We now look at the values of w_1 and w_2 for the different steps of the two algorithms (see Fig. 3). We can see that VB-MCBR (b) quickly falls into a local maximum, while Gibbs-MCBR (a) visits the space and reaches the region of the correct set of parameters (red dot). VB-MCBR is not optimal in this case.

Methods	mean ζ	std ζ	p-value to Gibbs-MCBR
SVR	0.11	0.1	0.0 **
Elastic net	0.77	0.11	0.0004 **
BRR	0.19	0.14	0.0 **
ARD	0.79	0.06	0.0 **
Gibbs-MCBR	0.89	0.04	-
VB-MCBR	0.04	0.05	0.0 **

Table 1. *Simulated regression data*. Explained variance ζ for different methods (average of 15 different trials). The p-values are computed using a paired t-test.



(a) BRR *(b) Gibbs-MCBR* *(b) ARD*

Fig. 2. Results on simulated regression data. Probability density function of the weights distributions obtained with BRR (a), Gibbs-MCBR (b) and ARD (c). Each color represents a different component of the mixture model.

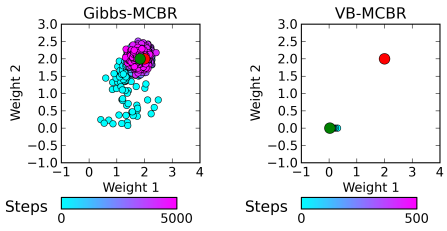


Fig. 3. Results on simulated regression data. Weights of the first two features found for the different steps of Gibbs-MCBR (a) and VB-MCBR (b). The red dot represents the ground truth of both weights, and the green dot represents the final state found by the two algorithms. VB-MCBR is stuck in a local maximum, and Gibbs-MCBR finds the correct weights.

3.2 Simulated neuroimaging data

Details on simulated neuroimaging data The simulated data set $\mathbf{X}$ consists of $n = 100$ images (size $12\times12\times12$ voxels) with a set of four square Regions of Interest (ROIs) (size $2\times2\times2$). We call $\mathcal{R}$ the support of the ROIs (*i.e.* the 32 resulting voxels of interest). Each of the four ROIs has a fixed weight in $\{-0.5, 0.5, -0.5, 0.5\}$. We call $w_{i,j,k}$ the weight of the (i, j, k) voxel. The resulting images are smoothed with a Gaussian kernel with a standard deviation of 2 voxels, to mimic the correlation structure observed in real fMRI data. To simulate the spatial variability between images (inter-subject variability, movement artifacts in intra-subject variability), we define a new support of the ROIs, called $\tilde{\mathcal{R}}$ such as, for each image l^{th}, 50% (randomly chosen) of the weights $\mathbf{w}$ are set to zero. Thus, we have $\tilde{\mathcal{R}} \subset \mathcal{R}$. We simulate the target $\mathbf{y}$ for the l^{th} image as:

$$y_l = \sum_{(i,j,k)\in\tilde{\mathcal{R}}} w_{i,j,k} X_{i,j,k,l} + \epsilon_l \tag{16}$$

with the signal in the (i, j, k) voxel of the l^{th} image simulated as:

$$X_{i,j,k,l} \sim \mathcal{N}(0, 1) \tag{17}$$

and $\epsilon_l \sim \mathcal{N}(0, \gamma)$ is a Gaussian noise with standard deviation $\gamma > 0$. We choose γ in order to have a signal-to-noise ratio of 5 dB.

Results on simulated neuroimaging data We compare VB-MCBR and Gibbs-MCBR with the different competing algorithms. The resulting images of weights are given

in Fig. 4, with the true weights (a) and resulting Anova F-scores (b). The reference methods can detect the truly informative regions (*ROIs*), but elastic net (f) and ARD (h) only retrieve part of the support of the weights. Moreover, elastic net yields an overly sparse solution. BRR (g) also retrieves the *ROIs*, but does not yield a sparse solution, as all the features are regularized in the same way. We note that the weights in the *feature space* estimated by SVR (e) are non-zero everywhere and do not outline the support of the ground truth. VB-MCBR (c) converges to a local maximum similar to the solution found by BRR (g), *i.e.* creates only one non-empty class, and thus regularizes all the feature similarly. We can thus clearly see that, in this model, the Variational Bayes approach is very sensitive to the initialization, and can fall into non-optimal local maxima, for very sparse support of the weights. Finally, Gibbs-MCBR (d) retrieves most of the true support of the weights by performing an adapted regularization.

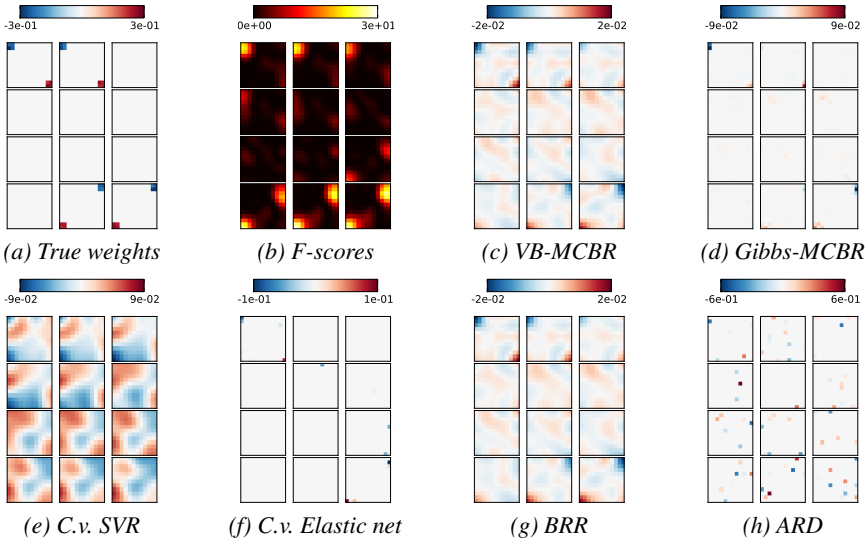


Fig. 4. Two-dimensional slices of the three-dimensional volume of simulated data. Weights found by different methods, the true target (a), and F-score (b). The Gibbs-MCBR method (d) almost retrieves the whole spatial support for the weights. The sparsity-promoting reference methods elastic net (f) and ARD (h) find an overly sparse support of the weights. VB-MCBR (c) converges to a local maximum similar to BRR (g), and thus does not yield a sparse solution. SVR (e) yields smooth maps that are not similar to the ground truth.

3.3 Experiments and results on real fMRI data

In this section, we assess the performance of MCBR in an experiment on the *mental representation of object size*, where the aim is to predict the size of an object seen by the subject during the experiment, in both intra-subject and inter-subject cases. The size (or scale parameter) of the object will be the target variable $\mathbf{y}$.

Details on real data We apply the different methods on a real fMRI dataset related to an experiment studying the representation of objects, on ten subjects, as detailed in [31]. During this experiment, ten healthy volunteers viewed objects of 4 shapes in 3 different sizes (yielding 12 different experimental conditions), with 4 repetitions of each stimulus in each of the 6 sessions. We pooled data from the 4 repetitions, resulting in a total of $n = 72$ images by subject (one image of each stimulus by session). Functional images were acquired on a 3-T MR system with eight-channel head coil (Siemens Trio, Erlangen, Germany) as T2*-weighted echo-planar image (EPI) volumes. Twenty transverse slices were obtained with a repetition time of 2 s (echo time, 30 ms; flip angle, $70°$; $2 \times 2 \times 2$-mm voxels; 0.5-mm gap). Realignment, normalization to MNI space, and General Linear Model (GLM) fit were performed with the SPM5 software (http://www.fil.ion.ucl.ac.uk/spm/software/spm5). The normalization is the conventional one of SPM (implying affine and non-linear transformations) and not the one using unified segmentation. The normalization parameters are estimated on the basis of a whole-head EPI acquired in addition, and are then applied to the partial EPI volumes. The data are not smoothed. In the GLM, the effect of each of the 12 stimuli convolved with a standard hemodynamic response function was modeled separately, while accounting for serial auto-correlation with an AR(1) model and removing low-frequency drift terms using a high-pass filter with a cut-off of 128 s. The GLM is fitted separately in each session for each subject, and we used in the present work the resulting session-wise parameter estimate images (the β-maps are used as rows of $\mathbf{X}$). The four different shapes of objects were pooled across for each one of the three sizes, and we are interested in finding discriminative information between sizes. This reduces to a regression problem, in which our goal is to predict a simple scalar factor (size of an object). All the analyzes are performed without any prior selection of regions of interest, and use the whole acquired volume.

Intra-subject regression analysis First, we perform an intra-subject regression analysis. Each subject is evaluated independently, in a 12-fold cross-validation. The dimensions of the real data set for one subject are $p \sim 7 \times 10^4$ and $n = 72$ (divided in 3 different sizes, 24 images per size). We evaluate the performance of the method by a leave-one-condition-out cross-validation (*i.e.*, leave-6-images-out), and doing so the GLM is performed separately for the training and test sets. The parameters of the reference methods are optimized with a nested leave-one-condition-out cross-validation within the training set, in the ranges given before.

Inter-subject regression analysis Additionally, we perform an inter-subject regression analysis on the sizes. The inter-subject analysis relies on subject-specific fixed-effects activations, *i.e.* for each condition, the 6 activation maps corresponding to the 6 sessions are averaged together. This yields a total of 12 images per subject, one for each experimental condition. The dimensions of the real data set are $p \sim 7 \times 10^4$ and $n = 120$ (divided into 3 different sizes). We evaluate the performance of the method by cross-validation (leave-one-subject-out). The parameters of the reference methods are optimized with a nested leave-one-subject-out cross-validation within the training set, in the ranges given before.

Results on real data

Intra-subject regression analysis The results obtained by the different methods are given in Table. 2. The *p-values* are computed using a paired t-test across subjects. VB-MCBR outperforms the other methods. Compared to the results on simulated data, VB-MCBR still falls in a local maximum similar to Bayesian Ridge Regression that performs well in this experiment. Moreover, both Gibbs-MCBR and VB-MCBR are more stable than the reference methods.

Inter-subject regression analysis The results obtained with the different methods are given in Table. 3. As in the intra-subject analysis, both MCBR approaches outperform the reference methods SVR, BRR and ARD. However, the prediction accuracy is similar to elastic net. In this case, Gibbs-MCBR performs slightly better than VB-MCBR, but the difference is not significant.

The maps of weights found by the different methods are detailed in Fig. 6. The methods are used combined with an Anova-based *univariate feature selection* (2500 voxels selected, in order to have a good support of the weights). As elastic net, Gibbs-MCBR yields a sparse solution, but extracts a few more voxels. The map found by elastic net is not easy to interpret, with very few informative voxels scattered in the whole occipital cortex. The map found by SVR is not sparse in the *feature space* and is thus difficult to interpret, as the spatial layout of the neural code is not clearly extracted. VB-MCBR does not yield a sparse map either, all the features having non-null weights.

One major asset of MCBR (and more particularly Gibbs-MCBR, as VB-MCBR often falls into a one-class local maximum) is that it creates a clustering of the features, based on the relevance of the features in the predictive model. This clustering can be accessed using the variable $\mathbf{z}$, that is implied in the regularization performed on the different features. In Fig. 5, we give the histogram of the weights of Gibbs-MCBR for the inter-subject analyzes. We keep the weights and the values of $\mathbf{z}$ of the last iteration, the different classes are represented as dots of different colors, and are superimposed on the histogram. We can notice than the pink distribution represented at the bottom of the histogram corresponds to relevant features. This cluster is very small (19 voxels), compared to the two blue classes represented at the top of the histogram that contain many voxels (746 voxels) which are highly regularized, as they are non-informative.

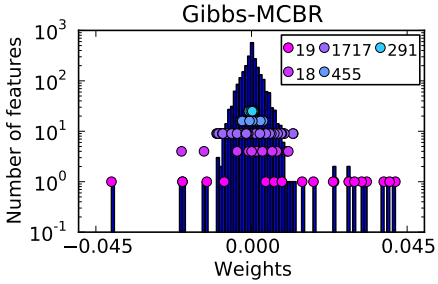


Fig. 5. *Inter-subject analysis*. Histogram of the weights found by Gibbs-MCBR, and corresponding $\mathbf{z}$ values (each color of dots represents a different class), for the inter-subject analyzes. We can see that Gibbs-MCBR creates clusters of informative and non informative voxels, and that the different classes are regularized differently, according to the relevance of the features in each of them.

Methods	mean ζ	std ζ	p-val / VB-MCBR
SVR	0.82	0.07	0.0003 **
Elastic net	0.9	0.02	0.0002 **
BRR	0.92	0.02	0.001 **
ARD	0.89	0.03	0.0003 **
Gibbs-MCBR	0.93	0.01	0.001 **
VB-MCBR	0.94	0.01	-

Table 2. *Intra-subject analysis*. Explained variance ζ for the three different methods. The p-values are computed using a paired t-test. VB-MCBR yields the best prediction accuracy, while being more stable than the reference methods.

Methods	mean ζ	std ζ	p-val / Gibbs-MCBR
SVR	0.77	0.11	0.14
Elastic net	0.78	0.1	0.75
BRR	0.72	0.1	0.01 **
ARD	0.52	0.33	0.02 *
Gibbs-MCBR	0.79	0.1	-
VB-MCBR	0.78	0.1	0.4

Table 3. *Inter-subject analysis*. Explained variance ζ for the different methods. The p-values are computed using a paired t-test. MCBR yields highest prediction accuracy than the two other Bayesian regularizations BRR and ARD.

4 Discussion

It is well known that in high-dimensional problems, regularization of feature loadings significantly increases the generalization ability of the predictive model. However, this regularization has to be adapted to each particular dataset. In place of costly cross-validation procedures, we cast regularization in a Bayesian framework and treat the regularization weights as hyper-parameters. The proposed approach yields an adaptive and efficient regularization, and can be seen as a compromise between a global regularization (Bayesian Ridge Regression) that does not take into account the sparse or focal distribution of the information, and Automatic Relevance determination. Additionally, MCBR creates a clustering of the features based on their relevance, and thus explicitly extracts groups of informative features.

Moreover, MCBR can cope with the different issues of ARD. ARD is subject to an underfitting in the hyper-parameters space, that corresponds to an underfitting in model selection (*i.e.* on the features to be pruned) [19]. Indeed, as ARD is estimated by maximizing evidence, models with less selected features are preferred, as the integration is done on less dimensions, and thus the evidence is higher. ARD will choose the sparsest model across models with similar accuracy. A contrario, MCBR requires far less hyper-parameters ($2 \times Q$, with $Q \ll p$), and suffers less from this issue, as the sparsity of the model is defined by groups. Moreover, a full Bayesian framework for estimating ARD requires to set some priors on the *hyper-parameters* (*e.g.* α_1 and α_2), and it may be sensitive to specific choice of these hyper-parameters. A solution is to use an *internal cross-validation* for optimizing these parameters, but this approach can be computationally expensive. In the case of MCBR, the distributions of the hyper-parameters are specific to a class and not to a specific feature, and thus, the proposed approach is less sensitive to the choice of the hyper-parameters. Indeed, the choice of good hyper-parameters for the features are dealt with at the class level.

On simulated data, our approach performs better than other classical methods such as SVR, BRR, ARD and elastic net and yields a more stable prediction accuracy. More-

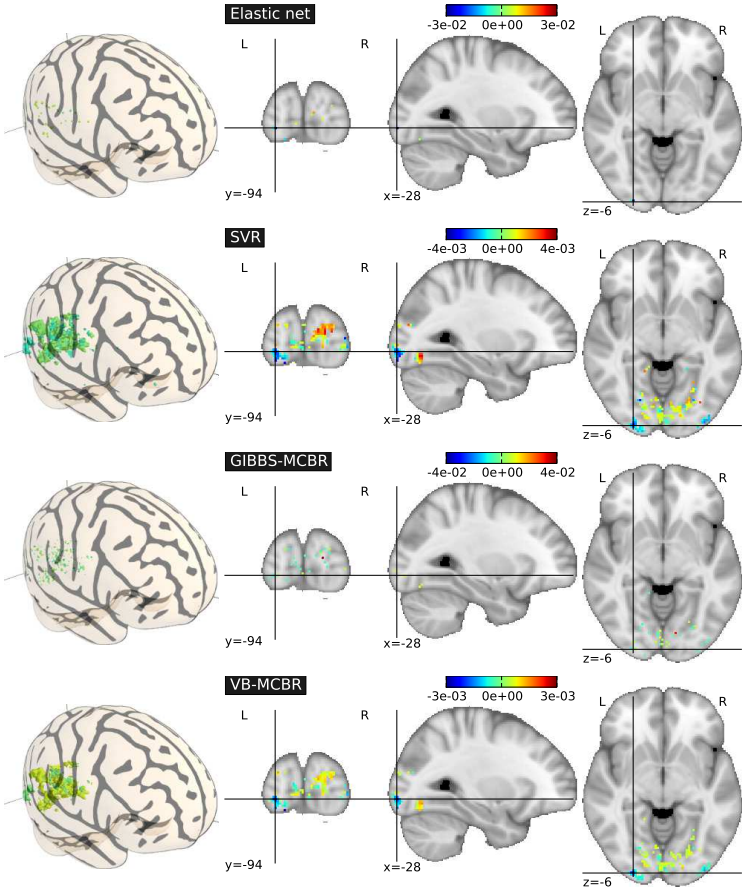


Fig. 6. *Inter-subject analysis*. Maps of weights found by the different methods on the 2500 most relevant features by Anova. The map found by elastic net is difficult to interpret as the very few relevant features are scattered within the whole brain. SVR and VB-MCBR do not yield a sparse solution. Gibbs-MCBR, by performing an adaptive regularization, draws a compromise between the other approaches, and yields a sparse solution, but also extract small groups of relevant features.

over, by adapting the regularization to different groups of voxels, MCBR retrieves the true support of the weights, and recovers a sparse solution. Results on real data show that MCBR yields more accurate predictions than other regularization methods. As it yields less sparse solution than elastic net, it gives access to more plausible loading maps which are necessary for understanding the spatial organization of brain activity, *i.e.* retrieving the spatial layout of the neural coding. On real fMRI data, the explicit clustering of Gibbs-MCBR is also an interesting aspect of the model, as it can extract few groups of relevant features from many voxels. In some experiments, the Variational Bayes algorithm yields less accurate predictions than the Gibbs sampling approach,

which can be explained by the difficulty of initializing the different variables (especially $\mathbf{z}$) when the support of the weight is overly sparse.

The question of model selection (i.e. the number of classes Q) has not been addressed in this paper. One can use the free energy in order to select the best model, but due to the instability of VB-MCBR, this approach does not seem promising. A more interesting method is the one detailed in [32], which can be used with Gibbs sampling algorithm. Here, model selection is performed implicitly by emptying classes that do not fit the data well. In that respect, the choice of heterogeneous priors for different classes is crucial: replacing our priors with class-independent priors (*i.e.* $\lambda_{1,q} = 10^{-2}$, $q \in [1, .., Q]$) in the inter-subject analysis on sizes prediction, leads Gibbs-MCBR to a local maximum similar to VB-MCBR.

Finally, this model is not restricted to Bayesian regularization, and can be used for classification, within a probit or logit model [33,34]. The proposed model may thus be used for diagnosis in medical imaging, for the prediction of both continuous or discrete variables.

Conclusion In this paper, we have proposed a model for adaptive regression, called *MCBR*. The proposed method integrates in the same Bayesian framework BRR and ARD, and performs a different regularization for relevant and irrelevant features. It can tune the regularization to the possible different level of sparsity encountered in fMRI data analysis, and yields interpretable information for fMRI inverse inference, namely the $\mathbf{z}$ variable (latent class variable). Experiments on both simulated and real data show that our approach is well-suited for neuroimaging, as it yields accurate and stable predictions compared to state-of-the-art methods.

Acknowledgments: The authors acknowledge support from the ANR grant ViMAGINE ANR-08-BLAN-0250-02.

References

1. K. Friston, A. Holmes, K. Worsley, J. Poline, C. Frith, R. Frackowiak, Statistical parametric maps in functional imaging: A general linear approach, Human Brain Mapping 2 (1995) 189–210. 1
2. S. Dehaene, G. Le Clec'H, L. Cohen, J.-B. Poline, P.-F. van de Moortele, D. Le Bihan, Inferring behavior from functional brain images, Nature Neuroscience 1 (1998) 549. 1
3. D. D. Cox, R. L. Savoy, Functional magnetic resonance imaging (fmri) "brain reading": detecting and classifying distributed patterns of fmri activity in human visual cortex., Neuroimage 19 (2003) 261–270. 1, 2
4. P. Dayan, L. F. Abbott, Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems, The MIT Press, 2001. 2
5. J.-D. Haynes, G. Rees, Predicting the stream of consciousness from activity in human visual cortex, Current Biology 15 (14) (2005) 1301 – 1307. 2
6. T. M. Mitchell, R. Hutchinson, R. S. Niculescu, F. Pereira, X. Wang, M. Just, S. Newman, Learning to decode cognitive states from brain images, Machine Learning V57 (1) (2004) 145–175. 2
7. S. LaConte, S. Strother, V. Cherkassky, J. Anderson, X. Hu, Support vector machines for temporal classification of block design fmri data, NeuroImage 26 (2) (2005) 317 – 329. 2

8. J. Mourao-Miranda, A. L. Bokde, C. Born, H. Hampel, M. Stetter, Classifying brain states and determining the discriminating activation patterns: Support vector machine on functional mri data, NeuroImage 28 (4) (2005) 980 – 995. 2
9. S. J. Hanson, Y. O. Halchenko, Brain reading using full brain support vector machines for object recognition: There is no face identification area, Neural Computation 20 (2) (2008) 486–503. 2
10. O. Yamashita, M. aki Sato, T. Yoshioka, F. Tong, Y. Kamitani, Sparse estimation automatically selects voxels relevant for the decoding of fmri activity patterns, NeuroImage 42 (4) (2008) 1414 – 1429. 2, 3
11. S. Ryali, K. Supekar, D. A. Abrams, V. Menon, Sparse logistic regression for whole-brain classification of fmri data, NeuroImage 51 (2) (2010) 752 – 764. 2
12. F. D. Martino, G. Valente, N. Staeren, J. Ashburner, R. Goebel, E. Formisano, Combining multivariate voxel selection and support vector machines for mapping and classification of fmri spatial patterns, NeuroImage 43 (1) (2008) 44 – 58. 2
13. I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 46 (1-3) (2002) 389–422. 2
14. C. Chu, Y. Ni, G. Tan, C. J. Saunders, J. Ashburner, Kernel regression for fmri pattern prediction, NeuroImage In Press, Corrected Proof (2010) –. 2, 3
15. H. Liu, M. Palatucci, J. Zhang, Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery, in: ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 649–656. 2
16. M. K. Carroll, G. A. Cecchi, I. Rish, R. Garg, A. R. Rao, Prediction and interpretation of distributed neural activity with sparse models, NeuroImage 44 (1) (2009) 112 – 122. 2
17. C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), 1st Edition, Springer, 2007. 2
18. M. Tipping, The relevance vector machine, Morgan Kaufmann, 2000. 2, 6
19. Y. Qi, T. P. Minka, R. W. Picard, Z. Ghahramani, Predictive automatic relevance determination by expectation propagation, in: ICML '04: Proceedings of the twenty-first international conference on Machine learning, ACM Press, 2004. 3, 13
20. D. Wipf, S. Nagarajan, A new view of automatic relevance determination, in: Advances in Neural Information Processing Systems 20, MIT Press, 2008, pp. 1625–1632. 3
21. Y. Ni, C. Chu, C. J. Saunders, J. Ashburner, Kernel methods for fmri pattern prediction, Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on (2008) 692–697. 3
22. K. Friston, C. Chu, J. Mouro-Miranda, O. Hulme, G. Rees, W. Penny, J. Ashburner, Bayesian decoding of brain images, NeuroImage 39 (1) (2008) 181 – 205. 3
23. E. I. George, R. E. McCulloch, Variable selection via gibbs sampling, Journal of the American Statistical Association 88 (423) (1993) 881–889. 4
24. S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, Morgan Kaufmann Publishers Inc., 1987. 6
25. H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. Roy. Stat. Soc. B 67 (2005) 301. 7
26. C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297. 7
27. G. Hughes, On the mean accuracy of statistical pattern recognizers, Information Theory, IEEE Transactions on 14 (1) (1968) 55–63. 7
28. J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, Journal of Statistical Software 33 (1). 7
29. C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm (2001). 7
30. scikit-learn, http://scikit-learn.sourceforge.net/, version 0.2 (downloaded in Apr. 2010). 7

31. E. Eger, C. Kell, A. Kleinschmidt, Graded size sensitivity of object exemplar evoked activity patterns in human loc subregions, J. Neurophysiol. 100(4):2038-47. 11
32. S. Chib, I. Jeliazkov, Marginal likelihood from the metropolis-hastings output, Journal of the American Statistical Association 96 (2001) 270–281. 15
33. J. H. Albert, S. Chib, Bayesian Analysis of Binary and Polychotomous Response Data, Journal of the American Statistical Association 88 (422) (1993) 669–679. 15
34. R. E. McCulloch, N. G. Polson, P. E. Rossi, A bayesian analysis of the multinomial probit model with fully identified parameters, Journal of Econometrics 99 (1) (2000) 173 – 193. 15

A VB-MCBR algorithm

The *Variational Bayes* approach yields the following variational distributions:

- $q(\mathbf{w}) \sim \mathcal{N}(\mathbf{w}|\mu, \mathbf{\Sigma})$ with:

$$\overline{\mathbf{A}} = \text{diag}(\overline{l_1}, \ldots, \overline{l_p}) \quad \text{with} \quad \overline{l_p} = \sum_{q=1}^{Q} q(z_j = q) \frac{l_{1,q}}{l_{2,q}} \tag{18}$$

$$\mathbf{\Sigma} = (\frac{a_1}{a_2}\mathbf{X}^T\mathbf{X} + \overline{\mathbf{A}})^{-1} \tag{19}$$

$$\mu = \frac{a_1}{a_2}\mathbf{\Sigma}\mathbf{X}^T\mathbf{y} \tag{20}$$

- $q(\lambda_q) \sim \Gamma(l_{1,q}, l_{2,q})$ with:

$$l_{1,q} = \lambda_{1,q} + \frac{1}{2}\sum_{j=1}^{p} q(z_j = q) \tag{21}$$

$$l_{2,q} = \lambda_{2,q} + \frac{1}{2}\sum_{j=1}^{p} (\mu_{jj}^2 + \Sigma_{jj}) q(z_j = q) \tag{22}$$

- $q(\alpha) \sim \Gamma(a_1, a_2)$ with:

$$a_1 = \alpha_1 + \frac{n}{2} \tag{23}$$

$$a_2 = \alpha_2 + \frac{1}{2}(\mathbf{y} - \mathbf{X}\mu)^T(\mathbf{y} - \mathbf{X}\mu) + \frac{1}{2}Tr(\mathbf{\Sigma}\mathbf{X}^T\mathbf{X}) \tag{24}$$

- $q(z_j = q) \sim \exp^{(\rho_{jq})}$ with:

$$\rho_{jq} = -\frac{1}{2}(\mu_j^2 + \Sigma_{jj})\frac{l_{1,q}}{l_{2,q}} + \ln(\pi_q) + \frac{1}{2}\left(\Psi(l_{1,q}) - \log(l_{2,q})\right) \tag{25}$$

$$\pi_q = \exp^{\left\{\Psi(d_q) - \Psi(\sum_{q=1}^{q=Q} d_q)\right\}} \tag{26}$$

$$d_q = \eta_q + \sum_{j=1}^{p} q(z_j = q) \tag{27}$$

where Ψ is the digamma function $\Psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$. The pseudo-code of the *VB-MCBR* algorithm is provided in pseudo-code 1

Algorithm 1: *VB-MCBR*

Initialize $a_1 = \alpha_1$, $a_2 = \alpha_2$, $l_1 = \lambda_1$, $l_2 = \lambda_2$ and $d_q = \eta_q$
Randomly initialize $q(z_j = q)$
Set a number of iterations *max steps*
repeat
  Compute A using Eq. 18, Σ using Eq. 19 and μ using Eq. 20.
  Compute l_1 using Eq. 21 and l_2 using Eq. 22.
  Compute a_1 using Eq. 23 and a_2 using Eq. 24.
  Compute ρ_{jq} using Eq. 25.
  Compute π_q using Eq. 26 and d_q using Eq. 27.
until *max steps* ;
return μ

Algorithm 2: *Gibbs-MCBR*

Initialize α_1, α_2, λ_1, λ_2 and η_q
Randomly initialize z
Set a number of iterations *burn number* for *burn-in*, and *max steps*.
repeat
  Compute Σ using Eq. 28 and μ using Eq. 29, sample $\mathbf{w}$ in $\mathcal{N}(\mathbf{w}|\mu, \mathbf{\Sigma})$.
  Compute l_1, l_2 using Eq. 30, Eq. 31, sample λ in $\Pi_{q=1}^{Q} \Gamma(\lambda_q|l_{1,q}, l_{2,q})$.
  Compute a_1 using Eq. 32 and a_2 using Eq. 33, sample α in $\Gamma(a_1, a_2)$.
  Compute ρ_{jq} using Eq. 34, sample $\mathbf{z}$ in mult$(\exp \rho_{j,1}, ..., \exp \rho_{j,Q})$.
  Compute d_q using Eq. 35, sample π_q in Dir(d_q).
until *max steps* ;
return Average value of $\mathbf{w}$ after *burn number* iterations.

B Gibbs-MCBR algorithm

With $\Theta = [\mathbf{w}, \lambda, \alpha, \mathbf{z}, \pi]$, we have the following candidate distributions (*i.e.* the distributions used for the sampling of the different parameters):

- $p(\mathbf{w}|\Theta - \{\mathbf{w}\}) \propto \mathcal{N}(\mathbf{w}|\mu, \mathbf{\Sigma})$ with:

$$\mathbf{\Sigma} = (\mathbf{X}^T\mathbf{X}\alpha + \mathbf{A})^{-1} \quad \text{with} \quad \mathbf{A} = \text{diag}(\lambda_{z_1}, ..., \lambda_{z_p}) \tag{28}$$

$$\mu = \mathbf{\Sigma}\alpha\mathbf{X}^T\mathbf{y} \tag{29}$$

- $p(\lambda|\Theta - \{\lambda\}) \propto \Pi_{q=1}^{Q} \Gamma(\lambda_q|l_{1,q}, l_{2,q})$ with:

$$l_{1,q} = \lambda_{1,q} + \frac{1}{2}\sum_{j=1}^{p} \delta(z_j = q) \tag{30}$$

$$l_{2,q} = \lambda_{2,q} + \frac{1}{2}\sum_{j=1}^{p} \delta(z_j = q) w_j^2 \tag{31}$$

- $p(\alpha|\Theta - \{\alpha\}) \propto \Gamma(a_1, a_2)$ with:

$$a_1 = \alpha_1 + \frac{n}{2} \tag{32}$$

$$a_2 = \alpha_2 + \frac{1}{2}(\mathbf{y} - \mathbf{X}\mu)^T(\mathbf{y} - \mathbf{X}\mu) \tag{33}$$

- $p(z_j|\Theta - \{\mathbf{z}\}) \propto \text{mult}(\exp \rho_{j,1}, ..., \exp \rho_{j,Q})$ with:

$$\rho_{jq} = -\frac{1}{2} w_j^2 \lambda_q + \ln(\pi_q) + \frac{1}{2}\log \lambda_q \tag{34}$$

- $p(\pi_q|\Theta - \{\pi\}) \propto \text{Dir}(d_q)$ with:

$$d_q = \eta_q + \sum_{j=1}^{p} \delta(z_j = q) \tag{35}$$

The algorithm is provided in pseudo-code 2.