A Sparse Generative Model and its EM Algorithm for Variable Selection in High-Dimensional Regression

Charles Bouveyron
Université Paris Descartes

Julien Chiquet
Université d'Evry & INRA

Pierre Latouche
Université Paris 1 Panthon-Sorbonne

Pierre-Alexandre Mattei
Université Paris Descartes

Abstract

We address the problem of Bayesian variable selection for high-dimensional linear regression. We consider a generative model that uses a spike-and-slab like prior distribution obtained by multiplying a deterministic binary vector, which traduces the sparsity of the problem, with a random Gaussian parameter vector. Such a model allows an expectation-maximization algorithm, optimizing a type-II log-likelihood, to be derived. This marginal log-likelihood involves an Occam's razor term, automatically penalizing the complexity, which is used for model selection. Albeit NP-hard, the algorithm we propose can be relaxed in order to infer a family of models. Model selection is eventually performed afterwards based on Occam's razor. We report numerical comparisons between our method, called spinyReg, and the most recent variable selection algorithms, including lasso, adaptive lasso and stability selection. SpinyReg turns out to perform well compared to those algorithms, especially regarding false detection rates.

1 Introduction

Over the past decades, parsimony has imposed itself as a very natural way to deal with high-dimensional data spaces. In the context of linear regression, finding a parsimonious parameter vector both prevents overfitting and allows to interpret easily the data by finding which predictors are relevant. The problem of finding such predictors is referred to as *variable selection* or *sparse regression*, and has mainly been considered either by likelihood penalization of the data, or by using Bayesian models.

Penalized Likelihood. The most natural sparsity-inducing penalty, the ℓ_0 norm, unfortunately leads to currently intractable problems as soon as the number of predictors exceeds about 30. To overcome this restriction, convex relaxation of the ℓ_0 norm – that is, ℓ_1 regularization, has become a basic tool in modern statistics. The most spread formulation of the ℓ_1-penalized linear regression is known as the lasso in the statistic community [13] or basis pursuit in the signal processing community [4]. However, the crude lasso is known not to be consistent in variable selection unless some cumbersome conditions on the design matrix [14]; moreover, it can be sensitive to highly correlated predictors [16]. There exists a large number of proposals to enhance the lasso as a selection operator: among these, the adaptive-lasso [15] is a weighted version enjoying nice oracle properties that works extremely well in practice and can be considered as state of the art. Another popular answer that builds on the lasso to achieve variable selection consistency in presence of correlated features is the stability selection approach [10], which applies many lasso procedures with randomized weights on subsamples of the original data.

Bayesian modelling. Bayesian models have also widely been studied in a variable selection context. Spike-and-slab models, first introduced by [11], use as priors for the regression coefficients a

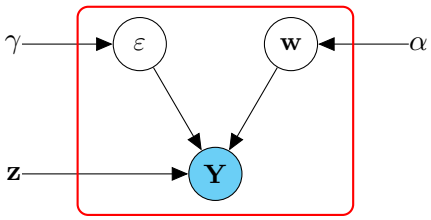


Figure 1: Graphical representation of the sparse generative model.

mixture of two distributions : a thin one, corresponding to irrelevant predictors (the *spike*, typically a Dirac law or a Gaussian distribution of small variance) and a thick one, corresponding to the relevant ones (the *slab*, typically a uniform or Gaussian distribution of large variance). Even though fast deterministic approaches have also recently been considered [12], MCMC methods have been usually chosen to select models with the highest posterior distributions. Some refined spike-and-slab models have also been very efficient even in very high-dimensional settings [8]. In practice MCMC methods for spike-and-slab may suffer from poor mixing properties.

As an alternative, our approach uses spike-and-slab priors induced by a binary vector which segregates the relevant from the irrelevant predictors. Such vectors, introduced by [6] have been widely used in the Bayesian literature, but have always been considered as random parameters (typically endowed with a Bernoulli prior). In this work, the originality is to induce the sparsity through a binary deterministic vector and to use an EM algorithm for type-II log-likelihood maximization. In order to avoid the NP-hard problem of maximizing over the binary vector in the E step, a combination of a relaxed quadratic optimization problem along with a model selection step relying on Occam's razor is proposed.

2 A Sparse Generative Model

Let us consider the following regression model

$$\begin{cases} \mathbf{Y} &= X\boldsymbol{\beta} + \boldsymbol{\varepsilon} \\ \boldsymbol{\beta} &= \mathbf{z} \odot \mathbf{w}, \end{cases} \tag{1}$$

where $\mathbf{Y} \in \mathbb{R}^n$ is the set of n observed responses, $X \in \mathcal{M}_{n,p}(\mathbb{R})$ is the design matrix with p input variables and $\odot$ denotes the Hadamard product, such that $\beta_j = z_j \times w_j$, for $j = 1, \ldots, p$. The vector $\boldsymbol{\varepsilon}$ is a noise term with $p(\boldsymbol{\varepsilon}|\gamma) = \mathcal{N}(\boldsymbol{\varepsilon}; 0, I_n/\gamma)$ where I_n denotes the $n \times n$ identity matrix. A prior distribution $p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}; 0, I_p/\alpha)$ with an isotropic covariance matrix is further assumed as in [9]. Moreover, we denote by $\mathbf{z} \in \{0,1\}^p$ a binary deterministic parameter vector, whose nonzero entries correspond to the active variables of the regression model. Such modeling induces a spike-and-slab like prior distribution for $\boldsymbol{\beta}$:

$$p(\boldsymbol{\beta}|\mathbf{z}, \alpha) = \prod_{j=1}^{p} p(\beta_j|z_j, \alpha) = \prod_{j=1}^{p} \delta_0(\beta_j)^{1-z_j} \mathcal{N}(\beta_j; 0, 1/\alpha)^{z_j},$$

where $\delta_0(\cdot)$ is the Dirac function at zero. However, we emphasize that, contrary to standard spike-and-slab models [11] which assume a prior Bernoulli distribution over $\mathbf{z}$, we see $\mathbf{z}$ here as a deterministic model parameter to be inferred from the data. As we shall see in Section 3, this allows us to work with a marginal log-likelihood which involves an Occam's razor term, allowing in turn model selection. In the same spirit, we do not put any prior distribution on γ nor α. Finally, the graphical model is presented in Figure 1 and we denote $q = \sum_{j=1}^{p} z_j$ the number of relevant variables.

3 Inference

In the following and in order to perform inference, $\mathbf{w}$ is seen as a latent variable while $Z = \mathrm{diag}(\mathbf{z})$, α as well as γ are parameters to be estimated from the data $(X, \mathbf{Y})$. To this end, we propose to use

an expectation-maximization (EM) approach [5] allowing us to iteratively find a local maximum of the type-II likelihood or *evidence* of the data:

$$p(\mathbf{Y}|X, Z, \alpha, \gamma) = \int_{\mathbb{R}^p} p(\mathbf{Y}|X, \mathbf{w}, Z, \alpha, \gamma)p(\mathbf{w}|\alpha)d\mathbf{w}. \tag{2}$$

To simplify the notations, the dependency on X will be omitted in the rest of the paper. All proofs are provided as supplementary materials.

3.1 E-step

Proposition 1. *The posterior distribution of* $\mathbf{w}$ *given the data is given by*

$$p(\mathbf{w}|\mathbf{Y}, Z, \alpha, \gamma) = \mathcal{N}(\mathbf{w}, \mathbf{m}, S), \tag{3}$$

where $S = (\gamma ZX^TXZ + \alpha I_p)^{-1}$ *and* $\mathbf{m} = \gamma SZX^T\mathbf{Y}$.

Notice that $\mathbf{m}$ and S also allow to compute a convenient expression of the evidence.

Proposition 2. *The type-II log-likelihood is given by*

$$\log p(\mathbf{Y}|Z, \alpha, \gamma) = -\frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\gamma) + \frac{p}{2}\log(\alpha) - \frac{\gamma}{2}\|\mathbf{Y}\|_2^2 + \frac{1}{2}\log\det S + \frac{1}{2}\mathbf{m}^T S^{-1}\mathbf{m}. \tag{4}$$

The vector $\mathbf{m}$ is the maximum-a-posteriori (MAP) estimator of $\mathbf{w}$. It can easily be shown (see [1] for instance) that computing the MAP estimator of the standard Bayesian linear regression model $\mathbf{Y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\beta}$ follows an isotropic Gaussian prior distribution, is equivalent to estimating the parameter vector of the frequentist ridge regression model. In our case, this implies that the nonzero coefficients of $\mathbf{z} \odot \mathbf{m}$ correspond to ridge estimates of the model

$$\mathbf{Y} = \tilde{X}\tilde{\boldsymbol{\beta}} + \boldsymbol{\varepsilon} \tag{5}$$

where $\tilde{X}$ is the submatrix of X with columns, corresponding to irrelevant variables, being deleted and $\tilde{\boldsymbol{\beta}} \in \mathbb{R}^q$.

3.2 M-step

At the M-step, the expectation of the complete data log-likelihood $\mathbb{E}_{\mathbf{w}}(\log p(\mathbf{Y}, \mathbf{w}, |Z, \alpha, \gamma))$, with respect to $p(\mathbf{w}|\mathbf{Y}, Z, \alpha, \gamma)$, is maximized over Z, α, γ.

Proposition 3. *Denoting* $\Sigma = S + \mathbf{m}\mathbf{m}^T$*, the expected complete data log-likelihood is given by*

$$\begin{aligned}\mathbb{E}_{\mathbf{w}}(\log p(\mathbf{Y}, \mathbf{w}|Z, \alpha, \gamma)) = \frac{n}{2}\log(\gamma) - \frac{\gamma}{2}\mathbf{Y}^T\mathbf{Y} - \frac{\gamma}{2}\mathbf{z}^T(X^TX \odot \Sigma)\mathbf{z} + \gamma\mathbf{z}^T(\mathbf{m} \odot (X^T\mathbf{Y})) \\ - \frac{\gamma}{2}\operatorname{Tr}(\Sigma) + \frac{p}{2}\log(\alpha) - \frac{p+n}{2}\log(2\pi).\end{aligned} \tag{6}$$

Maximizing the expectation of the complete data log-likelihood with respect to γ and α leads to the following estimates:

$$\hat{\gamma}^{-1} = \frac{1}{n}\left\{\mathbf{Y}^T\mathbf{Y} + \mathbf{z}^T(X^TX \odot \Sigma)\mathbf{z} - 2\mathbf{z}^T(\mathbf{m} \odot (X^T\mathbf{Y}))\right\} \quad \text{and} \quad \hat{\alpha} = \frac{p}{\operatorname{Tr}(\Sigma)}. \tag{7}$$

However and as expected, maximizing (6) with respect to the vector $\mathbf{z}$ is a binary quadratic problem and is therefore NP-hard. To tackle this issue, we use a simple relaxation of the problem by replacing the vector by a relaxed vector $\mathbf{z}^{\text{relaxed}}$ in $[0, 1]^p$. The relaxed optimization problem can be efficiently solved with the box constraint BFGS quasi-Newton method of [3]. The M-step update of $\hat{\mathbf{z}}^{\text{relaxed}}$ will consequently be

$$\hat{\mathbf{z}}^{\text{relaxed}} = \underset{\mathbf{u}\in[0,1]^p}{\operatorname{argmax}} \left\{-\frac{1}{2}\mathbf{u}^T(X^TX \odot \Sigma)\mathbf{u} + \mathbf{u}^T(\mathbf{m} \odot (X^T\mathbf{Y}))\right\}. \tag{8}$$

Note that we also relied on a branch-and-bound algorithm [2] for quadratic binary maximization. However, it can only be used with few input variables. Moreover, we emphasize that we obtained very similar results with both approaches in all the experiments we carried out.

4 Model Selection

In practice, the vector $\mathbf{z}^{\text{relaxed}}$ has to binarized in order to select the relevant input variables. A common choice would consist in relying on a threshold τ such that z_j is set to 1 if $z_j \geq \tau$, 0 otherwise. However, numerical experiments showed that such a procedure would lead to poor estimates of $\mathbf{z}$. In order to perform an efficient variable selection, we will use the outputs of the EM algorithm to create a path of models and, relying on Occam's razor, we will afterward maximize the type-II likelihood over this path to finally select the relevant variable.

4.1 Occam's Razor

One of the key advantages of the model we consider along with the EM algorithm is the fact that we maximize a marginal log-likelihood, which automatically penalizes the model complexity.

Proposition 4. *The type-II log-likelihood can be written as*

$$\begin{aligned} \log p(\mathbf{Y}|Z,\alpha,\gamma) &= \log p(\mathbf{Y}|\mathbf{m},Z,\alpha,\gamma) + \text{pen}(\mathbf{z},\alpha,\gamma) \\ &= -\frac{\gamma}{2}||\mathbf{Y}-\mathbf{X}Z\mathbf{m}||^2 - \frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\gamma) + \text{pen}(\mathbf{z},\alpha,\gamma) \end{aligned} \tag{9}$$

where

$$\begin{aligned} \text{pen}(\mathbf{z},\alpha,\gamma) &= \log p(\mathbf{m}|\alpha) + \frac{1}{2}\log\det S + \frac{p}{2}\log(2\pi) \\ &= -\frac{\alpha}{2}\left\|\mathbf{m}\right\|_2^2 + \frac{p}{2}\log\alpha + \frac{1}{2}\log\det S \end{aligned}$$

is the Occam factor.

Interestingly, the term $\text{pen}(\mathbf{z},\alpha,\gamma)$ corresponds exactly to Occam's razor as described in [9] and detailed in [1]. Such a term is known to penalize the model complexity and has been widely used for model selection purposes (see for instance [9]). Let us emphasize that $\text{pen}(\mathbf{z},\alpha,\gamma)$ is related to the penalization term of the Bayesian information criterion (BIC). Indeed, if a broad Gaussian prior distribution for the vector $\mathbf{w}$ is considered and if the corresponding matrix S is assumed to have full rank, then Occam's razor is approximately $(-1/2)p\log n$. Contrary to BIC which relies on an asymptotic Laplace approximation, we obtained here an analytical expression of the evidence.

In regression, Eq. (9) can be computed for models with various input variables and the model which realizes a trade off between the log-likelihood and the penalty term is then selected. In our case, the dependency on $\mathbf{z}$ is explicit and therefore relying on the EM algorithm to optimize the marginal log-likelihood over $\mathbf{z}$ induces a search over linear models with various input variables and various complexities.

4.2 Path of Models

As mentioned previously, we rely on $\hat{\mathbf{z}}^{\text{relaxed}}$ to find a path of models which are likely to have a high evidence. More precisely, we build a path by assuming that the larger the coefficients of $\hat{\mathbf{z}}^{\text{relaxed}}$ are, the more likely they are to correspond to relevant variables.

We define the set of vectors $(\hat{\mathbf{z}}^{(k)})_k$ as the binary vectors such that, for each k, the k top coefficients of $\hat{\mathbf{z}}^{\text{relaxed}}$ are set to 1 and the others to 0. For example, $\hat{\mathbf{z}}^{(1)}$ contains only zeros and a single 1 at the position of the highest coefficient of $\hat{\mathbf{z}}^{\text{relaxed}}$. The set of vectors $(\hat{\mathbf{z}}^{(k)})_k$ defines a path of models to look at for model selection. Note that this path allows us to deal with a family of p models (ordered by sparsity) instead of 2^p, allowing our approach to deal with a large number of input variables. Thus, the evidence is evaluated for all $\hat{\mathbf{z}}^{(k)}$ and the number $\hat{q}$ of relevant variables is chosen such that the evidence is maximized:

$$\hat{q} = \text{argmax}_{1\leq k\leq p}\, p(\mathbf{Y}|\hat{\mathbf{z}}^{(k)},\hat{\alpha},\hat{\gamma}) \quad \text{and} \quad \hat{\mathbf{z}} = \hat{\mathbf{z}}^{(\hat{q})}. \tag{10}$$

As shown in the experiment section, such a heuristic leads to a particularly accurate estimate of $\mathbf{z}$ and therefore of the model complexity.

5 SpinyReg: an Algorithm for Sparse Regression

5.1 Pseudo-code

Algorithm 1 presents a pseudo-code for the spinyReg algorithm.

Algorithm 1: The spinyReg algorithm

Input: X,$\mathbf{Y}$
Output: $\mathbf{z}$

Initialize $\gamma = 1$, $\alpha = 1$, $\mathbf{z}^{\text{relaxed}} = (1, ..., 1)$;

// EM algorithm to infer the path of models
repeat
  // E-step
  $S = (\gamma \operatorname{diag}(\mathbf{z}^{\text{relaxed}}) X^T X \operatorname{diag}(\mathbf{z}^{\text{relaxed}}) + \alpha I_p)^{-1}$;
  $\mathbf{m} = \gamma S \operatorname{diag}(\mathbf{z}^{\text{relaxed}}) X^T \mathbf{Y}$; $\Sigma = S + \mathbf{m}\mathbf{m}^T$;
  // M-step
  Compute $\hat{\alpha}$ and $\hat{\gamma}$ using Eq. (7);
  Compute $\hat{\mathbf{z}}^{\text{relaxed}}$ using Eq. (8) and the L-BFGS-B method;
until *convergence of the evidence*;

// Model Selection
for *k = 1..p* **do**
  Compute $\hat{\mathbf{z}}^{(k)}$;

$\hat{q} = \operatorname{argmax}_{1 \leq k \leq p} p(\mathbf{Y} | \hat{\mathbf{z}}^{(k)}, \hat{\alpha}, \hat{\gamma})$;
$\hat{\mathbf{z}} = \hat{\mathbf{z}}^{(\hat{q})}$;

5.2 Some Algorithmic Considerations

The spinyReg algorithm is essentially a model selection algorithm. In order to perform prediction, the natural estimator of the model is $\hat{\mathbf{z}} \odot \hat{\mathbf{m}}$ where

$$\hat{\mathbf{m}} = \gamma(\gamma \operatorname{diag}(\hat{\mathbf{z}}) X^T X \operatorname{diag}(\hat{\mathbf{z}}) + \alpha I_p)^{-1} \operatorname{diag}(\hat{\mathbf{z}}) X^T \mathbf{Y}.$$

However, as it was stated at the end of subsection 3.1, this estimator is exactly the ridge estimator performed on a small model where only the predictors corresponding to nonzero coefficients of $\hat{\mathbf{z}}$ are kept. Since this would imply an unnecessary shrinkage of the nonzero coefficients of β, we would rather recommend to perform an ordinary least squares (OLS) estimation on the same small model. This is the choice we made on the numerical simulations of section 6.

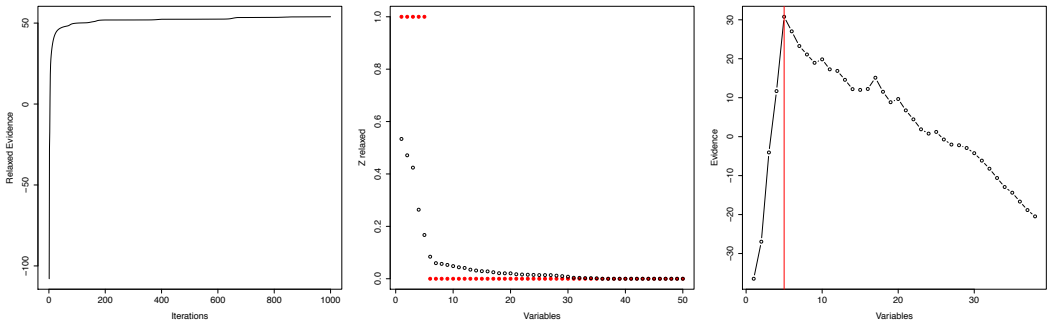
Moreover, after convergence of the EM algorithm, the indexes of the coefficients of $\hat{\mathbf{z}}$ that were exactly equal to 1 were automatically considered as relevant variables. This allows to avoid computing the first evidences of the path of models.

6 Numerical Experiments

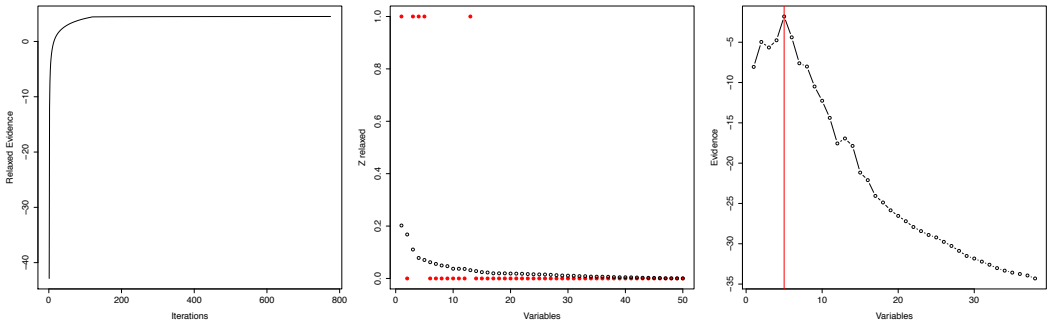
6.1 Simulation Setup

In this section, we illustrate and compare the behavior of the proposed method on simulated data sets. In order to consider a wide range of scenarios, we use three different simulation schemes: uniform, Toeplitz and blockwise. The simulation of the parameter $\mathbf{w}$ and of the noise $\boldsymbol{\varepsilon}$ is common for the three schemes: $\mathbf{w} \sim \mathcal{N}(0, I_p/\alpha)$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, I_n/\gamma)$. The design matrix X is simulated according to a Gaussian distribution with zero mean and a covariance matrix R depending on the chosen scheme. The correlation structure of $R = (r_{ij})_{i,j=1,...,p}$ is as follows:

- uniform: $r_{ii} = 1$ for all $i = 1, ...p$ and $r_{ij} = \rho$ for $i, j = 1, \ldots, p$ and $i \neq j$,
- Toeplitz: $r_{ii} = 1$ for all $i = 1, ...p$ and $r_{ij} = \rho^{|i-j|}$ for $i, j = 1, \ldots, p$ and $i \neq j$,



$p = 50$ and $n = 150$



$p = 50$ and $n = 30$

Figure 2: Variable selection with spinyReg on the two introductory examples ($p = 50$ and $n = 150$ or $n = 30$): evidence according to the iterations of the EM algorithm (left), values of $\hat{\mathbf{z}}^{\text{relaxed}}$ and actual binary values for z (center) and evidence computed on the path of models (right).

- blockwise: $R = \text{diag}(R_1, ..., R_4)$ is a 4-blocks diagonal matrix where R_ℓ is such that $r_{\ell ii} = 1$ and $r_{\ell ij} = \rho$ for $i, j = 1, \ldots, p/4$ and $i \neq j$.

Then, Z is simulated by randomly picking q active variables among p. Y is finally computed according to Equation (1).

6.2 An Introductory Example

This section first consider an introductory example which aims at highlighting the main features of the proposed approach. For this experiment, the uniform simulation setup is used with $p = 50$, $q = 5$, $\alpha = 1$ and $\gamma = 10$. From this setup, two data sets were simulated with respectively $n = 150$ and $n = 30$ observations. The second setting corresponds therefore to a sparse scenario with $n < p$ whereas the first one should be easier to fit. Notice that the dimensionality is kept relatively low mainly for visualization purpose.

Figure 2 present the results of the application of spinyReg on those two data sets. The left panel of each row shows the behavior of the evidence according to the iterations of the (relaxed) EM algorithm. The second panel presents the values of $\hat{\mathbf{z}}^{\text{relaxed}}$ (sorted in decreasing order) and the corresponding true values for $\mathbf{z}$ (red filled points) used in the simulations. Finally, the right panel of both rows shows the evidence on the path of models.

First, on the first row of Figure 2, the left panel allows to verify that the EM algorithm succeeds in maximizing the evidence, even in its relaxed version. On the second panel, one can see that the five largest values of $\hat{\mathbf{z}}^{\text{relaxed}}$ actually correspond to the five active variables. This confirms that spinyReg succeeds here in finding the relevant variables in the regression model. The third panel confirms that spinyReg would select five variables among the 50 original ones. On this quite simple example, spinyReg yields a true positive rate (TPR) equals to 1 and a false positive rate (FPR) equals to 0.

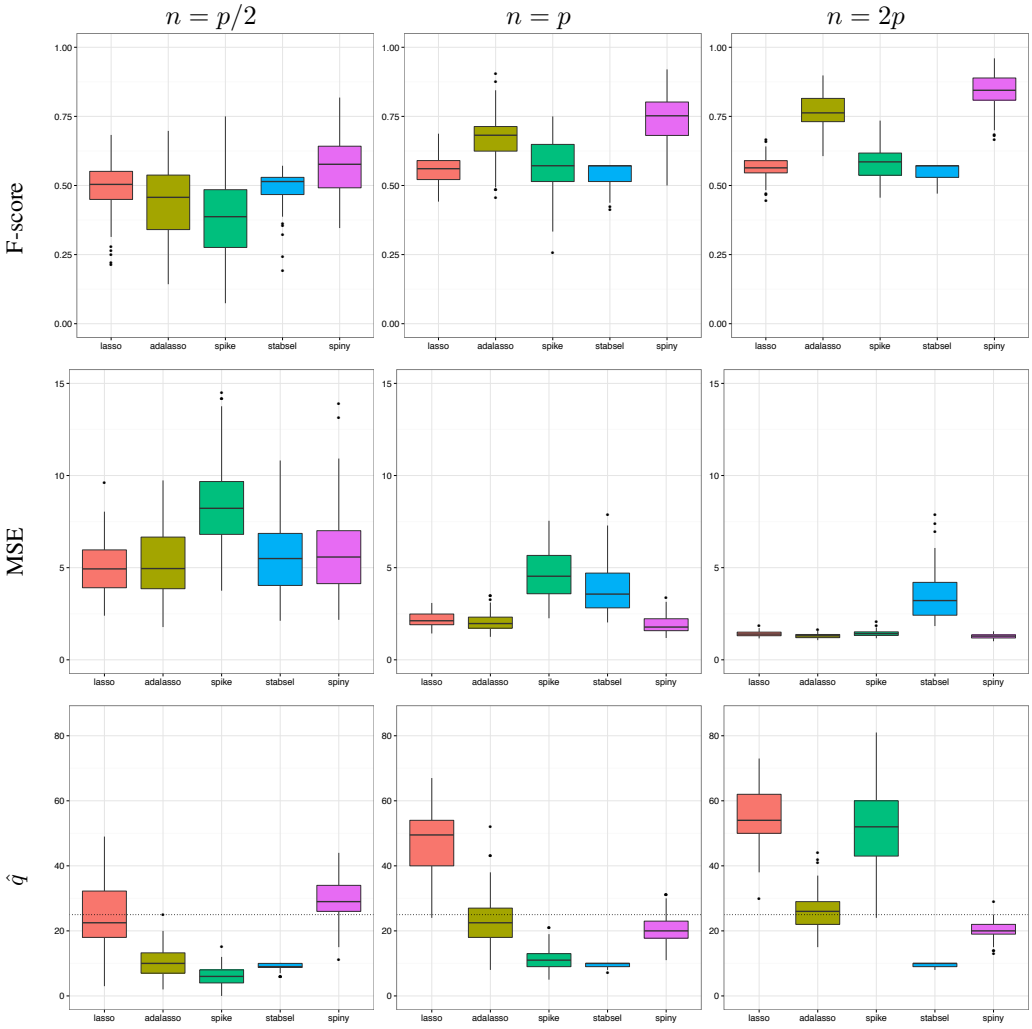


Figure 3: Performances of the 5 studied methods over 100 replications on the blockwise simulated data with $\rho = 0.75$, $p = 100$, $q = 25$ and for three data set sizes. Results for the uniform and Toeplitz schemes are available as supplementary material.

For the second and much difficult situation (bottom row of Figure 2), the behavior of the EM algorithm is still satisfying. One may however notice on the second panel that the estimated values for $\mathbf{z}^{\text{relaxed}}$ are less discriminative here. Indeed, the values for $\hat{\mathbf{z}}^{\text{relaxed}}$ are smaller and, among the five largest ones, the second variables does not correspond to an active one. Nevertheless, the evidence (right panel) actually allows to pick up the right number of active variables. On this more difficult data set, spinyReg yields a true positive rate (TPR) equals to 0.8 and a false positive rate (FPR) equals to 0.022 (1 false positive among 45 irrelevant variables).

6.3 Benchmark Study

We now compare the performance of spinyReg with the most recent variable selection methods: lasso, adaptive lasso, stability selection and spike-and-slab approach of [7]. To this end, we simulated 100 data sets for each of the three simulations schemes (uniform, Toeplitz and blockwise), for three data set sizes ($n = p/2$, $n = p$, $n = 2p$) and two values for the correlation parameter ($\rho = 0.25$ and $\rho = 0.75$). The other simulation parameters were $p = 100$, $q = 25$, $\alpha = 0.01$ and $\gamma = 1$. The measures used to evaluate the method performances are the prediction mean square error on test data (MSE, hereafter), the true positive rate (TPR), the false positive rate (FPR), the F

score, the Hamming distance between the predicted and actual vector z and the estimated value of q.

By lack of space, we present here only the results for the blockwise simulation setup. All the other results are available as supplementary material. Please note that very similar conclusions can be drawn on these other scenarios. Figure 3 presents the F score, MSE and $\hat{q}$ of the 5 studied methods for the blockwise simulation setup with $\rho = 0.75$ and for the three data set sizes.

The first row of Figure 3 gives the F score which is the harmonic mean of precision and recall. This measure allows us to figure out how the methods behave in term of detection of the relevant variables. Over the different data sizes, lasso, spike-and-slab and stability selection turn out to perform less than adaptive lasso and spinyReg. This is mainly due to a high number of false detections for these methods, especially when n is large. Adaptive lasso does not have this drawback and perform well in the different situations. Finally, SpinyReg has here a satisfying behavior and outperforms all other methods on the three data sizes. When looking at the detailed results (supplementary material), one can see that this good behavior is explained by a good precision and a very low number of false detections.

The second row of Figure 3 provides the MSE values for the studied methods. Most of the methods perform well except stability selection and spike-and-slab when $n \leq p$. In particular, spinyReg has the best prediction performance for $n \geq p$.

The last row of Figure 3 gives the number q of active variables estimated by the 5 methods. We remind that the actual number of active variables is $q = 25$ for these simulations (represented by the dashed lines on Figure 3). It is worth noticing that lasso has a clear tendency to overestimate the number of active variables, particularly when n becomes large. Conversely, stability selection has the opposite behavior and underestimate q. It turns out that spinyReg provides consistently a good estimate of the actual value for q. Adaptive lasso has a behavior close to the one of spinyReg but the latter provides a good estimate of q even for small data set sizes.

7 Conclusion

As a summary, we considered the problem of Bayesian variable selection for high-dimensional linear regression through a sparse generative model. The sparsity is induced by a deterministic binary vector which multiplies with the Gaussian regressor vector. The originality of the work was to consider its inference through a type-II log-likelihood maximization using an EM algorithm. The NP-hard problem of maximizing over the binary vector in the E step was recasted as the combination of a relaxed quadratic optimization problem and a model selection step relying on Occam's razor. Numerical experiments on high-dimensional simulated data have shown that spinyReg performs well compared to its competitors. SpinyReg has globally a conservative behavior (high precision and very low false detection rate). SpinyReg positions itself has a serious alternative to ℓ_1-penalized methods for variable selection, especially in contexts where false detections are particularly unwanted.

References

[1] C.M. Bishop. *Pattern recognition and machine learning*. Springer-Verlag, 2006.

[2] C. Buchheim, A. Caprara, and A. Lodi. An effective branch and bound algorithm for convex quadratic programming, integer programming and combinatorial optimization. *Lecture Notes in Computer Science*, 6080:285–298, 2010.

[3] R.H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *Journal on Scientific and Statistical Computing*, 16:1190–1208, 1995.

[4] Scott Shaobing Chen, David L. Donoho, Michael, and A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.

[5] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood for incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, B39:1–38, 1977.

[6] Edward I George and Robert E McCulloch. Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

[7] Hemant Ishwaran, Udaya B Kogalur, and J Sunil Rao. spikeslab: Prediction and variable selection using spike and slab regression. *R Journal*, 2(2), 2010.

[8] Hemant Ishwaran and J Sunil Rao. Spike and slab gene selection for multigroup microarray data. *Journal of the American Statistical Association*, 100(471):764–780, 2005.

[9] D.J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1998.

[10] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B*, 27, 2010.

[11] T.J. Mitchell and J.J. Beauchamp. Bayesian variable selection in linear regression (with discussion). *Journal of the American Statistical Association*, 83:1023–1036, 1988.

[12] Veronika Ročková and Edward I George. Emvs: The em approach to bayesian variable selection. *Journal of the American Statistical Association*, just-accepted, 2013.

[13] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 58(1):267–288, 1996.

[14] Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Mahcine Learning Research*, 7:2541–2563, 2006.

[15] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

[16] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 67(2):301–320, 2005.

A Sparse Generative Model and its EM Algorithm for Variable Selection in High-Dimensional Regression - Supplementary Material -

Anonymous Author(s)
Affiliation
Address
email

1 Proofs

1.1 Proof of Proposition 1

Proof. Using Bayes' rule, we have

$$\begin{aligned}\log p(\mathbf{w} \mid \mathbf{Y}, Z, \alpha, \gamma) &= \log p(\mathbf{Y} \mid \mathbf{w}, Z, \alpha, \gamma) + \log p(\mathbf{w} \mid \alpha) + K_1 \\ &= -\frac{\gamma}{2} \|\mathbf{Y} - XZ\mathbf{w}\|_2^2 - \frac{\alpha}{2} \|\mathbf{w}\|_2^2 + K_2 \\ &= -\frac{\gamma}{2}\mathbf{w}^T Z X^T X Z \mathbf{w} + \gamma \mathbf{w}^T Z X^T \mathbf{Y} - \frac{\alpha}{2} \|\mathbf{w}\|_2^2 + K_3 \\ &= -\frac{1}{2}\mathbf{w}^T S^{-1} \mathbf{w} + \mathbf{w}^T S^{-1} \mathbf{m} + K_3.\end{aligned}$$

where K_1, K_2 and K_3 are quantities that do not depend on $\mathbf{w}$. Therefore $p(\mathbf{w} \mid \mathbf{Y}, Z, \alpha, \gamma) = \mathcal{N}(\mathbf{w}, \mathbf{m}, S)$. □

1.2 Proof of Proposition 2

Proof. By directly computing the integrand of (2), we find

$$\begin{aligned}\log p(\mathbf{Y} \mid Z, \alpha, \gamma) &= -\frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\gamma) + \frac{p}{2}\log(\alpha) \\ &+ \log \int_{\mathbf{w} \in \mathbb{R}^p} \frac{1}{\sqrt{(2\pi)^p}} \exp\left(-\frac{\gamma}{2}\mathbf{Y}^T\mathbf{Y} + \gamma \mathbf{Y}^T X Z \mathbf{w} - \frac{\gamma}{2}\mathbf{w}^T Z X^T X Z \mathbf{w} - \frac{\alpha}{2}\mathbf{w}^T\mathbf{w}\right) d\mathbf{w}\end{aligned}$$

which leads to

$$\begin{aligned}\log p(\mathbf{Y} \mid Z, \alpha, \gamma) = -\frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\gamma) + \frac{p}{2}\log(\alpha) - \frac{\gamma}{2}\|\mathbf{Y}\|_2^2 \\ + \log \int_{\mathbf{w} \in \mathbb{R}^p} \frac{1}{\sqrt{(2\pi)^p}} \exp\left(-\frac{1}{2}\mathbf{w}^T S^{-1}\mathbf{w} + \mathbf{w}^T S^{-1}\mathbf{m}\right) d\mathbf{w}\end{aligned}$$

therefore

$$\log p(\mathbf{Y} \mid Z, \alpha, \gamma) = -\frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\gamma) + \frac{p}{2}\log(\alpha) - \frac{\gamma}{2}\|\mathbf{Y}\|_2^2 + \frac{1}{2}\log\det S + \frac{1}{2}\mathbf{m}^T S^{-1}\mathbf{m}.$$

□

1.3 Proof of Proposition 3

Proof. We have $\log p(\mathbf{Y}, \mathbf{w} \mid Z, \alpha, \gamma) = \log p(\mathbf{Y} \mid \mathbf{w}, Z, \alpha, \gamma) + \log p(\mathbf{w} \mid \alpha)$. Thus, since both the prior on $\mathbf{w}$ and the noise are Gaussian, we can write

$$\log p(\mathbf{Y}, \mathbf{w} \mid Z, \alpha, \gamma) = \frac{n}{2}\log\gamma + \frac{p}{2}\log(\alpha) - \frac{p+n}{2}\log(2\pi) - \frac{\gamma}{2}(\mathbf{Y} - XZ\mathbf{w})^T(\mathbf{Y} - XZ\mathbf{w}) - \frac{\alpha}{2}\mathbf{w}^T\mathbf{w}.$$

Therefore, by expanding and computing the expectation of the expression, we find :

$$\begin{aligned}\mathbb{E}_{\mathbf{w}}(\log p(\mathbf{Y}, \mathbf{w} \mid Z, \alpha, \gamma)) = \frac{n}{2}\log(\gamma) + \frac{p}{2}\log(\alpha) - \frac{p+n}{2}\log(2\pi) - \frac{\gamma}{2}\mathbf{Y}^T\mathbf{Y} \\ - \frac{\gamma}{2}\mathbb{E}_{\mathbf{w}}(\mathbf{w}^T Z X^T X Z \mathbf{w}) + \gamma \mathbf{Y}^T X Z \mathbb{E}_{\mathbf{w}}(\mathbf{w}) - \frac{\alpha}{2}\mathbb{E}_{\mathbf{w}}(\mathbf{w}^T\mathbf{w}).\end{aligned}$$

From (3), we have $\mathbb{E}_{\mathbf{w}}(\mathbf{w}) = \mathbf{m}$ and, by using the properties of the trace operator,

$$\mathbb{E}_{\mathbf{w}}(\mathbf{w}^T\mathbf{w}) = \mathbb{E}_{\mathbf{w}}(\mathrm{Tr}(\mathbf{w}\mathbf{w}^T)) = \mathrm{Tr}(\mathbb{E}_{\mathbf{w}}(\mathbf{w}\mathbf{w}^T)) = \mathrm{Tr}(S + \mathbf{m}\mathbf{m}^T) = \mathrm{Tr}(\Sigma).$$

Thus, we will also have

$$\mathbb{E}_{\mathbf{w}}(\mathbf{w}^T Z X^T X Z \mathbf{w}) = \mathbb{E}_{\mathbf{w}}(\mathrm{Tr}(Z X^T X Z \mathbf{w}\mathbf{w}^T)) = \mathrm{Tr}(Z X^T X Z \Sigma).$$

Moreover, since $Z = \mathrm{diag}(\mathbf{z})$, we can compute

$$\mathbf{Y}^T X Z \mathbf{m} = \mathbf{z}^T(\mathbf{m} \odot (X^T\mathbf{Y})) \quad \text{and} \quad \mathrm{Tr}(Z X^T X Z \Sigma) = \mathbf{z}^T(X^T X \odot \Sigma)\mathbf{z}.$$

By replacing the values of the terms we just computed, we eventually find the appropriate value of the evidence. □

1.4 Proof of Proposition 4

Proof. Let us compute each factor of the right-hand side of (9).

First,

$$\log p(\mathbf{Y} \mid \mathbf{m}, Z, \alpha, \gamma) = -\frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\gamma) - \frac{\gamma}{2}\|\mathbf{Y}\|_2^2 - \frac{\gamma}{2}\mathbf{m}^T Z X^T X Z \mathbf{m} + \gamma \mathbf{Y}^T X Z \mathbf{m}$$

therefore, since $\mathbf{m}^T S^{-1}\mathbf{m} = \gamma \mathbf{m}^T Z X^T \mathbf{Y} = \gamma \mathbf{Y}^T X Z \mathbf{m}$, we have

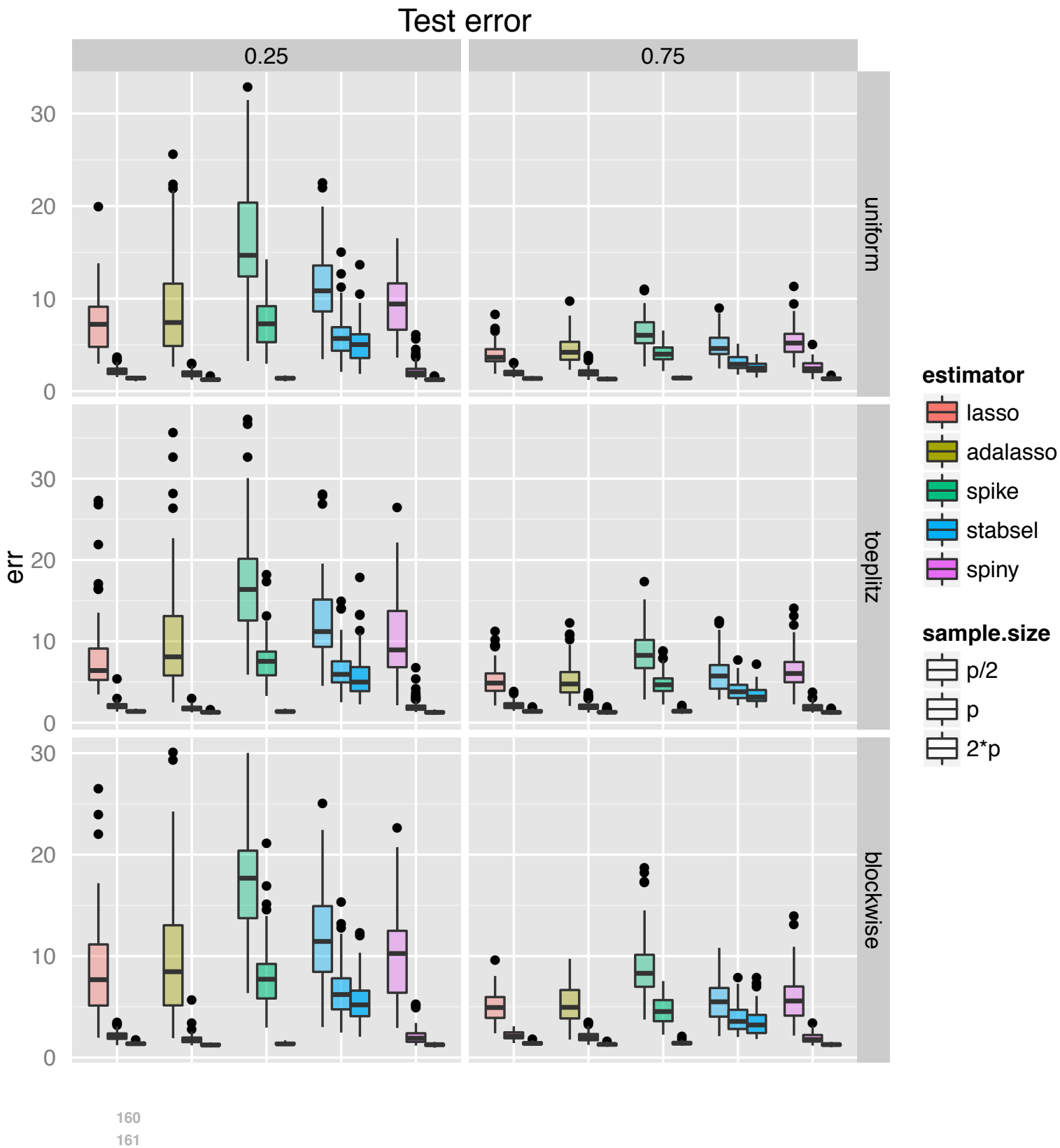
$$\log p(\mathbf{Y} \mid \mathbf{m}, Z, \alpha, \gamma) = -\frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\gamma) - \frac{\gamma}{2}\|\mathbf{Y}\|_2^2 - \frac{\gamma}{2}\mathbf{m}^T Z X^T X Z \mathbf{m} + \mathbf{m}^T S^{-1}\mathbf{m}.$$

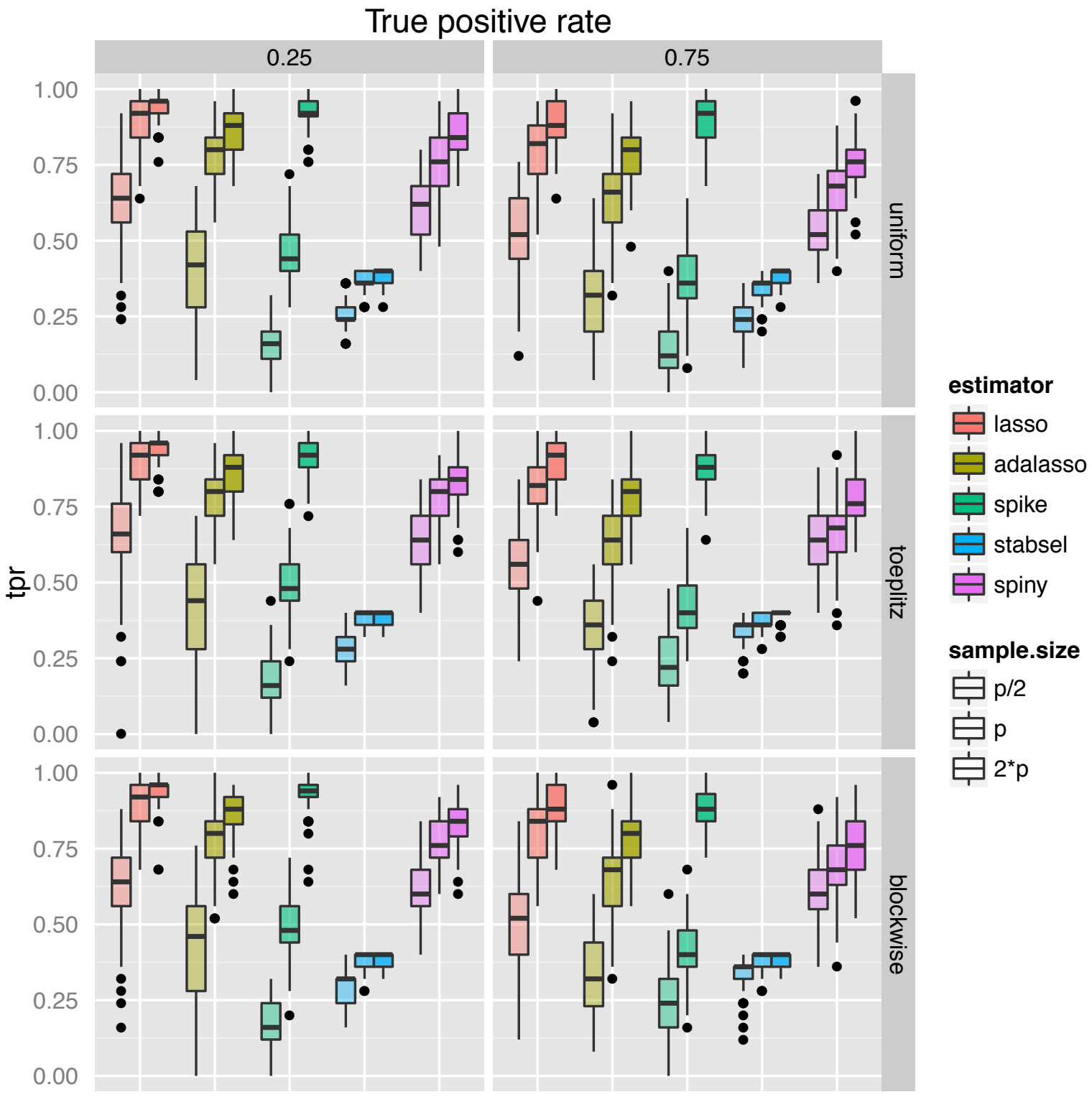
Furthermore, $\log p(\mathbf{m} \mid \alpha) = -\frac{p}{2}\log(2\pi) + \frac{p}{2}\log(\alpha) - \frac{\alpha}{2}\mathbf{m}^T\mathbf{m}$.

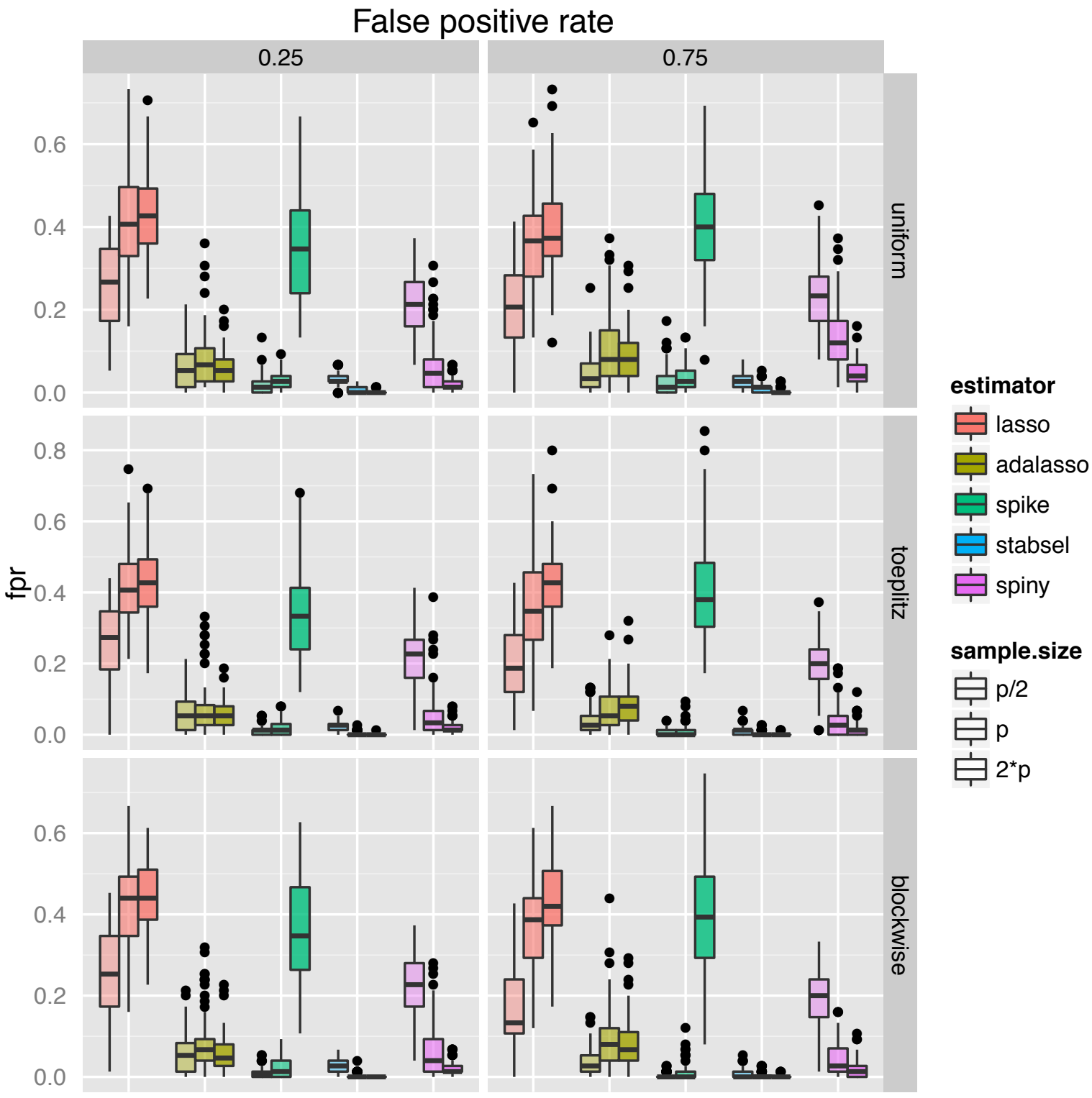
Thus, by summing the terms of the right-hand side of (9), we find the same expression of the type-II log-likelihood as in (4). □

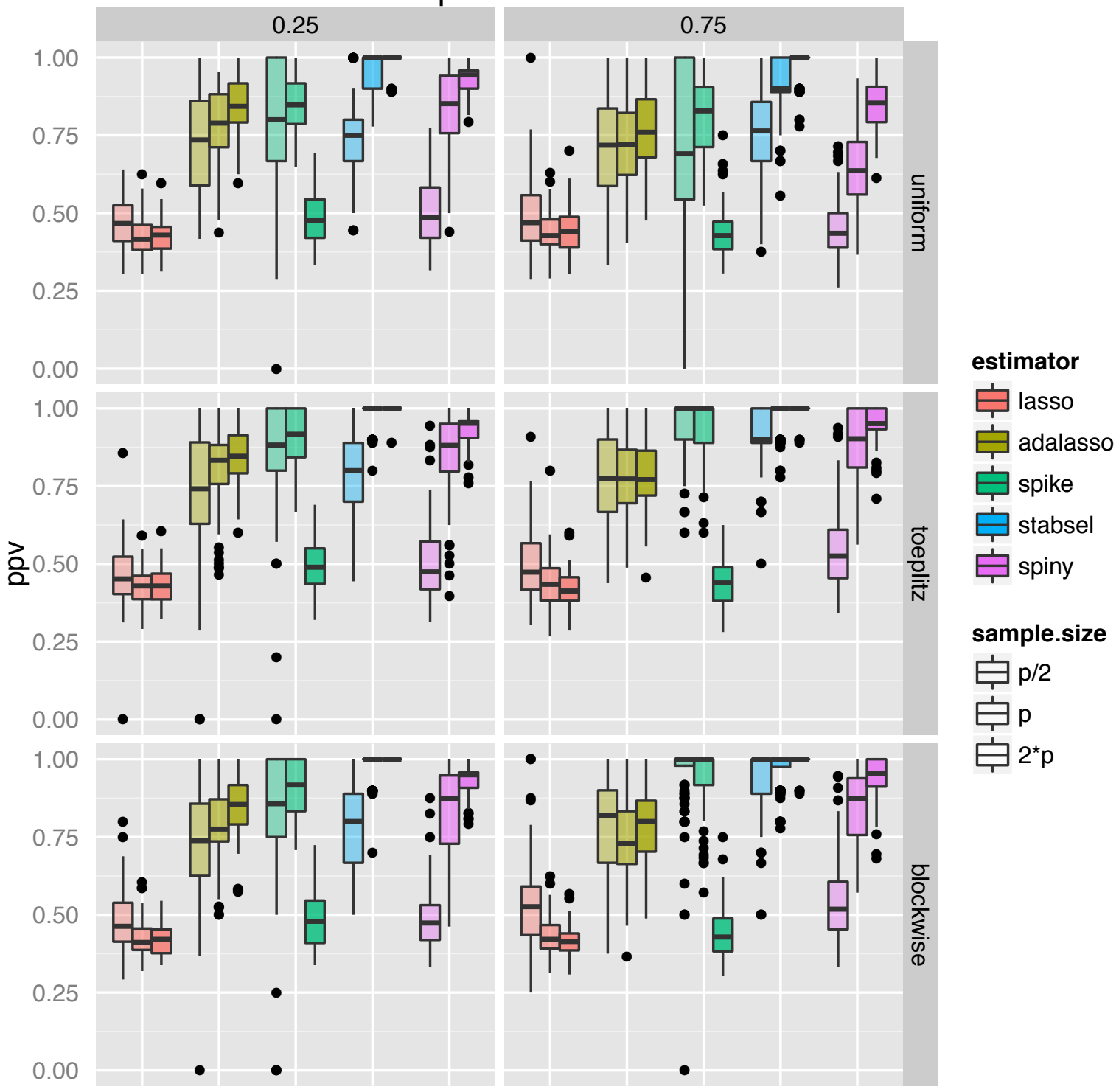
2 Additional Benchmark Study

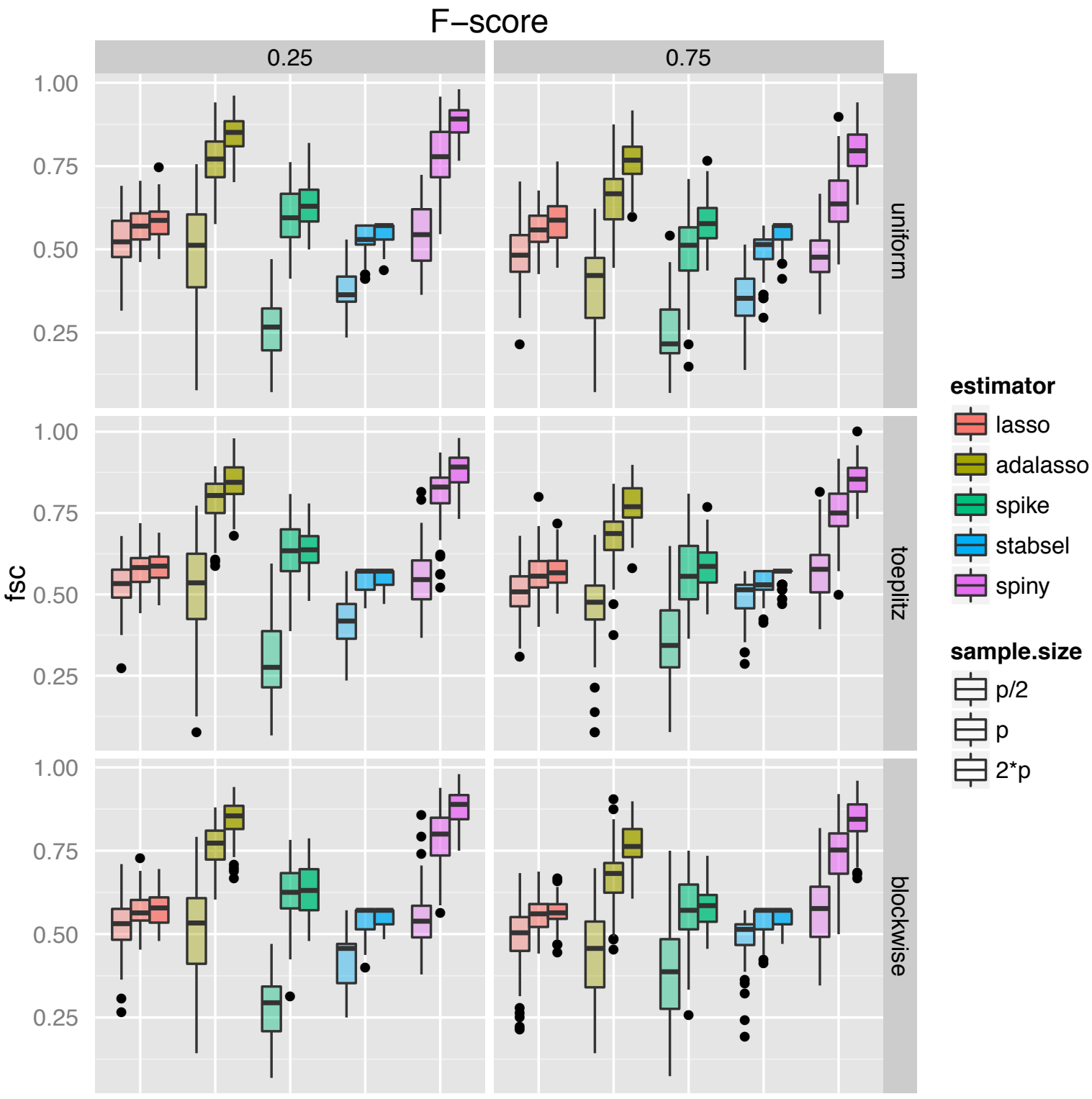
The following results are the ones announced in Section 6 of the main article.

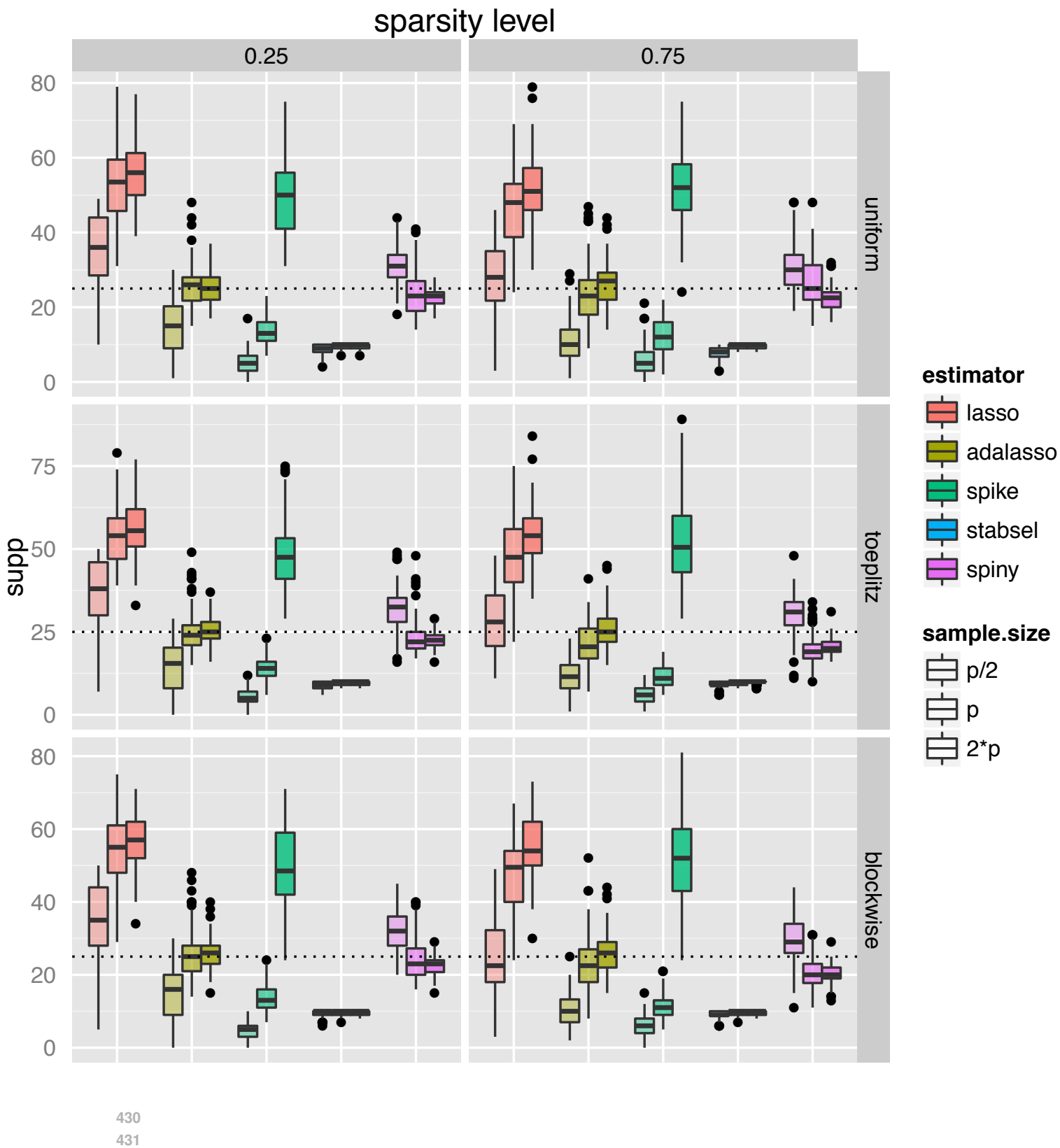

Test error
0.25
0.75
uniform
toeplitz
blockwise
err
0
10
20
30
estimator
lasso
adalasso
spike
stabsel
spiny
sample.size
p/2
p
2*p

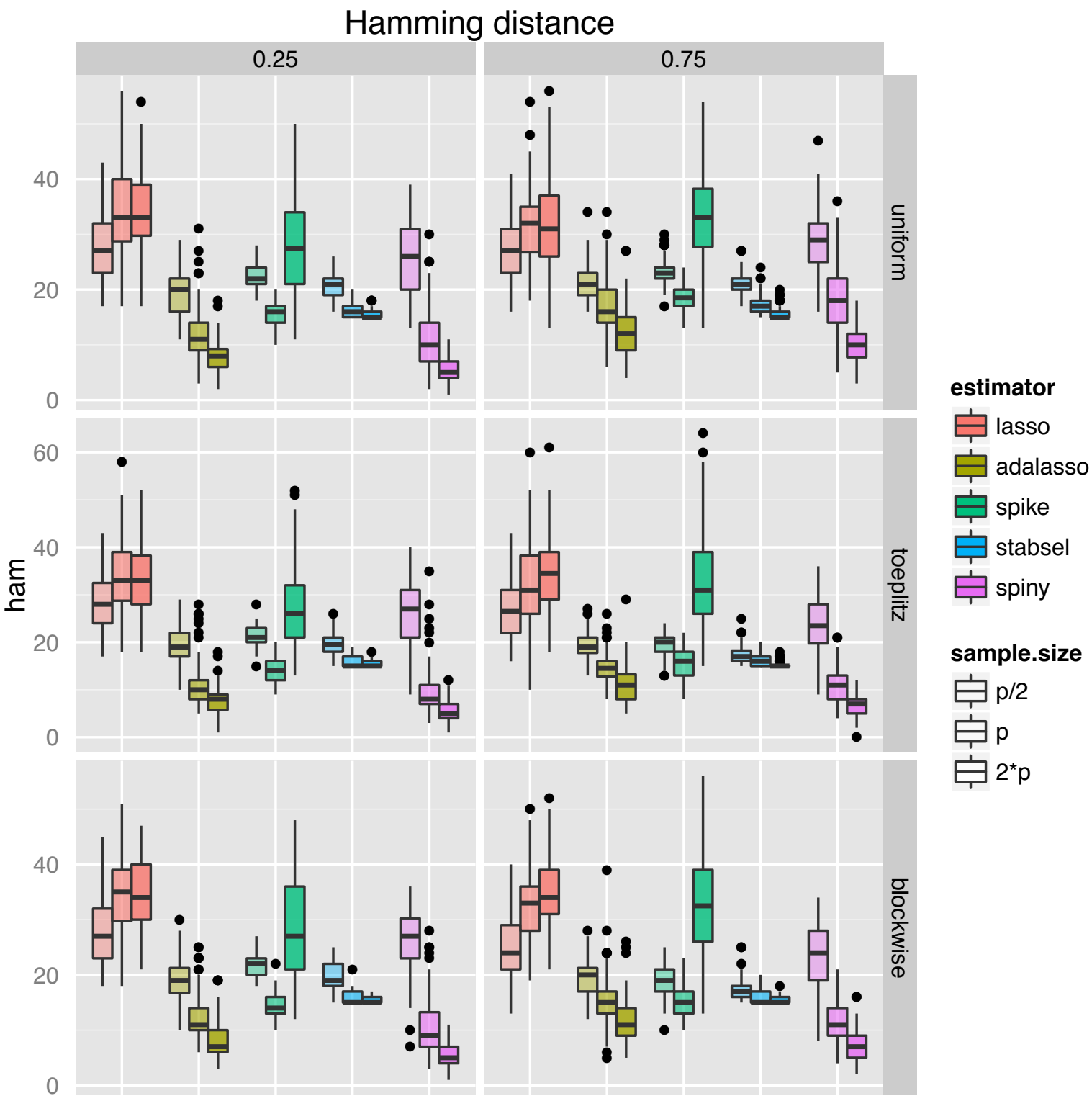

True positive rate
0.25
0.75
uniform
toeplitz
blockwise
tpr
1.00
0.75
0.50
0.25
0.00
estimator
lasso
adalasso
spike
stabsel
spiny
sample.size
p/2
p
2*p


False positive rate
0.25
0.75
uniform
toeplitz
blockwise
fpr
0.0
0.2
0.4
0.6
0.8
estimator
lasso
adalasso
spike
stabsel
spiny
sample.size
p/2
p
2*p


Positive predictive values
0.25
0.75
uniform
toeplitz
blockwise
ppv
1.00
0.75
0.50
0.25
0.00
estimator
lasso
adalasso
spike
stabsel
spiny
sample.size
p/2
p
2*p


F−score
0.25
0.75
uniform
toeplitz
blockwise
fsc
1.00
0.75
0.50
0.25
estimator
lasso
adalasso
spike
stabsel
spiny
sample.size
p/2
p
2*p


sparsity level
0.25
0.75
uniform
toeplitz
blockwise
pdns
0
20
40
60
80
0
25
50
75
estimator
lasso
adalasso
spike
stabsel
spiny
sample.size
p/2
p
2*p


Hamming distance
0.25
0.75
uniform
toeplitz
blockwise
ham
0
20
40
60
estimator
lasso
adalasso
spike
stabsel
spiny
sample.size
p/2
p
2*p