Research Article

Multiclass Sparse Bayesian Regression for fMRI-Based Prediction

Vincent Michel,1,2,3 Evelyn Eger,3,4 Christine Keribin,2,5 and Bertrand Thirion1,3

1 PARIETAL Team, INRIA Saclay-Île-de-France, 91191 Saclay, France
2 Laboratoire de Mathématiques, Université Paris-Sud 11, 91400 Orsay, France
3 CEA, DSV, I2BM, NeuroSpin, 91191 Gif-sur-Yvette, France
4 CEA, DSV, I2BM, INSERM U562, 91191 Gif-sur-Yvette, France
5 SELECT Team, INRIA Saclay-Île-de-France, 91400, France

Correspondence should be addressed to Bertrand Thirion, bertrand.thirion@inria.fr

Received 23 December 2010; Revised 3 March 2011; Accepted 7 April 2011

Academic Editor: Kenji Suzuki

Copyright © 2011 Vincent Michel et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Inverse inference has recently become a popular approach for analyzing neuroimaging data, by quantifying the amount of information contained in brain images on perceptual, cognitive, and behavioral parameters. As it outlines brain regions that convey information for an accurate prediction of the parameter of interest, it allows 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), and dimension reduction is thus a mandatory step. We introduce in this paper a new model, called Multiclass Sparse Bayesian Regression (MCBR), that, unlike classical alternatives, automatically adapts the amount of regularization to the available data. MCBR consists in grouping features into several classes and then regularizing each class differently in order to apply an adaptive and efficient regularization. We detail these 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], and it 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 for 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 learning methods have been used to extract information from brain images and compare it to the corresponding target. Among them, Linear Discriminant Analysis (LDA) [3, 5], Support Vector Machine (SVM) [6–9], or regularized prediction [10, 11] has been 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, leading to overfitting. Dimension reduction can be used to extract relevant information from the data, the standard approach in functional neuroimaging being feature selection (e.g., Anova) [3, 6, 11, 12]. However, by performing feature selection and parameter estimation separately, such approach is not optimal. Thus, a popular combined selection/estimation scheme, Recursive Feature Elimination [13], may be used. However, this approach relies on a specific heuristic, which does not guarantee the optimality of the solution and is particularly costly.
By contrast, there is great interest in sparsity-inducing regularizations, which 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 parameter but possibly also position, speed (assuming a 1-D space), or cardinality. Thus, we focus on regression problems and defer the generalization to classification to future work. Let us introduce the following predictive linear model:

$$ y = Xw + b, $$

where $y$ represents the behavioral variable and $(w, b)$ are the parameters to be estimated on a training set. A vector $w \in \mathbb{R}^p$ 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 $X \in \mathbb{R}^{n \times p}$ is the design matrix. Each row is a $p$-dimensional sample, that is, an activation map related to the observation. With $n \ll p$, the estimation of $w$ is ill posed.

To cope with the high dimensionality of the data, one can penalize the estimation of $w$, for example, based on the $\ell_2$ norm of the weights. Classical regularization schemes have been used in functional neuroimaging, such as the 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 the Bayesian regression techniques, which 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 $\ell_2$ norm is equivalent to setting the Gaussian priors on the weights $w$:

$$ w \sim \mathcal{N}(0, A^{-1}), \quad A = \text{diag}(\alpha_1, \ldots, \alpha_p), $$

where $\mathcal{N}$ is the Gaussian distribution and $\alpha_i$ the precision of the $i$th feature. The model in (2) defines two classical Bayesian regression schemes. The first one is Bayesian Ridge Regression (BRR) [17], which corresponds to the particular case $\alpha_1 = \cdots = \alpha_m$. 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], which 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 each feature separately, ARD is prone to underfitting when the model contains too many regressors [19] and also suffers from convergence issues [20].

These classical Bayesian regularization schemes have been used in fMRI inverse inference studies [10, 14, 21]. However, these studies used only 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 [22], 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 [23], which regularizes patterns of voxels differently. The weights of the model are defined by $w = U\eta$, where $U$ is a matrix defined as set of spatial patterns (one pattern by column) and $\eta$ are the parameters of the decomposition of $w$ in the basis defined by $U$. The regularization is controlled through the covariance of $\eta$, which is assumed to be diagonal with only $m$ possible different values $\text{cov}(\eta) = \exp(\lambda_1)I^{(1)} + \cdots + \exp(\lambda_m)I^{(m)}$.

The matrices $I^{(i)}$ are diagonal and defined subsets of columns of $U$ sharing similar variance $\exp(\lambda_i)$. Due to its class-based model, this approach is similar to the one proposed in this paper, but the construction of $I$ relies on ad hoc voxel selection steps, so that there is no proof that the solution is correct. A contrario, the proposed approach jointly optimizes, within the same framework, the construction of the pattern of voxels and the regularization parameter of each pattern.

In this paper, we detail a model for the Bayesian regression in which features are grouped into $K$ 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 Multiclass 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 (the variational Bayes or Gibbs sampling procedures), we show that the proposed algorithm yields better accuracy than reference methods, while providing more interpretable models.

## 2. Multiclass 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:

$$ y = f(X, w, b) = Xw + b. $$

We denote by $y \in \mathbb{R}^n$ the targets to be predicted and $X \in \mathbb{R}^{n \times p}$ 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 stimuli, while $n \sim 10$ to $10^2$.

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

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

$$ \alpha \sim \Gamma(\alpha_1, \alpha_2), $$

where $\alpha_1$ and $\alpha_2$ are the shape and scale parameters of the Gamma distribution.
where $\alpha$ is the precision parameter and $\Gamma$ stands for the gamma density with two hyperparameters $\alpha_1, \alpha_2$:

$$
\Gamma(x; \alpha_1, \alpha_2) = \alpha_2^\alpha_1 x^{\alpha_1-1} \exp(-x\alpha_2 / \Gamma(\alpha_1)).
$$

**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 $K$ indexed by a discrete variable $z_j$ ($z = \{z_1, \ldots, z_p\}$). All the features within a class $k \in \{1, \ldots, K\}$ share the same precision parameter $\lambda_k$, and we use the following prior on $z$:

$$
z \sim \prod_{j=1}^p \prod_{k=1}^K \delta_{jk}^{\lambda_{zk}},
$$

where $\delta$ is Kronecker’s $\delta$, defined as

$$
\delta_{jk} =
\begin{cases}
0 & \text{if } z_j \neq k, \\
1 & \text{if } z_j = k.
\end{cases}
$$

We finally introduce an additional Dirichlet prior [24] on $\pi$:

$$
\pi \sim \text{Dir}(\eta)
$$

with a hyperparameter $\eta$. By updating at each step the probability $n_k$ 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:

$$
w \sim \mathcal{N}(0, A^{-1}) \quad \text{with } \text{diag}(A) = \{\lambda_{z_1}, \ldots, \lambda_{z_p}\},
$$

where $\lambda_{z_j}$ is the precision parameter of the $j$th feature, with $z_j \in \{1, \ldots, K\}$. We introduce the following prior on $\lambda_k$:

$$
\lambda_{k} \sim \Gamma(\lambda_{k}; \lambda_{i,k}, \lambda_{2,k})
$$

with hyperparameters $\lambda_{1,k}, \lambda_{2,k}$. The complete generative model is summarized in Figure 1.

![Figure 1](image.png)

**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 $p(z \mid X, y)$ for this maximization. However, the posterior distribution of the latent variables given the data does not have 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 the variational Bayes, and the resulting algorithm is thus called VB-MCBR. We also detail an algorithm, called Gibbs-MCBR, based on a Gibbs sampling procedure.

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

$$
q(\Theta) = q(w)q(\lambda)q(\alpha)q(z)q(\pi).
$$

We introduce the Kullback-Leibler divergence $\mathcal{D}(q(\Theta))$ that measures the similarity between the true posterior
parameters. We set $\alpha$ the classes can be seen as a means of avoiding costly model
number of classes and letting the model automatically prune
4 International Journal of Biomedical Imaging

\[ \lambda \]

\[ \textit{2.2.2. Estimation by Gibbs Sampling: Gibbs-MCBR.} \]

We develop here an estimation of the MCBR model using Gibbs
sampling [26]. The resulting algorithm is called Gibbs-
MCBR; the pseudocode of the algorithm and the candidate
distributions are provided in Appendix B. The Gibbs
sampling algorithm is used for generating a sequence of
samples from the joint distribution to approximate marginal
distributions. The main idea is to use conditional distributions
that should be known and possibly easy to sample from, instead of directly computing the marginals from the
joint law by integration (the joint law may not be known
or hard to sample from). The sampling is done iteratively
among the different parameters, and the final estimation
of parameters is obtained by averaging the values of the
different parameters across the different iterations (one
may not consider the first iterations, this is called the burn in).

2.2.3. Initialization and Priors on the Model Parameters.

Our model needs few hyperparameters; we choose here to
use slightly informative and class-specific hyperparameters
in order to reflect a wide range of possible behaviors for the
weight distribution. This choice of priors is equivalent
to setting heavy-tailed centered Student's $t$-distributions with variance at different scales, as priors on the weight
parameters. We set $K = 9$, with weakly informative priors
$\lambda_{1,k} = 10^{k-4}$, $k \in [1, \ldots, K]$ and $\lambda_{2,k} = 10^{-2}$, $k \in [1, \ldots, K]$. 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 of avoiding 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 $q(z)$
for VB-MCBR (or $z$ for Gibbs-MCBR).

2.3. Validation and Model Evaluation

2.3.1. Performance Evaluation. Our method is evaluated with
a cross-validation procedure that splits the available data into
training and validation sets. In the following, $(X', y')$ is a
learning set $(X', y')$ is a test set, and $\hat{y'} = f(X'w)$ refers to
the predicted target, where $w$ is estimated from the training set.
The performance of the different models is evaluated using
$\zeta$, the ratio of explained variance:

\[ \zeta(y', \hat{y'}) = \frac{\text{var}(y') - \text{var}(y' - \hat{y'})}{\text{var}(y')} \]  

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).

2.3.2. Competing Methods. In our experiments, the proposed
algorithms are compared to different state-of-the-art regular-
ization methods.

(i) Elastic Net Regression [27], which requires setting
two parameters $\lambda_1$ and $\lambda_2$. 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
$\tilde{\lambda} = \|X'X\|_2$, and $\lambda_2 \in \{0.1, 0.5, 1., 10, 100.\}$ Note that $\lambda_1$ and $\lambda_2$ parame-
etrize heterogeneous norms.

(ii) Support Vector Regression (SVR) with a linear kernel
[28], 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.

(iii) Bayesian Ridge Regression (BRR), which is equivalent
to MCBR with $K = 1$ and $\lambda_1 = \lambda_2 = \alpha_1 = \alpha_2 = 10^{-6}$,
that is, weakly informative priors.

(iv) Automatic Relevance Determination (ARD), which is
equivalent to MCBR with $K = p$ and $\lambda_1 = \lambda_2 = \alpha_1 = \alpha_2 = 10^{-6}$, that is, 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 [29]. The optimal number of
voxels is selected within the range $[50, 100, 250, 500]$, using
a nested cross-validation within the training set. We do not
directly select a threshold on $P$ value or cluster size, but
rather a predefined 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
Table 1: Simulated regression data. Explained variance $\zeta$ for different methods (average of 15 different trials). The $P$-values are computed using a paired $t$-test.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Mean $\zeta$</th>
<th>Std $\zeta$</th>
<th>$P$-value to Gibbs-MCBR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVR</td>
<td>0.11</td>
<td>0.1</td>
<td>.0**</td>
</tr>
<tr>
<td>Elastic net</td>
<td>0.77</td>
<td>0.11</td>
<td>.0004**</td>
</tr>
<tr>
<td>BRR</td>
<td>0.19</td>
<td>0.14</td>
<td>.0**</td>
</tr>
<tr>
<td>ARD</td>
<td>0.79</td>
<td>0.06</td>
<td>.0**</td>
</tr>
<tr>
<td>Gibbs-MCBR</td>
<td>0.89</td>
<td>0.04</td>
<td>—</td>
</tr>
<tr>
<td>VB-MCBR</td>
<td>0.04</td>
<td>0.05</td>
<td>.0**</td>
</tr>
</tbody>
</table>

*Level of significance of the $P$-values between 0.01 and 0.05.

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 [30], while SVR is based on LibSVM [31]. Methods are used from Python via the Scikit-learn open source package [32].

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. Preliminary results on both simulated and real data showed that these values are sufficient enough for an accurate inference of the model.

As explained previously, we set $K = 9$, with weakly informative priors $\lambda_{i,k} = 10^{-k}$, $k \in [1, \ldots, K]$ and $\lambda_{i,k} = 10^{-k}$, $k \in [1, \ldots, K]$. Moreover, we set $c_1 = c_2 = 1$, and we randomly initialize $q(z)$ for VB-MCBR (or $z$ for Gibbs-MCBR).

3. Experiments and Results

3.1. Experiments on Simulated Data. We now evaluate and illustrate MCBR on two different sets of simulated data.

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

$$X \sim \mathcal{N}(0, 1) \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, 1),$$

$$y = 2(X_1 + X_2 - X_3 - X_4) + 0.5(X_5 + X_6 - X_7 - X_8) + \epsilon.$$

We have $p = 200$ features, $n' = 50$ images for the training set, and $n'' = 50$ 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.

3.1.2. Results on Simulated Regression Data. We average the results of 15 different trials, and the average explained variance is shown in Table 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 the local maximum of $\mathcal{F}$ and does not yield an accurate prediction. BBR has a low prediction accuracy compared to other methods such as ARD. Indeed, it cannot finely adapt the weights of the relevant features, as these features are regularized similarly as the irrelevant ones. SVR has also low accuracy, due to the fact that we do not perform any feature selection. Thus, SVR suffers from the curse of dimensionality, unlike other methods such as ARD or elastic net, which performs feature selection and model estimation jointly.

In Figure 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 monomodal density. ARD is far more adaptive and sets lots of weights to zero. The Gibbs-MCBR algorithm creates a multimodal distribution, lots of weights being highly regularized (pink distributions), and informative features are allowed to have higher weights (blue distributions).

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

3.1.3. Comparison between VB-MCBR and Gibbs-MCBR. We now look at the values of $w_i$ and $q_i$ for the different steps of the two algorithms (see Figure 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.

3.2. Simulated Neuroimaging Data

3.2.1. Details on Simulated Neuroimaging Data. The simulated data set $X$ consists of $n = 100$ images (size $12 \times 12 \times 12$ voxels) with a set of four square regions of interest (ROI) (size $2 \times 2 \times 2$). We call $\mathcal{R}$ the support of the ROI (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 (intersubject variability, movement artifacts in intrasubject variability), we define a new support of the ROIs, called $\tilde{\mathcal{R}}$ such that, for each image $l$th, 50% (randomly chosen) of the weights $w$ are set to zero. Thus, we have $\tilde{\mathcal{R}} \subset \mathcal{R}$. We simulate the target $y$ for the $l$th image as

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

with the signal in the $(i, j, k)$ voxel of the $l$th image simulated as

$$X_{i,j,k,l} \sim \mathcal{N}(0, 1),$$
and $e_t \sim \mathcal{N}(0, \gamma)$ is a Gaussian noise with standard deviation $\gamma > 0$. We choose $\gamma$ in order to have a signal-to-noise ratio of 5 dB.

### 3.2.2 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 Figure 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) retrieve only 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 nonzero 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); that is, it creates only one nonempty class, and thus regularizes all the features similarly. We can thus clearly see that, in this model, the variational Bayes approach is very sensitive to the initialization and can fall into nonoptimal 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.

### 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 intrasubject and intersubject cases. The size (or scale parameter) of the object will be the target variable $y$.

#### 3.3.1 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 [33]. 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
Figure 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.

Figure 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) retrieves almost 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.
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 an 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 2s (echo time: 30 ms; flip angle: 70°; 2×2×2-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 method of SPM (implying affine and nonlinear 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 autocorrelation with an AR(1) model and removing low-frequency drift terms using a high-pass filter with a cutoff 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 \( \beta \)-maps are used as rows of \( X \)). The four different shapes of objects were pooled across for each one of the three sizes, and we are interested in finding discriminative information on sizes. This reduces to a regression problem, in which our model. This clustering can be accessed using the variable \( z \).

**Table 2: Intrasubject analysis.** Explained variance \( \zeta \) 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.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Mean ( \zeta )</th>
<th>Std ( \zeta )</th>
<th>( P )-val/Gibbs-MCBR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVR</td>
<td>0.82</td>
<td>0.07</td>
<td>.0006***</td>
</tr>
<tr>
<td>Elastic net</td>
<td>0.9</td>
<td>0.02</td>
<td>.001***</td>
</tr>
<tr>
<td>BRR</td>
<td>0.92</td>
<td>0.02</td>
<td>.0358**</td>
</tr>
<tr>
<td>ARD</td>
<td>0.89</td>
<td>0.03</td>
<td>.0015***</td>
</tr>
<tr>
<td>Gibbs-MCBR</td>
<td>0.93</td>
<td>0.01</td>
<td>—</td>
</tr>
<tr>
<td>VB-MCBR</td>
<td>0.94</td>
<td>0.01</td>
<td>.99</td>
</tr>
</tbody>
</table>

**Level of significance of the \( P \)-values between 0.01 and 0.05.**

**Level of significance of the \( P \)-values below 0.01.**

**Table 3: Intersubject analysis.** Explained variance \( \zeta \) 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.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Mean ( \zeta )</th>
<th>Std ( \zeta )</th>
<th>( P )-val/Gibbs-MCBR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVR</td>
<td>0.77</td>
<td>0.11</td>
<td>.14</td>
</tr>
<tr>
<td>Elastic net</td>
<td>0.78</td>
<td>0.1</td>
<td>.75</td>
</tr>
<tr>
<td>BRR</td>
<td>0.72</td>
<td>0.1</td>
<td>.01**</td>
</tr>
<tr>
<td>ARD</td>
<td>0.52</td>
<td>0.33</td>
<td>.02*</td>
</tr>
<tr>
<td>Gibbs-MCBR</td>
<td>0.79</td>
<td>0.1</td>
<td>—</td>
</tr>
<tr>
<td>VB-MCBR</td>
<td>0.78</td>
<td>0.1</td>
<td>0.4</td>
</tr>
</tbody>
</table>

**Level of significance of the \( P \)-values between 0.01 and 0.05.**

**Level of significance of the \( P \)-values below 0.01.**

Intrasubject Regression Analysis. Additionally, we perform an intersubject regression analysis on the sizes. The intersubject analysis relies on subject-specific fixed-effect activations that is, 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.

3.3.2. Results on Real Data

Intrasubject 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 the Bayesian ridge regression which performs well in this experiment. Moreover, both Gibbs-MCBR and VB-MCBR are more stable than the reference methods.

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

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 \( z \), which is implied in the regularization performed on the different features. In Figure 5, we give the histogram of the weights of Gibbs-MCBR for the intersubject analysis. We keep the weights and the values of \( z \) 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
approach yields an adaptive and e
regularization weights as hyperparameters. The proposed
dataset. In place of costly cross-validation procedures, we
this regularization has to be adapted to each particular
generalization ability of the predictive model. However,
ularization of feature loadings significantly increases the
It is well known that in high-dimensional problems, reg-

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
noninformative.

The maps of weights found by the different methods
are detailed in Figure 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 nonnull weights.

4. Discussion
It is well known that in high-dimensional problems, reg-
ularization 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 hyperparameters. 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 hyperpara-

ter 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 evidence is higher. ARD will
choose the sparsest model across models with similar accu-

A contrario, MCBR requires far less hyperparameter
(2 × K, with K ≪ 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 hyperparameters (e.g., α1 and α2), and it
may be sensitive to specific choice of these hyperparameters.
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
hyperparameters are bound to a class and not to each feature.
Thus, the proposed approach is less sensitive to the choice of
the hyperparameters. Indeed, the choice of good hyperpara-

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. Moreover,
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, that is, retrieving the spatial layout of the neural
coding. On real fMRI data, the explicit clustering of Gibbs-

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 z) when the
support of the weight is overly sparse. Moreover, the VB-
MCBR algorithm relies on a variational Bayes approach,
which may not be optimal, due to strong approximations
in model inference. A contrario Gibbs-MCBR is more time
consuming but yields a better model inference. Finally, the
variability in the results may be explained by the difficulty to
estimate the model (optimality is not ensured).

The question of model selection (i.e., the number of
classes K) 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 [34], which can be used with the 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

Figure 5: Intersubject analysis. Histogram of the weights found
by Gibbs-MCBR and corresponding z values (each color of dots
represents a different class), for the intersubject analyzes. We
can see that Gibbs-MCBR creates clusters of informative and
noninformative voxels and that the different classes are regularized
differently, according to the relevance of the features in each of
them.
5. 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

on size prediction leads Gibbs-MCBR to a local maximum similar to VB-MCBR. Finally, this model is not restricted to the Bayesian regularization and can be used for classification, within a probit or logit model [35, 36]. The proposed model may thus be used for diagnosis in medical imaging, for the prediction of both continuous or discrete variables.
level of sparsity encountered in fMRI data analysis, and it yields interpretable information for fMRI inverse inference, namely, the \( z \) 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 the state-of-the-art methods.

### Appendices

#### A. VB-MCBR Algorithm

The variational Bayes approach yields the following variational distributions:

(i) \( q(w) \sim \mathcal{N}(w \mid \mu, \Sigma) \)

\[
\Sigma = \left( \frac{a_1}{a_2}X^TX + \bar{A} \right)^{-1}, \tag{A.2}
\]

\[
\mu = \frac{a_1}{a_2} \Sigma X^Ty; \tag{A.3}
\]

(ii) \( q(\lambda_k) \sim \Gamma(l_{1,k}, l_{2,k}) \) with

\[
l_{1,k} = \lambda_{1,k} + \frac{1}{2} \sum_{j=1}^{p} q(z_j = k), \tag{A.4}
\]

\[
l_{2,k} = \lambda_{2,k} + \frac{1}{2} \sum_{j=1}^{p} (\mu^2_{jj} + \Sigma_{jj}) q(z_j = k); \tag{A.5}
\]

(iii) \( q(\alpha) \sim \Gamma(a_1, a_2) \) with

\[
a_1 = a_1 + \frac{n}{2}; \tag{A.6}
\]

\[
a_2 = a_2 + \frac{1}{2} \left( y - X\mu \right)^T (y - X\mu) + \frac{1}{2} \text{Tr}(\Sigma X^TX); \tag{A.7}
\]

(iv) \( q(z_j = k) \sim \exp(\rho_{jk}) \)

\[
\rho_{jk} = -\frac{1}{2} \left( \mu_j^2 + \Sigma_{jj} \right) l_{2,k} + \ln(\eta_k) + \frac{1}{2} \left( \Psi(l_{1,k}) - \log(l_{2,k}) \right), \tag{A.8}
\]

\[
\eta_k = \exp(\Psi(d_k) - \Psi(\sum_{k=1}^{K} d_k)); \tag{A.9}
\]

\[
d_k = \eta_k + \sum_{j=1}^{p} q(z_j = k), \tag{A.10}
\]

where \( \Psi \) is the digamma function \( \Psi(x) = \Gamma^\prime(x)/\Gamma(x) \). The VB-MCBR algorithm is provided in pseudo-code in Algorithm 1.

#### B. Gibbs-MCBR Algorithm

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

(i) \( p(w \mid \Theta - \{w\}) \propto \mathcal{N}(w \mid \mu, \Sigma) \) with

\[
\Sigma = (X^TX + A)^{-1} \quad \text{with} \quad A = \text{diag}(\lambda_1, \ldots, \lambda_p), \tag{B.1}
\]

\[
\mu = \Sigma \alpha X^Ty; \tag{B.2}
\]

(ii) \( p(\lambda_k \mid \Theta - \{\lambda\}) \propto \prod_{j=1}^{K} \Gamma(\lambda_k \mid l_{1,k}, l_{2,k}) \) with

\[
l_{1,k} = \lambda_{1,k} + \frac{1}{2} \sum_{j=1}^{p} \delta(z_j = k), \tag{B.3}
\]

\[
l_{2,k} = \lambda_{2,k} + \frac{1}{2} \sum_{j=1}^{p} \delta(z_j = k) w_j^2; \tag{B.4}
\]
(iii) \( p(\alpha | \Theta - \{\alpha\}) \propto \Gamma(a_1, a_2) \) with
\[
a_1 = a_1 + \frac{n}{2}, \quad (B.5)
\]
\[
a_2 = a_2 + \frac{1}{2} (y - X \mu)^T (y - X \mu); \quad (B.6)
\]
(iv) \( p(z_j | \Theta - \{z\}) \propto \text{mult}(\exp \rho_{j,1}, \ldots, \exp \rho_{j,k}) \) with
\[
\rho_{j,k} = -\frac{1}{2} w_j^2 \lambda_k + \ln(\pi_k) + \frac{1}{2} \log \lambda_k; \quad (B.7)
\]
(v) \( p(\pi_k | \Theta - \{\pi\}) \propto \text{Dir}(d_k) \) with
\[
d_k = \eta_k + \sum_{j=1}^p \delta(z_j = k). \quad (B.8)
\]

The algorithm is provided in pseudocode in Algorithm 2.

Acknowledgment

The authors acknowledge support from the ANR Grant ViMAGE ANR-08-BLAN-0250-02.

References


