Variable Selection Methods Comparison

Hongjie Liu, Jiajun Tao, Shaohan Chen

1 Background and Objectives

1.1 Background

In linear models, when facing high-dimensional data, variable selection method is a common practice to achieve a balance between model fitness and complexity. However, in modern high-dimensional data applications, traditional variable selection methods often struggle with the presence of "weak" predictors, i.e., predictors with small but non-zero coefficients.

1.2 Objectives

This project aims to compare two popular variable selection methods, the step-wise forward method using the Akaike information criterion (AIC) and the automated LASSO regression. We conducted simulations under scenarios with high-dimensional data and larger numbers of observations to investigate how well each method performs in identifying weak and strong predictors and how missing weak predictors affects parameter estimations.

2 Statistical Methods to be Studied

2.1 Step-wise forward method

The step-wise forward method is an iterative process that starts with an empty model and sequentially adds variables that best improve the model fit, usually by adding predictors with the largest reduction in AIC. For linear models,

$$AIC = n \log \left(\sum_{i=1}^{n} (y_i - \widehat{y}_i)^2 / n \right) + 2p,$$

where $\widehat{y}_i$ is the fitted values from a model.

2.2 Automated LASSO regression

Automated LASSO regression estimates the model parameters by optimizing a penalized loss function:

$$\min_{\boldsymbol{\beta}} \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_i x_{ij} \right)^2 + \lambda \sum_{k=1}^{p} |\beta_k|,$$

where $\lambda \geq 0$ is a tuning parameter. To select the optimal value of λ, we apply the 1SE (one-standard-error) rule, which involves computing the prediction error for each value of λ and choosing the simplest model (i.e., the one with the fewest nonzero coefficients) whose prediction error is within one standard error of the minimum.

3 Investigation Settings and Scenarios

3.1 Definitions of Signal Types

This project aims to simulate data with a combination of strong, weak-but-correlated, weak-and-independent, and null predictors. The definitions of these predictor types are as follows.

Strong signals:

$$S_{strong} = \{j : |\beta_j| > c\sqrt{\log(p)/n} \text{ for some } c > 0,\ 1 \le j \le p\}$$

Weak-but-correlated (WBC) signals:

$$S_{WBC} = \{j : 0 < |\beta_j| \le c\sqrt{\log(p)/n} \text{ and } \text{corr}(X_j, X_j') \ne 0 \text{ for some } j' \in S_1,\ 1 \le j \le p\}$$

Weak-and-independent (WAI) signals:

$$S_{WAI} = \{j : 0 < |\beta_j| \le c\sqrt{\log(p)/n} \text{ and } \text{corr}(X_j, X_j') = 0 \text{ for all } j' \in S_1,\ 1 \le j \le p\}$$

Null signals:

$$S_{null} = \{j : \beta_j = 0,\ 1 \le j \le p\}$$

Thus, all p signals can be partitioned as follows:

$$\{1, \cdots, p\} = S_{strong} \cup S_{WBC} \cup S_{WAI} \cup S_{null}.$$

Let p_S, p_{WBC}, and p_{WAI} denote the number of strong, WBC, and WAI predictors, respectively. The number of true predictors $p_S + p_{WBC} + p_{WAI}$ should not exceed the sample size n.

3.2 Fixed Settings

In this project, we fix the number of the total predictors as 100, and set the ratio of true and null predictors as 2:3, which means there will be 40 true predictors and 60 null predictors. Meanwhile, we set the correlation between strong and weak-but-correlated signals as 0.4.

3.3 Scenarios (Unfixed Settings)

In the following experiments, we modify some unfixed settings to see different model performances. We will change the number of observations from $n = 100$, to $n = 500$, $n = 2000$. When $n = 100$, the number of observations is the same as the number of predictors, which indicate it to be a high dimensional scenario. In addition, we will also try different ratios of strong, weak-but-correlated, and weak-and-independent signals as 1:4:5 and 3:3:4, in order to see if the ratio of different signals, especially the strong signals, will influence the model performances.

4 Methods for Data Generation

4.1 Data Generation - Response Vector

To generate the response vector $\mathbf{y}$, it is assumed that residuals are independent, normally distributed and have equal variances, so that

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}),$$

where $\mathbf{X}$ is the design matrix and $\boldsymbol{\beta}$ is the vector of coefficients.

For each $j \in S_{strong}$, the value of β_j is set to 20, while for each $j \in S_{WBC} \cup S_{WAI}$, the value of β_j is set to 0.5. We choose $c = 20$ such that $0.5 \le c\sqrt{\log p/n} < 20$ for all scenarios to be investigated. The value of σ for the error term is set to 8.

4.2 Data Generation - Design Matrix

For each observations $\mathbf{x}_i$ in the design matrix $\mathbf{X}$, it is assumed that

$$\mathbf{x}_i \overset{i.i.d.}{\sim} N(\boldsymbol{\mu}, \boldsymbol{\Sigma}),\ i = 1, \cdots, n,$$

where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix.

To ensure fairness in the penalty of LASSO, it is assumed that all predictors are standardized, resulting in $\boldsymbol{\mu} = \mathbf{0}$ and $\Sigma_{ii} = 1$ for all $i \in \{1, \cdots, n\}$, where Σ_{ii} represents the ith diagonal element of the covariance matrix.

When all predictors are standardized, the covariance matrix $\boldsymbol{\Sigma}$ becomes a correlation matrix, allowing for the creation of correlations between WBC predictors and strong predictors. We set $p_{WBC} \geq p_S$. For each strong predictor (excluding one), $\left[\frac{p_{WBC}}{p_S}\right]$ WBC predictors are set to be correlated with it. Each WBC predictor is set to be correlated with only one strong predictor, with all other elements of $\boldsymbol{\Sigma}$ set to 0.

To generate data following a multivariate normal distribution, the **R**-function `mvrnorm` from the `MASS` package is used, with $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ passed as arguments.

5 Model Evaluation

5.1 Evaluation Metrics

In this project, we define the true predictors as positive and null predictors as negative.

For signal identification, we use the following five metrics to compare the two models:

- Complexity: number of selected predictors in the model
- Sensitivity: $\dfrac{TP}{TP + FN}$
- Specificity: $\dfrac{TN}{TN + FP}$
- F1-score: $\dfrac{2 \cdot sensitivity \cdot precision}{sensitivity + precision}$
- Accuracy: $\dfrac{TP + TN}{TP + TN + FP + FN}$

For parameter estimation, we use the following to metrics to compare the two models:

- RMSE: $\sqrt{\dfrac{1}{p}\sum_{i=1}^{p}(\hat{\beta}_i - \beta_i)^2}$
- Variance: $\dfrac{1}{p}\sum_{i=1}^{p}(\hat{\beta}_i - \bar{\beta}_i)^2$

5.2 Signal Identification Performance

Complexity of the models is indicated by the number of selected predictors. We can see that in high dimensional scenario ($n = 100$), forward selection model tends to select lots of predictors and Lasso tends to select very few. One interesting thing is that, if we increase the ratio of strong predictors (i.e., more strong predictors), Lasso also tends to select more predictors too. When it comes to normal scenario, forward selection still tends to select more predictors than Lasso, but the discrepancy is smaller than high dimensional case, and will be further narrowed down with n or the ratio of strong predictors increasing. And as n increases, the number of selected parameters of both models are closer to the true number 40.

As for overall classification performance, if in high dimensional scenario, forward selection tends to be very assertive and much better at identifying weak signals, leading to an extremely high sensitivity but low specificity. Lasso in turn tends to be very conservative and much better at identifying null signals, leading to an extremely high specificity but low sensitivity. Like high dimensional case, Lasso becomes more sensitive and not that assertive when the ratio of strong predictors increases. Based on the above plot, we can conclude that both models do not perform too well based on F1-score and accuracy, because they are very radical and tend to identify most of the predictors either as positive or negative, but far away from the truth.

For normal scenario, both models become less radical under normal scenarios, but forward selection is still more sensitive and assertive than Lasso, while Lasso has higher specificity and is more conservative. Both models perform better on those metrics with n increasing. Overall, Lasso and forward selection has similar F1-score and accuracy performance. But when there are more strong predictors, Lasso performs obviously better than forward selection.

About the classification performance of different signals. Under all n values, both models perfectly identify the strong signals. In high dimensional scenario, forward selection performs much better on identifying weak predictors while Lasso performs much better on identifying null predictors, that's why we see the high sensitivity of forward selection and high specificity of Lasso in the previous section. When there are more strong predictors, Lasso also performs better on selecting weak predictors. When it comes to normal scenario, forward selection is still better at selecting null and Lasso is better at selecting weak predictors. But the discrepancy is smaller compared with high-dimensional data, and will continue to be smaller as n increases. When there are more strong predictors, Lasso performs much better at selecting weak-but-correlated signals.

5.3 Parameter Estimation Performance

In high dimensional scenario, Lasso performs much better than forward selection, with obviously much lower and centered RMSE and also lower variance. When it comes to normal scenario, when $n = 500$, Lasso outperforms forward selection, but with n increasing, forward selection starts to outperform Lasso model on RMSE and variance. And overall, Lasso tends to perform better when there are more strong signals. If there are more strong predictors, the variance is also larger.

5.4 Effect of Missing Weak Predictors

Here we define missing weak predictors as the true weak predictors but we estimated them as null predictors, and we used RMSE to evaluate. In each scenario, we picked up three kinds of missing situations: most missing, least missing, and middle missing. As the name suggests, the most missing are the ones that have the least non-null estimations. Foe each kind of situation, we picked 10% to draw the plot. For example, we fixed 100 parameters and 100 simulation times. In each simulation, we can get the number of non-null parameters. After arranging them, we can pick the top 10 simulation times that have the most non-null parameters. In the same way, we can pick the last 10 and the middle 10 which ranks 46 to 55. It is worth mentioning that, Lasso only picks which parameters to use and the coefficients of Lasso can not be used directly. In order to compare RMSE, we need to refit the linear regression model using the parameters that Lasso picks.

When in high dimensional scenario ($n = 100$), Lasso has a very small RMSE however forward selection's RMSE is big. But when the missing amount increases, the RMSE of forward selection drops dramatically. When the number of missing parameters increases, the RMSEs of both methods decrease.

When under normal scenario ($n = 500$ or 2000), the RMSEs of both methods are small. It's hard to tell which method is better since their differences are small. There seems to be no apparent patterns between different ratios as well.

Since we care more about high dimensional scenario, the conclusion should be Lasso performs better than forward selection when in high dimensional scenario according to RMSE, and the more missing parameters, the better the RMSE.

6 Discussions

6.1 Limitations

There is much freedom when designing the simulations. In our algorithm, we have 5 parameters: number of observations, number of parameters, the ratio of strong and weak signals, the definition of strong and weak signals and the correlation between strong and weak signals. However, even more parameters can be adjusted such as the correlation within WAI signals, or between null and strong signals, etc. We generated many versions of data and found that many things can affect the result. Here we only fixed p and c and the results and conclusions may not be comprehensive.

6.2 Future Work

For the future work, we could adjust other parameters to investigate this problem further. What's more, we reproduced the high-dimensional scenario and faced the struggle of choosing covariates. We still could not have a clear solution to deal with this difficulty. It would be hard to tackle the problem, but it can be a direction of effort.

Contributions

PLEASE INSERT CONTRIBUTIONS HERE.

Shaohan Chen mainly focused himself on task 1. He conducted model evaluation based on the simulation code and simulated data generated by Hongjie Liu, and gave visualization analysis on the signals identification performance as well as the parameter estimation performance of both models. He was also responsible for giving presentation of task 1 and writing the corresponding parts of the slides and report.

Reference

1. Li Y, Hong HG, Ahmed SE, Li Y. Weak signals in high-dimensional regression: Detection, estimation and prediction. Appl Stochastic Models Bus Ind. 2018;1–16. https://doi.org/10.1002/asmb.2340

Appendix

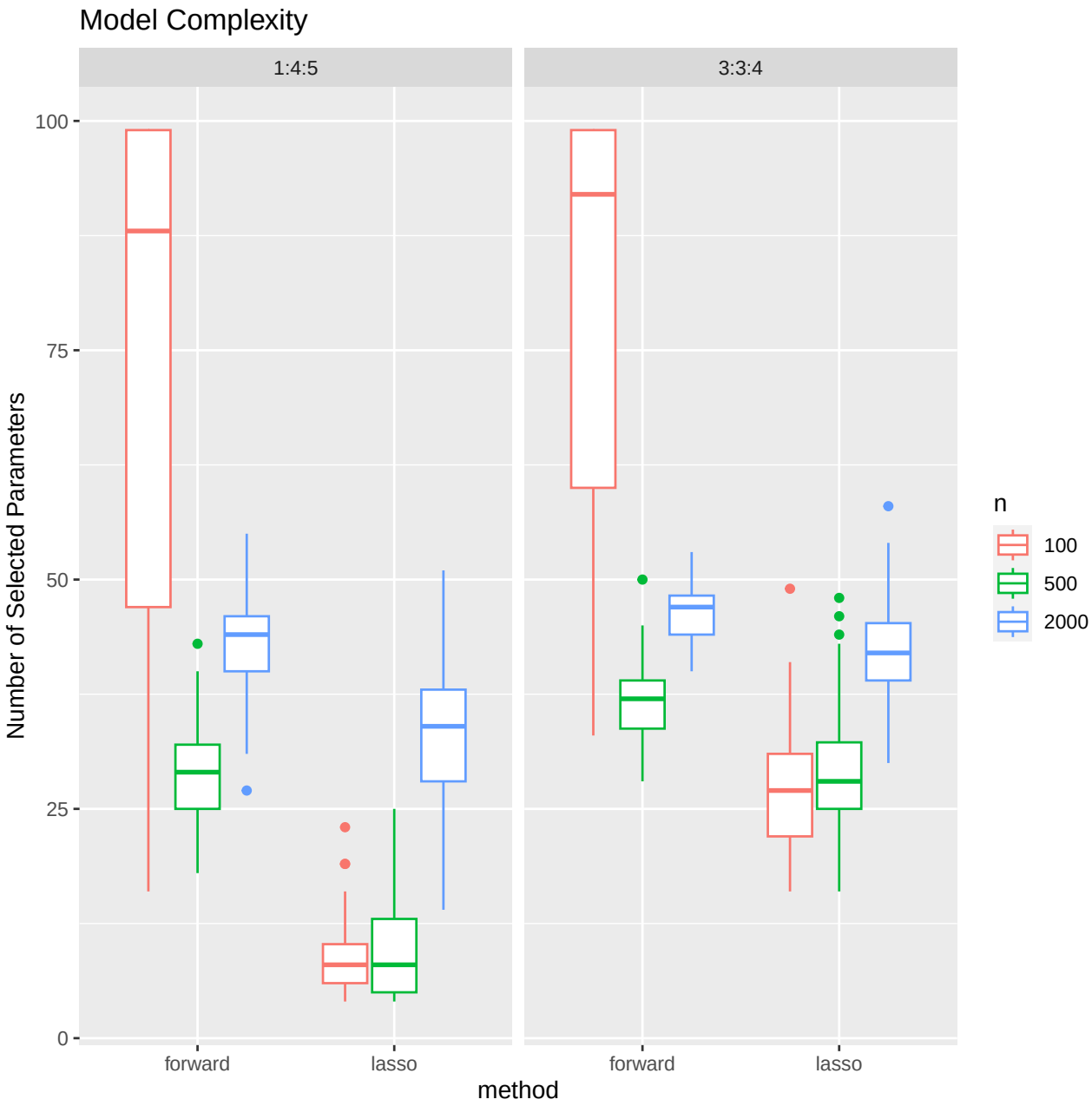

Figure 1: Model Complexity

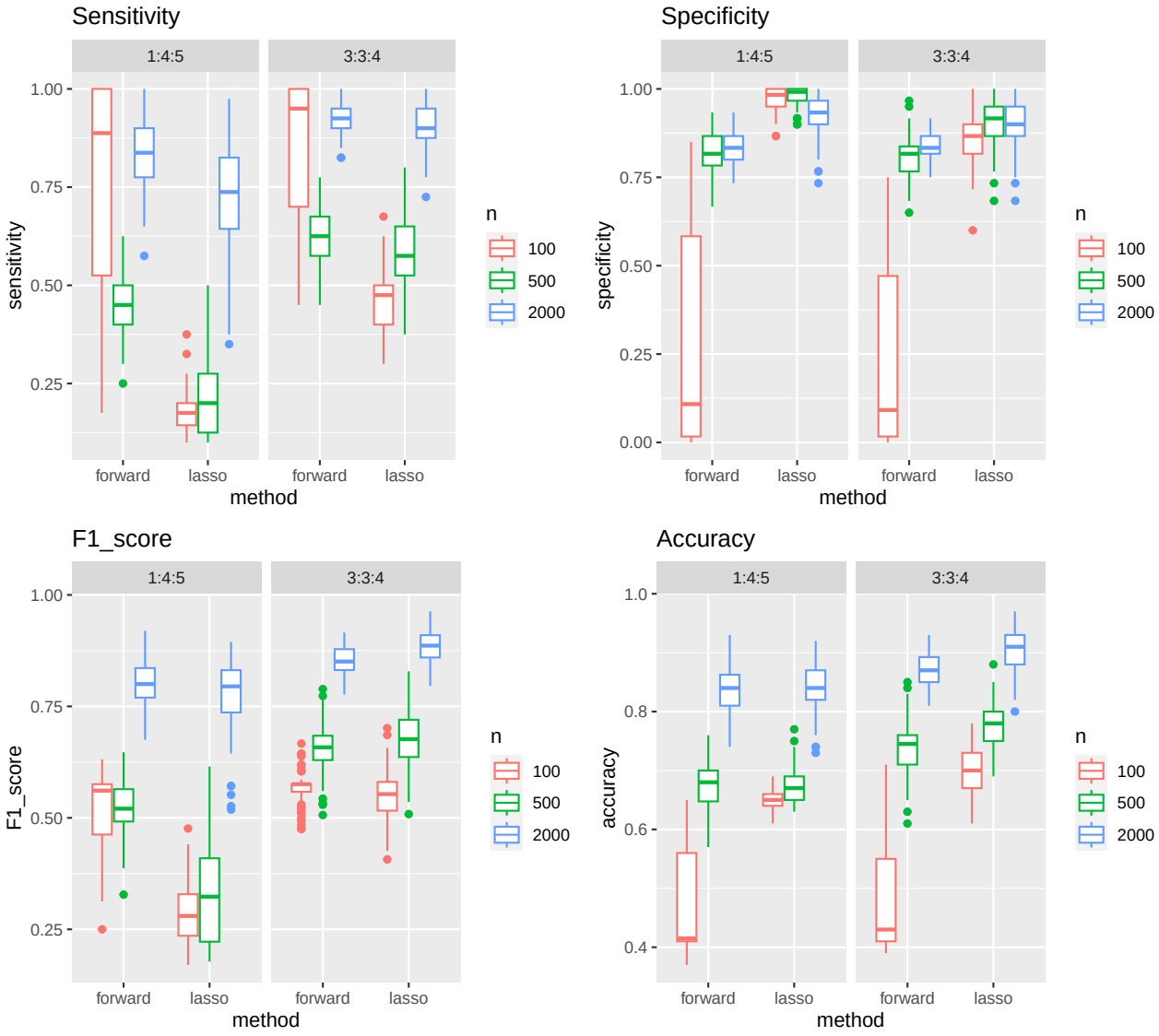

Figure 2: Overall Classification Performance

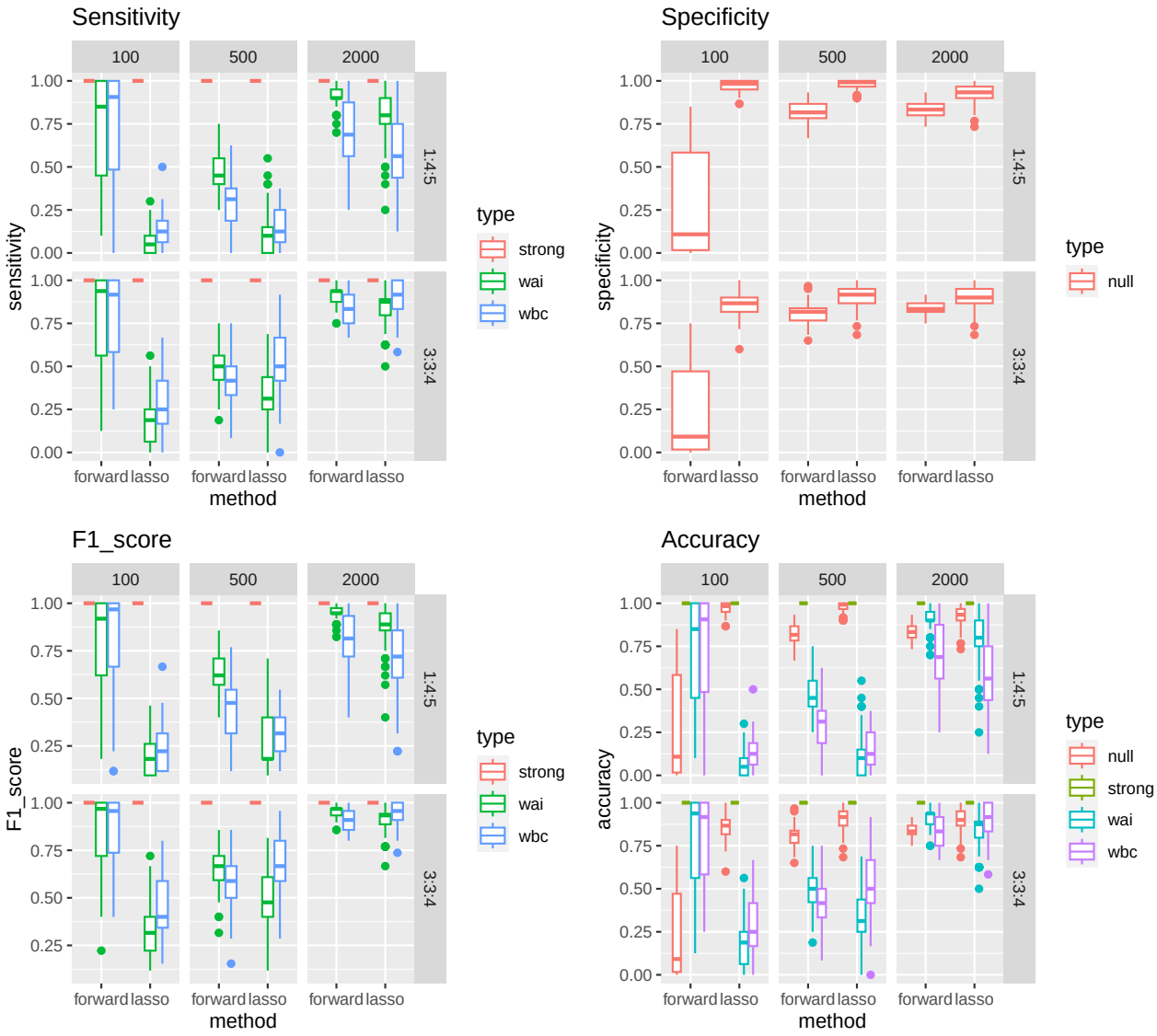

Figure 3: Classification Performance by Signals

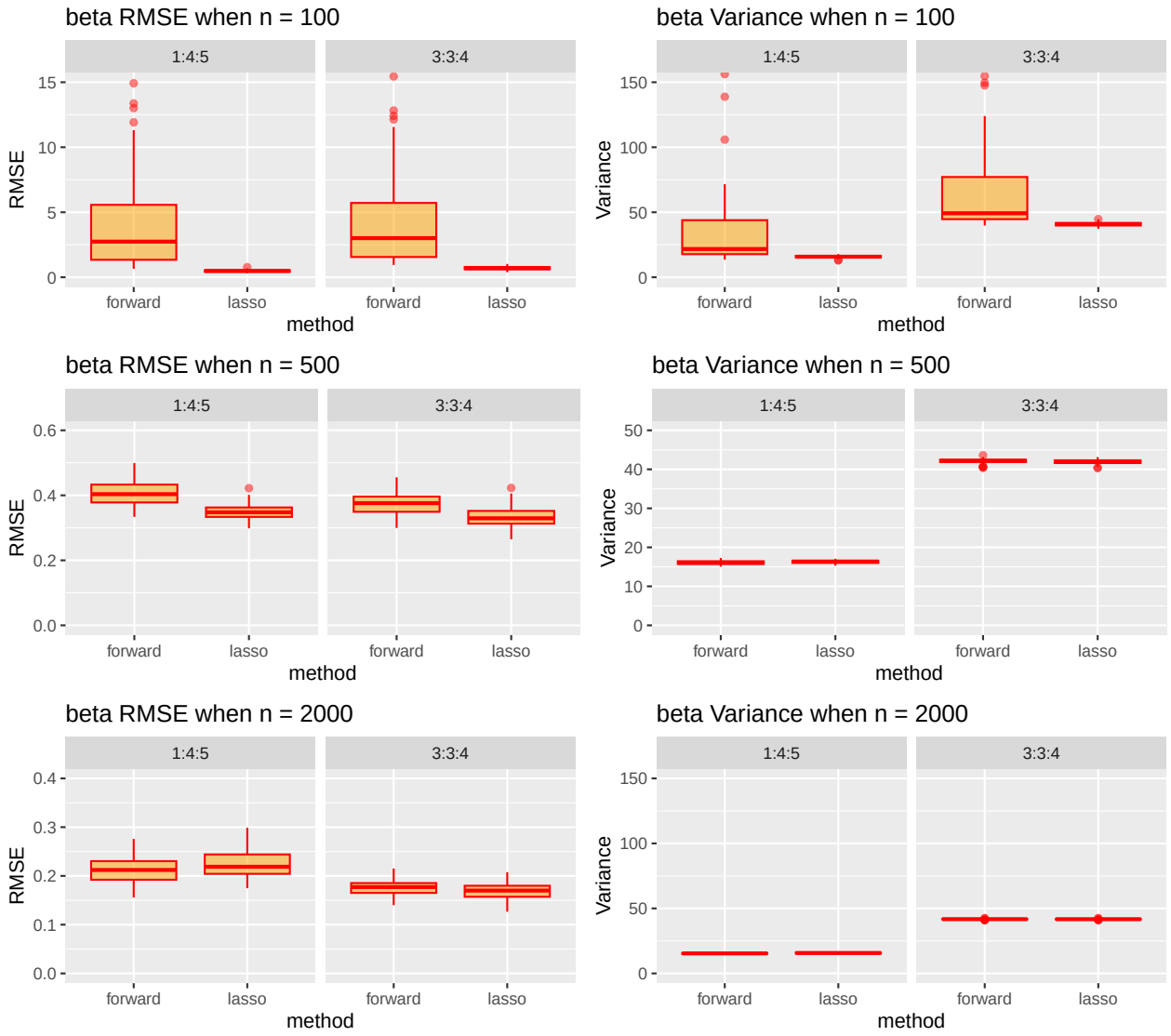

Figure 4: Parameter Estimation Performance

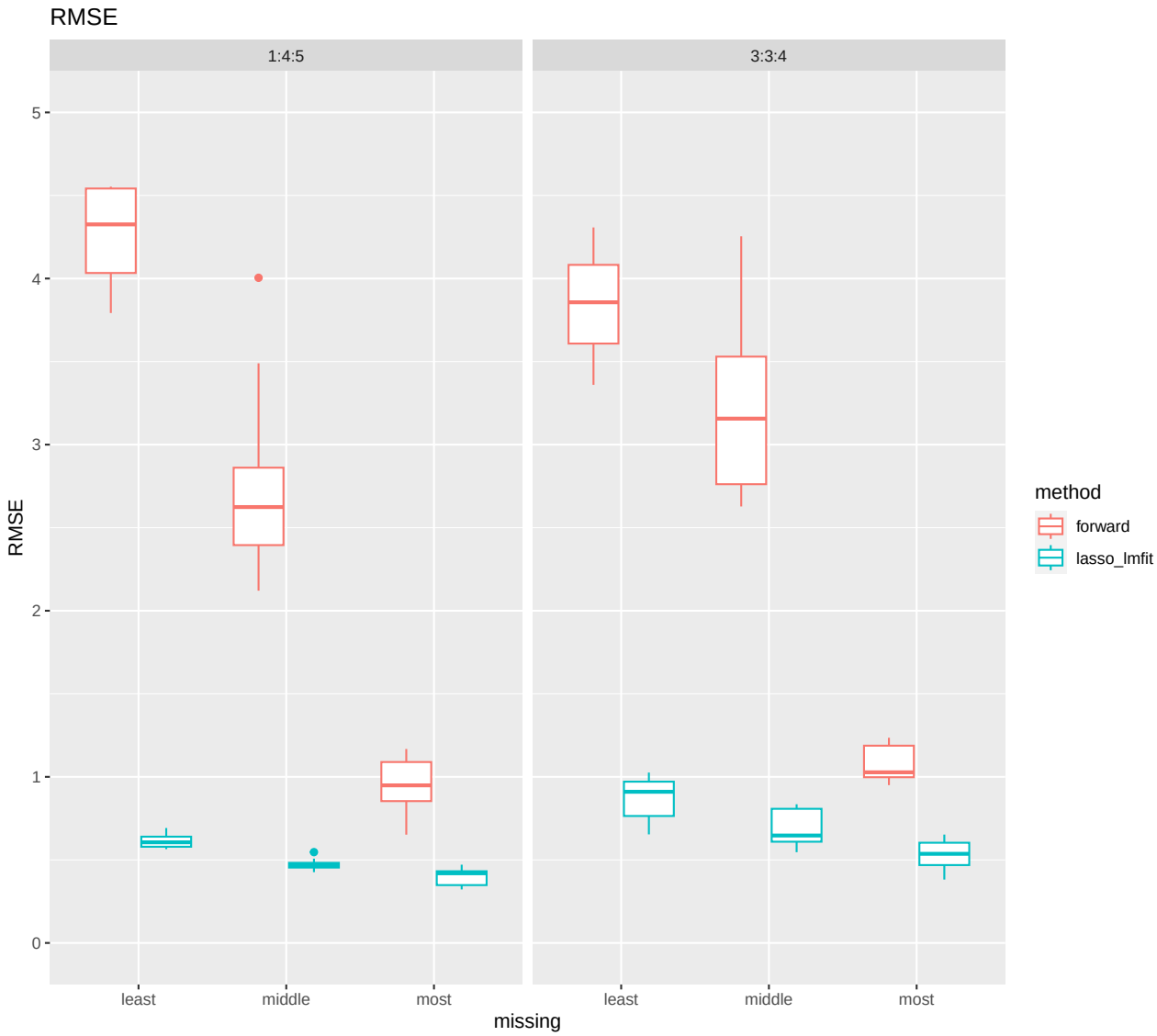

Figure 5: RMSE comparison when n = 100

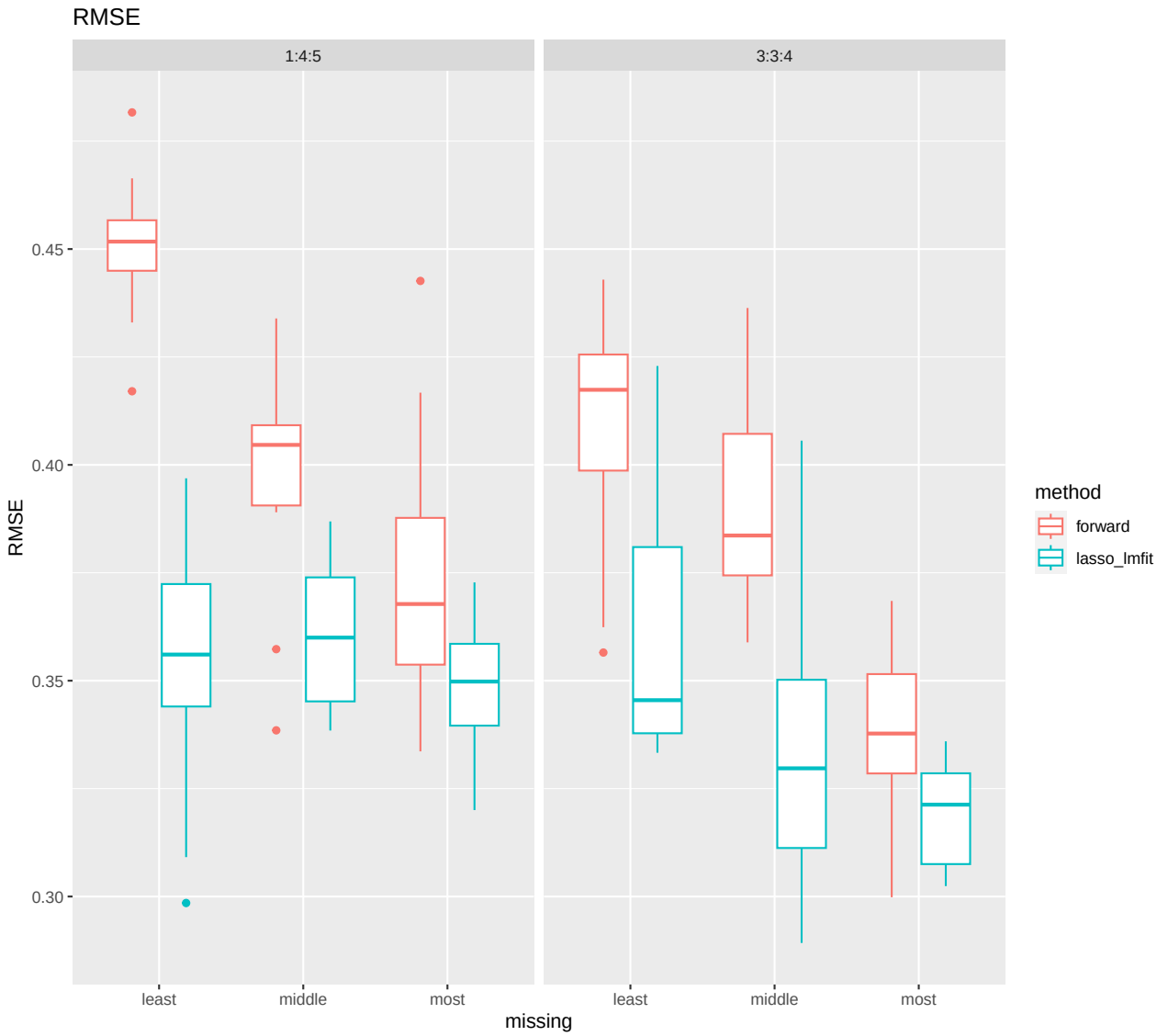

Figure 6: RMSE comparison when n = 500

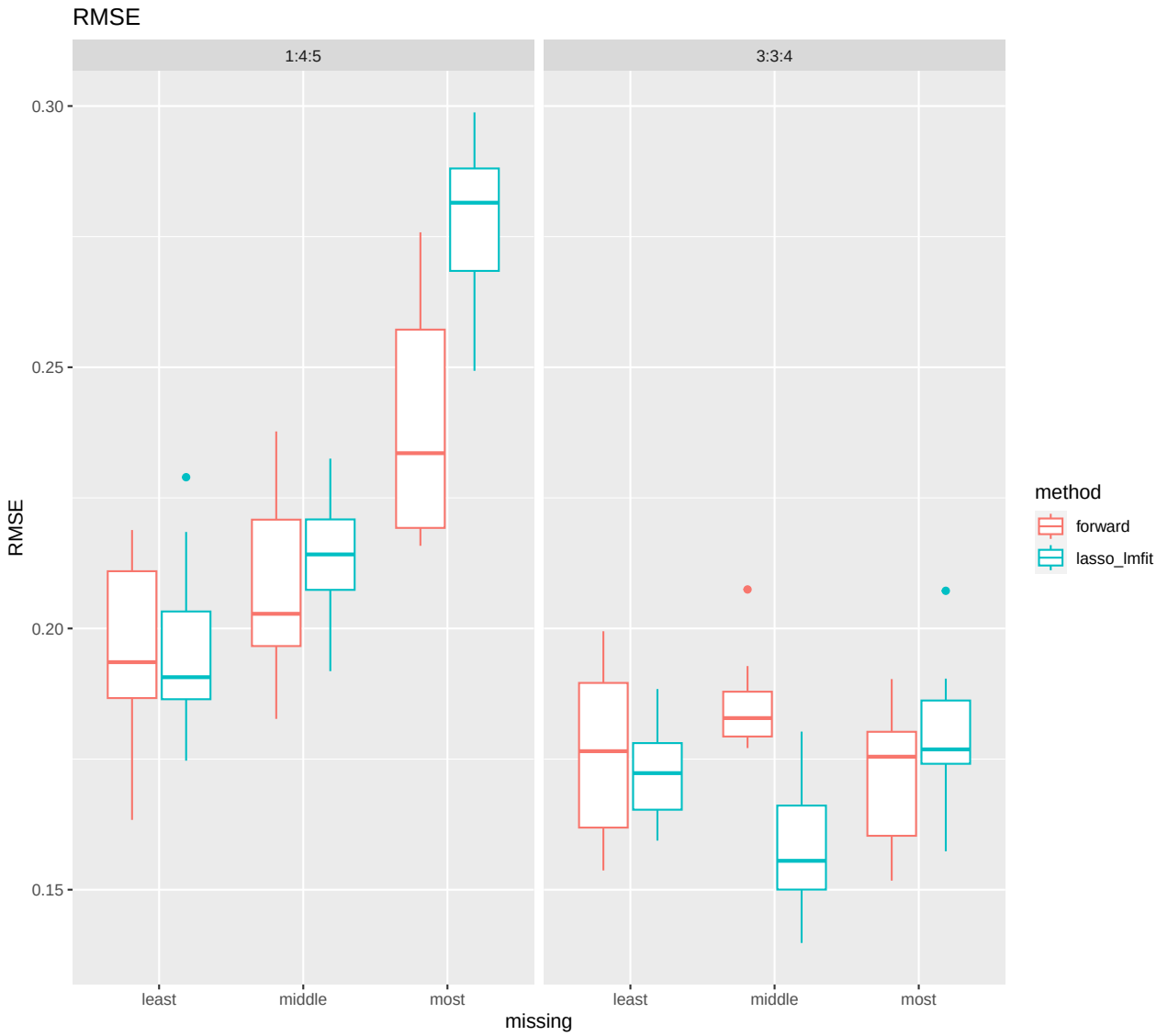

Figure 7: RMSE comparison when n = 2000