Explaining the Practical Success of Random Forests

Siyu Zhou

Department of Statistics, University of Pittsburgh

June 1, 2020

Introduction

The Random Forest (RF) procedure is a supervised learning tool introduced by Breiman [2001].

- Data of the form $\mathcal{D}_n = \{\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_n\}$ where $\boldsymbol{Z}_i = (\boldsymbol{X}_i, Y_i)$, $\boldsymbol{X}_i = \{X_{i,1}, \ldots, X_{i,p}\} \in \mathbb{R}^p$ is a vector of p features, $Y_i \in \mathbb{R}$ is the response
- True relationship given by $Y_i = f(\boldsymbol{X}_i) + \epsilon_i$.
- Given B resamples and a point $\boldsymbol{x}$, the RF prediction is given by

$$\hat{y} = \text{RF}(\boldsymbol{x}; \mathcal{D}_n, \Theta) = \frac{1}{B} \sum_{b=1}^{B} T(\boldsymbol{x}; \mathcal{D}_n, \Theta_b)$$

Introduction

Many variants of RFs have been developed in the last two decades. The original one by Breiman [2001] is characterized by

- Resamples are obtained via **bootstrapping** [Efron, 1982].
- Base-learners are CART-style trees [Breiman et al., 1984].
- $\Theta_b = (\Theta_{\mathcal{D}_n,b}, \Theta_{\texttt{mtry},b})$ where $\Theta_{\mathcal{D}_n,b}$ represents the randomness in resampling and $\Theta_{\texttt{mtry},b}$ serves to randomly select $\texttt{mtry} < p$ features as candidates for splits at each internal node of each tree.
- For classification problems, predicted labels are given by the majority vote.

This RF will be the one used in the following studies.

Record of Success

Random forests have remained among the most popular and off-the-shelf supervised learning methods since its inception.

- Bioinformatic: Díaz-Uriarte and De Andres [2006], Mehrmohamadi et al. [2016]
- Drug discovery: Svetnik et al. [2003]
- Ecology: Prasad et al. [2006], Cutler et al. [2007]
- 3D object recognition: Bernard et al. [2007], Huang et al. [2010], Guo et al. [2011], Fanelli et al. [2013]

In a recent large-scale empirical study [Fernández-Delgado et al., 2014], RFs were found to be the top classifiers against hundreds of alternatives compared on 121 datasets (the whole UCI database).

Theoretical Studies

The empirical success of RFs naturally led to lots of research investigating various properties and extensions:

- Consistency properties [Biau et al., 2008, Biau, 2012, Scornet et al., 2015, Scornet, 2016, Klusowski, 2019]
- Asymptotic normality [Mentch and Hooker, 2016, Wager and Athey, 2018]
- Stronger CLT [Peng et al., 2019]
- Testing procedures [Mentch and Hooker, 2016, 2017, Coleman et al., 2019]

Theoretical Studies

- Ensemble size vs stabilization [Lopes et al., 2019a,b]
- Estimating standard errors [Sexton and Laake, 2009, Wager et al., 2014]
- Variable importance and related issues [Breiman, 2001, Strobl et al., 2007, 2008, Toloşi and Lengauer, 2011, Nicodemus et al., 2010, Hooker and Mentch, 2019]
- Extensions to ...
  - Quantile regression [Meinshausen, 2006]
  - Reinforcement learning [Zhu et al., 2015]
  - Survival analysis [Hothorn et al., 2005, Ishwaran et al., 2008, Cui et al., 2017, Steingrimsson et al., 2019]

Lack of Explanation

Despite all of this progress there has been shockingly little work on actually explaining the underlying mechanisms at work in RFs that might explain their success.

- Main takeaways of studies experimenting tuning the RF procedure are high-level and heuristic:
  - Including more trees in the forest helps stabilize predictions.
  - Tuning the procedure can improve performance.

- "**present results are insufficient to explain in full generality the remarkable behavior of random forests**." [Biau and Scornet, 2016]

1 Introduction

2 Randomization as Regularization

- Existing Explanations
- Relative Performance of RFs
- Degrees of Freedom for RFs
- Randomized Forward Selection

Existing Explanations

1. Breiman [2001]: The additional randomness in RFs serves to de-correlate trees, thereby further reducing the variance of the ensemble

- Randomness trades off accuracy at the tree level for reduced correlation (akin to bias-variance tradeoff)
- Updated discussion given in Hastie et al. [2009]
- **More an motivation for why RFs can potentially work than an explanation for why they do work**

Existing Explanations

2. Biau and Scornet [2016]: "*The authors intuition is that tree aggregation models are able to estimate patterns that are more complex than classical onespatterns that cannot be simply characterized by standard sparsity or smoothness conditions.*"

- Informal, but perhaps the most popular among RF researchers
- Somewhat difficult to formalize and motivate when RFs might be expected to perform well

Existing Explanations

3. Wyner et al. [2017] Random forests (and AdaBoost [Freund et al., 1996]) work well "not in spite, but because of interpolation"

- The claim is that *because* RFs are interpolators, they naturally isolate "noisy" observations without affecting the perfect (or near perfect) fits nearby

Existing Explanations

But for any fixed B, as $n \to \infty$, the probability of interpolating all observations goes to 0. This can be fixed by simply letting B grow with n (though this isn't generally how RFs are constructed).

So the RF will interpolate (at least with high probability) if ...

- We're doing classification, and
- trees are fully grown, and
- we use bootstrapping (or at least subsamples $> 0.5n$), and
- we let $B \to \infty$,

Existing Explanations

However, RFs are also (at least as) successful in regression settings and have also been shown to work quite well with ...

- shallow trees [Duroux and Scornet, 2016]
- subsampling instead of bootstrapping [Zaman and Hirose, 2009, Mentch and Hooker, 2016, Wager and Athey, 2018]
- relatively few trees in the ensemble [Lopes et al., 2019b].

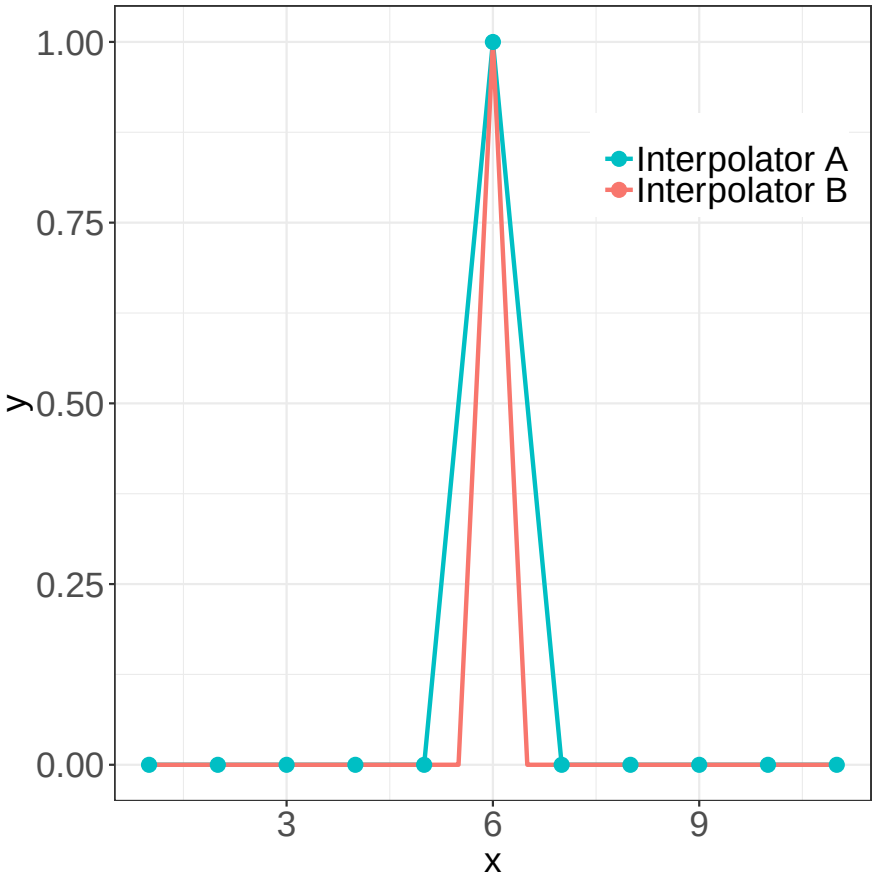
Thus, at best the interpolation theory is only potentially useful in this narrow scope of problem type.

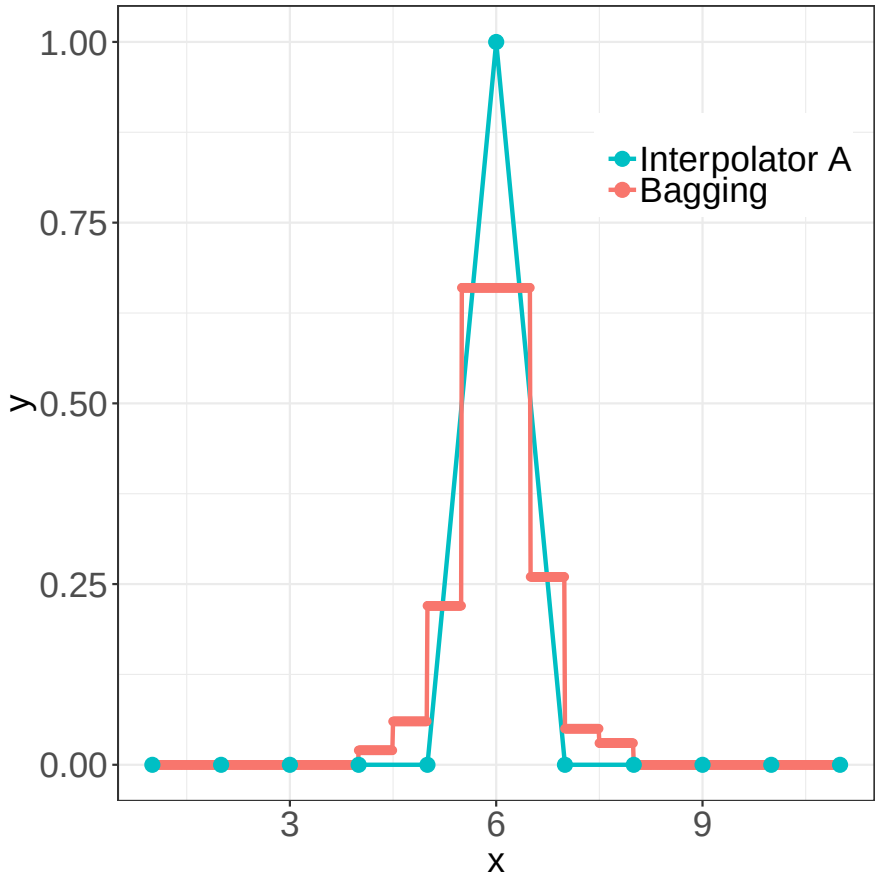
Existing Explanations

Why doesn't this explanation (or something in the same spirit) apply to regression problems?

Let's consider the same toy example we saw before and recall what Wyner et al. [2017] say that RFs are doing ...

Existing Explanations





The RF is not interpolating and in fact looks to be doing something nearly opposite – trying to "smooth out" the influence of the outlying point

Existing Explanations

Both Breiman [2001] and Wyner et al. [2017] seem to largely agree that in general, random forests substantially outperform bagging.

This is where we begin to take issue:

- Certainly not a "universal truth" and seems like a potentially naive foundation for building an explanation for RF success
- Not easy to find simulation settings where bagging performs drastically worse
- Nonetheless considered popular wisdom and simply taken as fact in most cases

Ingredients for a good explanation

In our view, a "good" explanation for RF success should ...

- Be specific enough to determine (at least roughly) when (in what settings) RFs should be expected to perform well relative to other methods
- Identify an intuitive role for the randomness (`mtry` parameter)
- Either extend to other kinds of methods (base-learners) or provide intuition into why this is a tree-based phenomenon

Ingredients for a good explanation

In our view, a "good" explanation for RF success should ...

- Be specific enough to determine (at least roughly) when (in what settings) RFs should be expected to perform well relative to other methods
- Identify an intuitive role for the randomness (`mtry` parameter)
- Either extend to other kinds of methods (base-learners) or provide intuition into why this is a tree-based phenomenon

Ingredients for a good explanation

In our view, a "good" explanation for RF success should ...

- Be specific enough to determine (at least roughly) when (in what settings) RFs should be expected to perform well relative to other methods
- Identify an intuitive role for the randomness (`mtry` parameter)
- Either extend to other kinds of methods (base-learners) or provide intuition into why this is a tree-based phenomenon

Ingredients for a good explanation

In our view, a "good" explanation for RF success should ...

- Be specific enough to determine (at least roughly) when (in what settings) RFs should be expected to perform well relative to other methods
- Identify an intuitive role for the randomness (`mtry` parameter)
- Either extend to other kinds of methods (base-learners) or provide intuition into why this is a tree-based phenomenon

Relative Performance of RFs

Relative to bagging where no additional randomness injected in the construction of trees, RFs perform *best* when the data is very noisy (low SNRs).

Let's first look at some simulated data:

1. **Linear models:** same setup as in Hastie et al. [2017]
   - Rows of $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ independently drawn from $N_p(0, \Sigma)$, where $\Sigma \in \mathbb{R}^{p \times p}$ has entry $(i, j) = \rho^{|i-j|}$ with $\rho = 0.35$.
   - Response $\boldsymbol{Y} = \boldsymbol{X}\beta + \epsilon$ where $\epsilon \sim N(0, \sigma^2 I)$ and σ^2 is calculated to satisfy corresponding SNR level ν, i.e.

$$\sigma^2 = \frac{\beta^T \Sigma \beta}{\nu}$$

   - First s components of β are equal to 1 and the rest are equal to 0 (beta-type 2 in Hastie et al. [2017])

Relative Performance of RFs

Relative to bagging where no additional randomness injected in the construction of trees, RFs perform *best* when the data is very noisy (low SNRs).

Let's first look at some simulated data:

1. **Linear models:** same setup as in Hastie et al. [2017]
   - Rows of $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ independently drawn from $N_p(0, \Sigma)$, where $\Sigma \in \mathbb{R}^{p \times p}$ has entry $(i, j) = \rho^{|i-j|}$ with $\rho = 0.35$.
   - Response $\boldsymbol{Y} = \boldsymbol{X}\beta + \epsilon$ where $\epsilon \sim N(0, \sigma^2 I)$ and σ^2 is calculated to satisfy corresponding SNR level ν, i.e.

$$\sigma^2 = \frac{\beta^T \Sigma \beta}{\nu}$$

   - First s components of β are equal to 1 and the rest are equal to 0 (beta-type 2 in Hastie et al. [2017])

Relative Performance of RFs

2. **MARS:** models studied in Friedman [1991] and recent RF papers where

$$\boldsymbol{Y} = 10\sin(\pi X_1 X_2) + 20(X_3 - 0.05)^2 + 10X_4 + 5X_5 + \boldsymbol{\epsilon}$$

- Features drawn independently from Unif(0,1); errors drawn in same fashion with σ^2 chosen to produce a particular SNR.

Relative Performance of RFs

- For the linear model, we take $n = 500$, $p = 100$, and $s = 5$
- For the MARS model, we take $p = s = 5$ with $n = 200, 500$ or 10000
- Consider SNRs ranging from 0.05 to 6 equally spaced on the log scale
- Models built via the R package `randomForest` at default settings except for `mtry`
  - **Note:** Here we adopt a slightly different convention and let `mtry` denote the *proportion* of the p features available for splitting, rather than the raw number
- Differences of test error of random forests (`mtry` $= 0.33$) and bagging (`mtry` $= 1$) calculated on separate test set with sample size $n_t = 1000$ is calculated and averaged over $N = 500$ simulations.

Relative Performance of RFs

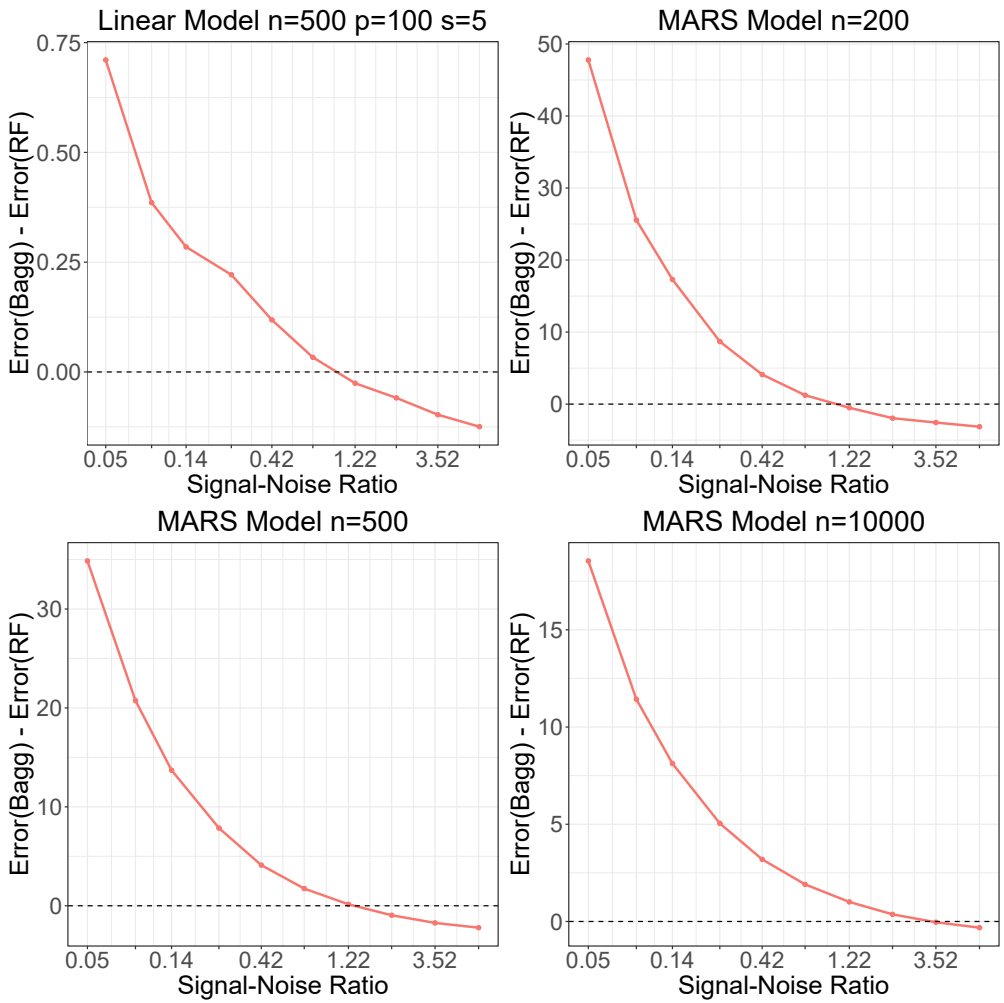


Figure 1: Error(Bagg) - Error(RF) vs SNR. Positive values indicate better performance by RFs.

Relative Performance of RFs

In each case we see a clear pattern: as the SNR goes up, the advantage offered by RFs dies out.

How about on real-world data? Here we take 15 datasets intended for regression to compare performance.

- Since we don't know the true SNRs, we inject additional random noise $\epsilon \sim N(0, \sigma^2)$ into the response
- σ^2 chosen as a proportion α of the sample variance of the response for $\alpha = 0, 0.01, 0.05, 0.1, 0.25, 0.5$
- Consider the relative test error defined by

$$\text{RTE} = \frac{\widehat{Err}(\text{Bagg}) - \widehat{Err}(\text{RF})}{\hat{\sigma}_y^2} \times 100\%$$

where $\widehat{Err}(\text{Bagg})$ and $\widehat{Err}(\text{RF})$ correspond to 10-fold CV error and $\hat{\sigma}_y^2$ is the empirical variance of the original response.

Relative Performance of RFs

Dataset	p	n
Abalone Age [abalone]	8	4177
Bike Sharing [bike]	11	731
Bioston Housing [boston]	13	506
Concrete Compressive Strength [concrete]	8	1030
CPU Performance [cpu]	7	209
Conventional and Social Movie [csm]	10	187
Facebook Metrics [fb]	7	499
Parkinsons Telemonitoring [parkinsons]	20	5875
Servo System [servo]	4	167
Solar Flare [solar]	10	1066

Table 1: Summary of low dimensional data utilized.

Relative Performance of RFs

Dataset	p	n
Aquatic Toxicity [AquaticTox]	468	322
Molecular Descriptor Influencing Melting Point [mtp2]	1142	274
Weighted Holistic Invariant Molecular Descriptor [pah]	112	80
Adrenergic Blocking Potencies [phen]	110	22
PDGFR Inhibitor [pdgfr]	320	79

Table 2: Summary of high dimensional data utilized.

Relative Performance of RFs

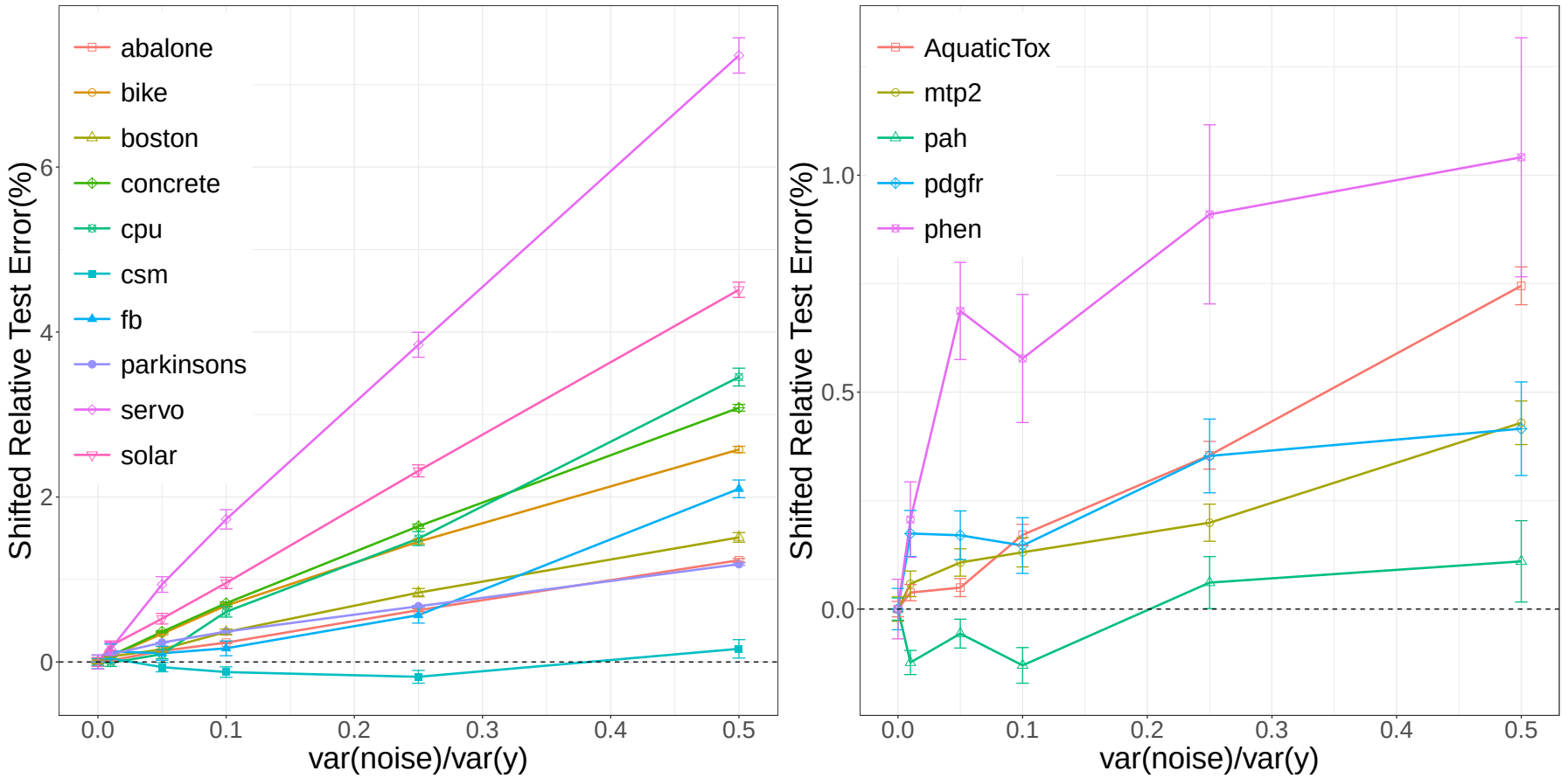


Figure 2: Shifted RTE on real data where additional noise is added. The left plot shows results on low-dimensional datasets taken from the UCI repository; the right plot shows results on high-dimensional datasets.

Relative Performance of RFs

Note that in comparing RFs to bagging, we're really just comparing RFs with different values of `mtry`. Suppose we reverse the direction of the problem and estimate the optimal value of `mtry` at various SNRs ...

- Here again we consider the same MARS and linear model setups as above with the same sampling, covariate, and noise settings (here $p = 20$, $s = 10$ for linear model)
- For both models, we consider $n = 50$ and $n = 500$
- Optimal `mtry` determined on independent test sets of same size; results averaged over 500 repetitions at each SNR level for each setting

Relative Performance of RFs

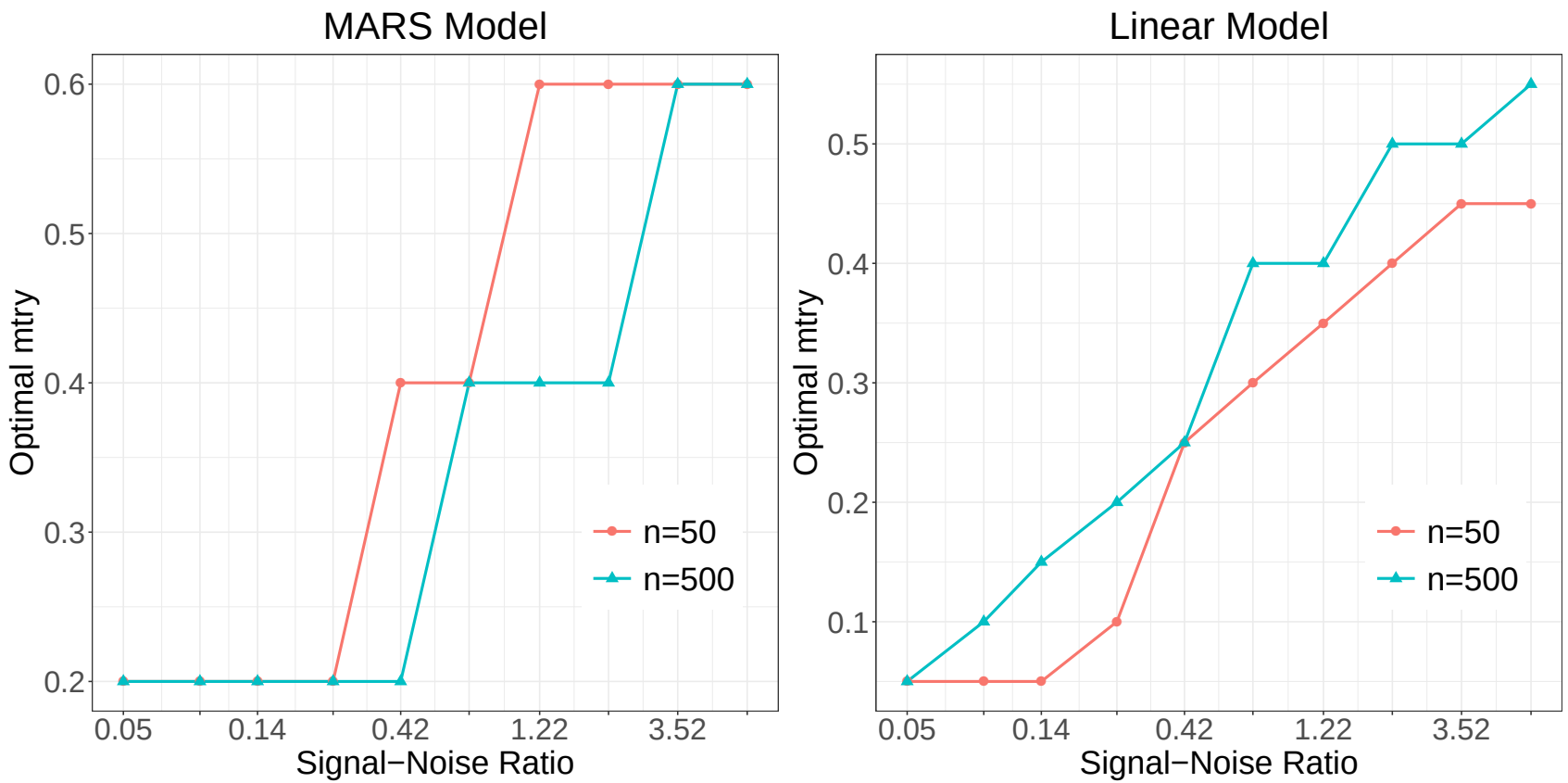


Figure 3: Optimal `mtry` vs SNR as measured by lowest average test error across 500 replicates.

Degrees of Freedom for RFs

So what leads to RFs' advantage in low SNR settings? In their recent empirical study comparing best subset selection (BSS), forward selection (FS), lasso, and relaxed lasso, Hastie et al. [2017] observe similar patterns of relative performance and attribute this to differences in degrees of freedom (dof).

Degrees of Freedom for RFs

- We consider two models: 'MARSadd' Friedman [1991]

$$Y = 0.1e^{4X_1} + \frac{4}{1 + e^{-20(X_2 - 0.5)}} + 3X_3 + 2X_4 + X_5 + \epsilon\,,$$

  with features sampled independently from Unif$(0, 1)$ and linear models drawn in same fashion as before with the following settings from Hastie et al. [2017]
  - **Low**: $n = 100$, $p = 10$, $s = 5$
  - **Medium**: $n = 500$, $p = 100$, $s = 5$
  - **High-10**: $n = 100$, $p = 1000$, $s = 10$
- SNR fixed at 3.52 (same findings at different SNRs)
- `mtry` $= 1/10, 1/3, 2/3, 1$
- dof estimated across 500 repetitions and plotted against `maxnodes` (maximum number of terminal nodes)

Degrees of Freedom for RFs

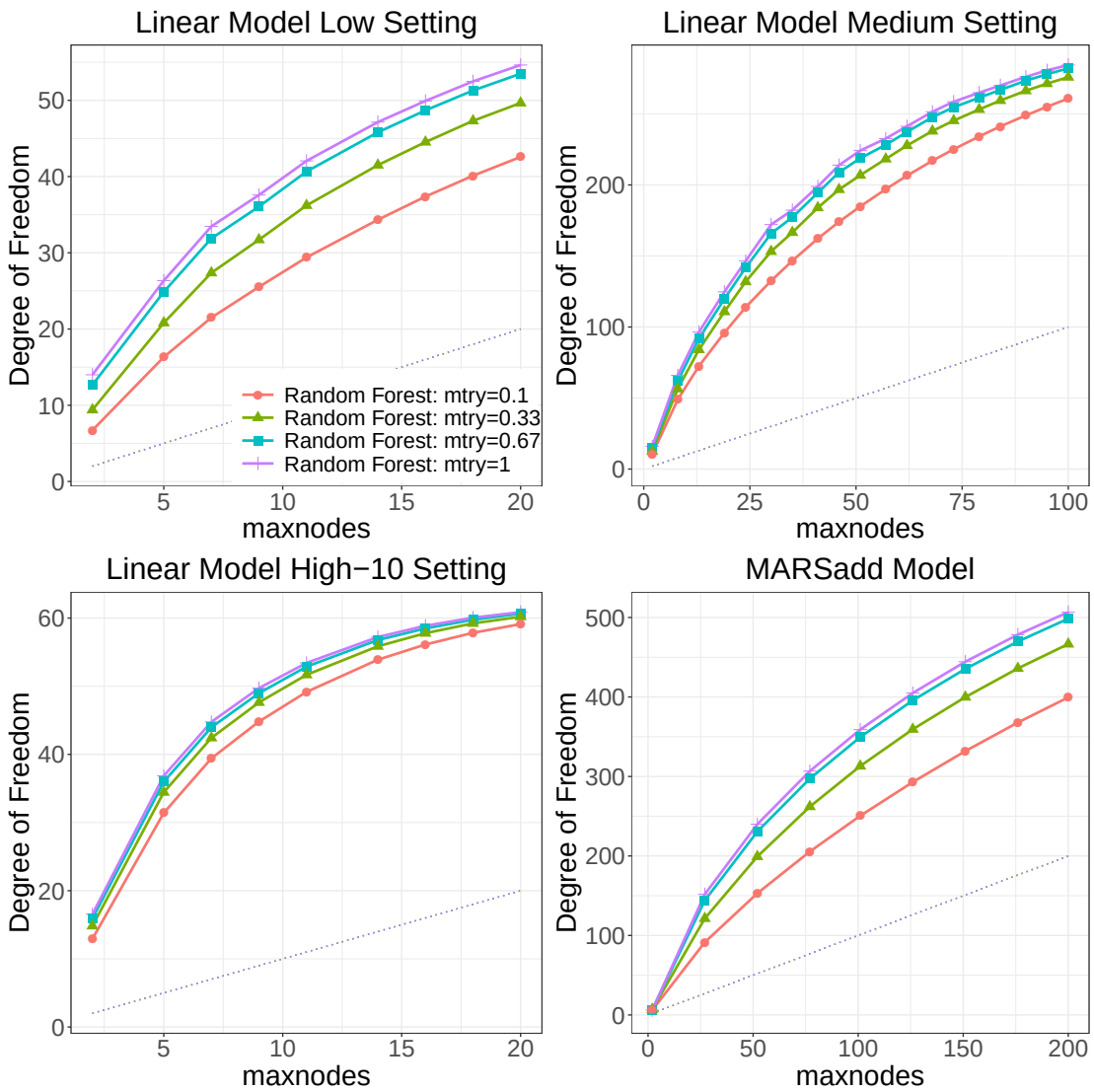


Figure 4: Estimated dof of RFs with different values of *mtry*

Extensions to RandFS

Results thus far are really interesting and helpful, but perhaps not shocking: more randomness $\implies$ lower variance / less overfitting $\implies$ improved performance at low SNRs.

But ... there seems to be **nothing tree-specific about this**. Trees are just sequentially partitioning the feature space into response-homogeneous regions ... can think of this as "building up" a model in the same fashion as forward selection.

So, if we were to create ensemble-ized versions of classical forward selection – analogues to the classic tree-based bagging and random forest procedures – would we see the same pattern? How would they compare to classical procedures (FS, lasso, relaxed lasso)?

Extensions to RandFS

Results thus far are really interesting and helpful, but perhaps not shocking: more randomness $\implies$ lower variance / less overfitting $\implies$ improved performance at low SNRs.

But ... there seems to be **nothing tree-specific about this**. Trees are just sequentially partitioning the feature space into response-homogeneous regions ... can think of this as "building up" a model in the same fashion as forward selection.

So, if we were to create ensemble-ized versions of classical forward selection – analogues to the classic tree-based bagging and random forest procedures – would we see the same pattern? How would they compare to classical procedures (FS, lasso, relaxed lasso)?

Extensions to RandFS

Bagging and random forests analogues for forward selection:

- **Bagged Forward Selection (BaggFS)**
  - Draw B bootstrap samples
  - Perform forward selection on each to depth of d
  - Average across the B models
- **Randomized Forward Selection (RandFS)**
  - Draw B bootstrap samples
  - Perform forward selection on each to depth of d, but at each step, randomly choose $\texttt{mtry} \times p$ features as candidates for selection. Such candidates are selected uniformly at random without replacement.
  - Average across the B models

Extensions to RandFS

- We consider linear models (constructed as before) in the following settings from Hastie et al. [2017]

	n	p	s
Low setting	100	10	5
Medium setting	500	100	5
High-5 setting	50	1000	5
High-10 setting	100	1000	10

- SNR equally spaced from 0.05 to 6 on log scale

Extensions to RandFS

- Following Hastie et al. [2017], performance measured as the test error relative to the Bayes error rate. Specifically, given a test point $(\boldsymbol{x_0}, y_0)$ with $y_0 = \boldsymbol{x_0}^T \beta + \epsilon_0$ and $\epsilon_0 \sim N(0, \sigma^2)$, the relative test error (RTE) to Bayes of a regression estimate $\hat{\beta}$ is given by

$$\text{RTE}(\hat{\beta}) = \frac{\mathbb{E}(y_0 - \boldsymbol{x_0}^T \hat{\beta})^2}{\sigma^2} = \frac{(\hat{\beta} - \beta)^T \Sigma (\hat{\beta} - \beta) + \sigma^2}{\sigma^2}$$

- Results averaged over 100 repetitions

Extensions to RandFS

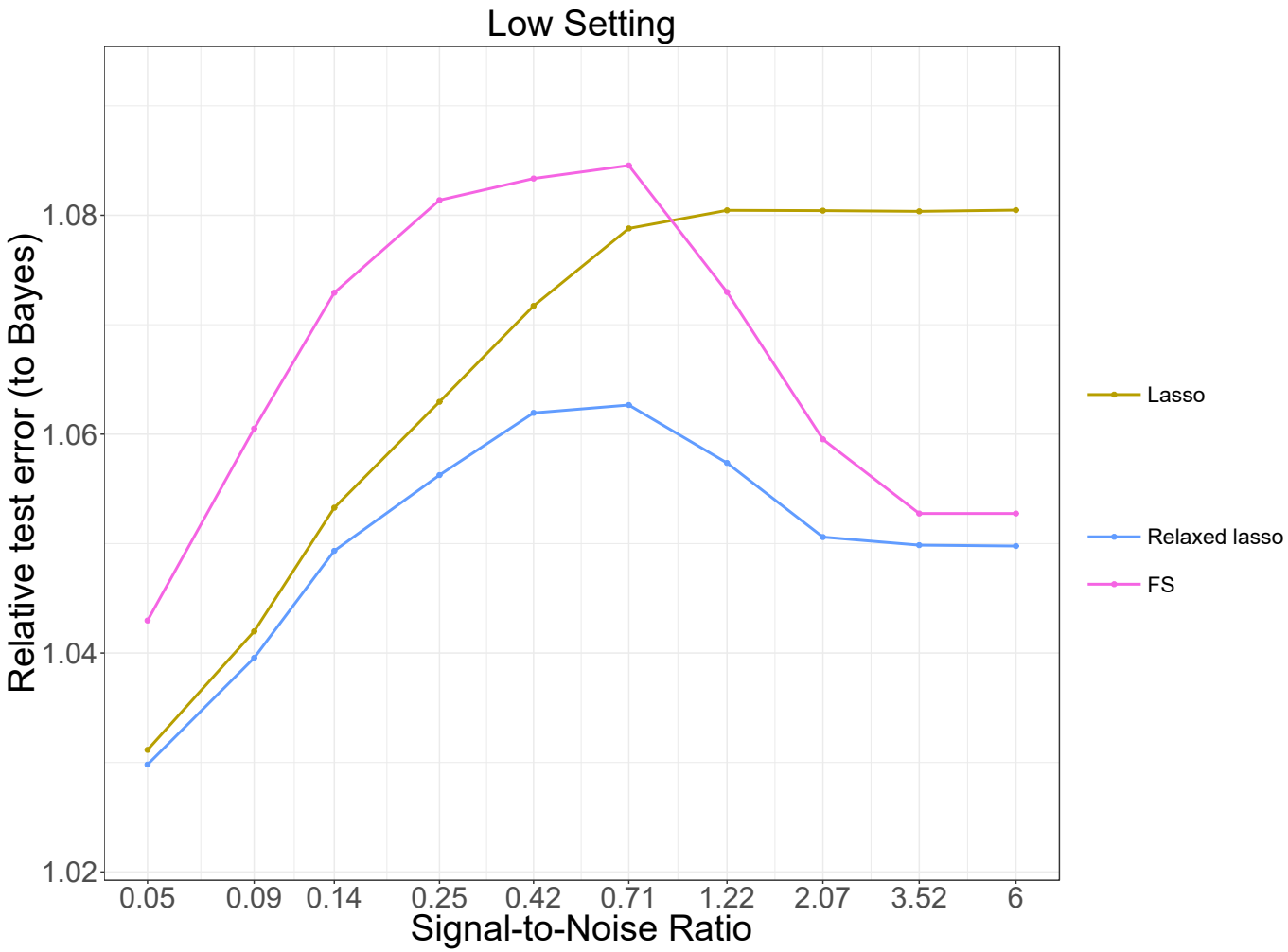


Figure 5: Performance Comparisons in low setting.

Extensions to RandFS

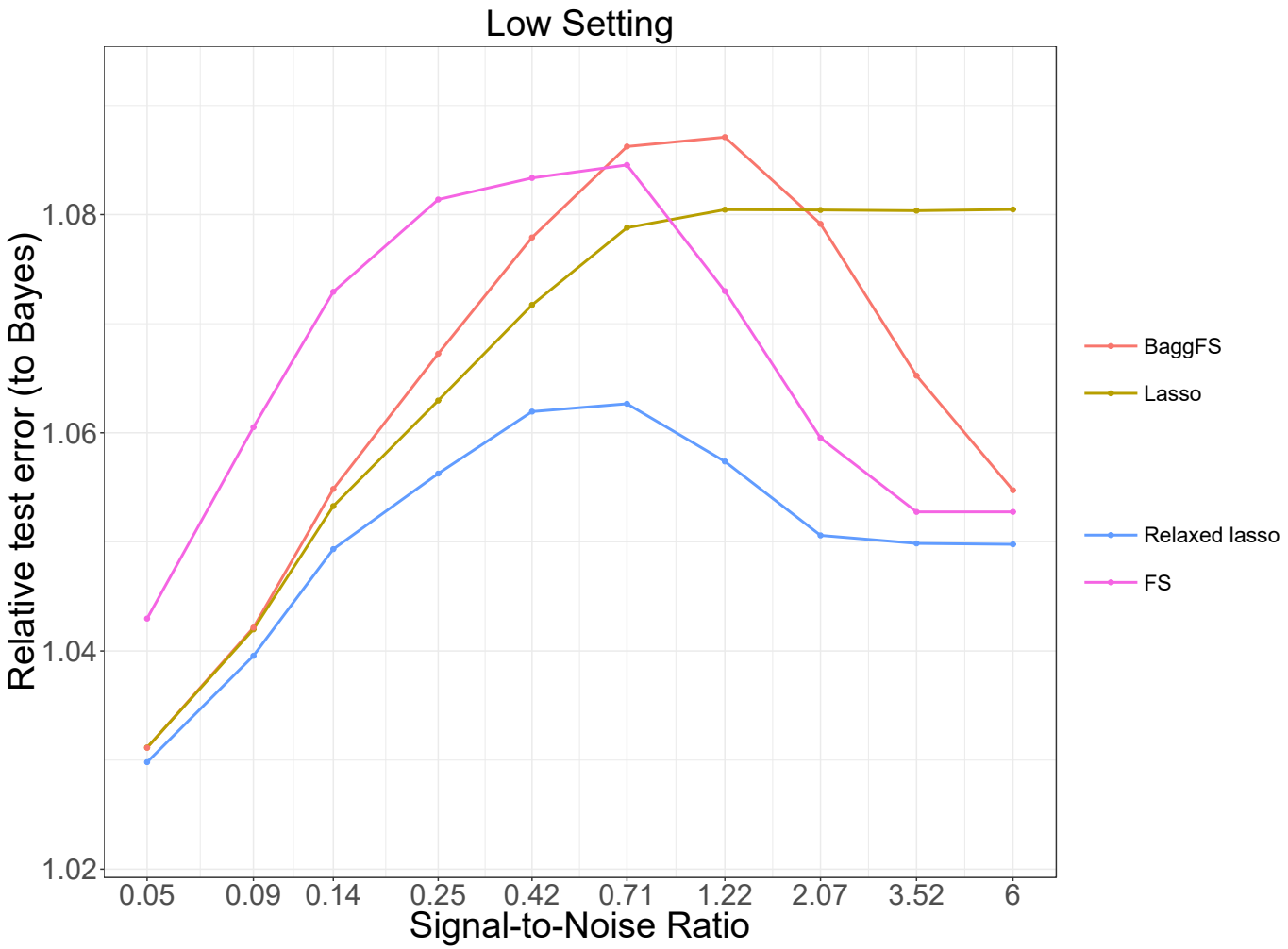


Figure 6: Performance Comparisons in low setting.

Extensions to RandFS

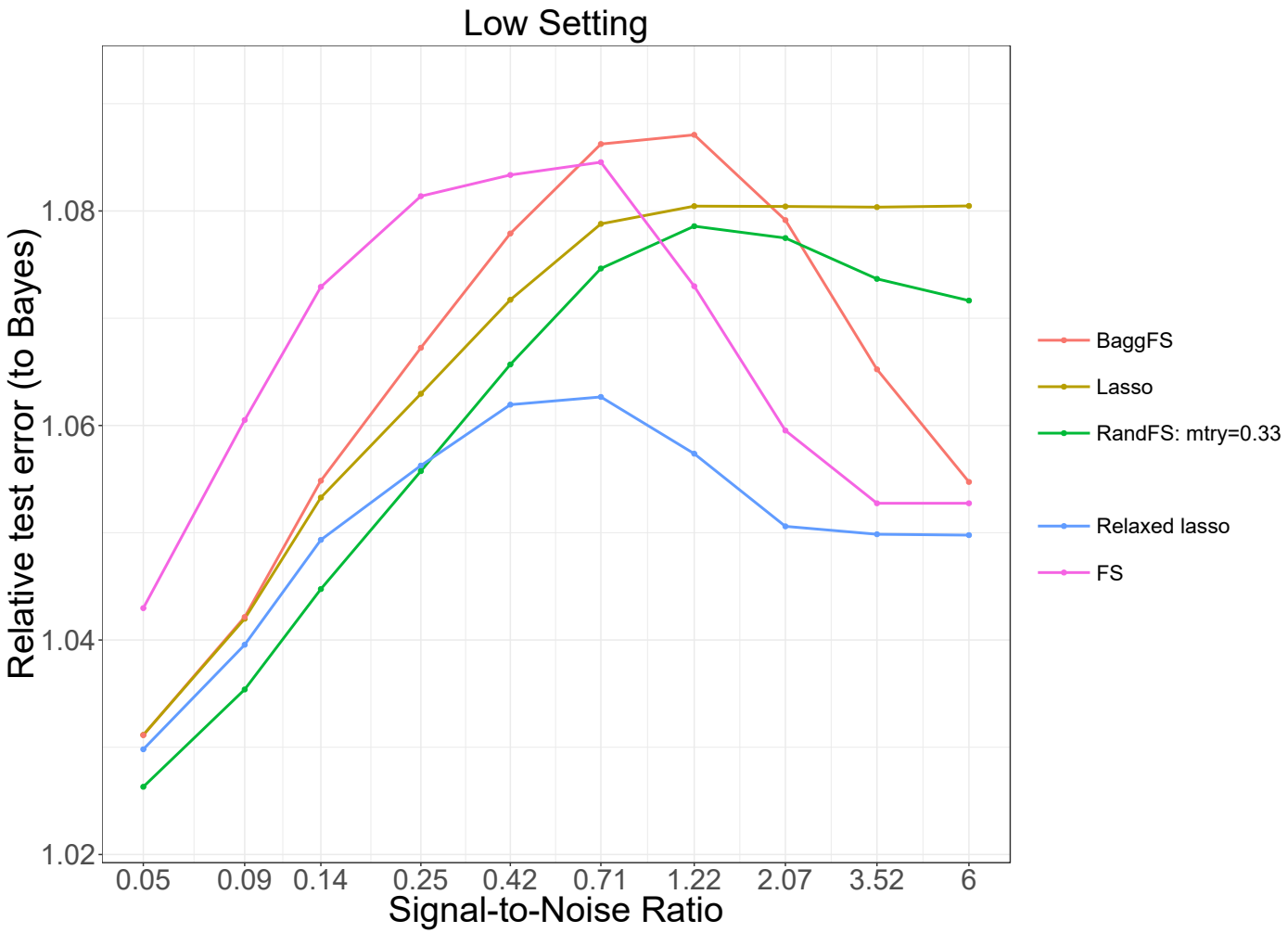


Figure 7: Performance Comparisons in low setting.

Extensions to RandFS

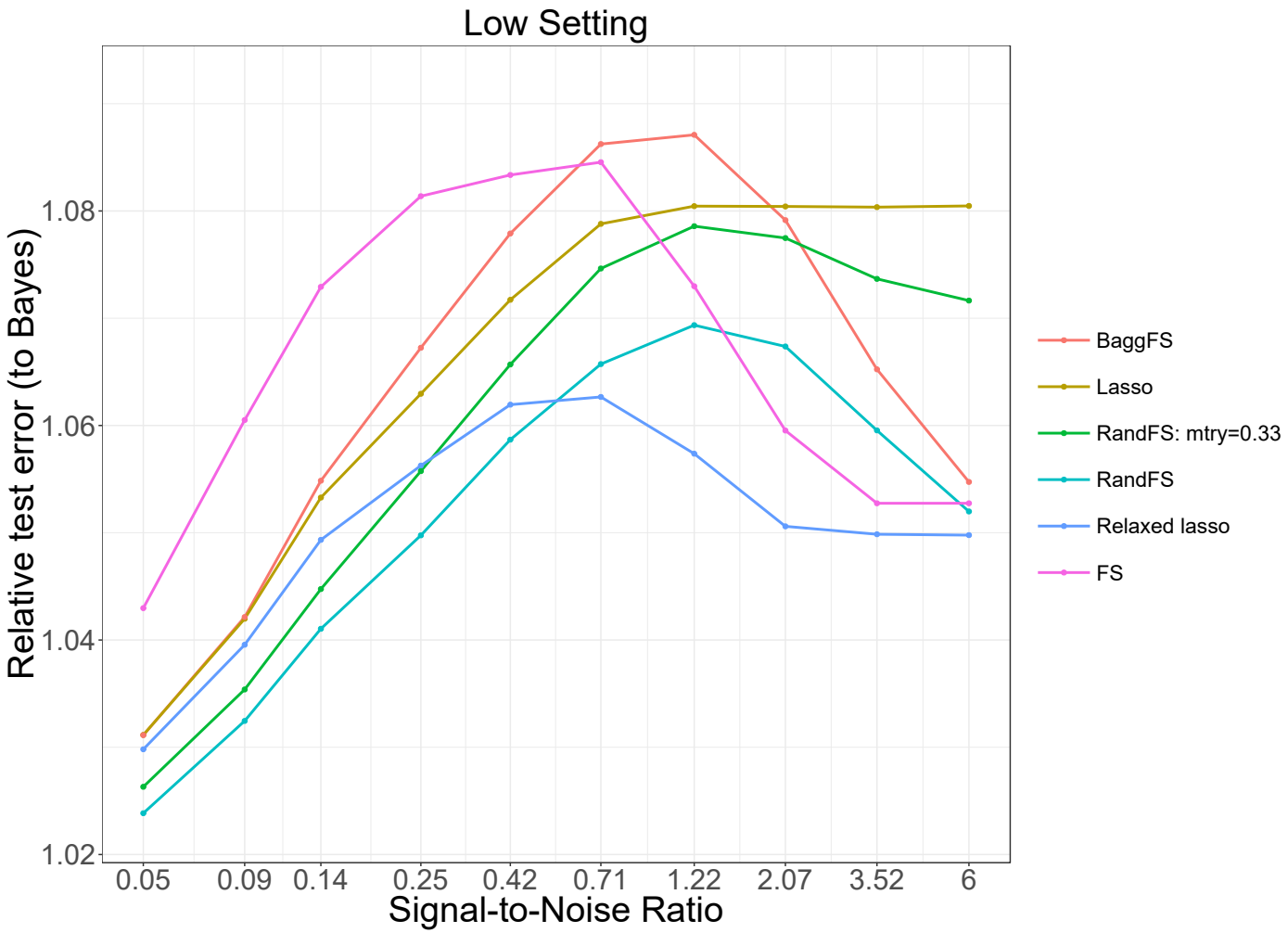


Figure 8: Performance Comparisons in low setting.

Extensions to RandFS

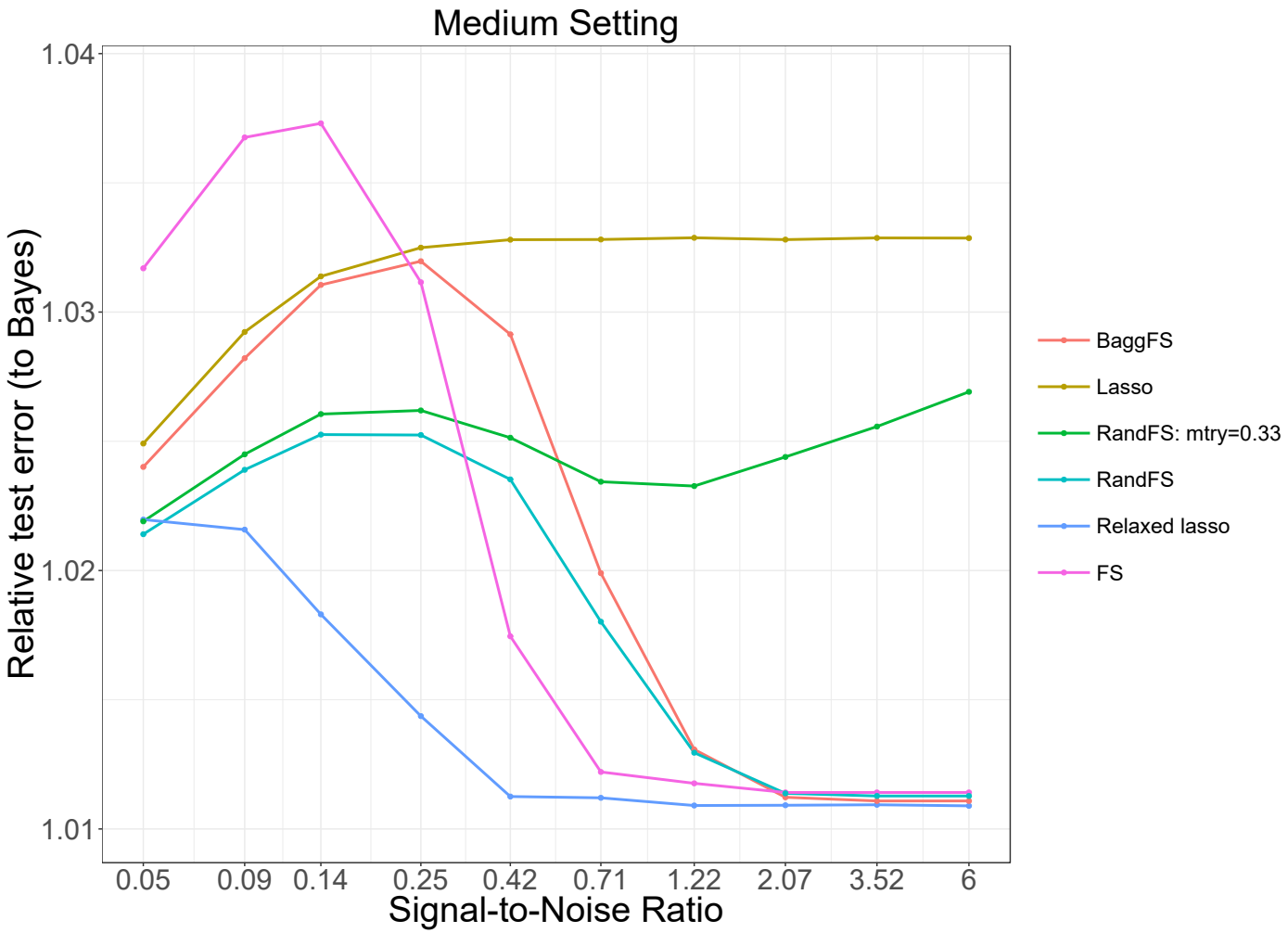


Figure 9: Performance Comparisons in medium setting.

Extensions to RandFS

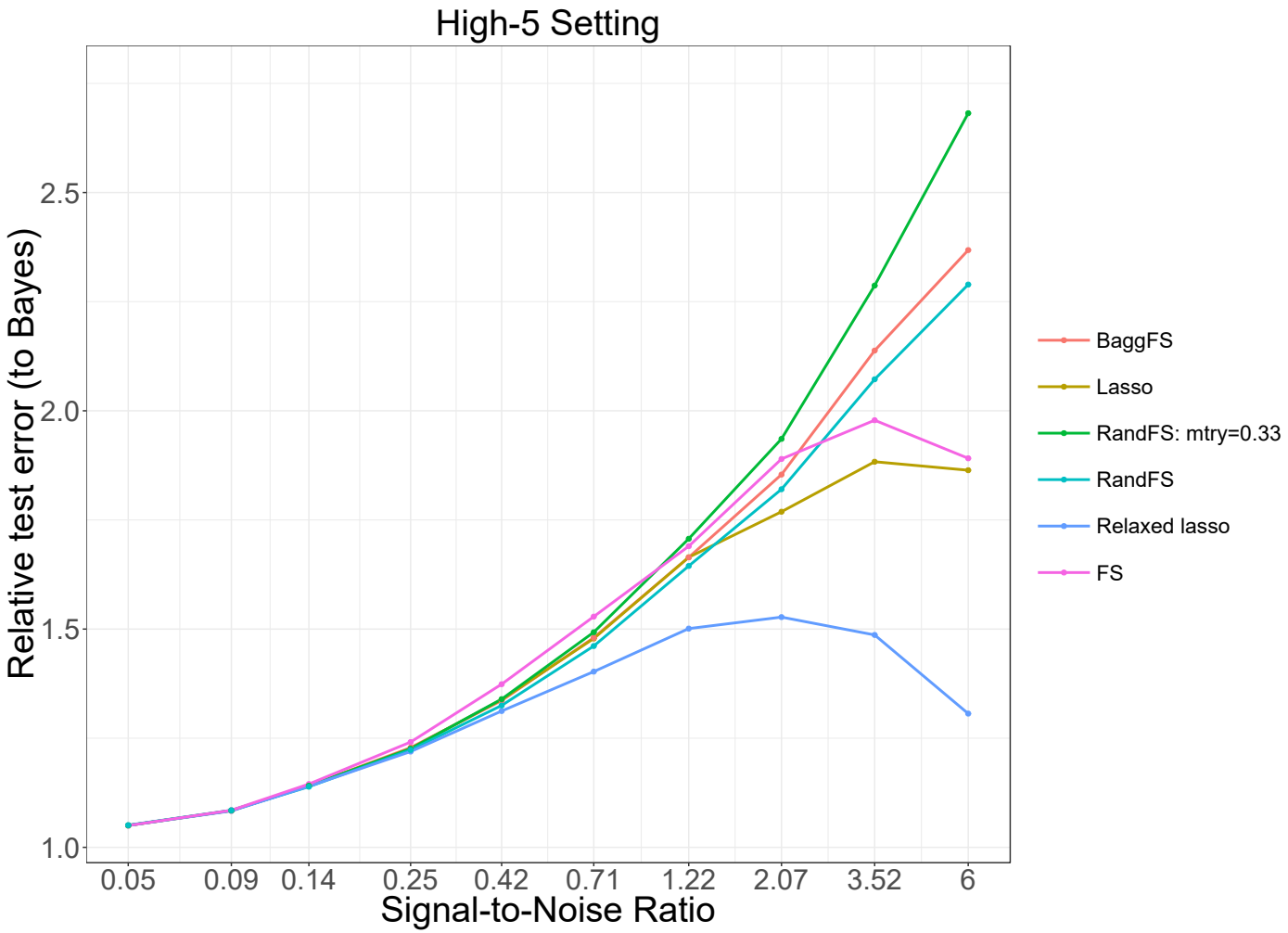


Figure 10: Performance Comparisons in high-5 setting.

Extensions to RandFS

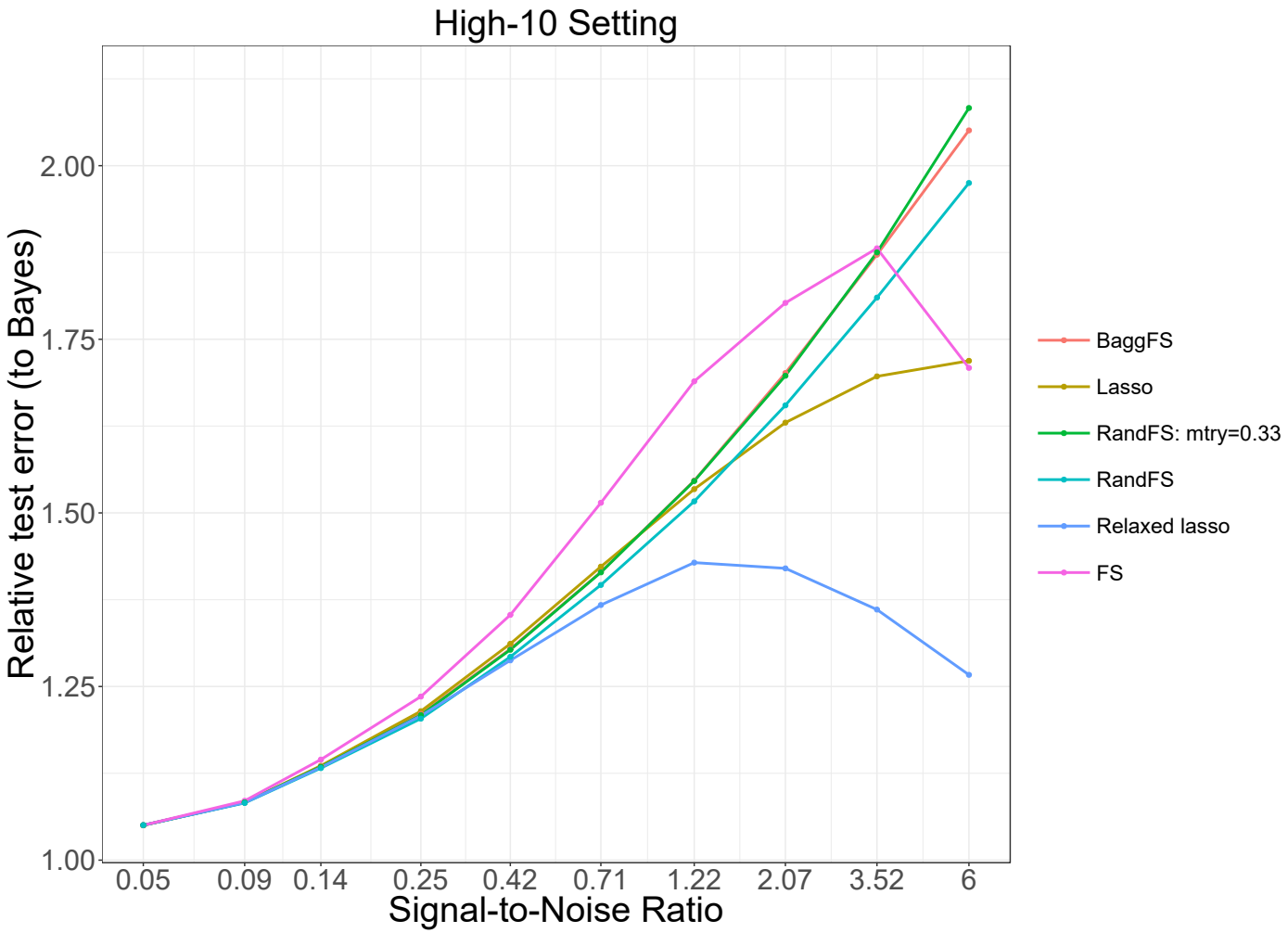
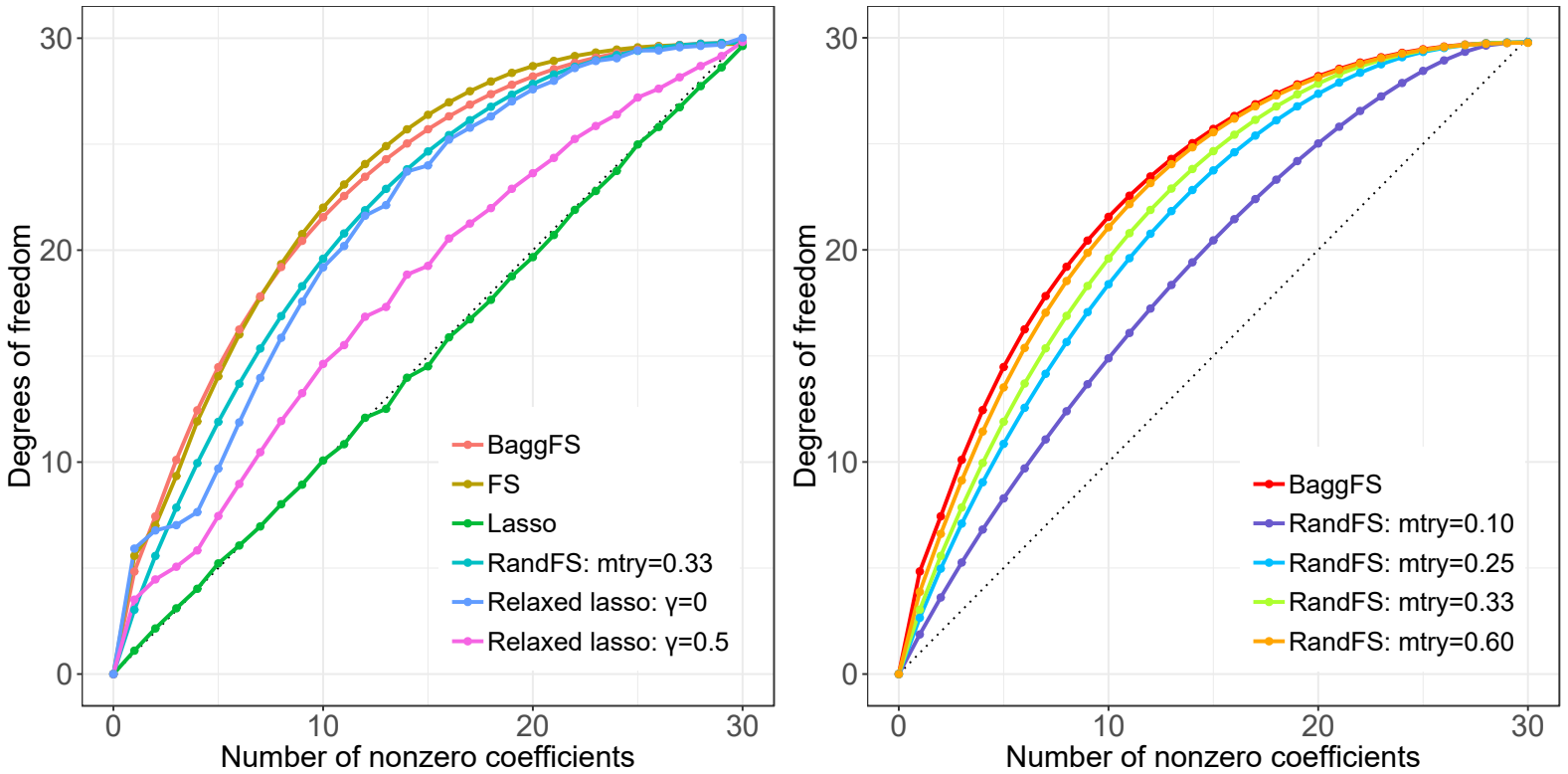


Figure 11: Performance Comparisons in high-10 setting.

Degrees of Freedom for RandFS

Dof estimates for RandFS follow the general pattern you would expect. Here we use a linear model with $n = 70, p = 30, s = 5$ and SNR $= 0.7$. Results are averaged over 500 repetitions.



Randomization as Regularization?

RandFS seems to be doing a sort of *implicit* regularization:

- For each of the B models, features are either selected or not
- For each feature X_k, its selection proportion α_k depends on the original data (and true relative importance), bootstrap samples, depth d to which models are grown, and `mtry`
- Coefficient estimates are effectively shrunk by amount proportional to that selection proportion
- In a simple case where $\boldsymbol{X}$ is orthogonal and the B models are built with $m < p$ features uniformly selected at random and n observations, estimates given by RandFS converges to ridge estimates with $\lambda = \frac{p-m}{p}$ as $B \to \infty$.
- LeJeune et al. [2019] showed that the optimal risk of ensembles of linear models built nonadaptively converges to the optimal risk of ridge regression.

Discussions

- Strong empirical evidence that relative improvement with RFs is a direct function of the SNR
- RFs not "just better" than bagging; `mtry` parameter really ought to be tuned (though we saw good performance even with $\texttt{mtry} = 0.33$ fixed)
- Reason to think that `mtry` serves much the same regularization role as, e.g. λ in ridge/lasso.
- This is a general principle rather than a tree-specific result.

Reference I

Simon Bernard, Sébastien Adam, and Laurent Heutte. Using random forests for handwritten digit recognition. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 1043–1047. IEEE, 2007.

Gérard Biau. Analysis of a Random Forests Model. *The Journal of Machine Learning Research*, 98888:1063–1095, 2012.

Gérard Biau and Luc Devroye. On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. *Journal of Multivariate Analysis*, 101(10):2499–2518, 2010.

Gérard Biau and Erwan Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016.

Gérard Biau, Luc Devroye, and Gábor Lugosi. Consistency of Random Forests and Other Averaging Classifiers. *The Journal of Machine Learning Research*, 9:2015–2033, 2008.

Leo Breiman. Randomizing outputs to increase prediction accuracy. *Machine Learning*, 40(3):229–242, 2000.

Leo Breiman. Random Forests. *Machine Learning*, 45:5–32, 2001.

Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. *Classification and Regression Trees*. Wadsworth, Belmont, CA, 1st edition, 1984.

Tim Coleman, Wei Peng, and Lucas Mentch. Scalable and efficient hypothesis testing with random forests. *arXiv preprint arXiv:1904.07830*, 2019.

Yifan Cui, Ruoqing Zhu, Mai Zhou, and Michael Kosorok. Some asymptotic results of survival tree and forest models. *arXiv preprint arXiv:1707.09631*, 2017.

D Richard Cutler, Thomas C Edwards Jr, Karen H Beard, Adele Cutler, Kyle T Hess, Jacob Gibson, and Joshua J Lawler. Random forests for classification in ecology. *Ecology*, 88(11):2783–2792, 2007.

Ramón Díaz-Uriarte and Sara Alvarez De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.

Thomas G Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157, 2000.

Roxane Duroux and Erwan Scornet. Impact of subsampling and pruning on random forests. *arXiv preprint arXiv:1603.04261*, 2016.

Bradley Efron. *The jackknife, the bootstrap, and other resampling plans*, volume 38. Siam, 1982.

Reference II

Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati, and Luc Van Gool. Random forests for real time 3d face analysis. *International Journal of Computer Vision*, 101(3):437–458, 2013.

Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15(1):3133–3181, 2014.

Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.

Jerome H Friedman. Multivariate Adaptive Regression Splines. *The annals of statistics*, pages 1–67, 1991.

Li Guo, Nesrine Chehata, Clément Mallet, and Samia Boukir. Relevance of airborne lidar and multispectral image data for urban scene classification using random forests. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(1):56–66, 2011.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2nd edition, 2009.

Trevor Hastie, Robert Tibshirani, and Ryan J Tibshirani. Extended comparisons of best subset selection, forward stepwise selection, and the lasso. *arXiv preprint arXiv:1707.08692*, 2017.

Giles Hooker and Lucas Mentch. Please stop permuting features: An explanation and alternatives. *arXiv preprint arXiv:1905.03151*, 2019.

Torsten Hothorn, Peter Bühlmann, Sandrine Dudoit, Annette Molinaro, and Mark J Van Der Laan. Survival ensembles. *Biostatistics*, 7(3):355–373, 2005.

Chen Huang, Xiaoqing Ding, and Chi Fang. Head pose estimation based on random forests for multiclass classification. In *2010 20th International Conference on Pattern Recognition*, pages 934–937. IEEE, 2010.

Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, Michael S Lauer, et al. Random survival forests. *The annals of applied statistics*, 2(3):841–860, 2008.

Jason M. Klusowski. Sharp analysis of a simple model for random forests. *arXiv preprint 1805.02587v6*, 2019.

Daniel LeJeune, Hamid Javadi, and Richard G Baraniuk. The implicit regularization of ordinary least squares ensembles. *arXiv preprint arXiv:1910.04743*, 2019.

Reference III

Yi Lin and Yongho Jeon. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101 (474):578–590, 2006.

Miles E Lopes, Suofei Wu, and Thomas Lee. Measuring the algorithmic convergence of randomized ensembles: The regression setting. *arXiv preprint arXiv:1908.01251*, 2019a.

Miles E Lopes et al. Estimating the algorithmic variance of randomized ensembles via the bootstrap. *The Annals of Statistics*, 47(2):1088–1112, 2019b.

Mahya Mehrmohamadi, Lucas K Mentch, Andrew G Clark, and Jason W Locasale. Integrative modelling of tumour dna methylation quantifies the contribution of metabolism. *Nature communications*, 7:13666, 2016.

Nicolai Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7(Jun):983–999, 2006.

Lucas Mentch and Giles Hooker. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *The Journal of Machine Learning Research*, 17(1):841–881, 2016.

Lucas Mentch and Giles Hooker. Formal hypothesis tests for additive structure in random forests. *Journal of Computational and Graphical Statistics*, 26(3):589–597, 2017.

Kristin K Nicodemus, James D Malley, Carolin Strobl, and Andreas Ziegler. The behaviour of random forest permutation-based variable importance measures under predictor correlation. *BMC bioinformatics*, 11(1):110, 2010.

Wei Peng, Tim Coleman, and Lucas Mentch. Asymptotic distributions and rates of convergence for random forests and other resampled ensemble learners. *arXiv preprint arXiv:1905.10651*, 2019.

Anantha M Prasad, Louis R Iverson, and Andy Liaw. Newer classification and regression tree techniques: bagging and random forests for ecological prediction. *Ecosystems*, 9(2):181–199, 2006.

Erwan Scornet. Random forests and kernel methods. *IEEE Transactions on Information Theory*, 62(3):1485–1500, 2016.

Erwan Scornet, Gérard Biau, Jean-Philippe Vert, et al. Consistency of random forests. *The Annals of Statistics*, 43(4): 1716–1741, 2015.

Joseph Sexton and Petter Laake. Standard errors for bagged and random forest estimators. *Computational Statistics & Data Analysis*, 53(3):801–811, 2009.

Reference IV

Jon Arni Steingrimsson, Liqun Diao, and Robert L Strawderman. Censoring unbiased regression trees and ensembles. *Journal of the American Statistical Association*, 114(525):370–383, 2019.

Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics*, 8(1):25, 2007.

Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. Conditional variable importance for random forests. *BMC bioinformatics*, 9(1):307, 2008.

Vladimir Svetnik, Andy Liaw, Christopher Tong, J Christopher Culberson, Robert P Sheridan, and Bradley P Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43(6):1947–1958, 2003.

Laura Toloşi and Thomas Lengauer. Classification with correlated features: unreliability of feature ranking and solutions. *Bioinformatics*, 27(14):1986–1994, 2011.

Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

Stefan Wager, Trevor Hastie, and Bradley Efron. Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *The Journal of Machine Learning Research*, 15(1):1625–1651, 2014.

Abraham J Wyner, Matthew Olson, Justin Bleich, and David Mease. Explaining the success of adaboost and random forests as interpolating classifiers. *The Journal of Machine Learning Research*, 18(1):1558–1590, 2017.

Faisal Zaman and Hideo Hirose. Effect of subsampling rate on subbagging and related ensembles of stable classifiers. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 44–49. Springer, 2009.

Ruoqing Zhu, Donglin Zeng, and Michael R Kosorok. Reinforcement learning trees. *Journal of the American Statistical Association*, 110(512):1770–1784, 2015.