# Improving Group Lasso for high-dimensional categorical data

Szymon Nowakowski[1,2], Piotr Pokarowski[1],
Wojciech Rejchel[3], and Agnieszka Sołtys[1]

[1] Institute of Applied Mathematics and Mechanics, University of Warsaw, Warsaw, Poland sd.nowakowski2@uw.edu.pl, pokar@mimuw.edu.pl, agnieszkaprochenka@gmail.com

[2] Faculty of Physics, University of Warsaw, Warsaw, Poland

[3] Faculty of Mathematics and Computer Science, Nicolaus Copernicus University, Toruń, Poland, wrejchel@gmail.com

**Abstract.** Sparse modeling or model selection with categorical data is challenging even for a moderate number of variables, because roughly one parameter is needed to encode one category or level. The Group Lasso is a well known efficient algorithm for selection of continuous or categorical variables, but all estimates related to a selected factor usually differ. Therefore, a fitted model may not be sparse, which makes the model interpretation difficult. To obtain a sparse solution of the Group Lasso, we propose the following two-step procedure: first, we reduce data dimensionality using the Group Lasso; then, to choose the final model, we use an information criterion on a small family of models prepared by clustering levels of individual factors. In the consequence, our procedure reduces dimensionality of the Group Lasso and strongly improves interpretability of the final model. What is important, this reduction results only in the small increase of the prediction error. In the paper we investigate selection correctness of the algorithm in a sparse high-dimensional scenario. We also test our method on synthetic as well as the real data sets and show that it outperforms the state of the art algorithms with respect to the prediction accuracy, model dimension and execution time. Our procedure is contained in the R package `DMRnet` and available in the CRAN repository.

**Keywords:** Information criterion · Model reduction · Penalized likelihood · Regression · Sparse prediction

## 1 Introduction

Data sets containing categorical variables (factors) are common in statistics and machine learning. Sparse modeling for such data is much more challenging than for those having only numerical variables. There are two main reasons for that:
(i) a factor with $k$ levels is usually encoded as $k-1$ dummy variables, so $k-1$ parameters are needed to learn it,
(ii) a reduction of the dimensionality for a factor is much more sophisticated than

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_47

for a numerical predictor (leave or delete), because we can either delete this factor or merge some of its levels (a number of possibilities grows very quickly with a number of levels). Let us consider as the example the factor corresponding to a continent that a person (client, patient etc.) lives on, or a company (university etc.) is located. This factor has 6 levels (Antarctica is not considered), so there are 203 possibilities to merge its levels (they are usually called *partitions*).

It is really difficult to develop efficient algorithms for categorical data and investigate their statistical properties, when a number of factors and/or a total number of their levels is large. Thus, this topic has not been intensively studied so far and the corresponding literature has been relatively modest. However, categorical data are so common that it had to change. We have found many papers investigating categorical data from the last few years, among others [7], [15], [17], [18], [19], [21].

In the paper we consider high-dimensional selection and sparse prediction for categorical data, where a number of active variables is significantly smaller than a learning sample size $n$ and a number of all variables $p$ significantly exceeds $n$. The goal is to develop a procedure, whose outputs have small prediction errors and easy interpretation. For categorical predictors the latter property means that all non-active factors should be discarded. What is more, if an active factor contains equal levels, then they should be merged.

Neural networks or random forests have good predictive properties, but their outputs are difficult to interpret. They often fail when $n$ is small and/or $p$ is larger than $n$ (cf. the experimental results in Section 4). In the paper we focus on such scenarios and consider penalized likelihood methods as a family of Lasso algorithms, which are commonly used for sparse prediction. Some of the methods can discard non-active predictors for high-dimensional data, but they cannot merge the levels of active factors, which strongly limits their intepretability. For instance, the Lasso [22] treats dummy variables as separate, binary predictors, the Group Lasso [25] can only leave or delete a whole factor and the Sparse Group Lasso [20] additionally removes levels of selected factors. The Fused Lasso [23] can merge levels, but only in a simplified case when variables are ordered. These methods significantly reduce a number of parameters, but they do not realize *partition selection*, i.e. they cannot choose models consisting of subsets of numerical variables and partitions of levels of factors.

In the mainstream research on the Lasso-type algorithms, the CAS-ANOVA method [3] fits sparse linear models with the fusion of factor levels using the $l_1$ penalty imposed on the differences between the parameters corresponding to levels of a factor. The implementation of CAS-ANOVA has been provided in [8] and [14]. An alternative approach is a greedy search algorithm, called DMR, from [13]. The growing interest in partition selection has been noticed recently. In [15] a Bayesian method for linear models is introduced based on a prior inducing fusion of levels. Another approach trying to solve the problem from the Bayesian perspective is considered in [7]. The frequentist method using the linear mixed models was presented in [19], where factors were treated as random effects. A partition selection algorithm called SCOPE, which is based on a regularized

DOI: 10.1007/978-3-031-36021-3_47

likelihood, can be found in [21]. This procedure uses a minimax concave penalty on differences between consecutive, sorted estimators of coefficients for levels of a factor. Finally, tree-based algorithms are applied to categorical data in [18].

Let us note that all the aforementioned partition selection methods are restricted to a classical scenario $p < n$, except SCOPE and DMR. The new implementation of the latter is based on variables screened by the Group Lasso [16]. In this paper, we present an improved as well as simplified version of the DMR algorithm, called PDMR (Plain DMR). Our main contributions are as follows:

1. We propose the following two-step PDMR procedure: first, we reduce data dimensionality using the Group Lasso. Next, a small family of models is constructed by clustering levels of individual factors. The final model is chosen from the family using an information criterion. In DMR the Group Lasso model is refitted and dissimilarity matrices in clustering are computed by likelihood ratio statistics. In PDMR we eliminate this step by employing the Group Lasso coefficients, which simplifies and improves the DMR. The PDMR not only works better in numerical experiments, but we are able to mathematically confirm its properties (Theorem 1).

2. We prove in Theorem 1 that PDMR returns a sparse linear model containing the true model, even if $p >> n$. The proof is based on a new bound of a number of partitions by generalized Poisson moments. It is worth to note that so far there are no theoretical results regarding the correctness of the DMR selection for high-dimensional data, while for SCOPE a weaker property than selection consistency was proved in [21]. Our result is also weaker than selection consistency, but it relates directly to any output of our algorithm, while results from [21, Theorem 6] concern only one of blockwise optima of their objective function. We discuss this issue in Section 3.

3. In theoretical considerations, the Lasso-type algorithms are defined for one penalty and return one estimator. However, practical implementations usually use nets of data-driven penalties and return lists of estimators. Our next contribution is an analogous implementation of PDMR. In numerical experiments on simulated and real data, we compare PDMR, DMR and SCOPE. We show that PDMR performs better than SCOPE and DMR with respect to a prediction error and model simplicity/sparsity. Moreover, PDMR is computationally faster than DMR and several dozen times faster than SCOPE. Our procedure is contained in the R package `DMRnet` [16] and available in the CRAN repository.

In the rest of this paper we describe the considered models and the PDMR algorithm. The main theoretical result, which establishes properties of our method, is given in Section 3. Finally, we compare PDMR to the other methods in numerical experiments. The proof of Theorem 1, auxiliary theoretical results and additional descriptions of experiments are relegated to the online supplement[4].

[4] https://github.com/SzymonNowakowski/ICCS-2023

DOI: 10.1007/978-3-031-36021-3_47

## 2 Linear models and the algorithm

We consider independent data $(y_1, x_{1.}), (y_2, x_{2.}), \dots, (y_n, x_{n.})$, where $y_i \in \mathbb{R}$ is a response variable and $x_{i.} \in \mathbb{R}^p$ is a vector of predictors. Every vector of predictors $x_{i.}$ can consist of continuous predictors as well as categorical predictors. We arrange them in the following way $x_{i.} = (x_{i1}^T, x_{i2}^T, \dots, x_{ir}^T).^T$ Suppose that subvector $x_{ik}$ corresponds to a categorical predictor (factor) for some $k \in \{1, \dots, r\}$. Let a set of levels of this factor be given by $\{0, 1, 2, \dots, p_k\}$. In that case we usually use $x_{ik} \in \{0, 1\}^{p_k}$, so $x_{ik}$ is a dummy vector corresponding to the $k$-th predictor of the $i$-th object in a data set. Notice that we do not include a reference level (say, the zero level) in $x_{ik}$. The only exception relates to the first factor, whose reference level is contained in $x_{i1}$. This special level plays a role of an intercept. If necessary, we can rearrange vectors of predictors to have the first factor with $k = 1$. If $x_{ik}$ corresponds to a continuous predictor, then simply $x_{ik} \in \mathbb{R}^{p_k}$ and $p_k = 1$. Therefore, a dimension of $x_{i.}$ is $p = 1 + \sum_{k=1}^{r} p_k$. Finally, let $X = [x_{1.}, \dots, x_{n.}]^T$ be a $n \times p$ design matrix and by $x_{j,k}$ we denote its column corresponding to the $j$-th level of the $k$-th factor.

We consider a linear model

$$y_i = x_{i.}^T \mathring{\beta} + \varepsilon_i \quad \text{for} \quad i = 1, 2, \dots, n. \tag{1}$$

Coordinates of $\mathring{\beta}$ correspond to coordinates of a vector of predictors, that is $\mathring{\beta} = (\mathring{\beta}_1^T, \mathring{\beta}_2^T, \dots, \mathring{\beta}_r^T)^T$, where we have $\mathring{\beta}_1 = (\mathring{\beta}_{0,1}, \mathring{\beta}_{1,1}, \dots, \mathring{\beta}_{p_1,1})^T \in \mathbb{R}^{p_1+1}$ and $\mathring{\beta}_k = (\mathring{\beta}_{1,k}, \mathring{\beta}_{2,k}, \mathring{\beta}_{3,k}, \dots, \mathring{\beta}_{p_k,k})^T \in \mathbb{R}^{p_k}$ for $k = 2, \dots, r$. Moreover, we suppose that noise variables $\varepsilon_i$ have a *subgaussian distribution* with the same parameter $\sigma > 0$, that is for $i = 1, 2, \dots, n$ and $u \in \mathbb{R}$ we have

$$\mathbb{E} \exp(u\varepsilon_i) \le \exp(\sigma^2 u^2 / 2). \tag{2}$$

The main examples of subgaussian noise variables are normal variables or those having bounded supports.

### 2.1 Notations

For $\beta \in \mathbb{R}^p$ and $q \ge 1$ let $|\beta|_q = (\sum_{j=1}^{p} |\beta_j|^q)^{1/q}$ be the $\ell_q$ norm of $\beta$. The only exception is the $\ell_2$ norm, for which we use the special notation $||\beta||$.

A feasible model is defined as a sequence $M = (P_1, P_2, \dots, P_r)$. If the $k$-th predictor is a factor, then $P_k$ is a particular partition of its levels. If the $k$-th predictor is continuous, then $P_k \in \{\emptyset, \{k\}\}$. To make the notation coherent and concise we *artificially* augment each $\beta \in \mathbb{R}^p$ by $\beta_{0,k} = 0, k = 2, \dots, r$. Notice that every $\beta$ determines a model $M_\beta$ : if the $k$-th predictor is a factor, then partition $P_k$ is induced by equalities of coefficients, i.e. $\beta_{j_1,k} = \beta_{j_2,k}, j_1 \neq j_2$ means that levels $j_1$ and $j_2$ belong to the same cluster of the $k$-th factor. If the $k$-th predictor is continuous, then $P_k = \{k\}$ when $\beta_k \neq 0$ and $P_k = \emptyset$ otherwise.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_47

### 2.2 The algorithm

To simplify notations (and without losing the generality), we suppose that all considered predictors are categorical. For estimation of $\mathring{\beta}$ we consider a quadratic loss function as in the maximum likelihood estimation:

$$\ell(\beta) = \frac{1}{n}\sum_{i=1}^{n}[(x_{i.}^T\beta)^2/2 - y_i x_{i.}^T\beta]. \tag{3}$$

We present the PDMR algorithm, which consists of two steps:
**(1) Screening**: we compute the weighted Group Lasso

$$\hat{\beta} = \arg\min_{\beta} \ell(\beta) + \lambda\sum_{k=1}^{r}||W_k\beta_k||,$$

where $W_k$ is a diagonal matrix with $(W_k)_{jj} = ||x_{j,k}||/\sqrt{n}$ playing roles of weights. Such a choice of weights was suggested in the seminal paper [25]. It is also explained in Proposition 1 in the online supplement. Number $\lambda > 0$ is a tuning parameter whose choice is discussed in Section 3;
**(2) Selection**: this step is divided into three parts:
**(2a) construction of the nested family of models $\mathcal{M}$:** let $\hat{S} = \{1 \leq k \leq r : \hat{\beta}_k \neq 0\}$ and $\hat{\beta}_{0,k} = 0$ for $k \in \hat{S} \setminus \{1\}$. So, $\hat{S}$ is a set of factors which are not discarded by the Group Lasso. For each $k \in \hat{S}$ we separately perform complete linkage hierarchical clustering of levels of those factors. Each clustering starts with a dissimilarity matrix $(D_k)_{j_1,j_2} = |\widehat{\beta}_{j_1,k} - \widehat{\beta}_{j_2,k}|, 0 \leq j_1, j_2 \leq p_k$. This matrix is consecutively updated as follows: a distance between two clusters $A$ and $B$ of levels of the $k$-th factor is defined as $\max_{a\in A, b\in B}|\hat{\beta}_{a,k} - \hat{\beta}_{b,k}|$. Each clustering begins with disjoint factor's levels and then two most *similar* clusters are merged. Finally, we obtain the *empty* factor with all levels merged. Cutting heights from this clustering are contained in $h_k^T$. Then we create a vector $h$, which consists of elements of a vector $(0, h_1^T, h_2^T, \ldots, h_{\hat{S}}^T)^T$ sorted increasingly. Next, we construct a family $\mathcal{M} = \{M_0 = \hat{S}, M_1, M_2, \ldots, \{\emptyset\}\}$, where $M_{j+1}$ is $M_j$ with one additional merging of appropriate clusters corresponding to the $(j+1)$-th element in $h$,
**(2b) Generalized Information Criterion**

$$\hat{M}_{PDMR} = \arg\min_{M\in\mathcal{M}} \quad \ell(\hat{\beta}_M) + \lambda^2/2|M|,$$

where $\hat{\beta}_M$ is a minimum loss estimator over $\mathbb{R}^p$ with constraints determined by a model $M$ and $|M|$ equals to a number of distinct levels in $M$. Technical details of this constrained minimization is given in Section 3 of the online supplementary materials. We also show there that it can be considered as an unconstrained minimization over a smaller space.
**(2c) Estimation of parameters in the model $\hat{M}_{PDMR}$**:

$$\hat{\beta}_{PDMR} = \arg\min_{\beta_{\hat{M}_{PDMR}}} \quad \ell(\beta_{\hat{M}_{PDMR}})$$

To study data with binary responses we can extend the PDMR algorithm to logistic regression. From the practical point of view this generalization is relatively easy. We apply the Group Lasso for logistic regression with $\ell(\beta) = \sum_{i=1}^{n} [\log(1+\exp(x_{i.}^T\beta)) - y_i x_{i.}^T\beta]/n$. A similar modification should be done when applying the information criterion step. We have contained it in the R package `DMRnet` and apply it in Section 4. This extension is more difficult from the theoretical perspective, so the result in Section 3 is restricted to linear models (1).

## 3 Statistical properties of PDMR

In this section we state the main theoretical result concerning our algorithm.

Assume that $M_{\mathring{\beta}}$ is the true model, i.e. the one which is determined by the true parameter $\mathring{\beta}$. In Theorem 1 we establish that our procedure is able to *partially* find $M_{\mathring{\beta}}$. It is possible only if $M_{\mathring{\beta}}$ is sufficiently distinguishable from other models, which is sometimes called *identifiability* of $M_{\mathring{\beta}}$. This issue is quite involved, so we move its precise description to Section 2 in the online supplement. Here, the identifiability of $M_{\mathring{\beta}}$ is expressed by a positive value $\kappa$ such that the larger $\kappa$ is, the model $M_{\mathring{\beta}}$ is easier to identify. Finally, $|M_{\mathring{\beta}}|$ denotes a number of distinct levels in $M_{\mathring{\beta}}$.

**Theorem 1.** *Suppose that assumptions* (1) *and* (2) *are satisfied and there exists* $0 < a < 1$ *such that*

$$\frac{2a^{-2}\sigma^2 \log p}{n} \leq \lambda^2 < \frac{\kappa}{16(1+a)^2}. \tag{4}$$

*Then*

$$P(\hat{M}_{PDMR} \subsetneq M_{\mathring{\beta}}) \leq 3p \exp\left(-\frac{a^2 n \lambda^2}{2\sigma^2}\right). \tag{5}$$

Theorem 1 states that PDMR computes consistent screening, which means that with high probability it is able to reduce a model returned by the Group Lasso without losing any active variables. The only parameter of PDMR can be chosen as $\lambda^2 = 2a^{-2}\sigma^2 \log p(1+q)/n$ for some $q > 0$. Then the right-hand side in (5) behaves like $1/p^q$. So, the larger $q$ is, the faster probability in (5) goes to 0. However, increasing $q$ restricts the usefulness of Theorem 1 only to $M_{\mathring{\beta}}$ having large $\kappa$. Therefore, the reasonable choice of $q$ would be a small value which still ensures that probability in (5) goes to zero, for instance $q \to 0$ but $q \log p \to \infty$. Notice that it is the same choice of $\lambda$ as in the *Risk Inflation Criterion* [5].

PDMR can successfully work in the case that $p$ is large. From (4) we see that $\kappa$ has to be larger than $\log(p)/n$, which goes to zero even if $p = \exp(n^\alpha), \alpha < 1$ for $n \to \infty$. So, as $n$ increases, Theorem 1 can be applied to data having smaller value of $\kappa$. Notice that there are no similar theoretical results for other partition selection competitors of PDMR in high-dimensions. Guarantees for DMR from [13] relates only to the $p < n$ scenario. In [21, Theorem 6] it is shown that the output of SCOPE and the true model are some blockwise optima of the SCOPE objective function. This conclusion is very weak, because there might be plenty of such blockwise optima. Consider the function $f(x, y) = |x - y| + |x + y|$, which has one global minimum $(0, 0)$. Notice that the point $(100, 100)$ is one of the blockwise optima of $f$ but it is useless when estimating $(0, 0)$. Therefore, the output of SCOPE might be very poor estimator of $M_{\mathring{\beta}}$, while Theorem 1 states that *any* output of PDMR computes consistent screening in high-dimensions.

Calculating a value of $\kappa$ is difficult. In Section 2.1 of the online supplement we show that for the special case that $X^T X$ is an orthogonal matrix, $\kappa$ can be determined and condition (4) leads to $\Delta^2 \succeq \sigma^2 \log p \max_k p_k/n$, where $\Delta = \min\limits_{1\leq k\leq r} \min\limits_{0\leq j_1, j_2 \leq p_k : \mathring{\beta}_{j_1,k} \neq \mathring{\beta}_{j_2,k}} |\mathring{\beta}_{j_1,k} - \mathring{\beta}_{j_2,k}|$. The value of $\Delta$ states how much distinct levels belonging to the same factor in $M_{\mathring{\beta}}$ differ. Therefore, this condition shows a clear relation between characteristics of the data (i.e. $n, p, p_k, \sigma, \Delta$) that gives the sufficient distinguishability of $M_{\mathring{\beta}}$.

The proof of Theorem 1 is given in Section 2 in online supplementary materials. It can be sketched as follows: first we establish that the Group Lasso is a consistent estimator, which implies that the family $\mathcal{M}$, defined in the step (2a) of PDMR, contains the true set $M_{\mathring{\beta}}$. Then we establish that the probability of choosing a submodel of $M_{\mathring{\beta}}$ can be expressed as a Touchard polynomial and estimated using the recent combinatorial results from [1]. Proving an upper bound on $P(M_{\mathring{\beta}} \subsetneq \hat{M}_{PDMR})$ still remains an open problem.

## 4 Experiments

We start with the implementation details of PDMR and competing procedures. Then simulated and real data sets are investigated.

In the theoretical analysis of Lasso-type estimators one usually considers only one value of the tuning parameter $\lambda$. We have also followed this way in Section 3. However, the practical implementations can efficiently return estimators for a data driven net of $\lambda$'s, as in the R package `glmnet` [6]. Similarly, using a net of $\lambda$'s, the Group Lasso and the Group MCP algorithms have been implemented in the R package `grpreg` [4]. We also propose a net modification of PDMR:
1. For $\lambda$ belonging to the grid: calculate the Group Lasso estimator $\hat{\beta}(\lambda)$ and then perform complete linkage for each factor and get a nested family of models $M_1(\lambda) \subset M_2(\lambda) \subset \ldots$
2. For a fixed model dimension $c$, select a model $M_c$ from the family $(M_c(\lambda))_\lambda$, which has the minimal prediction loss.
3. Select a final model from the sequence $(M_c)_c$ using the Risk Inflation Criterion (RIC), see [5], i.e. using the tuning parameter $2\sigma^2 \log(p)/n$, which is the same as suggested in Theorem 1. Obviously, $\sigma$ is unknown, but it can be quite easily estimated: as usual we take an appropriately scaled residual sum of squares on a model returned by the Group Lasso.

The above implementation of PDMR is available in the `DMRnet` package starting with its 0.3.4 version with `algorithm="PDMR"` argument.

In experiments we also evaluate the following methods:
(i) DMR, which refits $\hat{\beta}$ after the Group Lasso with the maximum likelihood and computes dissimilarity matrices in the clustering step as the likelihood ratio statistics. It is also contained in the `DMRnet` package,
(ii) Group Lasso (gL) with a tuning parameter `lambda` chosen by cross-validation as implemented in `cv.grpreg` from the `R` package `grpreg` with `penalty="grLasso"`,
(iii) Group MCP (gMCP) with a tuning parameter `lambda` chosen by cross-validation as implemented in `cv.grpreg` from the `R` package `grpreg` with `gamma` set as the default and `penalty="grMCP"`,
(iv) SCOPE from the R package `CatReg` [21]. A tuning parameter `gamma` is chosen as 8 or 32, which is suggested in that paper. We denote them as S-8, S-32, respectively. For real data we consider also the case of binary responses and then `gamma` is 100 or 250 as in [21]. We denote them S-100 and S-250, respectively,
(v) Random Forest (RF) - we use the `randomForest` function from the `R` package `randomForest`.

All results presented in this section can be reproduced using our codes, which are publicly available at *https://github.com/SzymonNowakowski/ICCS-2023.*

### 4.1 Simulation study

This section contains the experiments with simulated high-dimensional linear models. Design matrices $X$ and parameter vectors $\mathring{\beta}$ are the same as in [21], but additionally we systematically change the signal to noise ratio (SNR).

In the training data we have $n = 500$ observations. Every vector of predictors consists of $r = 100$ factors, each with 24 levels. Thus, after deleting 99 reference levels we obtain $p = 2301$. A design matrix $X$ is generated as in [21], namely: first, we draw matrix $Z$, whose rows $z_{i.}, i = 1, \ldots, 500$ are independent 100-dimensional vectors having normal distribution $N(0, \Sigma)$. The off-diagonal elements of $\Sigma$ are chosen such that correlation between $\Phi(z_{ij})$ and $\Phi(z_{ik})$ equals 0.5 for $j \neq k$, where $\Phi$ is a cdf of the standard normal distribution. Then we set $x_{ij} = \lceil 24\Phi(z_{ij}) \rceil$. Finally, $X$ is recoded into dummy variables and reference levels of each factors, except the first one, are deleted.

Errors $\varepsilon_i$ are independently distributed from $N(0, \sigma^2)$, where $\sigma$ is chosen in such a way to realize distinct SNR values. The performance of the estimators is measured using the root-mean-square errors (RMSE), which are calculated using the test data consisting of $10^5$ observations. To work on the universal scale we divide the RMSE of procedures by the RMSE of the oracle, which knows the true model in advance, i.e. a maximum likelihood estimator computed for the true model. Final results are averages over 200 draws of training and testing data. We consider the following six models, which are the same as in [21, Section 6.1.2]. However, we renumber them with respect to true model dimensions (True MD), i.e. a number of distinct levels among their consecutive factors in $M_{\mathring{\beta}}$ (recall that we do not count reference levels of factors, except the first one):

**Setting 1**: $\mathring{\beta}_k = (\underbrace{0, \ldots, 0}_{7 \text{ times}}, \underbrace{2, \ldots, 2}_{8 \text{ times}}, \underbrace{4, \ldots, 4}_{8 \text{ times}})$ for $k = 2, 3$, $\mathring{\beta}_k = (\underbrace{0, \ldots, 0}_{15 \text{ times}}, \underbrace{5, \ldots, 5}_{8 \text{ times}})$

for $k = 4, 5, 6$, $\mathring{\beta}_1 = (0, \mathring{\beta}_2)$, and $\mathring{\beta}_k = 0$ otherwise. So, True MD=10,
**Setting 2**: $\mathring{\beta}_k = (\underbrace{0, \ldots, 0}_{7 \text{ times}}, \underbrace{2, \ldots, 2}_{8 \text{ times}}, \underbrace{4, \ldots, 4}_{8 \text{ times}})$ for $k = 2, 3$, and for $k = 4, 5, 6$ we have $\mathring{\beta}_k = (\underbrace{0, \ldots, 0}_{9 \text{ times}}, \underbrace{2, \ldots, 2}_{4 \text{ times}}, \underbrace{4, \ldots, 4}_{10 \text{ times}})$, $\mathring{\beta}_1 = (0, \mathring{\beta}_2)$, and $\mathring{\beta}_k = 0$ otherwise. So, True MD=13,
**Setting 3**: $\mathring{\beta}_k = (\underbrace{0, \ldots, 0}_{5 \text{ times}}, \underbrace{2, \ldots, 2}_{6 \text{ times}}, \underbrace{4, \ldots, 4}_{6 \text{ times}}, \underbrace{6, \ldots, 6}_{6 \text{ times}})$ for $k = 2, ..., 5$, $\mathring{\beta}_1 = (0, \mathring{\beta}_2)$, and $\mathring{\beta}_k = 0$ otherwise. So, True MD=16,
**Setting 4**: $\mathring{\beta}_k = (\underbrace{0, \ldots, 0}_{4 \text{ times}}, \underbrace{1, \ldots, 1}_{5 \text{ times}}, \underbrace{2, \ldots, 2}_{4 \text{ times}}, \underbrace{3, \ldots, 3}_{5 \text{ times}}, \underbrace{4, \ldots, 4}_{5 \text{ times}})$ for $k = 2, ..., 5$, $\mathring{\beta}_1 = (0, \mathring{\beta}_2)$, and $\mathring{\beta}_k = 0$ otherwise. So, True MD=21,
**Setting 5**: $\mathring{\beta}_k = (\underbrace{0, \ldots, 0}_{3 \text{ times}}, \underbrace{2, \ldots, 2}_{12 \text{ times}}, \underbrace{4, \ldots, 4}_{8 \text{ times}})$ for $k = 2, \ldots, 10$, $\mathring{\beta}_1 = (0, \mathring{\beta}_2)$, and $\mathring{\beta}_k = 0$ otherwise. So, True MD=21,
**Setting 6**: $\mathring{\beta}_k = (\underbrace{0, \ldots, 0}_{15 \text{ times}}, \underbrace{5, \ldots, 5}_{8 \text{ times}})$ for $k = 2, ..., 25$, $\mathring{\beta}_1 = (0, \mathring{\beta}_2)$, and $\mathring{\beta}_k = 0$ otherwise. So, True MD=26.
In [21, Section 6.1.2] the above Settings 1-6 are numbered as 3, 1, 8, 4, 7 and 5, respectively. In each model we consider distinct SNR values, in particular Settings 2 and 6 from that paper are also studied.

The results of experiments are presented in Figures 1 and 2. In the former we present the RMSE of procedures divided by the RMSE of the oracle. On the $x$-axis there are SNR values for the interval $[1, 5]$. In the second figure we observe numbers of distinct levels that are recognized by procedures (model dimension, MD) divided by a number of distinct level in the true model (i.e. True MD). Recall that the goal is to find an easily interpretable model (i.e. MD should be small), which has also good predictive properties. From Figures 1 and 2 we observe that PDMR is the clear winner. Notice that MD of PDMR is similar to SCOPE-8 and smaller than for other competitors. However, RMSE for SCOPE-8 is larger than for PDMR in Figure 1. Besides, predictive properties of PDMR are the best (Scenario 1-2) or close to the best (Scenario 3-5). The main competitors of PDMR with respect to prediction are SCOPE-32, Group Lasso and Group MCP. However, Group Lasso and Group MCP do not return interpretable model. Their MD is more than 7 times larger than True MD, so they are not presented in Figure 2 (except Scenario 4). SCOPE-32 has similar prediction errors to PDMR but it is worse in model interpretation, because its MD is usually 2-3 times larger than for PDMR. Finally, notice that Random Forests fail in our experiments, which confirms the fact that this procedure often works poorly with data having many unknown parameters and small sample sizes.

In Scenario 6 PDMR works worse than for other settings. It comes from the fact that Scenario 6 does not satisfy the assumptions required for the Group Lasso operation, i.e. the true factors have more parameters in total than the

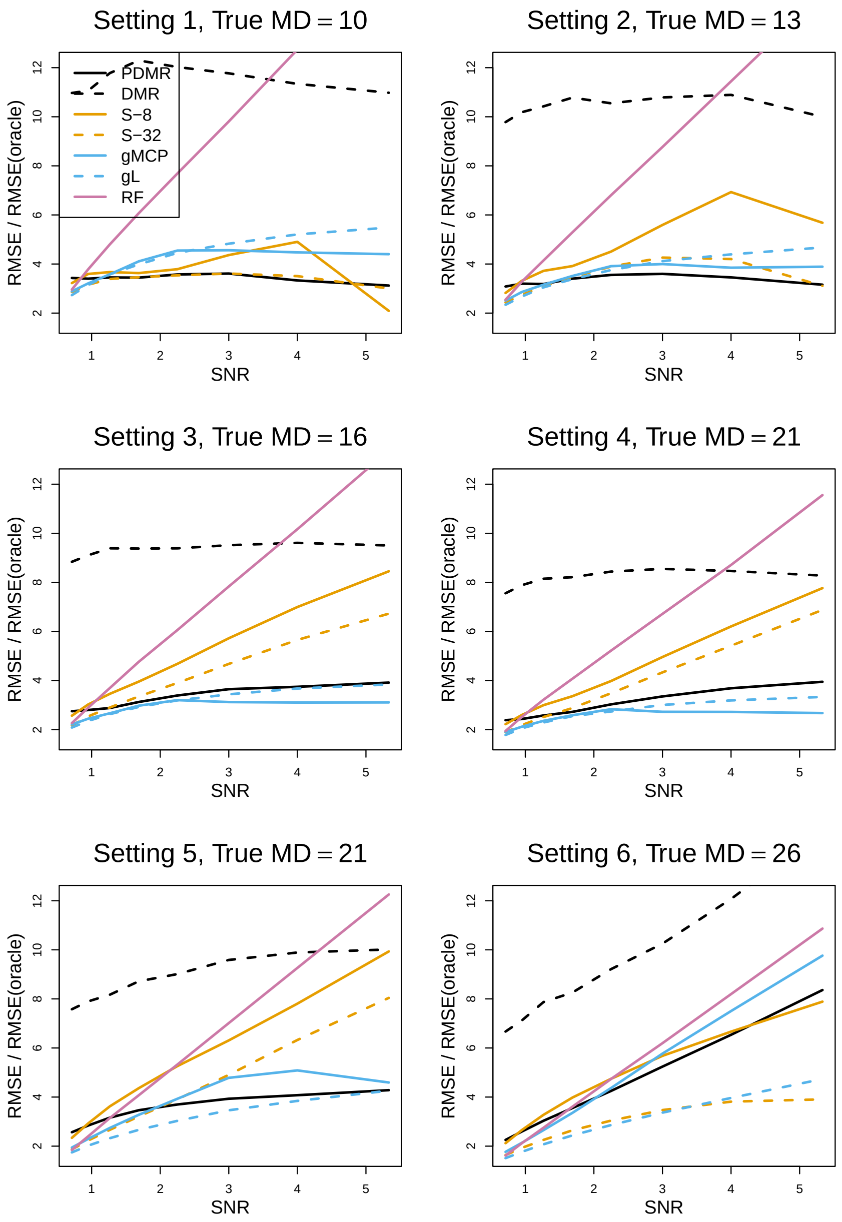

**Fig. 1.** Relative prediction errors of the considered methods in six settings. SNR - the signal to noise ratio, RMSE - the root-mean-square error. The remaining details are given in the text.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_47

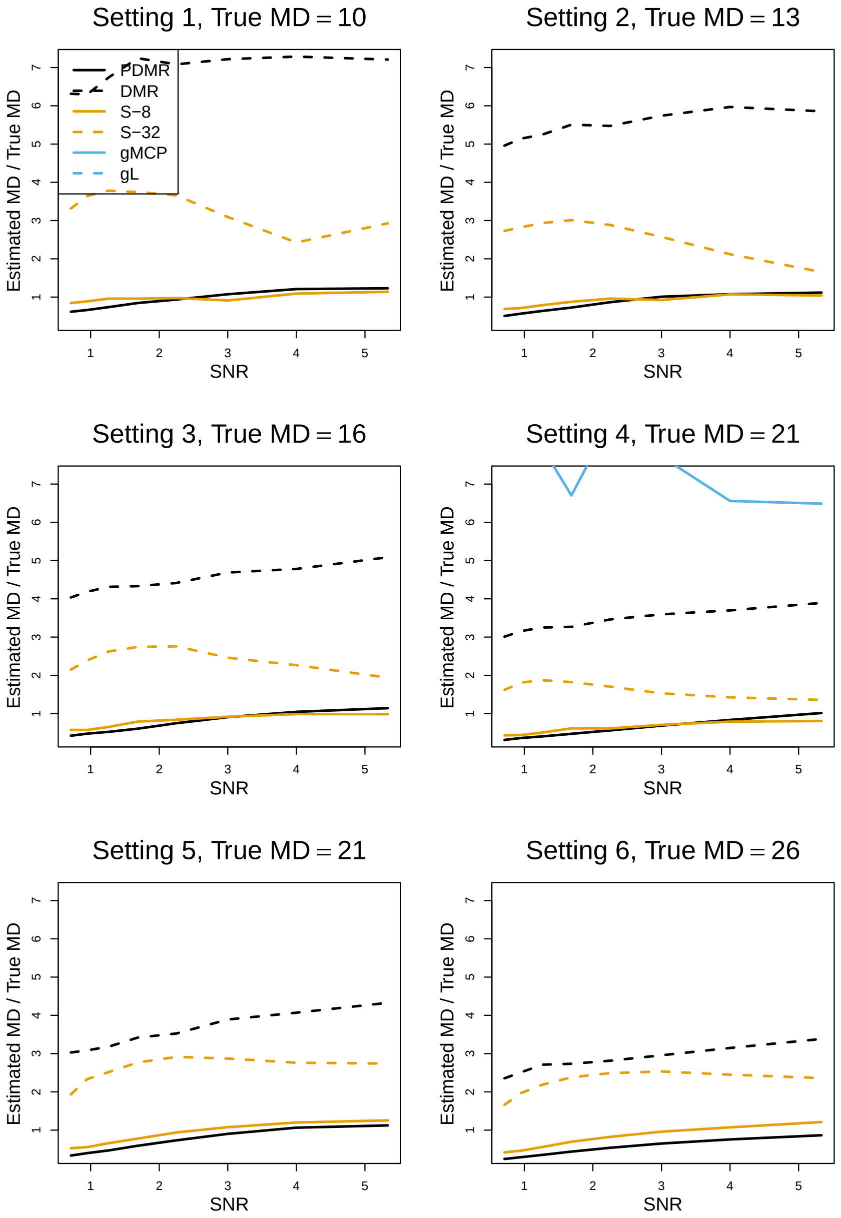

**Fig. 2.** Relative model dimensions (MD) of the considered methods in six settings. SNR - the signal to noise ratio. The remaining details are given in the text.

number of observations ($n = 500$). This loss in the prediction quality is observed not only for PDMR, but also for the other methods.

### 4.2 Real data study

We investigate five real data sets: the first two with binary responses and the next three with continuous responses. Here we present only short descriptions of these data sets. A full discussion is given in Section 5 in the online supplement. The preprocessing step is also thoroughly explained there.
**1.** The Adult data set [11] contains data from the 1994 US census. Preprocessing results in $n = 45,222$ with 2 continuous and 8 categorical variables with $p = 93$;
**2.** The Promoter data set [9], [24] contains E. Coli genetic sequences of length 57. The data set consists of 106 observations with 57 categorical variables, each with 4 levels representing 4 nucleotides, thus with $p = 172$;
**3.** The Airbnb data set is available from *insideairbnb.com.* Preprocessing results in $n = 49,976$ with 765 variables (out of which 3 are categorical) with $p = 40668$;
**4.** The Insurance data set [10] contains a response, which is an 8-level ordinal variable measuring insurance risk of an applicant. We treat as continuous. Preprocessing results in 59,381 observations with 5 continuous and 108 categorical variables with $p = 823$;
**5.** The Antigua data set [2] is available at the `R` package `DAAG` [12]. It consists of 287 observations with 1 continuous and 4 categorical variables with $p = 58$.

We conduct experiments analogously to the ones described in Section 4.1: each data set is divided (200 times repeated) into a training, and testing sets. The training set contains 70% of observations for Promoter and Antigua. To adapt the remaining data to the case $p >> n$ we take only $n_i$ observations to their training sets in the $i$-th repetition. We present values of $n_i$'s in Table 1. We also show the values of $p_i$'s, which are new parameter space sizes in the $i$-th repetition. Notice that $p_i < p$, because predictors constant for a given training set get dropped. Estimators are computed on the training sets and their prediction errors (PE) are calculated on testing sets. For continuous responses PE is RMSE as in Section 4.1, while for binary responses PE is a misclassification error. PE and MD of methods are given in Figure 3.

The prediction error of PDMR is similar or slightly worse than of gL, gMCP and RF (for Adult and Promoter), but PDMR is much better in model interpretability. MD of PDMR is at least a few times smaller than for gL and gMCP. PDMR is definitely better than DMR on Insurance and similar on the others. Comparing to SCOPE, PDMR wins on Adult and Promoter, because its MD is smaller and more stable, while their PE are similar. On Antigua and Insurance we have a tie: PDMR has smaller PE, but SCOPE has smaller MD. We did not complete computations for RF on Insurance nor on Airbnb, because `randomForest` function cannot handle categorical predictors with more than 53 levels. We did not complete SCOPE computations on Airbnb for any considered `gamma` value (we tried 4 values: 8, 32, 100 and 250 for `gamma`, all with no success).

Finally, in Table 1 we present median values of execution time (in seconds) for particular procedures. We can observe that PDMR is faster than DMR and

DOI: 10.1007/978-3-031-36021-3_47

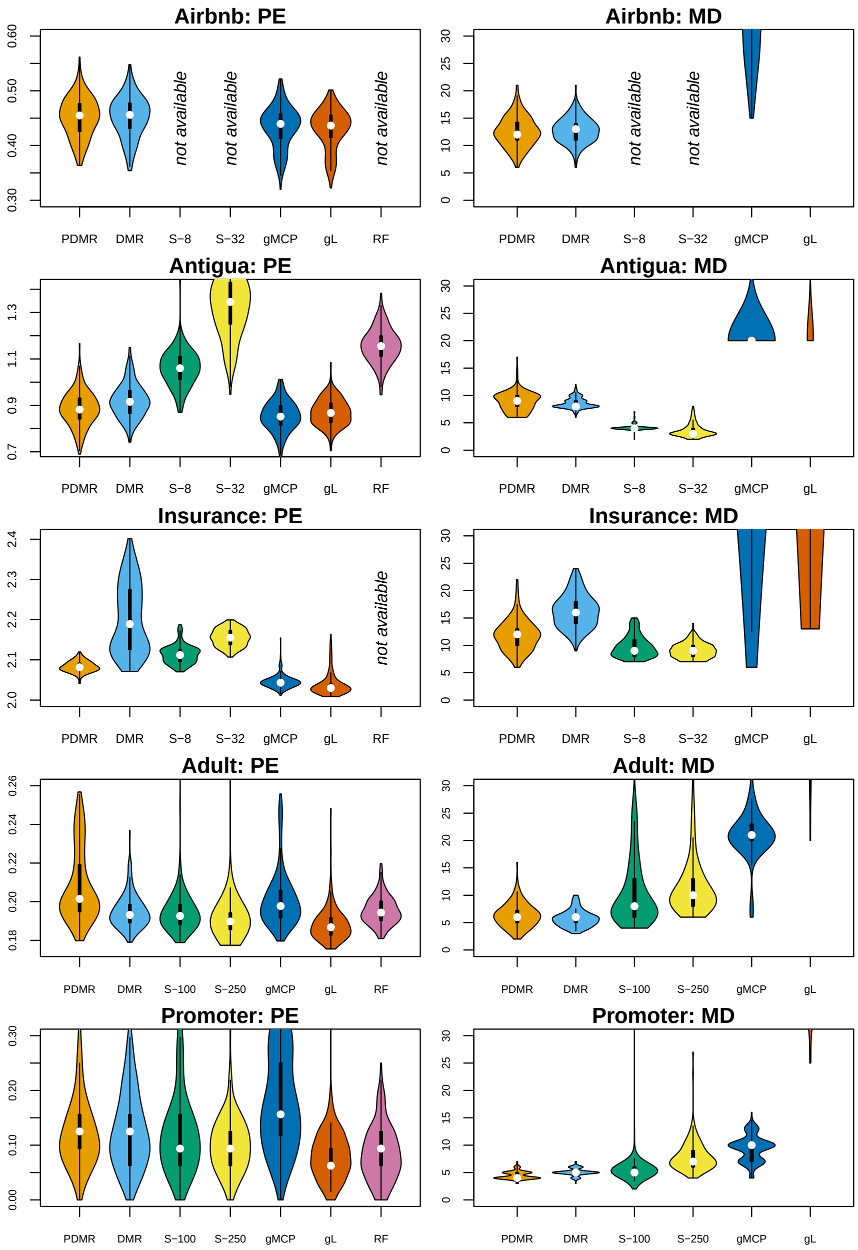

**Fig. 3.** The prediction error (PE) and the model dimension (MD) of the considered methods for five real data sets. The remaining details are given in the text.

**Table 1.** Median execution time of procedures (in seconds) and $n_i$, $p_i$ (real data sets).

| Dataset | $n_i$ | $p_i$ range | PDMR | DMR | S-8 | S-32 |
|---|---|---|---|---|---|---|
| Airbnb | 999 | $1410, \ldots, 1504$ | 5.19 | 5.41 | N/A | N/A |
| Antigua | 200 | 57, 58 | 0.16 | 0.19 | 1.65 | 0.69 |
| Insurance | 1781 | $313, \ldots, 365$ | 46.1 | 113.82 | 27.59 | 25.48 |
| Adult | 452 | $60, \ldots, 78$ | 5.66 | 10.2 | 300.53 | 322.96 |
| Promoter | 74 | 172 | 3.29 | 4.7 | 604.86 | 3616.55 |
| Setting 1, SNR=3 | | | 11.67 | 15.25 | 335.6 | 332.03 |
| Setting 2, SNR=3 | | | 11.57 | 15.56 | 371.91 | 353.68 |
| Setting 3, SNR=3 | | | 11.47 | 15.11 | 352.84 | 370.06 |
| Setting 4, SNR=3 | | | 12.71 | 16.81 | 504.25 | 764.15 |
| Setting 6, SNR=3 | | | 12.17 | 16.49 | 712.94 | 1253.64 |

significantly faster than SCOPE. There is only one data set (Insurance) that execution time of SCOPE is shorter. For simulated data we present results for SNR=3, but they look pretty the same for other SNR values, for instance in Section 5 of the online supplement we show results for SNR=4. For binary response data (Adult and Promoter) columns „S-8" and „S-32" in Table 1 correspond to SCOPE with `gamma` equal to 100 and 250, respectively.

## 5 Conclusions

By merging levels of factors the PDMR algorithm can significantly reduce a dimension of the Group Lasso model with only a small loss of prediction accuracy. PDMR is better than DMR with respect to the prediction error and model dimension. PDMR also simplifies DMR, which enables us to rigorously confirm consistency of PDMR in the high-dimensional scenario (Theorem 1). Such results are not available for DMR nor other partition selection algorithms. Numerical experiments show that PDMR is several dozen times faster than SCOPE and is comparable or better with respect to the prediction error and model dimension.

Our paper can be extended in a few directions. The first one is to prove an analogous bound to that in Theorem 1 but concerning supermodels of $M_{\mathring{\beta}}$. The second one is to generalize this theorem to GLMs. Finally, we obtain optimal weights for the Group Lasso, which are different from those recommended by the authors of this method (cf. Section 2.1 in the supplement). Possibly, the new weights can improve asymptotics of the Group Lasso in the general scenario (not necessarily orthogonal) and its practical performance as well.

## References

1. Ahle, T.D.: Sharp and simple bounds for the raw moments of the binomial and poisson distributions. Statist. Probab. Lett. **182**, 109–306 (2022)
2. Andrews, D.F., Herzberg, A.M.: Data. A Collection of Problems from Many Fields for the Student and Research Worker. Springer, New York (1985)

3. Bondell, H.D., Reich, B.J.: Simultaneous factor selection and collapsing levels in anova. Biometrics **65**, 169–177 (2009)
4. Breheny, P., Huang, J.: Group descent algorithms for nonconvex penalized linear and logistic regression models with grouped predictors. Stat. Comput. **25**, 173–187 (2015)
5. Foster, D., George, E.: The risk inflation criterion for multiple regression. Ann. Statist. **22**, 1947–1975 (1994)
6. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. **33**, 1–22 (2010)
7. García-Donato, G., Paulo, R.: Variable selection in the presence of factors: a model selection perspective. J. Amer. Statist. Assoc. **117**, 1847–1857 (2022)
8. Gertheiss, J., Tutz, G.: Sparse modeling of categorial explanatory variables. Ann. Appl. Stat. **4**, 2150–2180 (2010)
9. Harley, C., Reynolds, R.: Analysis of e. coli promoter sequences. Nucleic Acids Res. **15**, 2343–2361 (1987)
10. Kaggle: Prudential life insurance assessment. https://www.kaggle.com/c/prudential-life-insurance-assessment/data (2015)
11. Kohavi, R.: Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In: Proceedings of KDD. pp. 202–207 (1996)
12. Maindonald, J., Braun, W.: Data Analysis and Graphics Using R. Cambridge University Press (2010)
13. Maj-Kańska, A., Pokarowski, P., Prochenka, A.: Delete or merge regressors for linear model selection. Electron. J. Stat. **9**, 1749 – 1778 (2015)
14. Oelker, M.R., Gertheiss, J., Tutz, G.: Regularization and model selection with categorical predictors and effect modifiers in generalized linear models. Stat. Modelling **14**, 157–177 (2014)
15. Pauger, D., Wagner, H.: Bayesian effect fusion for categorical predictors. Bayesian Anal. **14**, 341–369 (2019)
16. Prochenka-Soltys, A., Pokarowski, P., Nowakowski, S.: DMRnet: Delete or Merge Regressors Algorithms for Linear and Logistic Model Selection and High-Dimensional Data (2022), https://cran.r-project.org/web/packages/DMRnet
17. Rabinowicz, A., Rosset, S.: Cross-validation for correlated data. J. Amer. Statist. Assoc. **117**, 718–731 (2022)
18. Rabinowicz, A., Rosset, S.: Trees-based models for correlated data. J. Mach. Learn. Res. **23**, 1–31 (2022)
19. Simchoni, G., Rosset, S.: Using random effects to account for high-cardinality categorical features and repeated measures in deep neural networks. In: Proceedings of NIPS. pp. 25111–25122 (2021)
20. Simon, N., Friedman, J., Hastie, T., Tibshirani, R.: A sparse-group lasso. J Comput Graph Stat **22**, 231–245 (2013)
21. Stokell, B.G., Shah, R.D., Tibshirani, R.J.: Modelling high-dimensional categorical data using nonconvex fusion penalties. J. Roy. Statist. Soc. Ser. B **83**, 579–611 (2021)
22. Tibshirani, R.: Regression shrinkage and selection via the Lasso. J. Roy. Statist. Soc. Ser. B **58**, 267–288 (1996)
23. Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., Knight, K.: Sparsity and smoothness via the fused lasso. J. Roy. Statist. Soc. Ser. B **67**, 91–108 (2005)
24. Towell, G., Shavlik, J., Noordewier, M.: Refinement of approximate domain theories by knowledge-based artificial neural networks. In: Proceedings of AAAI (1990)
25. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. J. Roy. Statist. Soc. Ser. B **68**, 49–67 (2006)