IEOR 262B: Mathematical Programming II
Final Project Report

Review and Comparison of Subset Selection Methods for Linear Regression

Miguel Fernández-Montes
University of California, Berkeley

Abstract

Many applications of regression modeling benefit from sparse parameter estimates, making the best subset selection problem a desired, although difficult, solution, which has been approximated by popular heuristics such as the Lasso.

This work presents results of extensive experiments comparing four methods for subset selection in linear regression, namely the Lasso, relaxed Lasso, a discrete first-order method presented by Bertsimas et al. [2] and a mixed integer programming formulation of the best subset selection problem.

Both statistical performance and support recovery are evaluated, across different dimensionalities, noise levels, sparsity patterns and correlation factors, leading to the following main insights:

1. The Lasso yields good statistical performance in noisy settings
2. The relaxed Lasso performs well across a variety of scenarios, both in terms of statistical accuracy as well as support recovery
3. Best subset selection and the above-mentioned discrete first-order method provide the best support recovery, specially in high dimensional settings.
4. The discrete first-order method can give the same results as an MIP solution and may prove useful in other applications beyond linear regression

1 Introduction

In many statistical modeling scenarios we desire to estimate a *sparse* parameter vector. In particular, it may be desirable to select the best k features among all possible p predictors (with $k < p$). Sparsity leads to more interpretable results, eliminates data redundancies and, if the underlying data generating process is actually sparse, it will lead to better statistical performance. In the high dimensional setting, that is, when the number of features is similar or greater than the number of observations, subset selection becomes necessary for most statistical inference needs.

The problem of interest for this work is linear regression, in which we assume the following model:

$$y = X\beta + \epsilon \tag{1}$$

where $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$, $\beta \in \mathbb{R}^p$ and ϵ is a random noise term.

The best subset selection problem for linear regression can be formulated as the following optimization problem:

$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 \quad \text{s.t.} \quad \|\beta\|_0 \leq k \tag{2}$$

Note that this problem is non-convex and, in fact, NP-hard

A popular heuristic that aims to yield an approximate solution to the best subset selection problem is the Lasso [7], which solves the following convex optimization problem:

$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1; \quad \text{with } \lambda \in \mathbb{R}_+ \tag{3}$$

However, the Lasso introduces shrinkage, and therefore bias, through its L_1-norm penalty term. An approach that finds a compromise between the *shrunk* Lasso solution and the "free" least-squares solution is the relaxed Lasso [6]. Other, more sophisticated approaches to approximate the best subset solution are those that use non-convex penalties, such as the smoothly clipped absolute deviations (SCAD) [4].

This work presents a comparison between several methods, namely the Lasso, relaxed Lasso, a discrete first-order method for subset selection and a mixed integer formulation of the best subset problem, via a series of experiments on synthetic data with varying characteristics as well as on real datasets.

1.1 Main contributions

This project aims to elaborate on the comparisons of exact and approximate methods for the best subset selection problem developed in previous literature [2], [5]. We consider that this work expands on the previous work through the following contributions:

- Experimentation across a wider range of signal-to-noise ratios than previous works
- Experimentation across a wider range of correlation factors between predictors than previous works
- Direct comparison of a discrete first-order method with other heuristics (Lasso, relaxed Lasso) and the best subset mixed integer solution across a variety of scenarios
- Evaluation of support recovery metrics such as missed detection rate or false alarm rate

- Implementation of methods and experiments in the Python programming language

Note that this project, which developes a comparison of, essentially, a series of optimization problems and algorithms, is intimately tied to statistical modeling and thus relies on several concepts from statistics, specially when it comes to method performance evaluation.

2 Background

The work presented here is primarily inspired by the results of Bertsimas et al. [2] as well as the results of Hastie et al. [5]. In fact, this project can be seen as re-implementation and combination of many of the experiments described in these two works with the additional contributions mentioned above.

Bertsimas et al. develop a discrete first-order method for best subset selection and show that a mixed integer formulation of the problem at hand yields good solutions relatively quickly. Both of these methods will be included in our experiments.

Hastie et al. provide an extensive set of comparisons between several methods, including the best subset as a mixed integer program, in a wider range of scenarios than those presented by Bertsimas et al. The authors conclude that there is no clear winner between best subset, which performs specially well in very low noise cases, and Lasso, which performs specially well in noisier scenarios. Additionally, Hastie et al. conclude that the relaxed Lasso is able to perform well across all settings, which has motivated the inclusion of this method in our own experiments.

3 Methodology

We have carried out several experiments in which various methods for subset selection in linear regression have been applied to datasets of varying characteristics. First, we give a description of the relevant methods. Then, we describe the data generation process and the experimental setup.

3.1 Methods for the best subset selection problem in regression

3.1.1 Discrete first-order method for subset selection

Bertsimas et al. [2] present a discrete first-order method that yields k-sparse solutions to the linear regression estimation problem. Basically, the first-order approximation of the objective function (in this case, the least squares loss) is minimized under a cardinality constraint at each iteration. This work employs the variant of the algorithm presented below, which has been implemented as a Python routine.

Algorithm 1: Discrete first-order method

initialization: β_1 such that $\|\beta\|_0 \leq k$;
while $g(\beta_t) - g(\beta_{t+1}) > \epsilon$ **do**

$$\beta_{t+1} \leftarrow \texttt{threshold}_k\left(\beta_t - \frac{1}{L}\nabla g(\beta)\right)$$

end

Here, we have left $g(\cdot)$ as a general, L-smooth differentiable function, as this method is applicable to more scenarios than linear regression. Note that the $\texttt{threshold}_k(\cdot)$ operator retains the top k values of its input (in absolute value) and sets the rest to zero.

In practice, we run the algorithm for fifty different random initializations in the following manner. For each set of 50 runs, the first initial β_{init} is set to either a thresholded version of the least squares solution or to a thresholded version of the marginal regression coefficient vector, depending on the dimensionality of the data. Then, subsequent initializations are created by adding a random vector to β_{init}, inspired by a similar implementation from the authors of [5]. The solution yielding the lowest objective function is kept.

3.1.2 Best subset mixed integer formulation

The best subset problem can be formulated as a mixed integer program, presented below:

$$\begin{aligned} \min_{\beta,z} \quad & \tfrac{1}{2}\|y - X\beta\|_2^2 \\ \text{s.t.} \quad & -\mathcal{M}_U z_i \leq \beta_i \leq \mathcal{M}_U z_i \quad \forall i = 1, ..., p \\ & \textstyle\sum_{i=1}^{p} z_i = k \\ & z_i \in \{0, 1\} \qquad\qquad \forall i = 1, ..., p \end{aligned} \tag{4}$$

Note that the objective is a quadratic function, which can also be expressed as:

$$\frac{1}{2}\beta^T X^T X \beta - (X^T y)^T \beta + \frac{1}{2}\|y\|_2^2$$

In equation 4, $\mathcal{M}_U$ denotes an upper bound on the absolute value of the coefficients, which can be provided via the solution of the first-order method shown earlier or through theoretical considerations described in [2].

For the case where $p > n$, the formulation can be manipulated into a problem with n-dimensional decision variables:

$$\begin{aligned} \min_{\beta,z,\zeta} \quad & \tfrac{1}{2}\zeta^T\zeta - (X^T y)^T \beta + \tfrac{1}{2}\|y\|_2^2 \\ \text{s.t.} \quad & \zeta = X\beta \\ & -\mathcal{M}_U z_i \leq \beta_i \leq \mathcal{M}_U z_i \qquad \forall i = 1, ..., p \\ & -\mathcal{M}_U^{\zeta} \leq \zeta_i \leq \mathcal{M}_U^{\zeta} \qquad \forall i = 1, ..., n \\ & \textstyle\sum_{i=1}^{p} z_i = k \\ & z_i \in \{0, 1\} \qquad\qquad \forall i = 1, ..., p \end{aligned} \tag{5}$$

In equation 5, bounds for the auxiliary variable ζ are provided as prescribed by Bertsimas et al., as it improves the quality of the solution. Additional bounds are recommended, as well as formulating the sparsity constraints via specially ordered sets, but these modifications have not been included in the formulation employed in this work

The optimization problem can be solved using the popular Gurobi solver, which employs branch and cut techniques. For quicker convergence, the problem benefits from *warm-starts* and bounds given by the solution of the first-order method mentioned above.

3.1.3 Lasso

The Lasso optimization problem (3) will be solved by using the "out-of-the-box" implementations provided by the `scikit-learn` Python package[1]. For the high dimensional setting the Least Angle Regression solver is used whereas for the low dimensional setting the default coordinate descent method is applied.

3.1.4 Relaxed Lasso

The relaxed Lasso solution can be expressed as

$$\hat{\beta}_{relax} = \gamma\hat{\beta}_{lasso} + (1-\gamma)\hat{\beta}_{LS}, \quad \gamma \in [0,1] \tag{6}$$

where $\hat{\beta}_{LS}$ is the least-squares estimate for the problem considering *only* the active set of coefficients detected by the Lasso (note that since the vector will be padded with zeros for all coefficients that do not belong in the active set). The parameter γ measures how much weight is given to the regularized solution from the Lasso as opposed to the unregularized least-squares solution.

3.2 Synthetic datasets

For our experimentation purposes, regression data has been artificially generated in the following manner. A design matrix X is drawn from a multivariate Gaussian distribution $X \sim N(\mathbf{0}, \Sigma)$ where the covariance matrix has entries equal to $\Sigma_{ij} = \rho^{|i-j|}$. The response vector is then generated as $y = X\beta_0 + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and β_0 is the true parameter vector, with a sparsity level (number of non-zero entries) of k_0. The variance of the error terms can be related of the signal-to-noise ratio (SNR), a useful measure of the "noisiness" in the data:

$$\text{SNR} = \frac{\text{Var}(\beta_0^T x)}{\sigma^2} \tag{7}$$

Just like in previous research, we are interested in evaluating the performance of the aforementioned approaches at different dataset dimensionalities, sparsity patterns, noise levels and correlation between predictors.

Thus, we define three different settings, according to the dimensionality of our data:

- **low**: $n = 100, \quad p = 10, \quad k_0 = 5$
- **mid**: $n = 500, \quad p = 100, \quad k_0 = 5$
- **high**: $n = 50, \quad p = 1000, \quad k_0 = 5$

Besides, emulating the work of Hastie et al., we define different sparsity patterns:

- **Type 1**: the first k_0 elements of β_0 are set equal to one and the rest to zero
- **Type 2**: k_0 elements of β_0 at equally spaced intervals are set equal to one and the rest to zero
- **Type 3**: the first k_0 elements of β_0 are set to equally spaced values ranging from 10 to 0.5 and the rest are set to zero

[1] `https://scikit-learn.org/stable/`

- **Type 5** (weak sparsity): the first k_0 elements of β are set equal to one and the rest are set to 0.5^{i-k_0} for $i = k_0, ..., p$

For all experiments, ten values of the SNR, equally spaced in a logarithmic scale between 0.05 and 8, have been tried for each setting and sparsity pattern. For the experiments in the low and mid settings, the correlation factor ρ takes on four values: 0, 0.35, 0.7 and 0.9. Due to computational constraints, experiments on the high dimensional setting were done using correlation factors of 0 and 0.7, and for exclusively one sparsity pattern.

3.2.1 Metrics

In order to evaluate the statistical performance of each method we employ the following metrics, which mirror those employed in [5]:

- **Relative risk**

$$\mathrm{RR}(\hat{\beta}) = \frac{\mathbb{E}(x^T\hat{\beta} - x^T\beta_0)^2}{\mathbb{E}(x^T\beta_0)^2} = \frac{(\hat{\beta} - \beta_0)^T\Sigma(\hat{\beta} - \beta_0)}{\beta_0^T\Sigma\beta_0} \tag{8}$$

- **Relative test error**

$$\mathrm{RTE}(\hat{\beta}) = \frac{\mathbb{E}(y - x^T\hat{\beta})^2}{\sigma^2} = \frac{(\hat{\beta} - \beta_0)^T\Sigma(\hat{\beta} - \beta_0) + \sigma^2}{\sigma^2} \tag{9}$$

- **Proportion of variance explained**

$$\mathrm{PVE}(\hat{\beta}) = 1 - \frac{(\hat{\beta} - \beta_0)^T\Sigma(\hat{\beta} - \beta_0) + \sigma^2}{\beta_0^T\Sigma\beta_0 + \sigma^2} \tag{10}$$

This work also aims to evaluate the sparsity and quality of the support recovery yielded by each method, that is, how well the estimated parameter vector resembles the true parameter vector. For this purpose we define the following metrics:

- **Number of non-zeros** which can be expressed as the L_0-(pseudo)norm $\|\hat{\beta}\|_0$
- **False alarm rate**

$$\mathrm{FAR}(\hat{\beta}) = \frac{1}{\|\hat{\beta}\|_0}\sum_{i=1}^{p}\mathbf{1}\{\hat{\beta}_i \neq 0, \beta_{0i} = 0\} \tag{11}$$

- **Missed detection rate**

$$\mathrm{MDR}(\hat{\beta}) = \frac{1}{\|\beta_0\|_0}\sum_{i=1}^{p}\mathbf{1}\{\hat{\beta}_i = 0, \beta_{0i} \neq 0\} \tag{12}$$

- **Relative beta (parameter) error**

$$\mathrm{RBE}(\hat{\beta}) = \frac{\|\hat{\beta} - \beta_0\|^2}{\|\beta_0\|^2} \tag{13}$$

Note that the last three metrics only make sense if we know the true parameter vector, which in real applications is rarely the case. These metrics simply aim to give an idea of how well our methods would recover sparse regression coefficients.

3.2.2 Experimental setup

The experimental setup developed in this work follows that of Hastie et al. [5] with minor variations:

1. Generate training data according to dimensionality setting, sparsity pattern, SNR and correlation factor. We also create a validation set that is generated in the same manner.

2. Run the four methods of interest: Lasso, relaxed Lasso, discrete first-order method and best subset (MIP) on the generated data. For each method a range of parameters is tried and the combination that yields lower MSE on the validation set is kept.

3. Evaluate metrics of interest for statistical accuracy and support recovery. Metrics are defined in the next subsection.

4. Repeat this process 10 times and average metrics results.

3.3 Real datasets

The methods have also been tried on two semi-synthetic datasets, generated from real gene microarray datasets: the `lymphoma` dataset and the `prostate` dataset, both included in the SPLS R package. The lymphoma dataset contains $n = 62$ records and $p = 4026$ features. The prostate dataset contains $n = 102$ records and $p = 6033$ features. In both cases, the top 1000 features, according to their correlation with the response, have been kept[2]. A synthetic continuous response vector has been generated as $y = X\beta + \epsilon$ where the first 10 elements of β are set to one and the rest to zero. The noise term is drawn from a Gaussian distribution so as to yield a SNR of 5.

The experiments in the real datasets are limited to running the four methods of interest, choosing their parameters via validation on a hold out set. Then, the mean squared error and out-of-sample R^2 are measured in a test set that represents 15% of the original observations. Additionally, we measure the support recovery metrics defined earlier (MDR, FAR, relative parameter error and number of nonzeros). Note that the size of these datasets, in terms of number of records, is very small, which makes validation and testing in hold out sets relatively unreliable. A better approach for future work would involve using k-fold cross validation.

4 Results

Several visualizations representing the evaluation metrics for each method on every setting are included in the Appendix. In the following subsection we show the results for a particular setting and discuss the main insights gathered from all the experiments. Subsequently, we show the evaluation results from the experiments on the real datasets.

4.1 Results on synthetic data

Figure 1 shows the relative risk and relative test error for the low dimensional setting with sparsity pattern type 2, at two different correlation levels.

[2] a `scikit-learn` out-of-the-box method has been employed for the selection of the top 1000 predictors

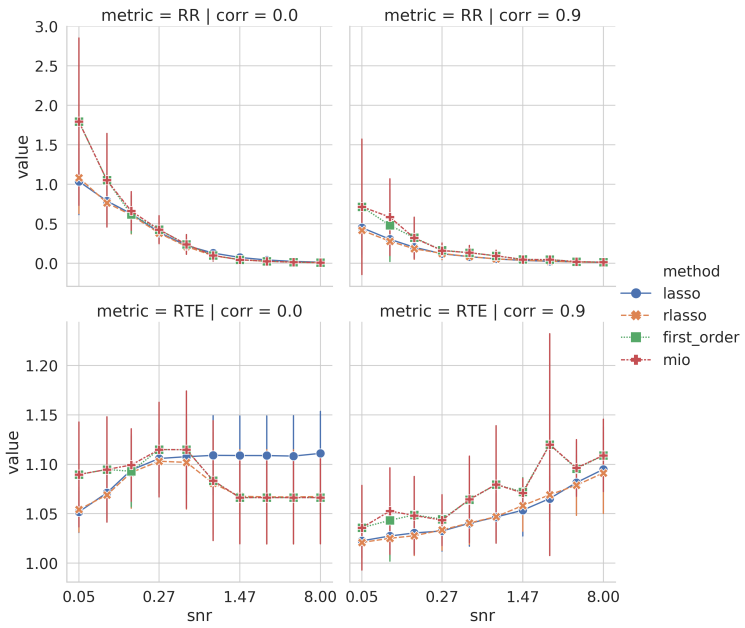

Figure 1: Statistical performance in the low dimensional setting, β type 2.

After reviewing our results, in the **low dimensional** setting, for practically all sparsity patterns we see that:

- Lasso and relaxed Lasso show better statistical performance than best subset and first-order for low SNR values
- Best subset (MIP) and the discrete first-order method significantly improve their statistical performance at high SNR values.
- The relaxed Lasso is able to give similar relative test errors than best Subset and first-order at high SNRs whereas the Lasso does not. This highlights the inherent *bias* in the Lasso estimator as opposed to the unbiased estimator given by best subset.
- The performance of the first-order method is nearly dentical to that of the best subset MIP formulation. In fact the regression coefficient estimates are identical in most cases, which leads to think that for the low dimensional case the first-order method can be a good approximation for the best subset problem, in case we care about sparsity.
- At very high correlation values ($\rho = 0.9$), the Lasso and Relaxed Lasso are the winners, even when noise levels are low. It appears that best subset and first-order become "confused" given the high correlation among predictors and end up yielding a high relative test error, which in the case of Lasso is smaller, perhaps given its regularization properties.
- The proportion of variance explained is very similar for all methods.

In the **mid dimensional** we see similar statistical results but we note that the Lasso starts giving worse performance. In fact, there are less differences in relative risk at lower SNRs between the four methods.

Let us now look at the support recovery metrics for an example in the mid dimensional setting, as depicted in figure 2. We see that

- Lasso always overestimates the number of nonzero coefficients.
- Best subset and first-order method are able to yield the right sparsity level, with very low (and normally

zero) false alarm rate, even at low SNR levels and high correlations, performing better than the other two methods

- The relaxed Lasso is able to fix some the problems of the Lasso when it comes to support recovery but only gets close to the performance of best subset at low-noise levels (high SNR)

This behavior is in general true for all sparsity patterns. It is interesting to note that with weak sparsity (sparsity pattern type 5) at high correlations all methods give less sparse solutions with growing SNR. Best subset and the first-order method, however, yield always sparser representations than the Lasso and relaxed Lasso. The reader is referred to the appendix for a closer inspection of these results.

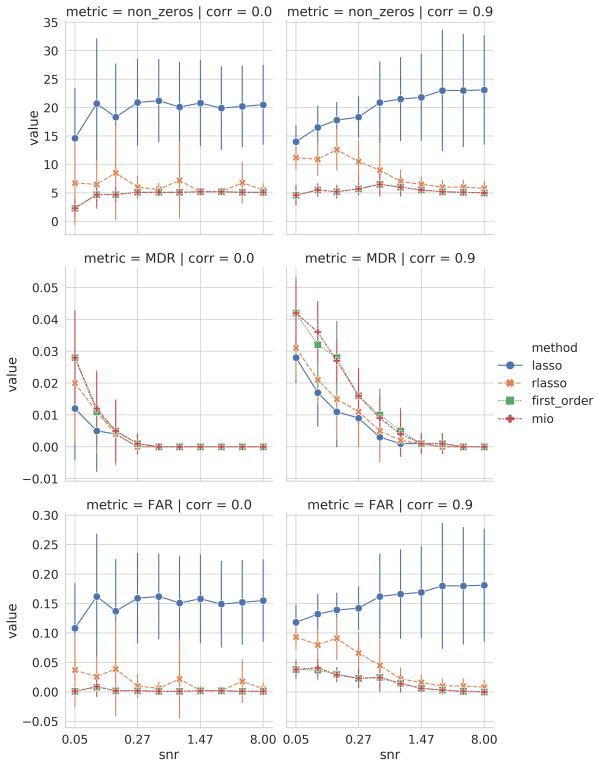

Figure 2: Support recovery performance in the mid dimensional setting, β type 2.

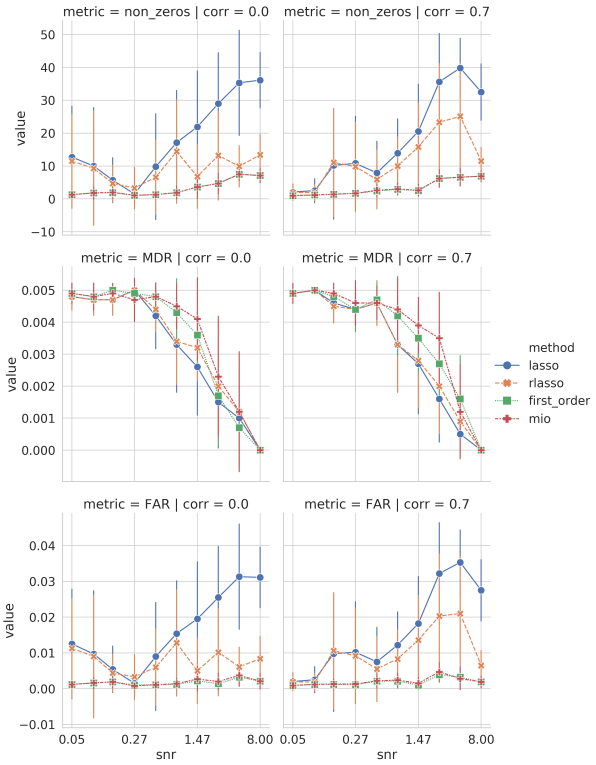

Figure 3: Support recovery performance in the high dimensional setting, β type 2.

Finally, the **high dimensional** case brings up interesting insights. Figure 4 shows the proportion of variance explained for a particular setting. It can be seen that best subset and discrete first-order method yield *negative* values of the PVE for low SNR regimes, and either zero or slightly negative values for the Lasso and relaxed Lasso. As SNR increases, best subset and first-order are able to match and finally surpass the Lasso and relaxed Lasso in terms of explained variance

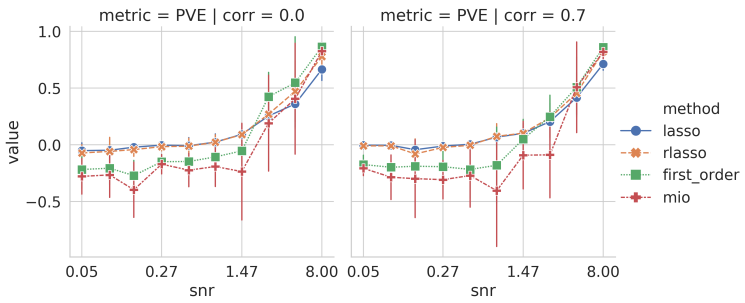

Figure 4: Proportion of variance explained in the high dimensional setting, β type 2.

However, when it comes to support recovery, only the best subset and first-order solutions give an appropriate sparsity level for the high dimensional case, at least in our experiments, as shown in figure 3. Note that not even the relaxed Lasso is able to give an appropriate number of nonzero coefficients or a low false alarm rate.

4.2 Results on the `lymphoma` and `prostate` datasets

In table 1 we see the results for each of our four methods of interest in the semi-synthetic generated from the real `lymphoma` dataset

Table 1: Results from each method on the `lymphoma` data

method	**test_mse**	**OSR2**	**non_zeros**	**beta_error**	**MDR**	**FAR**
lasso	5.4692	0.8084	2	0.9244	0.008	0
rlasso	5.4692	0.8084	2	0.9244	0.008	0
first_order	5.9758	0.7907	9	1.7777	0.009	0.008
best subset	10.0168	0.6491	9	3.5622	0.008	0.007

We see that the Lasso and relaxed Lasso, which yield the same solution in this case, show significantly better results than the best subset method. The first-order method falls closely behind Lasso in terms of statistical performance. It is surprising to see that the best subset solution performs the previous three in this case, even when warm-started with a better performing solution, given by the first-order method. Note, however, that the best subset method is a *high variance* approach and recall the small size of our validation and test sets, which may explain these results. Finally, it is important to notice that the first two methods are yielding surprisingly sparse solutions, with only two non-zero parameters, whereas first-order and best subset closely approximate the true sparsity level ($k_0 = 10$).

In table 2 we see the results for each of our four methods of interest in the semi-synthetic generated from the real `prostate` dataset.

Table 2: Results from each method on the `prostate` data

method	**test_mse**	**OSR2**	**non_zeros**	**beta_error**	**MDR**	**FAR**
lasso	6.6120	0.7820	24	0.9630	0.007	0.021
rlasso	6.6171	0.7819	22	0.9678	0.007	0.019
first_order	4.6099	0.8480	20	4.3538	0.007	0.017
mio	14.1268	0.5343	20	8.3086	0.01	0.02

Like before, the best subset method yields, by far, the worst performance in this case, perhaps due to similar

reasons. The discrete first-order method is the winner, at least regarding its out-of-sample R^2. Note that this is true despite the fact that the method does not recover the true parameter vector, as demonstrated by the high relative parameter error (beta_error), and with a higher number of nonzero coefficients than the ground truth.

5 Conclusions and further work

This work has compared four relevant method for subset selection in the linear regression setting, evaluating their statisical performance as well as their support recovery. We summarize the conclusions drawn from our results as follows:

Relaxed Lasso is a good compromise. In terms of overall statistical performance and support recovery, the relaxed Lasso appears to give good results across all scenarios, coming from its ability to couple the robustness of Lasso and the "unbiasedness" of a free least-squares estimator (after proper parameter tuning).

Lasso for noisy (and real) settings. In general, the Lasso yields better statistical results for low SNR levels and high correlation factors. It also appears to yield better out-of-sample performance on data coming from real applications. The regularization properties inherent to this method probably play a big role in its superior performance in these settings.

Best subset for best support recovery. The best subset solution (and the discrete first-order method) give the best support recovery in mid and high dimensional settings, even at high correlations and noise levels.

Insights regarding the discrete first-order method. Perhaps one of the more surprising results is the fact that in most cases, the discrete first-order method yields the exact same solution than the best subset (MIP) method and it yields better performance in some scenarios. Several aspects might explain the previous result. First, the Gurobi solver was only allowed a limited amount of time (3 minutes) to find a solution. Second, the validation scheme selected the optimal value of k (the sparsity constraint) by using the performance of the first-order method on the validation sets, rather than trying the best subset method for each possible value. Finally, our implementation of the best subset MIP did not include all the tips and tricks detailed by the original authors of [2] in order to improve performance.

That said, these results shed light on the benefits that the discrete first-order method presented in 3.1.1 could bring. In the low and mid dimensional settings this method is relatively fast and it is able to find solutions with good statistical performance at low noise levels. Moreover, in applications where sparsity and good support recovery are crucial, the first-order method is able to yield solutions that correctly recover the right sparsity level and that significantly lower the false alarm rate.

5.1 Further work

This work has only looked at a small pool of methods for subset selection, and limited its analysis to the case of linear regression, mainly on synthetic data. Nowadays, more complicated and efficient formulations of the best subset problem exist, such as those shown in [3], [1]. Further comparisons of these methods with the Lasso and other heuristics could be researched. Additionally, other statistical learning problems, such as classification, that can benefit from sparsity, may be explored. Finally, on a different note, different applications of the discrete first-order method described in 3.1.1 and Algorithm 1 could be explored, for instance, for sparse logistic regression or portfolio optimization problems.

References

[1] Alper Atamtürk and Andres Gomez. Rank-one convexification for sparse regression. *ArXiv*, abs/1901.10334, 2019.

[2] Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *arXiv: Methodology*, pages 813–852, 2015.

[3] Dimitris Bertsimas, Jean Pauphilet, and Bart Van Parys. Sparse regression: Scalable algorithms and empirical performance. *arXiv: Methodology*, 2019.

[4] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

[5] Trevor Hastie, Robert Tibshirani, and Ryan J. Tibshirani. Extended comparisons of best subset selection, forward stepwise selection, and the lasso, 2017.

[6] Nicolai Meinshausen. Relaxed lasso. *Comput. Stat. Data Anal.*, 52:374–393, 2007.

[7] Robert Tibshirani. Regression shrinkage and selection via the lasso. 1996.

A Appendix. Experiment results

A.1 Low dimensional setting

A.1.1 Low dimensional setting - Type 1 Sparsity - Statistical performance metrics

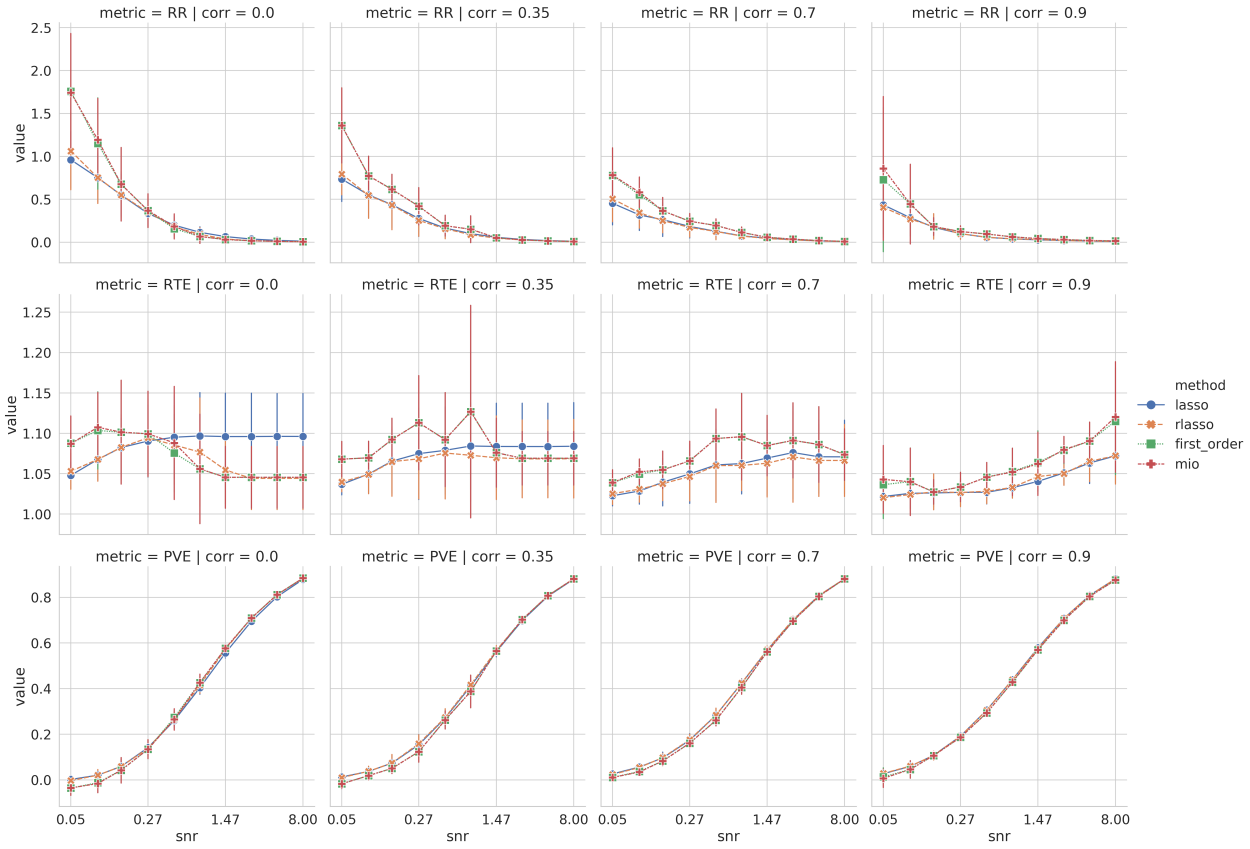

Figure 5: Statistical evaluation metrics for experiments in the low dimensional setting β type 1. RR: relative risk, RTE: relative test error, PVE: proportion of variance explained

A.1.2 Low dimensional setting - Type 1 Sparsity - Support recovery metrics

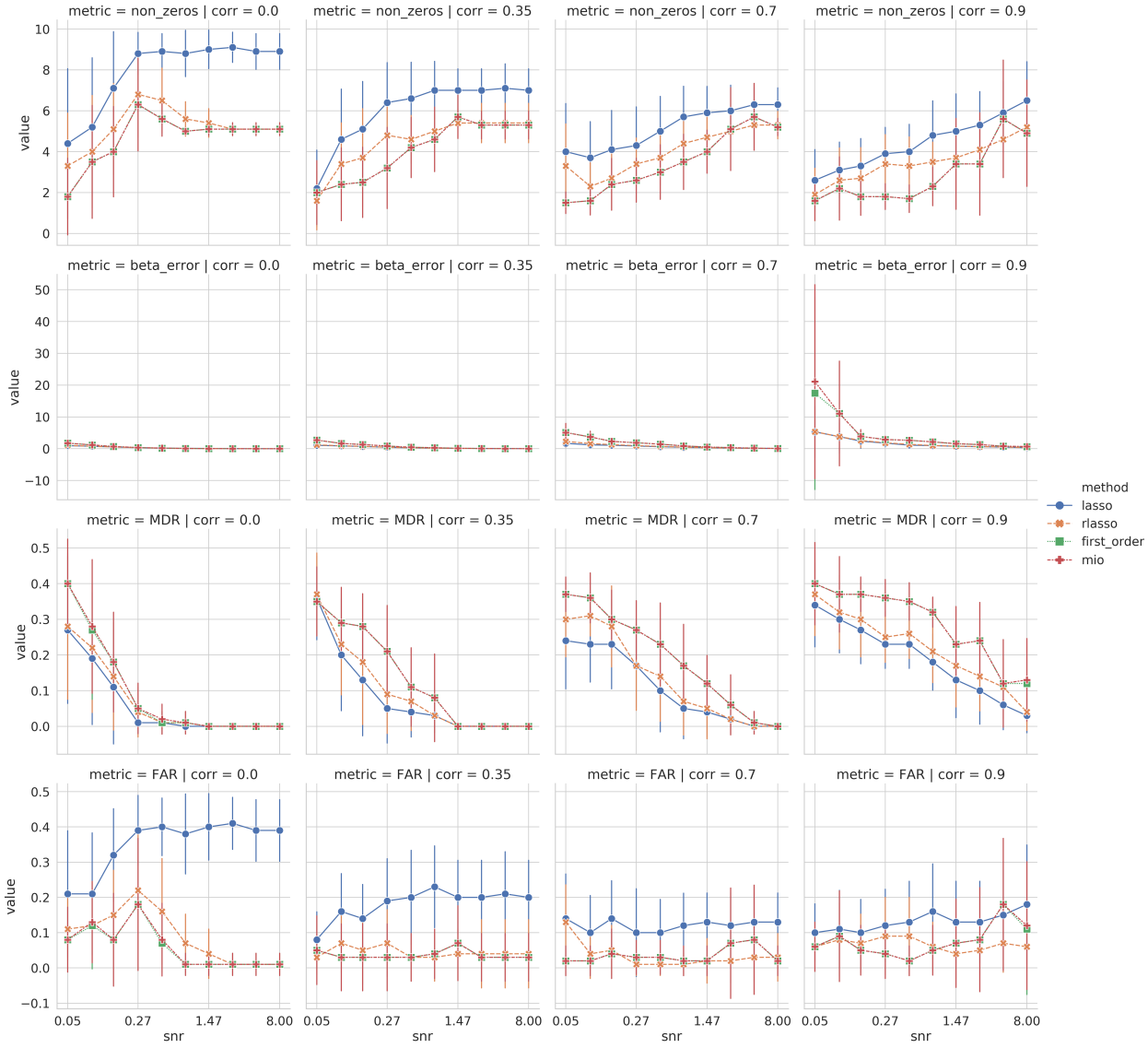

Figure 6: Support recovery evaluation metrics for experiments in the low dimensional setting β type 1. FAR: false alarm rate, MDR: missed detection rate, RBE: relative error of parameter vector

A.1.3 Low dimensional setting - Type 2 Sparsity - Statistical performance metrics

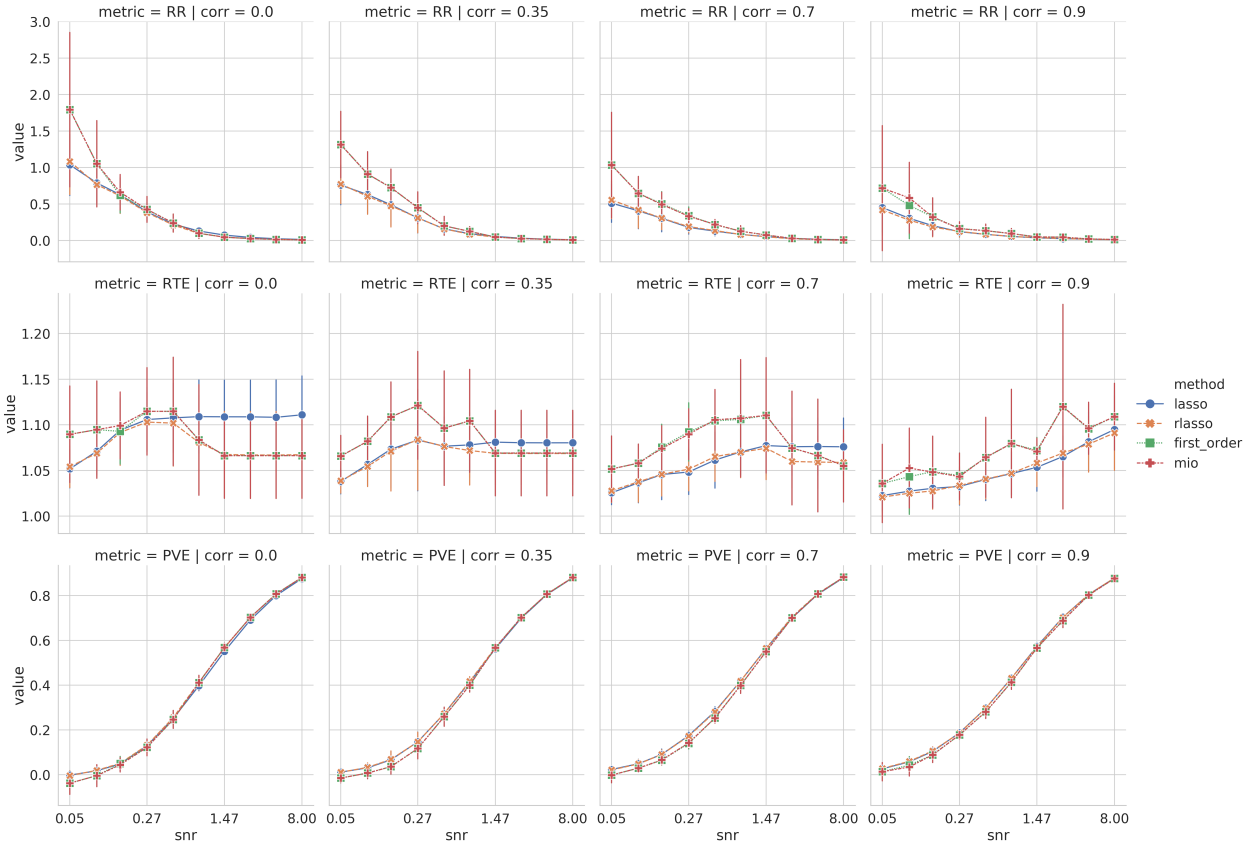

Figure 7: Statistical evaluation metrics for experiments in the low dimensional setting β type 2. RR: relative risk, RTE: relative test error, PVE: proportion of variance explained

A.1.4 Low dimensional setting - Type 2 Sparsity - Support recovery metrics

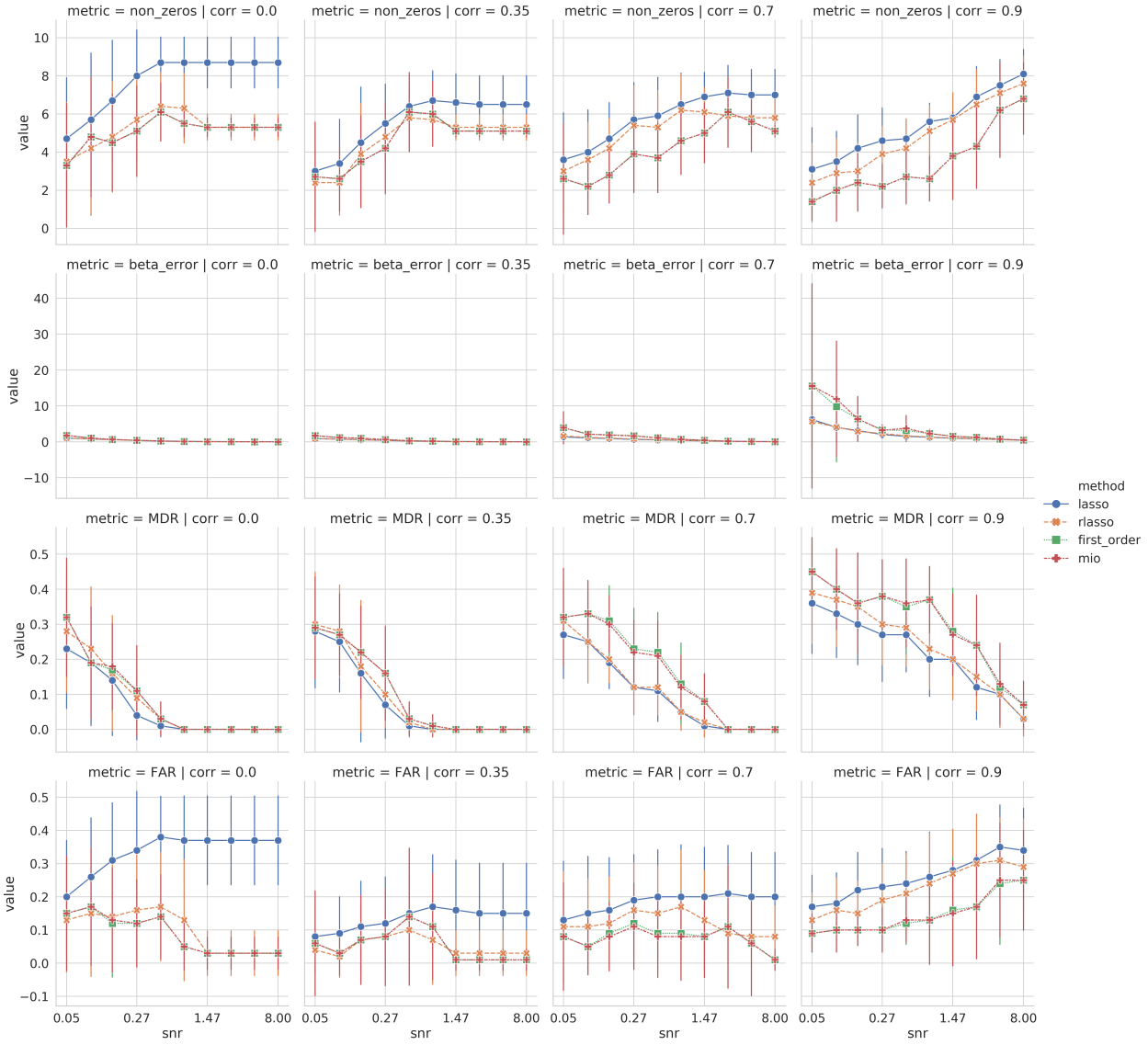

Figure 8: Support recovery evaluation metrics for experiments in the low dimensional setting β type 2. FAR: false alarm rate, MDR: missed detection rate, RBE: relative error of parameter vector

A.1.5 Low dimensional setting - Type 3 Sparsity - Statistical performance metrics

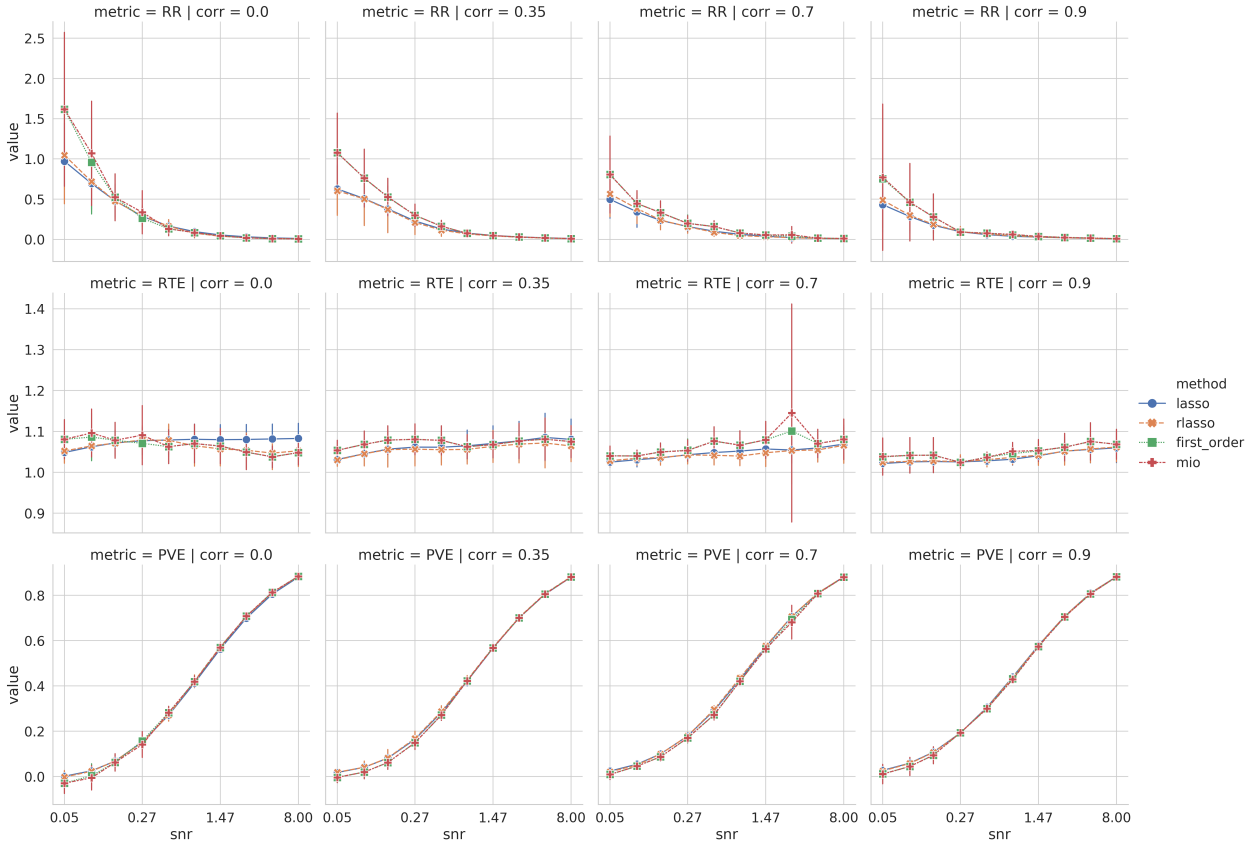

Figure 9: Statistical evaluation metrics for experiments in the low dimensional setting β type 3. RR: relative risk, RTE: relative test error, PVE: proportion of variance explained

A.1.6 Low dimensional setting - Type 3 Sparsity - Support recovery metrics

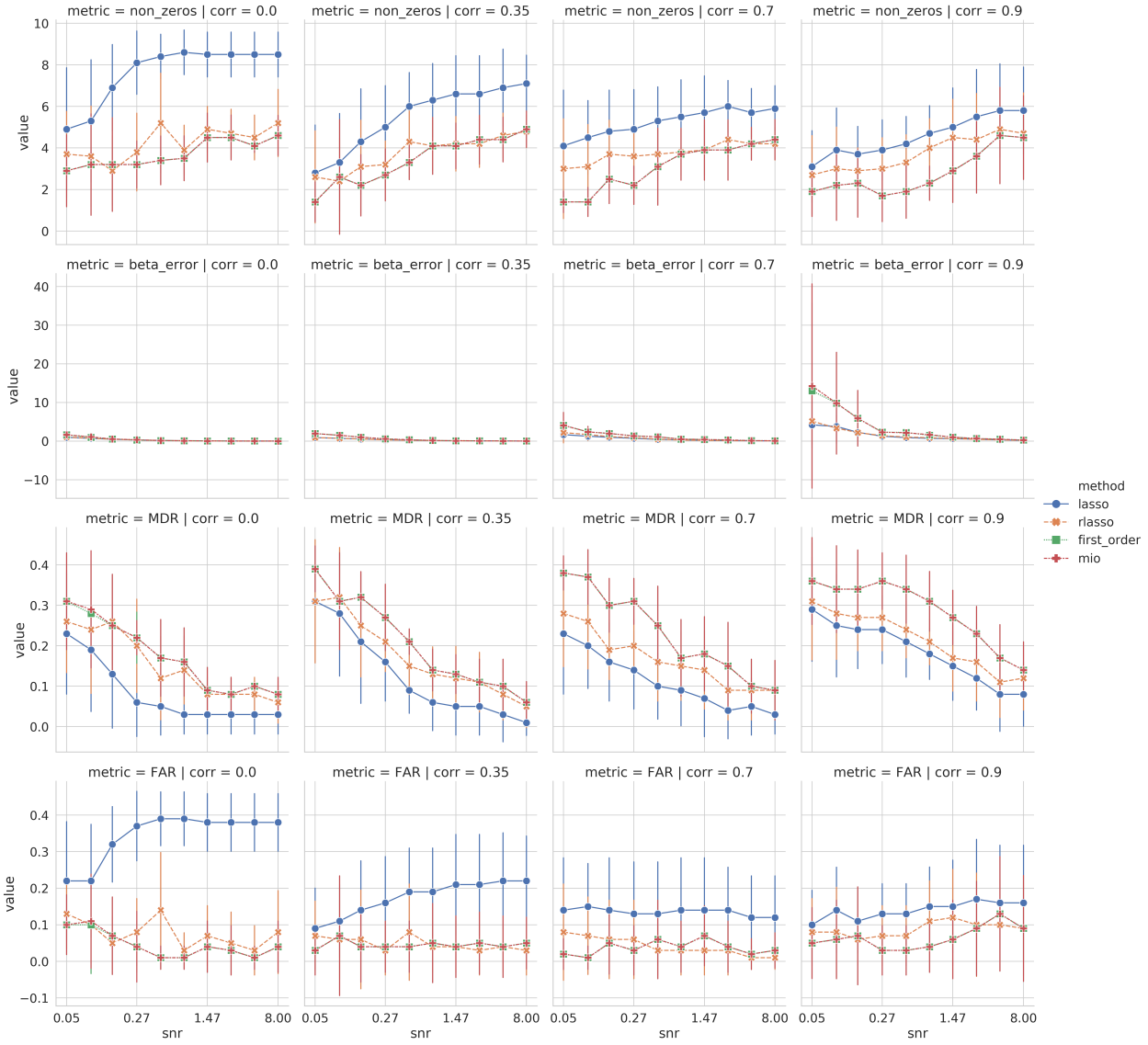

Figure 10: Support recovery evaluation metrics for experiments in the low dimensional setting β type 3. FAR: false alarm rate, MDR: missed detection rate, RBE: relative error of parameter vector

A.1.7 Low dimensional setting - Type 5 Sparsity - Statistical performance metrics

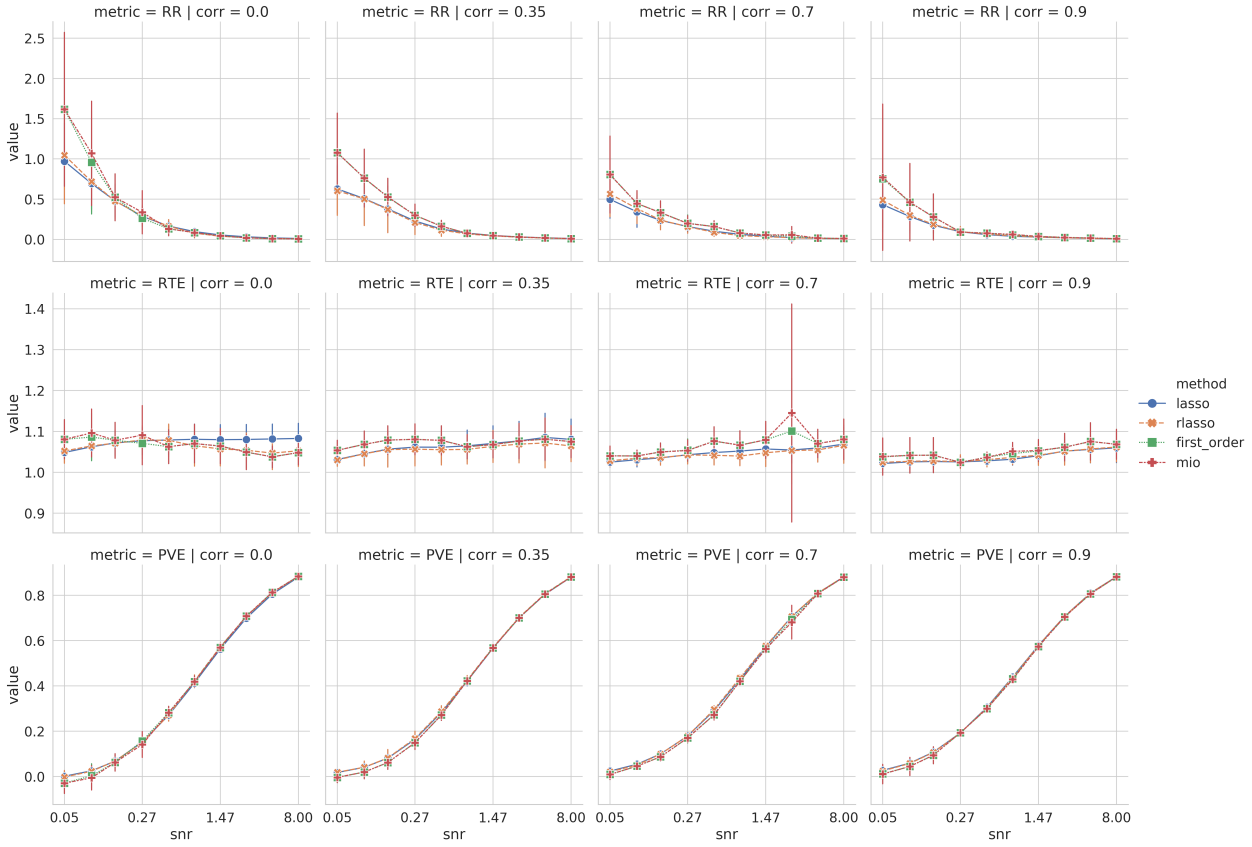

Figure 11: Statistical evaluation metrics for experiments in the low dimensional setting β type 3. RR: relative risk, RTE: relative test error, PVE: proportion of variance explained

A.1.8 Low dimensional setting - Type 5 Sparsity - Support recovery metrics

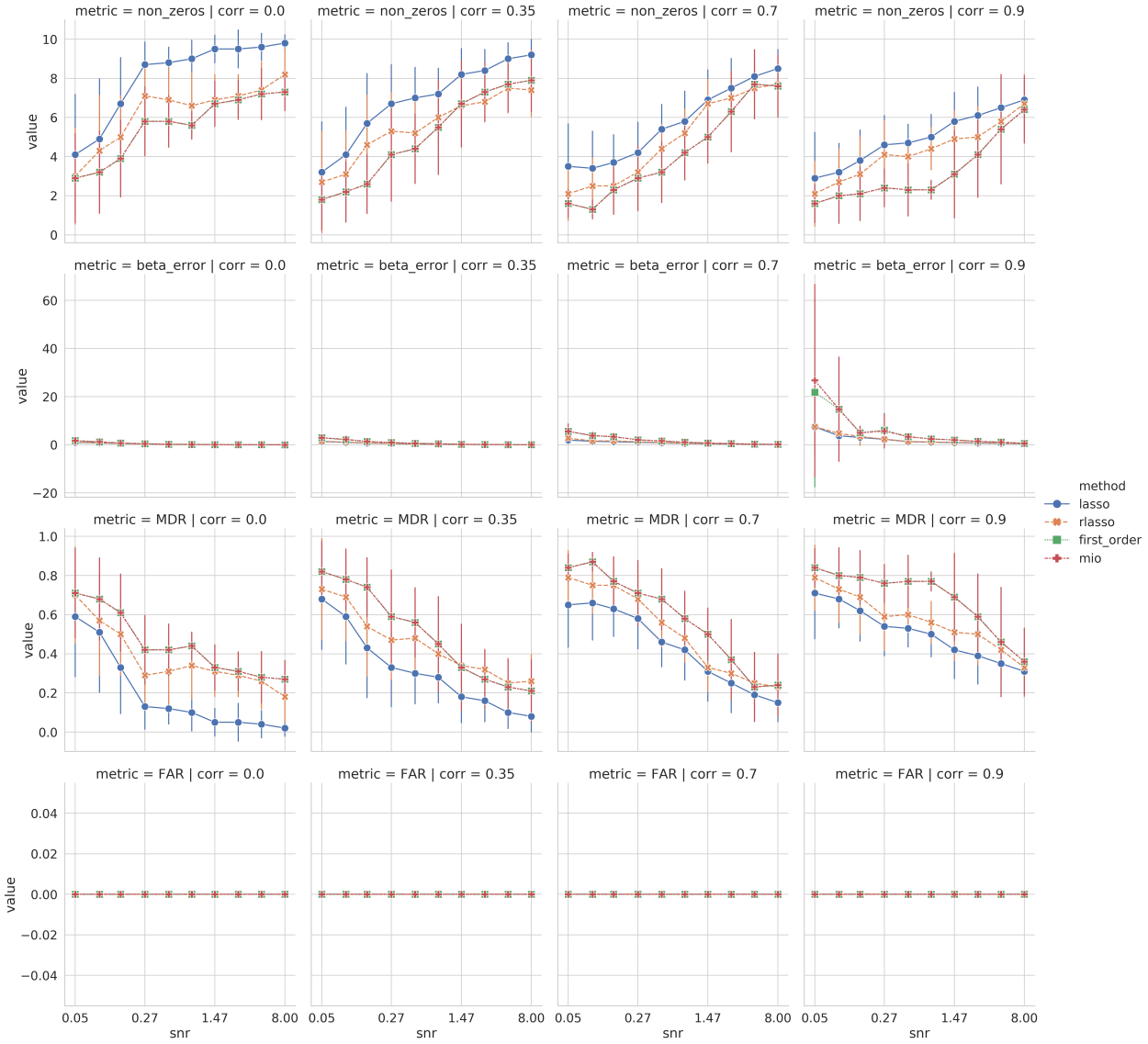

Figure 12: Support recovery evaluation metrics for experiments in the low dimensional setting β type 5. FAR: false alarm rate, MDR: missed detection rate, RBE: relative error of parameter vector

A.2 Mid dimensional setting

A.2.1 Mid dimensional setting - Type 1 Sparsity - Statistical performance metrics

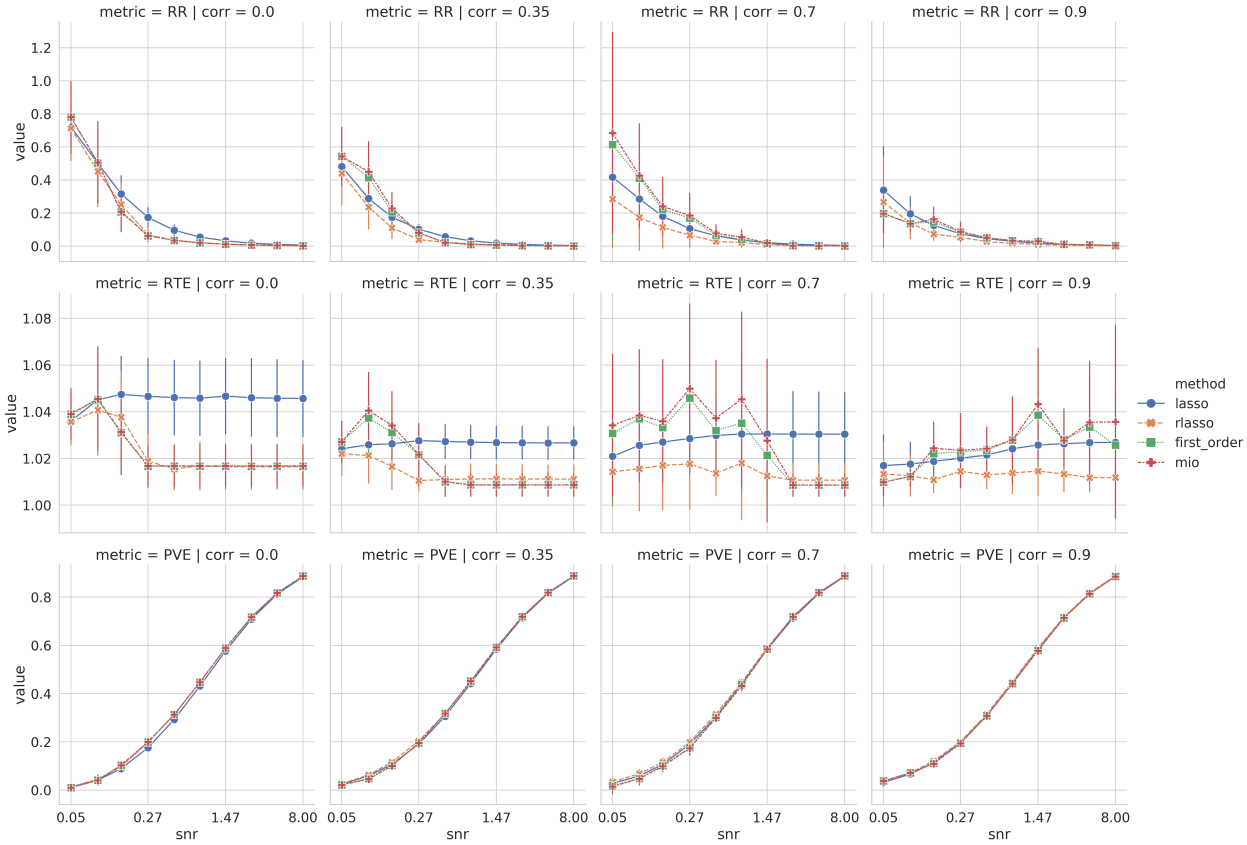

Figure 13: Statistical evaluation metrics for experiments in the mid dimensional setting β type 1. RR: relative risk, RTE: relative test error, PVE: proportion of variance explained

A.2.2 Mid dimensional setting - Type 1 Sparsity - Support recovery metrics

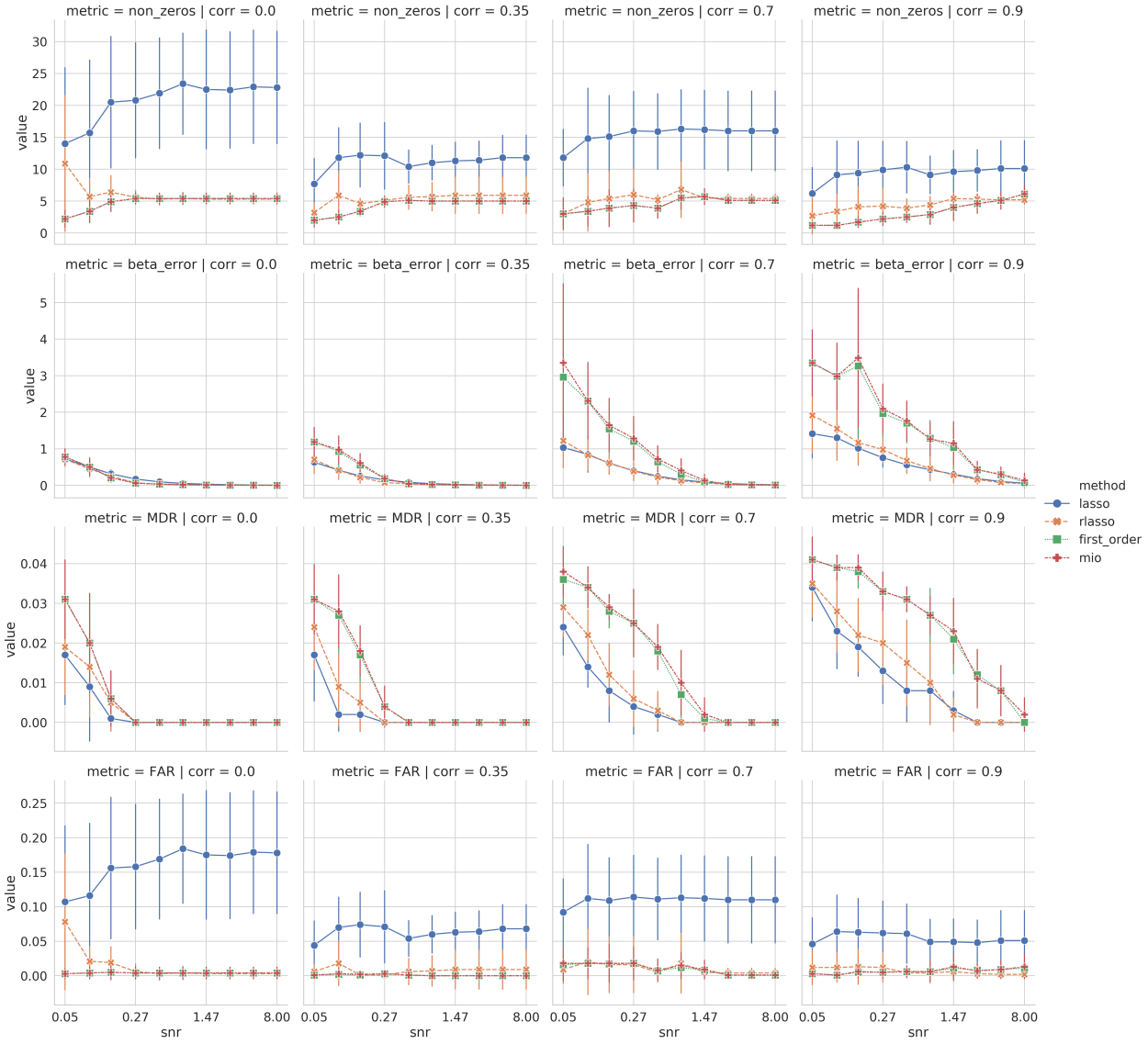

Figure 14: Support recovery evaluation metrics for experiments in the mid dimensional setting β type 1. FAR: false alarm rate, MDR: missed detection rate, RBE: relative error of parameter vector

Mid dimensional setting - Type 2 Sparsity - Statistical performance metrics

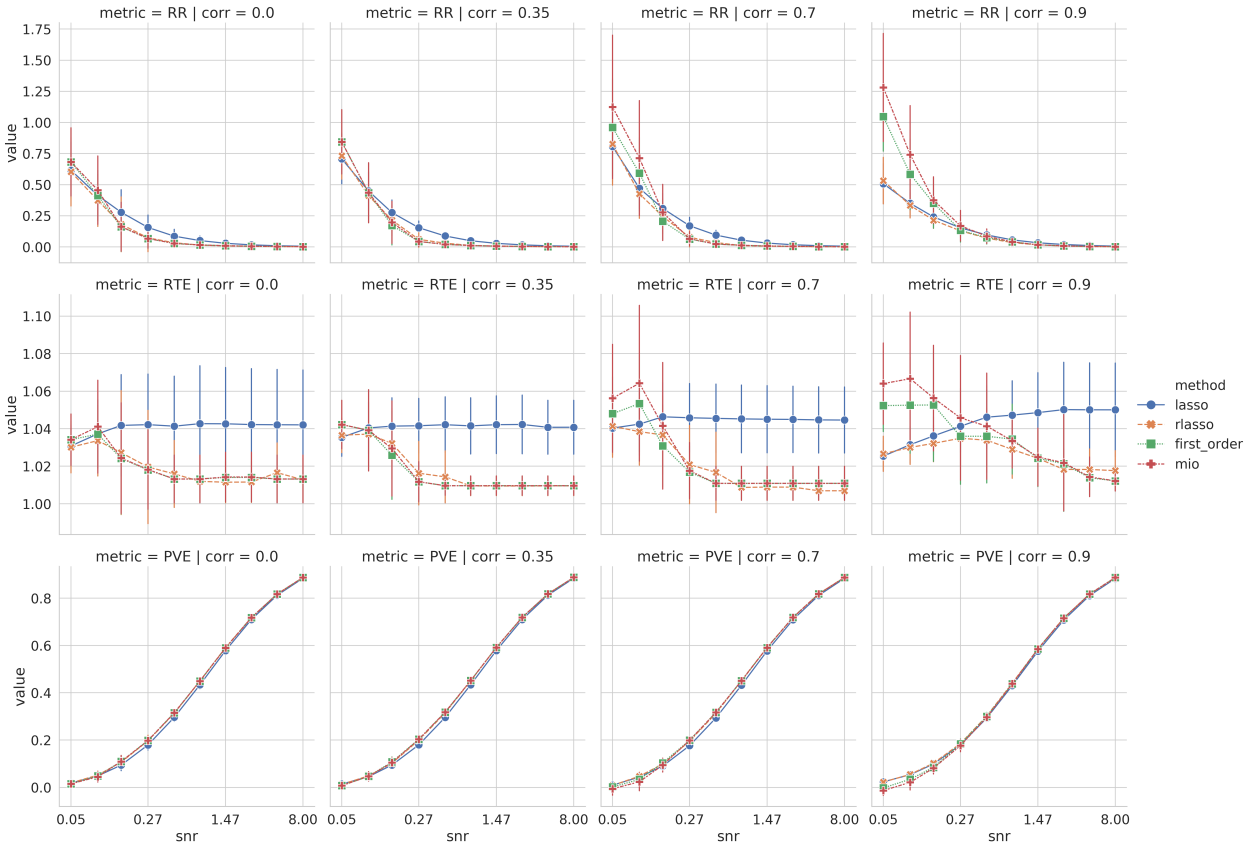

Figure 15: Statistical evaluation metrics for experiments in the mid dimensional setting β type 2. RR: relative risk, RTE: relative test error, PVE: proportion of variance explained

A.2.3 Mid dimensional setting - Type 2 Sparsity - Support recovery metrics

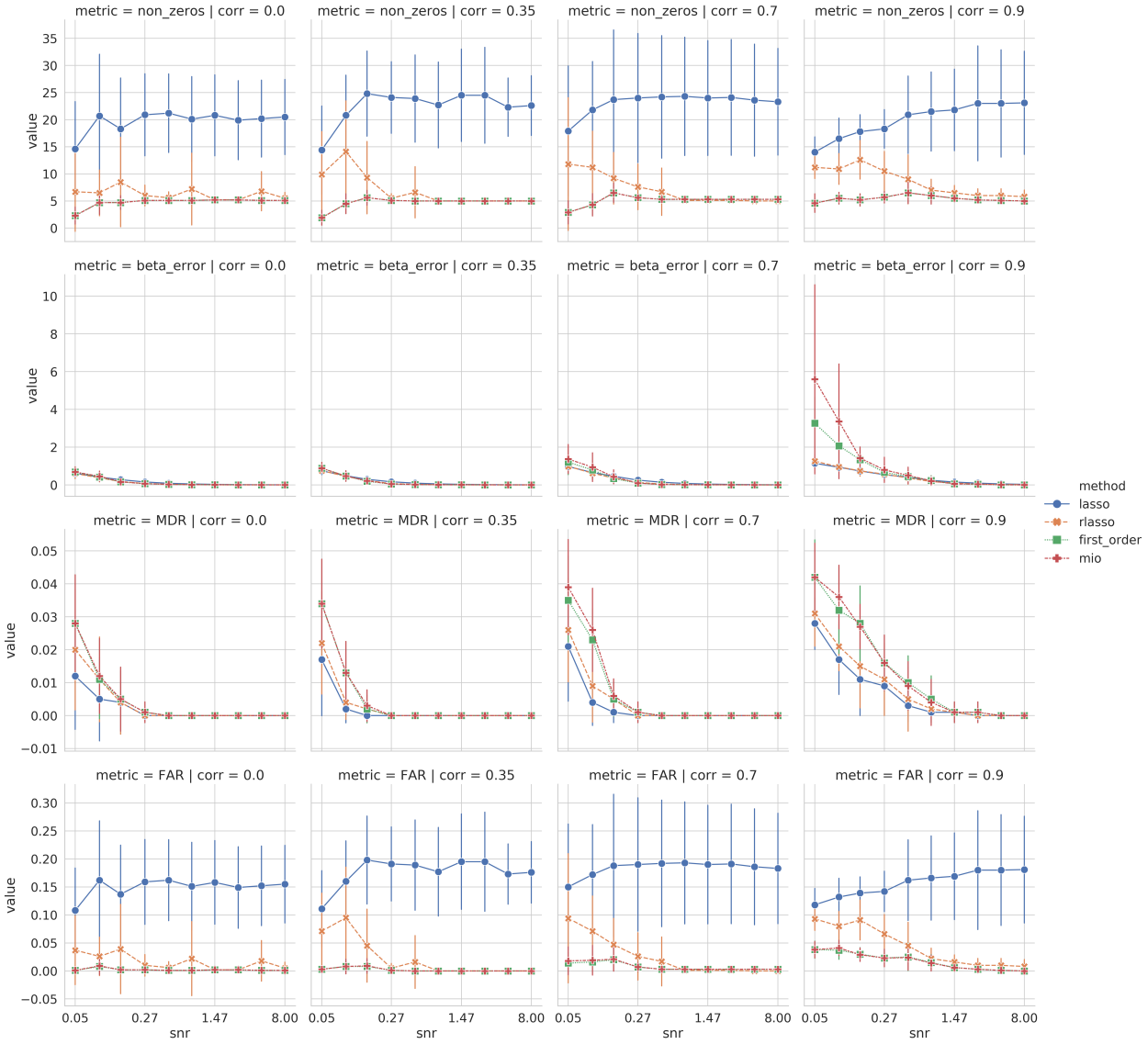

Figure 16: Support recovery evaluation metrics for experiments in the mid dimensional setting β type 2. FAR: false alarm rate, MDR: missed detection rate, RBE: relative error of parameter vector

A.2.4 Mid dimensional setting - Type 3 Sparsity - Statistical performance metrics

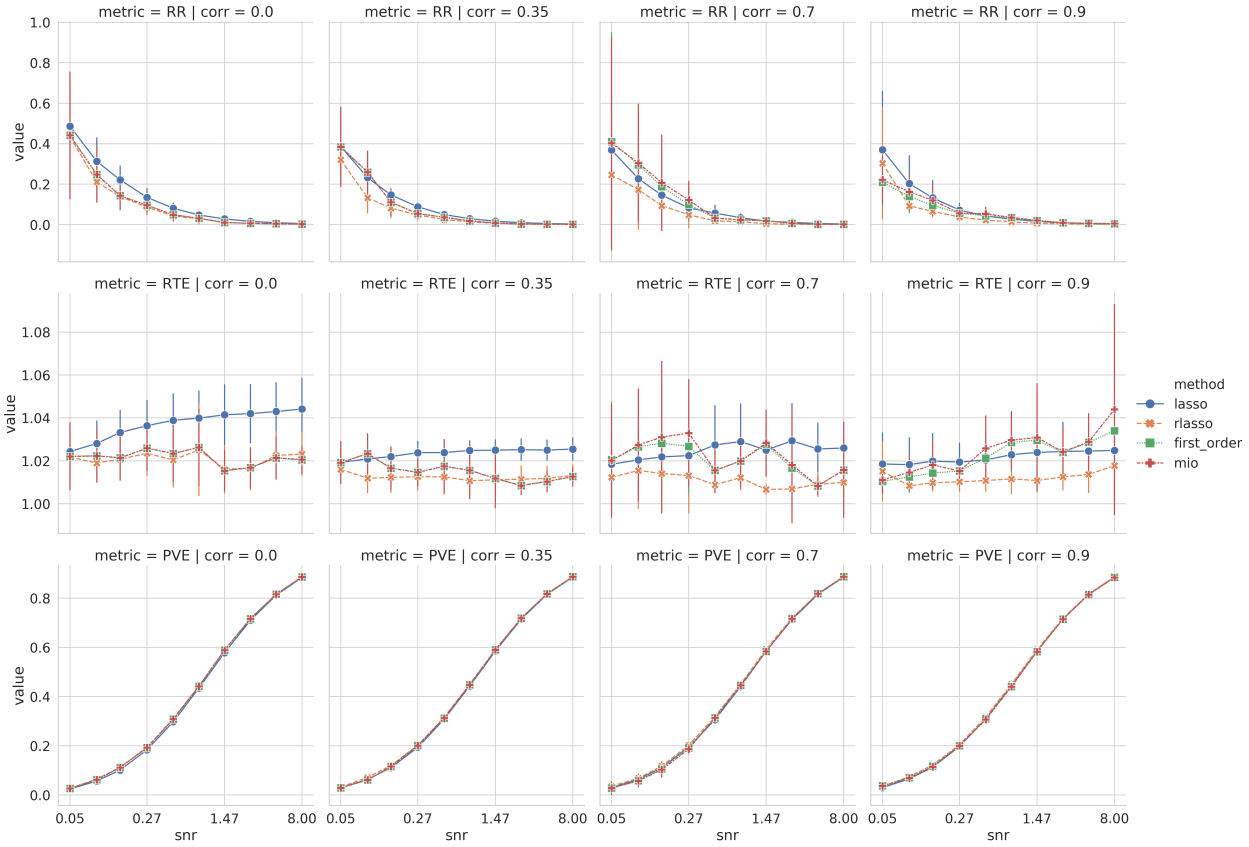

Figure 17: Statistical evaluation metrics for experiments in the mid dimensional setting β type 3. RR: relative risk, RTE: relative test error, PVE: proportion of variance explained

A.2.5 Mid dimensional setting - Type 3 Sparsity - Support recovery metrics

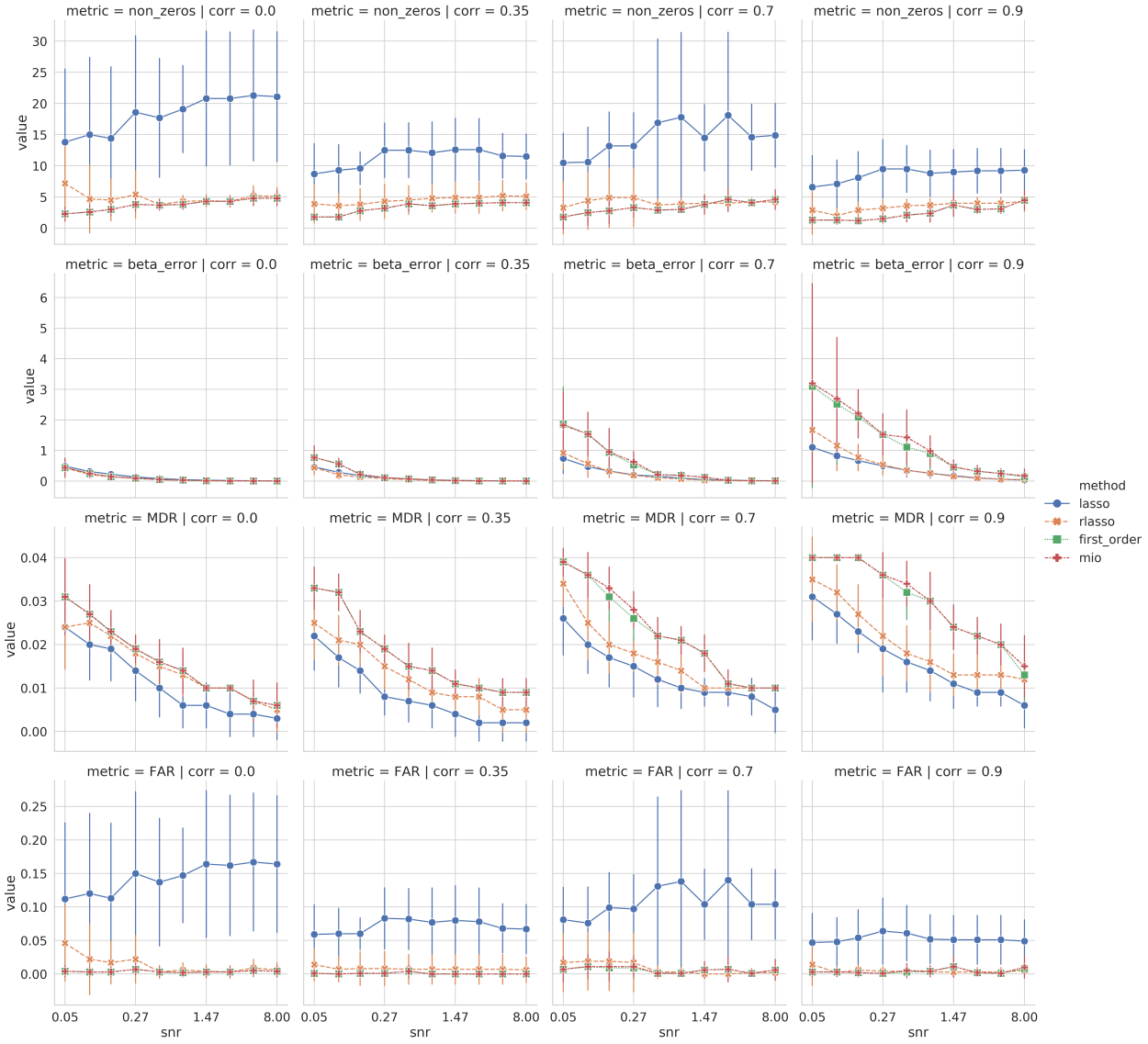

Figure 18: Support recovery evaluation metrics for experiments in the mid dimensional setting β type 3. FAR: false alarm rate, MDR: missed detection rate, RBE: relative error of parameter vector

A.2.6 Mid dimensional setting - Type 5 Sparsity - Statistical performance metrics

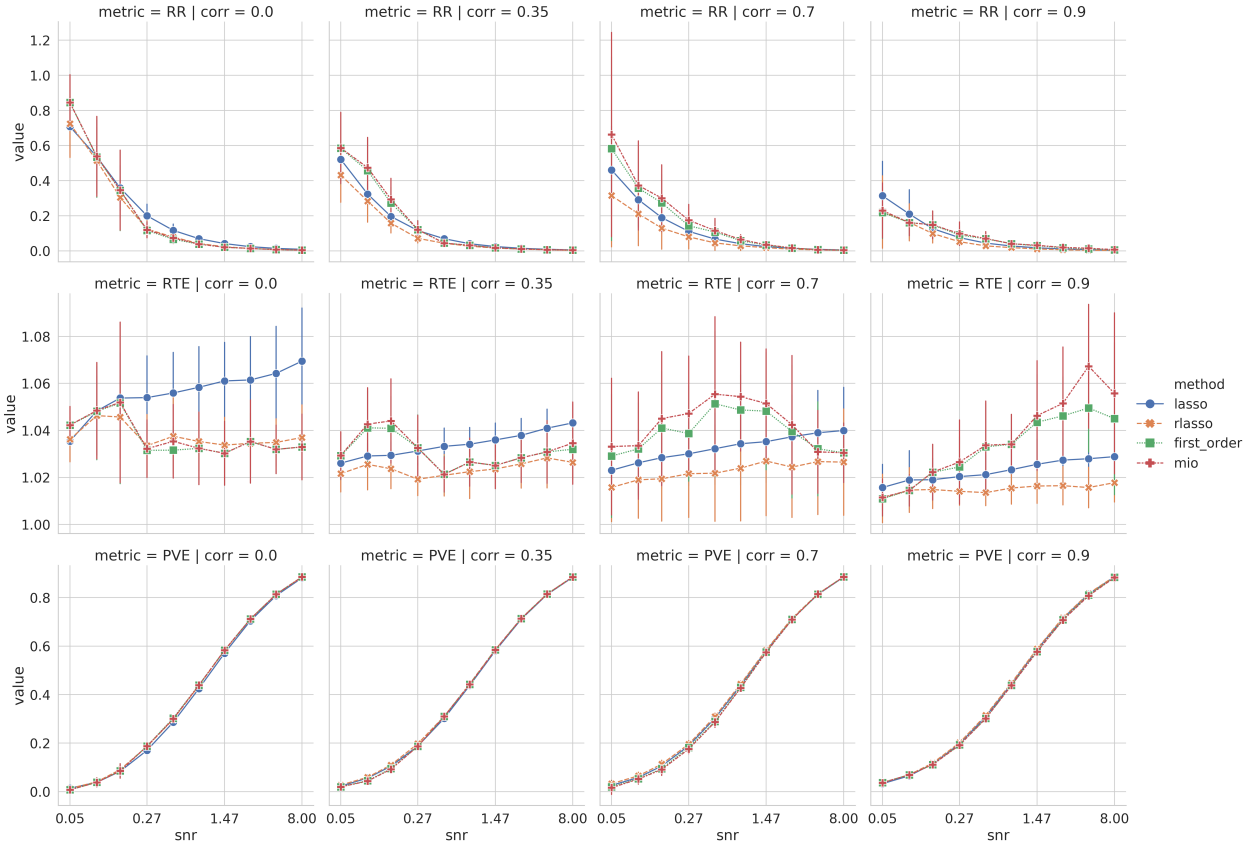

Figure 19: Statistical evaluation metrics for experiments in the mid dimensional setting β type 5. RR: relative risk, RTE: relative test error, PVE: proportion of variance explained

A.2.7 Mid dimensional setting - Type 5 Sparsity - Support recovery metrics

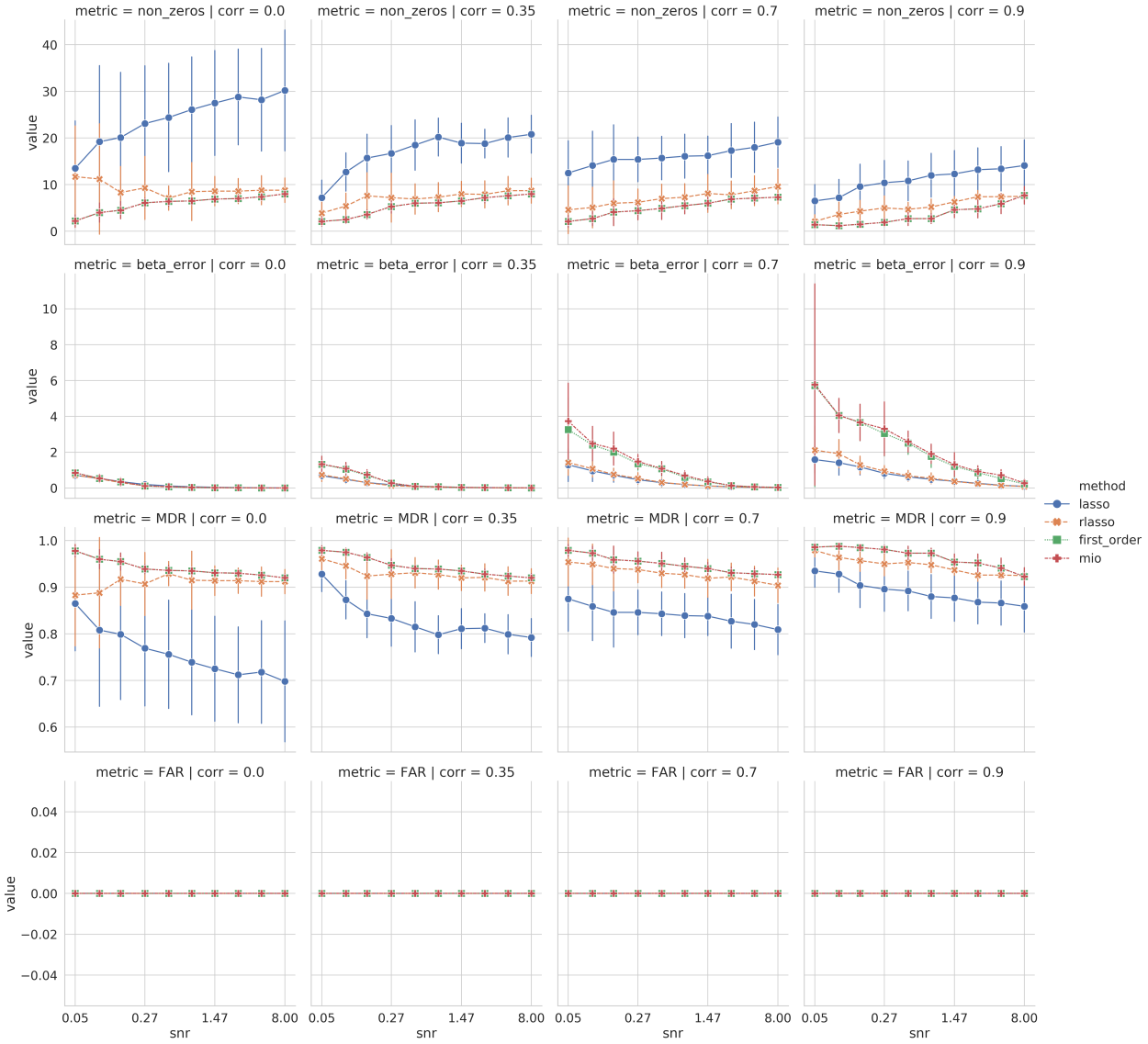

Figure 20: Support recovery evaluation metrics for experiments in the mid dimensional setting β type 5. FAR: false alarm rate, MDR: missed detection rate, RBE: relative error of parameter vector

A.3 High dimensional setting

A.3.1 High dimensional setting - Type 2 Sparsity - Statistical performance metrics

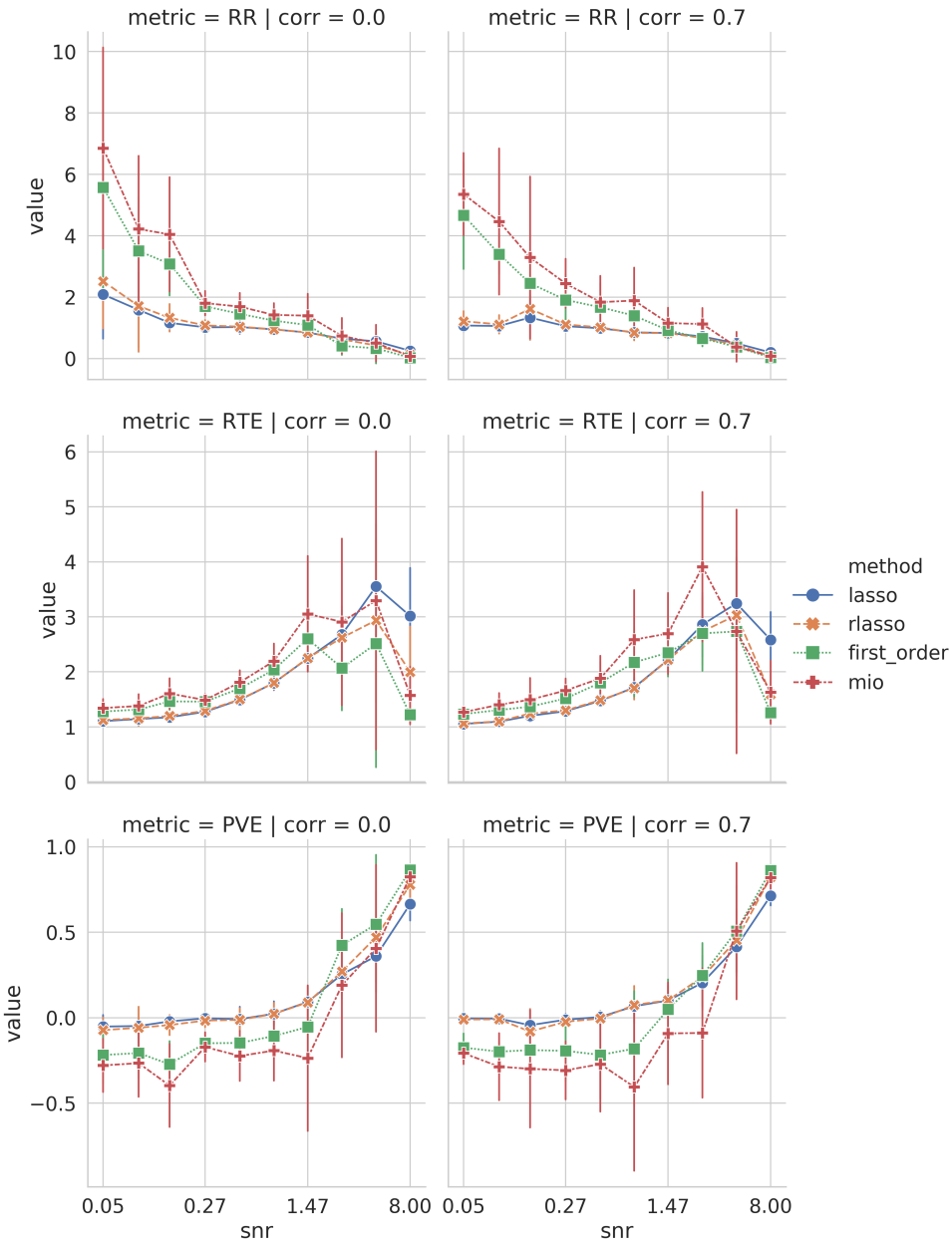

Figure 21: Statistical evaluation metrics for experiments in the high dimensional setting β type 2. RR: relative risk, RTE: relative test error, PVE: proportion of variance explained

A.3.2 High dimensional setting - Type 2 Sparsity - Support recovery metrics

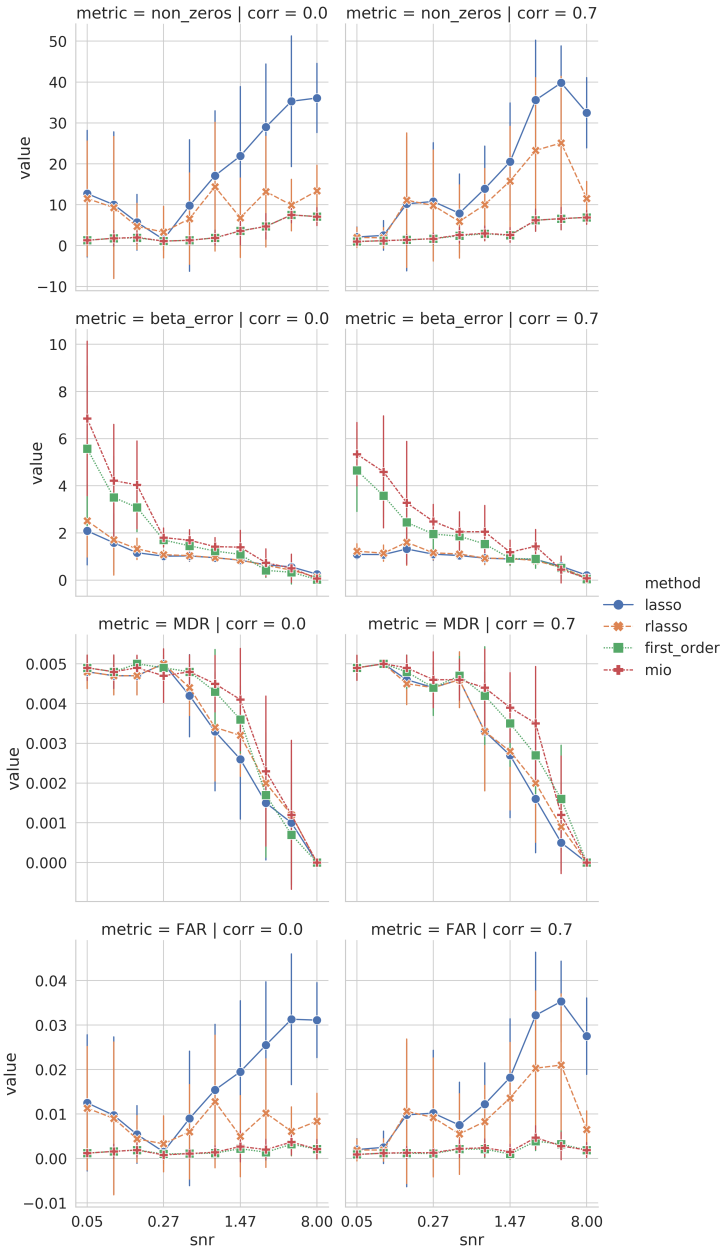

Figure 22: Support recovery evaluation metrics for experiments in the high dimensional setting β type 2. FAR: false alarm rate, MDR: missed detection rate, RBE: relative error of parameter vector