Statistics and Computing (2020) 30:697–719
https://doi.org/10.1007/s11222-019-09914-9

High-dimensional regression in practice: an empirical study of finite-sample prediction, variable selection and ranking

Fan Wang[1] · Sach Mukherjee[2] · Sylvia Richardson[1] · Steven M. Hill[1]

Received: 27 July 2018 / Accepted: 16 October 2019 / Published online: 19 December 2019

Abstract

Penalized likelihood approaches are widely used for high-dimensional regression. Although many methods have been proposed and the associated theory is now well developed, the relative efficacy of different approaches in finite-sample settings, as encountered in practice, remains incompletely understood. There is therefore a need for empirical investigations in this area that can offer practical insight and guidance to users. In this paper, we present a large-scale comparison of penalized regression methods. We distinguish between three related goals: prediction, variable selection and variable ranking. Our results span more than 2300 data-generating scenarios, including both synthetic and semisynthetic data (real covariates and simulated responses), allowing us to systematically consider the influence of various factors (sample size, dimensionality, sparsity, signal strength and multicollinearity). We consider several widely used approaches (Lasso, Adaptive Lasso, Elastic Net, Ridge Regression, SCAD, the Dantzig Selector and Stability Selection). We find considerable variation in performance between methods. Our results support a "no panacea" view, with no unambiguous winner across all scenarios or goals, even in this restricted setting where all data align well with the assumptions underlying the methods. The study allows us to make some recommendations as to which approaches may be most (or least) suitable given the goal and some data characteristics. Our empirical results complement existing theory and provide a resource to compare methods across a range of scenarios and metrics.

Keywords Simulation study · High-dimensional regression · Penalized regression · Lasso · Variable selection · Prediction

1 Introduction

In a wide range of applications, it is now routine to encounter regression problems where the number of features or covariates p exceeds the sample size n, often greatly. Even in the simple case of linear models with independent Gaussian noise, estimation is nontrivial and requires specific assumptions. A common and often appropriate assumption is that of sparsity, where only a subset of the variables (the active set) have nonzero coefficients, with the number s_0 of such active variables usually assumed much smaller than p.

Penalized methods augment the regression log-likelihood with a penalty term that encodes a structural assumption such as sparsity. Recent years have seen much progress in theory and methodology for penalized regression (see Bühlmann and van de Geer (2011), for a lucid account). However, while the theoretical developments have been remarkable and insightful, they cannot go as far as telling the user which method to use in a given finite-sample setting. Meanwhile, rapid methodological progress has meant a wide range of plausible approaches to choose among.

The present study performs a systematic empirical comparison of a number of penalized regression methods, which could provide some guidance for users when selecting methods for specific applications. We consider seven popular approaches (Lasso, Adaptive Lasso, Elastic Net, Ridge Regression, SCAD, the Dantzig Selector and Stability Selection) and a range of data-generating scenarios. It is obvious that large departures from modeling assumptions can pro-

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s11222-019-09914-9) contains supplementary material, which is available to authorized users.

✉ Steven M. Hill
steven.hill@mrc-bsu.cam.ac.uk

[1] MRC Biostatistics Unit, University of Cambridge, Cambridge, UK

[2] German Centre for Neurodegenerative Diseases (DZNE), Bonn, Germany

duce poor results. Here, our intention is not so much to look at robustness to such departures, but rather to look at variation in performance even in the favorable case where assumptions broadly hold (i.e., for data generated from sparse linear models).

In the simulations, we vary a number of factors in a relatively fine-grained manner within an essentially full factorial design (i.e., all combinations of factors). Furthermore, in addition to synthetic data (covariates and responses are simulated), we also consider semisynthetic data (real covariates but simulated responses, using gene expression data from cancer samples) which allows us to study method performance under a more realistic covariate correlation structure. We distinguish between three goals: prediction, variable selection and variable ranking. We consider variable ranking in addition to selection due to the fact that in many applications, users are interested in guidance for follow-up studies or data acquisition. Then, highlighting variables in a suitable rank order is particularly important.

We find that for many scenarios there is substantial variation in performance between methods (i.e., choice of method is influential). However, there is no unambiguous winner across scenarios (i.e., details of the data-generating setup matter), and this is despite the fact that we focus on a relatively narrow class of scenarios broadly favorable to the approaches employed. Relative performance also depends on the specific goal.

Our study allows some broad recommendations to be made based on the goal and on characteristics of the data that are known, or can be determined, by the user (e.g., correlation structure). We find that Lasso and Adaptive Lasso are usually competitive for ranking when there is no or very weak correlation between variables, and Ridge Regression is often a good choice in more highly correlated scenarios. For prediction, Lasso is competitive in most scenarios (correlated or uncorrelated). Choice of method for selection depends on whether the user would rather keep false positives low or maximize the number of active variables discovered. For the former, our results suggest Stability Selection is the best option, and for the latter, Adaptive Lasso performs well when variables have no or very weak correlation and Elastic Net when variables are more highly correlated. Lasso typically offers a reasonable compromise between controlling false positives and discovering true positives.

We also find evidence of an interesting “phase transition”-like behavior for SCAD, where it goes from being the best performing approach to the worst as scenario difficulty increases. SCAD is therefore highly variable and so carries more risk as a choice of approach. Ridge Regression and Adaptive Lasso can also perform particularly poorly relative to other approaches in some scenarios for prediction. Furthermore, our results and associated simulation and plotting code (see “Code and data availability” section) provide a resource, allowing users to check in detail how the methods considered here fare against each other across many scenarios and also to extend the study with other (existing or novel) approaches.

In addition to the main simulation study, we extend some data-generating scenarios in specific directions to further explore properties of the methods. Specifically, we investigate how performance changes under a different covariate correlation structure to that explored in the main study, we explore sensitivity of Stability Selection to its tuning parameters and we examine the impact of heterogeneous regression coefficients on selection performance.

A number of previous papers have examined the empirical performance of penalized regression methods. Meinshausen and Bühlmann (2010) consider large p problems from a selection perspective. Bühlmann and Mandozzi (2014) is a more comprehensive study using semisynthetic data and evaluating screening or ranking properties in high-dimensional settings. Hastie et al. (2017) consider both low- and high-dimensional settings with a focus on prediction. In contrast to previous work, our design is considerably more comprehensive and systematic. We use finer grids on factors including n, p, s_0 and signal-to-noise ratio (SNR) so that our results cover a wider range of designs, allowing us to more fully investigate the trends in relative performance. We also consider several types of multicollinearity, so we can better understand this practically important factor. Furthermore, we evaluate all three of prediction, selection and ranking, using specific performance metrics for each. To limit scope, we do not consider Bayesian approaches here but note that there have been some interesting empirical comparisons of frequentist and Bayesian methods (including Celeux et al. 2012; Bondell and Reich 2012; Perrakis et al. 2019).

The remainder of the paper is organized as follows. In Sect. 2, we outline the methods compared and describe our simulation strategy, including the data-generating factors considered. We also give details of how the methods are implemented and the performance metrics used. Section 3 presents the results from our main simulation study. For each goal, we present some key observations and provide a summary with some recommendations. Results from additional simulations appear in Sect. 4. We conclude with a discussion in Sect. 5.

2 Methods

2.1 Model setting and notation

We focus on the best studied high-dimensional regression setting, namely the sparse linear model with independent Gaussian noise. That is, we consider models of the form

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

where $\mathbf{y} = (y_1, y_2, \ldots, y_n)^{\mathrm{T}}$ is a vector of responses, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_p]$ a $n \times p$ design matrix, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^{\mathrm{T}}$ a vector of (true) coefficients and $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)^{\mathrm{T}}$ are the errors. We use $S = \{j : \beta_j \neq 0\}$ to denote the active set with $s_0 = |S|$ the number of active variables. (Below, we also refer to active variables as "signals.") We focus on the case where $p > n$ and where s_0 is small (i.e., a sparse setting). Unless otherwise noted, $\boldsymbol{\epsilon} \sim N_n(\mathbf{0}, \sigma^2 I_n)$, $\sigma > 0$, where N_n is the n-dimensional Gaussian and I_n the $n \times n$ identity matrix.

2.2 The methods considered

A general penalized estimate for linear regression takes the following form:

$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\beta}{\operatorname{argmin}} \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p} P_\lambda(\beta_j) \qquad (2)$$

where $P_\lambda(\beta_j)$ is a penalty function applied to each component of $\boldsymbol{\beta}$ and $\lambda \geq 0$ is a tuning parameter that controls the amount of penalization. We consider several specific methods outlined below.

Lasso The Lasso estimator (Tibshirani 1996) takes the form given in (2) with an L_1-norm penalty: $P_\lambda(\beta_j) = \lambda|\beta_j|$. This shrinks coefficients toward zero, with some set to exactly zero, and λ controls the amount of shrinkage and degree of sparsity.

The theoretical properties of the Lasso have been well studied, and an extensive treatment can be found in Bühlmann and van de Geer (2011). We provide a very brief summary of the conditions for consistent selection and prediction. Allowing $p \gg n$, under a sparsity assumption on $\boldsymbol{\beta}$, Lasso is consistent for prediction for values of λ in a suitable range of the order $\sqrt{\log(p)/n}$. Additional assumptions can be made on the design matrix $\mathbf{X}$ to improve the rate of convergence for prediction error and to obtain consistency for estimation. For consistent variable selection, further non-trivial assumptions need to be made. One is a "beta-min" assumption that requires coefficients for active variables to be sufficiently large. If we then further assume a restrictive assumption on the design matrix $\mathbf{X}$, called the irrepresentable condition (Zhao and Yu 2006) (or equivalently the neighborhood stability assumption; Meinshausen and Bühlmann 2006), which places restrictions on correlation between variables, then Lasso is consistent for variable selection for $\lambda \gg \sqrt{\log(p)/n}$.

We highlight three important points arising from the above: First, that the conditions required for consistent selection are much stronger than those for consistent prediction; second, that λ should be larger for consistent variable selection than for consistent prediction; and third, that the prediction-optimal λ (estimated using, for example, cross-validation) can lead to inclusion of many false positives (Meinshausen and Bühlmann 2006).

Ridge Regression Ridge Regression (Hoerl and Kennard 1970) uses an L_2-norm penalty in (2): $P_\lambda(\beta_j) = \lambda\beta_j^2$. This shrinks coefficients toward zero, but results in non-sparse solutions because it is not singular at the origin. It also has a grouping effect where correlated variables have similar estimates. Note that Ridge Regression is the only method considered here that does not perform variable selection *per se*.

Elastic Net The Elastic Net estimator (Zou and Hastie 2005) is (2) with a penalty

$$P_\lambda(\beta_j) = \lambda\left(\alpha|\beta_j| + (1-\alpha)\beta_j^2\right). \qquad (3)$$

That is, L_1- and L_2-norm penalties combined with an additional parameter $\alpha \in [0, 1]$ ($\alpha = 1$ and $\alpha = 0$ correspond to Lasso and Ridge, respectively). This combines some of the benefits of Ridge while giving sparse solutions. In the $p > n$ setting, Lasso can select at most n variables, but Elastic Net has no such limitation.

SCAD SCAD (Fan and Li 2001) uses the following penalty in (2):

$$P_\lambda(\beta_j) = \begin{cases} \lambda|\beta_j|, & \text{if } |\beta_j| \leq \lambda \\ -\frac{|\beta_j|^2 - 2a\lambda|\beta_j| + \lambda^2}{2(a-1)}, & \text{if } |\beta_j| \in (\lambda, a\lambda] \\ \frac{(a+1)\lambda^2}{2}, & \text{if } |\beta_j| > a\lambda \end{cases} \qquad (4)$$

where $a > 2$ and $\lambda > 0$. This is a non-convex, quadratic spline function by which small coefficients are shrunk toward zero with a Lasso penalty, while large coefficients are not penalized. The resulting estimator is, unlike Lasso, nearly unbiased for large coefficients. Fan and Li (2001) and Fan et al. (2004) also show that SCAD enjoys an oracle property (assuming some regularity conditions)—it is simultaneously consistent for variable selection and estimation, where the latter is as efficient (asymptotically) as the ideal case when the true model is known in advance. For further details on the properties of SCAD, see Fan and Lv (2010) and references therein.

Adaptive Lasso Adaptive Lasso (Zou 2006) uses a Lasso penalty with weights in (2): $P_\lambda(\beta_j) = \lambda\omega_j|\beta_j|$. Similar in spirit to SCAD, Adaptive Lasso aims to eliminate the bias in the Lasso by shrinking larger coefficients less than smaller ones. This coefficient-specific regularization is achieved using the weights ω_j, which are taken to have the form $\omega_j = 1/|\tilde{\beta}_j|^\gamma$, where $\tilde{\beta}_j$ is an initial estimate for β_j and $\gamma > 0$. Larger initial estimates give rise to smaller weights and so receive less shrinkage. The ordinary least squares estimate and Ridge Regression estimate are suggested as initial estimates by Zou (2006). Adaptive Lasso also enjoys the oracle property (for suitable choices of λ).

Dantzig Selector The Dantzig Selector estimator (Candes and Tao 2007) takes a different form to that in (2), namely:

$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \|\boldsymbol{\beta}\|_1 : \|\mathbf{X}^{\mathrm{T}} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})\|_\infty \leq \lambda \right\}. \tag{5}$$

The Dantzig Selector and the Lasso are closely connected as discussed in Bickel et al. (2009), and under certain conditions on the design matrix, Lasso and Dantzig provide the same solution (Meinshausen et al. 2007; James et al. 2009).

Stability Selection This is a general approach by which to combine variable selection with data subsampling to obtain more stable selection and control the number of false positives. Specifically, M random data subsamples of size $\tilde{n} < n$ are generated by sampling without replacement. Applying a variable selection procedure, with regularization parameter λ, to these datasets gives a score $\hat{\Pi}_{\lambda,j}$ indicating the frequency with which variable j is selected among the M iterations. Let Λ denote the set of considered values for the regularization parameter. Then, a set of "stable variables" is obtained by choosing those variables that have selection probabilities larger than a cutoff value $\pi_{\mathrm{thr}} \in (0, 1)$ for any $\lambda \in \Lambda$.

In contrast to the methods described above, Stability Selection does not require setting of the parameter λ, but instead requires the cutoff π_{thr} to be chosen. Meinshausen and Bühlmann (2010) provide theoretical results showing how π_{thr} can be chosen to achieve a user-specified upper bound $\tilde{V}$ on the expected number of false positives $\mathbb{E}[V]$, assuming a fixed set of regularization parameters Λ. Alternatively, the user can fix π_{thr} and then the theory shows how Λ should be chosen to achieve the desired upper bound on $\mathbb{E}[V]$. In our study, we use the Lasso as the variable selection procedure with Stability Selection.

2.3 Simulation setup

We generate values for the response vector using model (1). We set $\boldsymbol{\beta}$ to have s_0 nonzero entries (all set to 3 except in Sect. 4.3 where we consider heterogeneous coefficients) and then set σ to obtain a desired SNR, defined here as SNR $= \sqrt{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta}/(n\sigma^2)}$.

We consider synthetic data, where both covariates and responses are simulated, and semisynthetic data, where covariates are real and responses are simulated.

2.3.1 Synthetic data

We consider the following two designs with synthetic covariates:

- *Independence design* All p covariates are i.i.d. standard normal.
- *Pairwise correlation design* The p covariates are partitioned into B blocks, each of size $p^B = p/B$. All covariates are standard normal but with correlation between any pair of covariates within the same block set to ρ. Covariates in different blocks are independent of each other. The number of active variables within a block is s_0^B for the first s_0/s_0^B blocks, with the remaining blocks containing no active variables.

2.3.2 Semisynthetic data

We consider semisynthetic data using real covariates from The Cancer Genome Atlas (TCGA) study. We use gene expression data from TCGA ovarian cancer samples (The Cancer Genome Atlas Research Network 2011).[1] The dataset contains 594 samples and expression levels for 22,277 genes. The samples are a mixture of primary tumor (569), recurrent tumor (17) and normal tissue (8). We randomly subsample the samples and genes to obtain a $n \times p$ design matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_p]$. Those samples not included in $\mathbf{X}$ are used as test data.

Signals are allocated among the p predictors to give either "low"- or "high"-correlation designs, using an approach similar to Bühlmann and Mandozzi (2014):

- *"Low"-correlation design* We allocate s_0 signals at random among $\mathbf{x}_1, \ldots, \mathbf{x}_p$.
- *"High"-correlation design* We use the following procedure to form correlated blocks:

  (i) Form a block of $p^B = 10$ predictors consisting of the two predictors $\tilde{\mathbf{x}}_1$ and $\tilde{\mathbf{x}}_2$ that are most correlated and the eight other predictors that are most correlated with $\tilde{\mathbf{x}}_1$

  (ii) Allocate s_0^B signals to this block by designating $\tilde{\mathbf{x}}_1$ and the $s_0^B - 1$ predictors that are most correlated with it as signals

  (iii) Repeat steps (i) and (ii), but remove from consideration any predictors already allocated to a block, and continue repeating until s_0 signals have been allocated.

[1] Specifically, we use the dataset provided in Supplementary Appendix of Tucker et al. (2014); the dataset is available at http://bioinformatics.mdanderson.org/Supplements/ResidualDisease.

Table 1 Factors varied in the simulation study and values considered

Factors	Values considered
All designs	
Sample size, n	100, 200, 300
Dimensionality, p	500, 1000, 2000, 4000
Sparsity, s_0	10, 20, 40
Signal-to-noise ratio, SNR	0.5[a], 1, 2, 4
Synthetic (pairwise) correlation design only	
Block size, p^B	10, 100
Pairwise correlation within a block, ρ	0.5, 0.7, 0.9
Number of signals per block, s_0^B	1, 2, 5
Semisynthetic ("low"/"high") correlation designs only	
Block size, p^B	10
Number of signals per block, s_0^B	1, 2, 5

Note that for the correlation designs, the s_0^B signals per block apply to the first s_0/s_0^B blocks only

[a] All designs except synthetic pairwise correlation design

Table 2 Combinations of p, p^B, s_0 and s_0^B explored in the (synthetic and semisynthetic) correlation designs

p	p^B	$B=\frac{p}{p^B}$	s_0	s_0^B		
				1	2	5
Synthetic (pairwise) correlation design only						
500	100	5	10	✗	✓	✓
			20	✗	✗	✓
			40	✗	✗	✗
1000	100	10	10	✓	✓	✓
			20	✗	✓	✓
			40	✗	✗	✓
2000	100	20	10	✓	✓	✓
			20	✓	✓	✓
			40	✗	✓	✓
4000	100	40	10	✓	✓	✓
			20	✓	✓	✓
			40	✓	✓	✓
All correlation designs						
*	10	*	*	✓	✓	✓

✓ indicates that the combination is included, and ✗ indicates that the combination is not included. For $p^B = 10$, * denotes all combinations of p and s_0

The p variables in each of our simulation scenarios are selected completely at random from the original dataset, so the correlation structure among the p variables is representative of the original data. In the "low"-correlation design, the correlation between a given signal and any other variable is, on average, the same as the average correlation between all p variables. In the "high"-correlation design, the average correlation between all p variables will follow the same distribution as in the "low"-correlation design. However, by identifying correlated blocks and allocating signals within these blocks as described above, a given signal is now more likely to have higher correlation with some non-signals and, for $s_0^B > 1$, with some other signals.

2.3.3 Systematic exploration of data-generating factors

We consider the effects of the various data-generating factors in a systematic way via 2394 simulation scenarios, each corresponding to a different configuration. The values considered for each factor are shown in Table 1, and we cover the majority of combinations of the factors. One exception is for correlation designs, we exclude some combinations of s_0^B and $B = p/p^B$ which violate the necessary constraint $s_0^B \geq s_0/B$ (see Table 2). Also, SNR = 0.5 is not considered for the synthetic correlation design.

2.4 Method implementation

Tuning parameters are set to reflect the way methods would typically be used by users. For Lasso, Elastic Net, Ridge Regression, SCAD, Adaptive Lasso and Dantzig Selector, λ is set via tenfold cross-validation (CV). Following Bühlmann and Mandozzi (2014), we implement two versions of Elastic Net with $\alpha = 0.3$ and $\alpha = 0.6$, referred to as heavy Elastic Net (HENet) and light Elastic Net (LENet), respectively. For SCAD, we set $a = 3.7$, as recommended by Fan and Li (2001). For Adaptive Lasso (AdaLasso), we use the Ridge Regression estimate as the initial estimate to calculate the weights and set $\gamma = 1$. For Stability Selection, we set the number of iterations to $M = 100$ with subsample size $\tilde{n} = \lfloor 0.632n \rfloor$ and selection probability cutoff $\pi_{\text{thr}} = 0.6$ (the R package defaults; see below). We do not place any explicit control on the expected number of false positives $\mathbb{E}[V]$ (i.e., we consider the full range of regularization parameters Λ). An exception to this is for selection in the semisynthetic data analysis, where we set $\tilde{V}$, the upper bound on $\mathbb{E}[V]$, to 10. However, we assess sensitivity to these tuning parameters in Sect. 4.

We use available R packages to implement the methods: `glmnet` for Lasso, Elastic Net, Ridge Regression and Adaptive Lasso (Friedman et al. 2010); `ncvreg` for SCAD (Breheny and Huang 2011); `flare` for Dantzig Selector (Li et al. 2015); and `c060` for Stability Selection (Sill et al. 2014). Covariates are standardized, and the response vector is centered. We run all methods on all simulation scenarios with the exception of Dantzig and AdaLasso: Dantzig is run only for the synthetic independence design, and synthetic correlated design with $p = 500$ and $p = 1000$, due to its computational demands under multicollinearity for large p; Adaptive Lasso is not run for the synthetic correlated design. For each simulation scenario, we show results averaged across 64 simulated datasets.

2.5 Performance metrics

We distinguish between prediction, variable selection and ranking and use the following metrics.

Prediction To assess predictive performance, we use the root-mean-squared error (RMSE). For each simulation scenario, we generate training data with sample size n and test data with sample size $n_{\text{test}} = 500$. Models are fitted on training data to obtain coefficient estimates $\hat{\boldsymbol{\beta}}_{cv}$ and prediction error, calculated as RMSE $= \|\mathbf{y}_{\text{test}} - \mathbf{X}_{\text{test}}\hat{\boldsymbol{\beta}}_{cv}\|_2/\sqrt{n_{\text{test}}}$, where $\mathbf{y}_{\text{test}}$ and $\mathbf{X}_{\text{test}}$ are the test responses and design matrix, respectively. Stability Selection focuses on variable selection, and we therefore do not include it in assessment of predictive performance.

Variable selection For assessment of variable selection, we use true positive rate (TPR) and positive predictive value (PPV):

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \in [0, 1]; \quad \text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} \in [0, 1], \quad (6)$$

where TP, FP and FN are the number of true positives, false positives and false negatives, respectively. Ridge Regression does not perform variable selection *per se* and is therefore excluded from this evaluation.

Variable ranking For ranking, we assess performance using the partial area under the receiver operating characteristic curve (pAUC). This is the area under the curve obtained when restricting to a maximum of 50 false positives (FPR $= \frac{50}{p - s_0}$). The pAUC calculation requires a score under which to rank variables j. For Ridge Regression, we rank by $s_j = |(\hat{\beta}_{cv})_j|$ and for Stability Selection by $s_j = \max_{\lambda \in \Lambda} \hat{\Pi}_{\lambda,j}$. For the other methods (Lasso, Elastic Net, SCAD and Dantzig Selector), we could use $|(\hat{\beta}_{cv})_j|$ as for Ridge, but due to sparsity this would involve ranking many covariates with $(\hat{\beta}_{cv})_j = 0$. We instead consider the set of estimated active sets $\{\hat{S}_\lambda : \lambda \in \Lambda\}$ where Λ is the set of candidate regularization parameters. We consider a covariate to be more important the longer it remains in $\hat{S}_\lambda$ as λ increases and more sparsity is induced. This motivates defining ranking scores as: $s_j = \max\{\tilde{\lambda} \in \Lambda : j \in \hat{S}_\lambda$ for all $\lambda \leq \tilde{\lambda}, \lambda \in \Lambda\}$ or $s_j = 0$ if $j \notin \hat{S}_{\lambda_{\min}}$, where $\lambda_{\min} = \min\{\lambda \in \Lambda\}$.

3 Main results

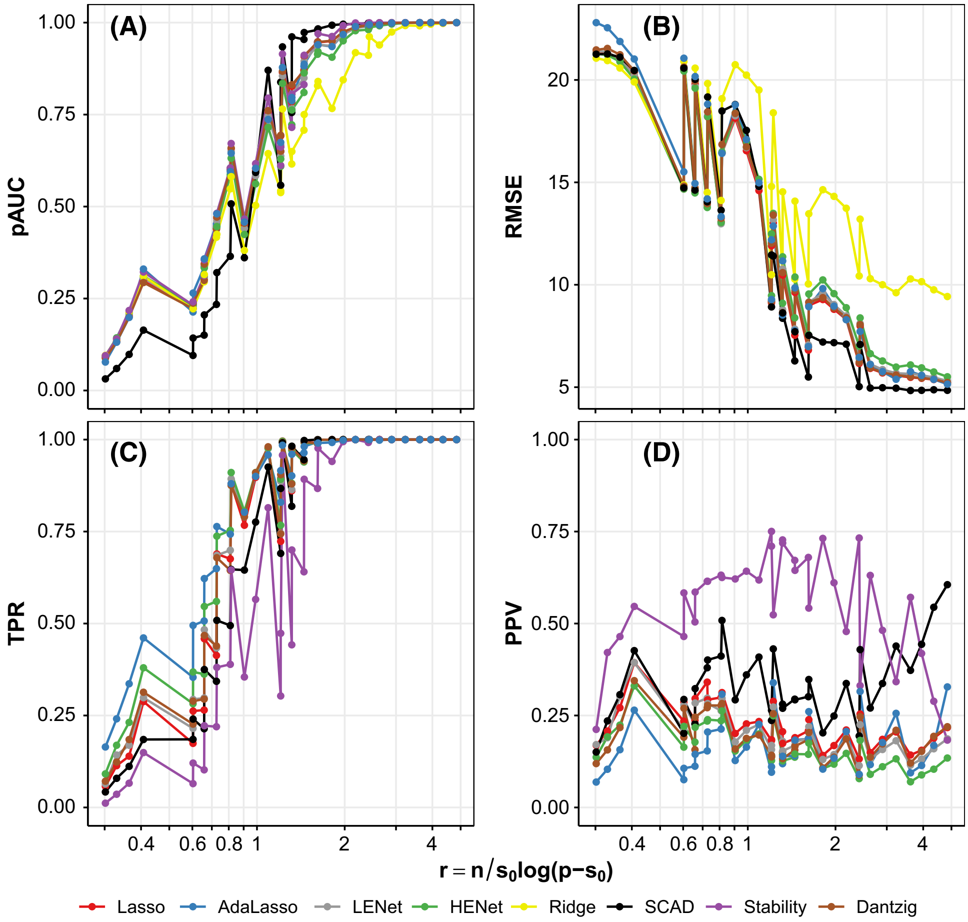
Due to the large number of simulation regimes, we focus below on the key patterns. All performance data and plotting code are made available on GitHub, allowing specific scenarios to be investigated further (see "Code and data availability" section). Figures S1–S21, referred to below, can be found in Supplementary Material.

We first present summary observations that hold across all the simulation scenarios. We then present results for each metric in turn: ranking in Sect. 3.2, prediction in Sect. 3.3 and selection in Sect. 3.4. In each of these three sections, we first present key observations for the synthetic independence design and then key observations for the correlation designs (both semisynthetic and synthetic designs). We then end each section by providing a summary with a recommendation regarding choice of method.

3.1 Observations from across all simulation scenarios

An approximate guide to simulation scenario difficulty Figure 1 shows the performance metrics versus rescaled sample

Fig. 1 Ranking (**a**), prediction (**b**) and selection (**c**, **d**) performance versus the rescaled sample size $r = n/(s_0 \log(p - s_0))$ for synthetic independence design scenarios with SNR = 2. Line color indicates method. Note that Stability Selection and Ridge Regression are not included in the assessment of prediction and selection performance, respectively. See Sect. 2.5 for details of metrics; pAUC = partial area under the receiver operating characteristic curve, RMSE = root-mean-squared error, TPR = true positive rate, PPV = positive predictive value



size r, for the synthetic independence design with SNR = 2. The quantity r equals $n/(s_0 \log(p - s_0))$ (see Wainwright 2009) and is motivated by scaling results for consistent Lasso variable selection. Large (small) values of r can be interpreted as large (small) sample size relative to dimensionality and sparsity. We observe a clear overall trend of better pAUC (Fig. 1a) and TPR (Fig. 1c) for all methods as r increases, with performance leveling off for larger values of r. The trend is similar for RMSE as r increases (Fig. 1b). The behavior of PPV is method-dependent and the overall trend is non-monotonic as r increases (Fig. 1d). Performance with varying r was qualitatively similar for other SNR values and also for correlation designs (see Fig. S1 for independence design with SNR = 0.5 and Fig. S2 for a semisynthetic correlation design with SNR = 2). Therefore, although the motivation for r lies in asymptotic theory for variable selection, we found that r and SNR together serve as a useful approximate guide to the difficulty of each simulation scenario for all three tasks (selection, ranking and prediction). We make use of this characterization below.

LENet is between Lasso and HENet The performance of LENet is invariably between that of Lasso and HENet for all metrics. For example, ranking performance of LENet lies between Lasso and HENet for 98% of synthetic data scenarios where there is a "salient" difference in pAUC between Lasso and HENet. (For our purposes here, we take a difference in pAUC of larger than 0.01 to be "salient.") We therefore exclude LENet below to aid presentation.

Dantzig Selector is similar to Lasso The Dantzig Selector mostly performed similarly to Lasso (see red and brown lines in Fig. 1 and see also Fig. S3), in line with theory (e.g., Meinshausen et al. 2007; Efron et al. 2007). However, Dantzig is more computationally expensive than Lasso (Meinshausen et al. 2007). For example, when $(n, p, s_0) = (100, 500, 10)$ and SNR = 1 in the synthetic independence design, Dantzig takes around 1500 s to compute the whole solution path, while Lasso takes less than one second. In the interest of brevity, we also exclude Dantzig in the presentation of results below.

No overall winner; large differences For all metrics, there is no one method that consistently performs best across all or the majority of the scenarios. Moreover, relative differences in performance can be large in some scenarios. Even in the textbook context of synthetic independence design scenarios shown in Fig. 1d, the median percentage relative decrease in PPV between the methods with the highest and lowest scores

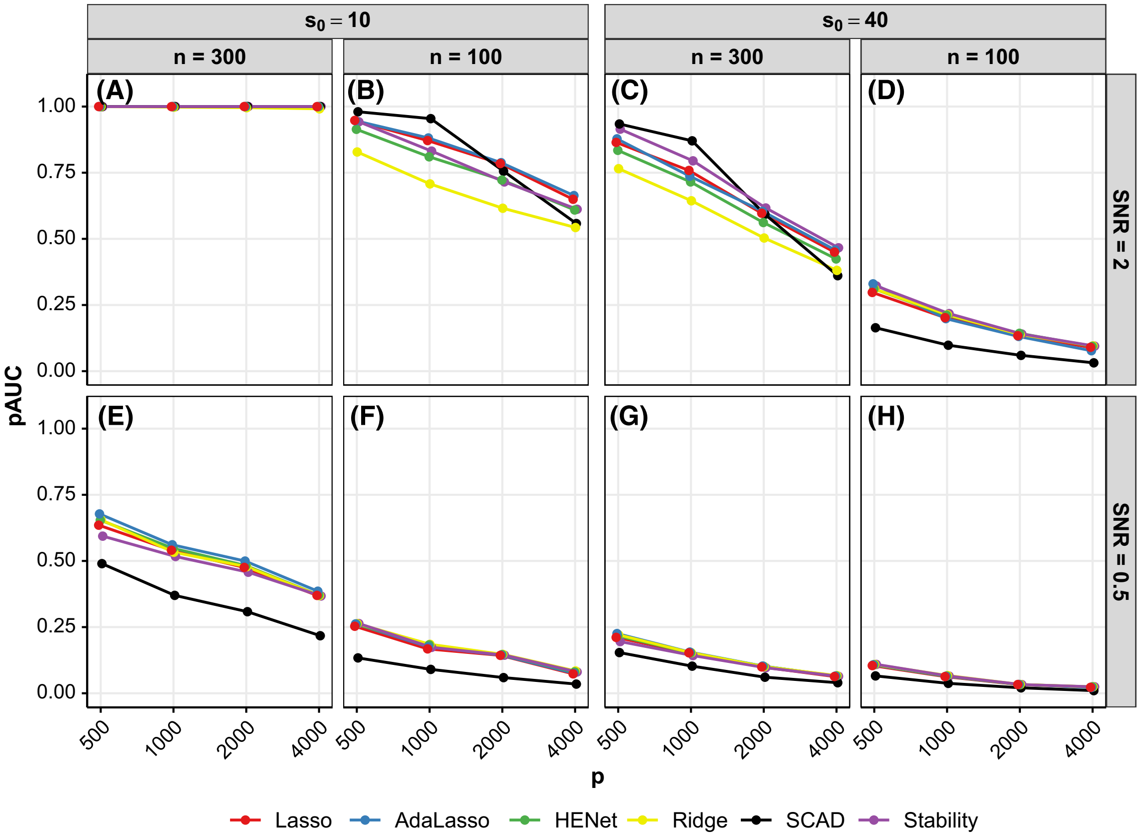


Fig. 2 Ranking performance (pAUC) versus p for a subset of synthetic independence design scenarios. Each panel represents a different combination of n, s_0 and SNR. Line color indicates method and x-axis is on a log scale. See also Figure S4

is 77%. Across all 2394 scenarios considered, the median percentage relative decrease is 46% for pAUC, 14% for RMSE, 61% for TPR and 68% for PPV.

3.2 Ranking

3.2.1 Independence design: synthetic data

Figure 2 shows ranking performance for a subset of independence design scenarios (see also Figure S4 where performance of pairs of methods is plotted against each other for all independence design scenarios).

SCAD transition in performance The performance of SCAD relative to other approaches varies substantially across scenarios. SCAD can offer the best performance in "easier" scenarios (e.g., Fig. 2b, black line), but does not retain this advantage as scenario difficulty increases. In particular, SCAD undergoes a transition from best to worst performing method with an unfavorable change in n, p, s_0 or SNR (see Fig. 2c for such a transition with increasing p).

An L_2 *penalty, AdaLasso and Stability Selection provide no substantive benefit over Lasso* Apart from SCAD in "easy" settings, none of the approaches perform notably better than Lasso (see Figs. 2 and S4). Moreover, Stability Selection, HENet and Ridge sometimes perform worse than Lasso (e.g., Fig. 2b). AdaLasso performs essentially the same as Lasso (Fig. S4), but can give small gains in pAUC over Lasso when SNR is small (see blue line in Fig. 2e for $p = 500$).

3.2.2 Correlation designs

For the semisynthetic data, we focus on the "high"-correlation design (see Sect. 2.3.2) because results for the "low"-correlation design are in good agreement with those from the synthetic independence design (see Figs. S5 and S6). This is because the covariates are very weakly correlated on average. (Mean absolute correlation coefficient between covariate pairs is 0.08.) Performance tends to be a bit worse for the "low"-correlation design than for the independence design (Fig. S5). We also note that, for ranking, AdaLasso typically performs slightly better than Lasso in the "low"-correlation design, whereas they mostly had equal performance in the independence design (Fig. S6).

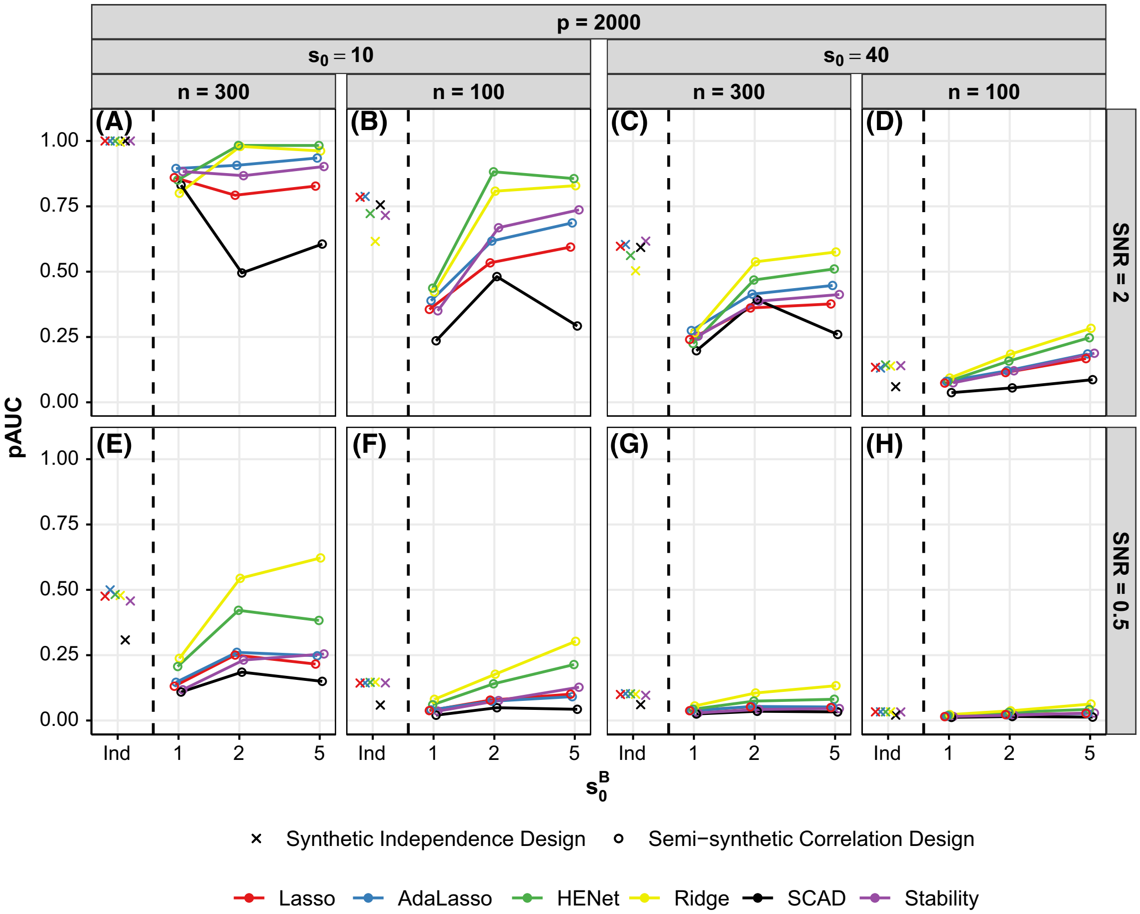


Fig. 3 Ranking performance (pAUC) versus s_0^B (number of signals per block) for a subset of semisynthetic "high"-correlation designs. Each panel represents a different combination of n, s_0 and SNR. All results shown are for $p = 2000$ (see Figure S7 for results with $p = 500$). For comparison, results for the corresponding independence design scenarios are also shown in each panel ("Ind"). Line color indicates method and x-axis is on a log scale

The semisynthetic and synthetic data results are broadly similar, so we focus on the semisynthetic data results and mainly use the synthetic data to investigate the impact of varying correlation strength ρ and block size p^B; these parameters were either fixed ($p^B = 10$) or not directly controlled (in the case of ρ) for the semisynthetic data.

Figure 3 shows ranking performance, as a function of number of signals per block s_0^B, for a subset of the "high"-correlation semisynthetic design scenarios with $p = 2000$. (analogous results for $p = 500$ are shown in Figure S7). Results for the synthetic independence design are also shown in each figure panel for reference (denoted by "Ind"). Figure 4 shows ranking performance, as a function of correlation strength ρ, for a subset of pairwise correlation synthetic design scenarios. To aid presentation of results, we fix $(n, p, s_0) = (200, 4000, 40)$ or $(200, 1000, 10)$ which give $r = 0.6$ ("hard") or $r = 2.9$ ("easy"), respectively, and also fix SNR = 1 (analogous results for SNR = 2 and 4 are shown in Figure S8).

Improved performance relative to the independence design for some scenarios Correlated covariates have a negative effect on ranking performance relative to the synthetic independence design when there is one signal per block (compare crosses with corresponding $s_0^B = 1$ circles in Figs. 3 and 4). Performance then often improves as s_0^B increases, particularly for HENet and Ridge Regression (e.g., yellow and green lines in Fig. 3b for semisynthetic data; contrast also the first and second columns in Fig. 4 for synthetic data). This can lead to an improvement in performance relative to the independence design when $s_0^B > 1$, with the largest improvements typically for HENet and Ridge in "harder" settings with small r or SNR. For example, in Fig. 3b where $r = 1.3$ and SNR = 2, HENet and Ridge have an increase in pAUC of 0.13 and 0.21, respectively relative to the independence design when $s_0^B = 5$.

For the synthetic data, we also find that an increase in block size p^B has an opposite effect to s_0^B, with a decrease in pAUC (contrast first and third columns in Fig. 4). Increasing

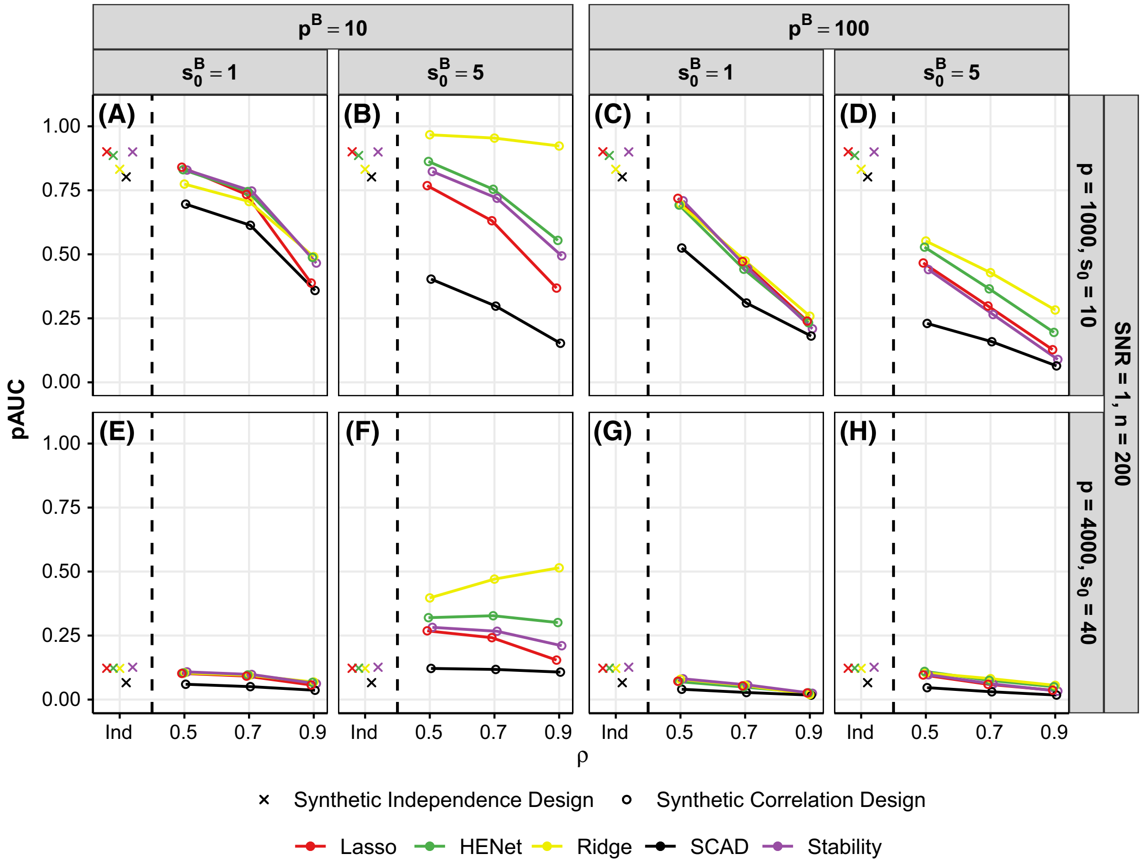


Fig. 4 Ranking performance (pAUC) versus ρ (correlation strength) for a subset of synthetic pairwise correlation designs. Each row represents a different combination of p and s_0, while each column represents a different combination of p^B (block size) and s_0^B (number of signals per block). All results shown are for SNR = 1 and $n = 200$ (see Figure S8 for SNR = 2 and 4). The top row has $(n, p, s_0) = (200, 1000, 10)$, giving $r = 2.9$, and the bottom row has $(n, p, s_0) = (200, 4000, 40)$, giving $r = 0.6$. For comparison, results for the corresponding independence design scenarios are also shown ("Ind"; these data points are identical across the panels in each row). Line color indicates method

correlation strength ρ typically has a detrimental effect. Only in the case of "harder" scenarios (small r or SNR) with small block size and several signals per block, performance can be enhanced by increasing ρ, most notably for Ridge Regression (see, e.g., yellow line in Fig. 4f where $r = 0.60$, $p^B = 10$ and $s_0^B = 5$).

Taken together, the above means that it is in "hard" scenarios when block size p^B is small and blocks consist of highly correlated variables of which several are active (i.e., large ρ and s_0^B) that we see the largest gains from correlation relative to the independence design, for HENet and Ridge (contrast yellow and green crosses and circles in Fig. 4f).

We also find that SCAD tends to be the most negatively affected by correlation (see, e.g., black in Fig. 3a).

HENet and Ridge Regression outperform other methods The positive influence of correlation on the ranking performance of HENet and Ridge Regression means that they now have the best pAUC scores in most scenarios with small block sizes and $s_0^B > 1$, with Ridge outperforming HENet. For example, for the semisynthetic data scenario in Fig. 3e where SNR = 0.5, Ridge substantially outperforms all other approaches when $s_0^B = 5$, with an improvement in pAUC of 0.24 over the second best method, HENet. HENet itself also improves over Stability Selection with a difference in pAUC of 0.13. There was no such benefit from an L_2 penalty in the corresponding independence design scenario (crosses in Fig. 3e).

We also observe in the small block size ($p^B = 10$) synthetic data results that the gains in pAUC from an L_2 penalty over Lasso become larger as correlation strength ρ increases (contrast yellow and red lines for $\rho = 0.5$ and $\rho = 0.9$ in Fig. 4b). These advantages from an L_2 penalty are either smaller or not present at all in the corresponding larger block size scenarios with $p^B = 100$ and $s_0^B > 1$ (fourth column in Fig. 4), suggesting that the *proportion* of covariates in a block that are signals is important. We investigated this by increasing s_0^B to 40 in the $p^B = 100$ scenarios shown in Fig. 4h (where $r = 0.60$, SNR = 1 and $s_0^B = 5$) and indeed found that salient improvements over Lasso are then obtained with an L_2 penalty: pAUC = 0.42, 0.13 and 0.07 for Ridge, HENet and Lasso, respectively when $\rho = 0.9$.

The largest benefits from an L_2 penalty are therefore for scenarios with small, highly correlated blocks with many signals per block. In general, benefits from an L_2 penalty appear to be more prevalent for the semisynthetic data than the synthetic data. This is likely due to the covariate correlation structure being less rigid for the semisynthetic data, with covariates being weakly correlated across blocks as opposed to independent.

SCAD transition in performance SCAD again displays its characteristic transition behavior with decreasing r or SNR in the correlation design (see, e.g., Fig. S7), but due to it typically being the most negatively affected by correlation, the number of "easy" scenarios where SCAD performs best is reduced.

SCAD's sensitivity to correlation means there can also be a transition with increasing s_0^B. In "easy" settings with large r or SNR, SCAD can perform best when there is only one signal per block (and also in the corresponding independence design), but perform worst when there are many signals per block (see, e.g., Fig. S7G). We also have a transition with increasing ρ for the synthetic data (e.g., Fig. S8D).

Stability selection and AdaLasso mostly outperform Lasso Stability Selection and AdaLasso remain competitive relative to Lasso, as in the independence design (Figs. 3, 4). Moreover, they now offer notable improvements over Lasso for some scenarios with sufficiently large s_0^B, and r or SNR (see, e.g., purple and blue lines vs. red line in Fig. 3b and purple vs. red line in Fig. 4b). However, they are usually outperformed by HENet and Ridge Regression, except in a few "easy" scenarios (e.g., $s_0^B = 1$ in Figs. 3a, S7C).

3.2.3 Summary and recommendations

For settings with uncorrelated or very weakly correlated covariates,[2] Lasso or AdaLasso are usually competitive for ranking and so can be considered as good choices. When one is confident of being in an "easy" scenario with sufficiently large r and SNR, SCAD could be considered here as it may perform notably better than Lasso and AdaLasso, but using SCAD carries more risk due to the high variability arising from its transition behavior.

For settings with more highly correlated covariates, we confirm that Ridge Regression is a good option since it outperforms or is competitive with the other approaches in most scenarios. Since SCAD rarely outperformed other methods and is very sensitive to changes in scenario properties, we would suggest it is not a good option for correlated settings.

3.3 Prediction

3.3.1 Independence design: synthetic data

Figure 5 shows predictive performance for a subset of independence design scenarios (see also Figure S9 where performance of pairs of methods is plotted against each other for all independence design scenarios).

An L_2 penalty and AdaLasso provide no substantive benefit over Lasso An L_2 penalty offers very little benefit for prediction, with Ridge performing substantially worse than all the other methods in many scenarios of moderate-to-large SNR (see, e.g., Fig. 5a). When SNR is small, HENet and Ridge perform similarly to Lasso (see, e.g., Fig. 5e–h). The exception is for small r scenarios, where small improvements in prediction error can be seen for HENet and Ridge relative to Lasso (see, e.g., $p = 500$ in Fig. 5d). AdaLasso performs similar to or worse than Lasso and performs particularly badly for smaller SNR, where it has the highest prediction error (see, e.g., blue line in Fig. 5e–h).

SCAD transition in performance SCAD has a similar transition property for prediction as for ranking (see above), but with the difference that SCAD does not become the worst performing method as scenario difficulty increases; Ridge or AdaLasso still performs worse (black line in Fig. 5c).

3.3.2 Correlation designs

For prediction performance in the "low"-correlation semisynthetic design, see Figure S10, where performance of pairs of methods is plotted against each other. Relative performance of methods agrees well with the synthetic independence design (Fig. S9).

Figure 6 shows predictive performance for a subset of the "high"-correlation semisynthetic design scenarios with $p = 2000$ (analogous results for $p = 500$ are shown in Figure S11), and Fig. 7 shows predictive performance for a subset of pairwise correlation synthetic design scenarios with SNR = 1 (analogous results for SNR = 2 and 4 are shown in Figure S12).

Performance improvements relative to the independence design when $s_0^B = 1$ Predictive performance worsens with increasing number of signals per block s_0^B (see Fig. 6), and this is primarily due to an increase in the variance of the response $\mathbf{y}$ as a result of the correlation between signals. For the same reason, in the synthetic data design, increasing correlation strength ρ leads to higher predictive error when blocks contain more than one signal (see second and fourth columns of Fig. 7).

[2] Here, we are assuming that uncorrelated variables are also independent, so that the independence design simulations apply. For very weak correlation, the semisynthetic "low"-correlation design applies.

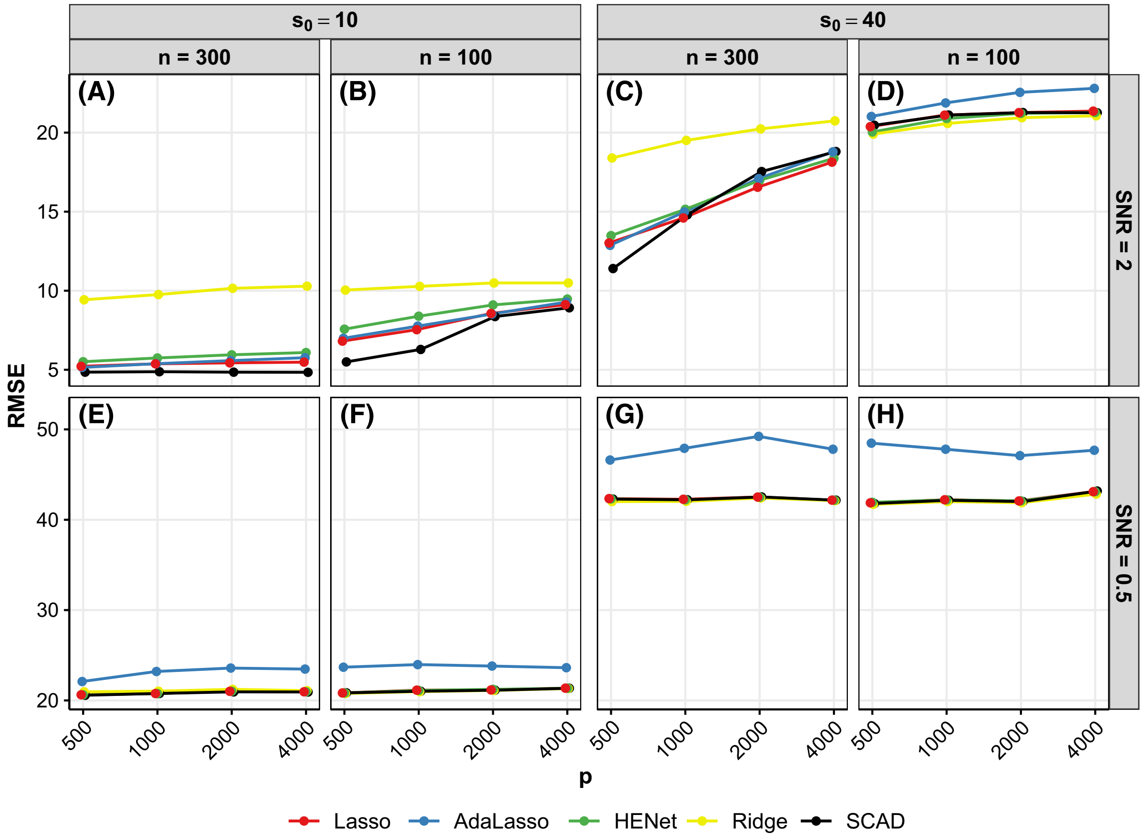


Fig. 5 Prediction performance (RMSE) versus p for a subset of synthetic independence design scenarios. Each panel represents a different combination of n, s_0 and SNR. Line color indicates method. Note that y-axis scales vary across rows. See also Figure S9

When there is one signal per block ($s_0^B = 1$), the signals are uncorrelated (or very weakly correlated for semisynthetic data) and so there is no or little increase in the variance of $\mathbf{y}$ relative to the independence design. The correlation between the signal and non-signals in each block can then result in a decrease in predictive error relative to the independence design, and we observe this for the semisynthetic data (compare crosses and $s_0^B = 1$ circles in Fig. 6d) and synthetic data (compare crosses and circles in Fig. 7g). For the latter, we also find that an increase in correlation strength ρ and increase in block size p^B lead to larger decreases in RMSE (compare Fig. 7e, g).

The method that shows the largest improvements relative to the independence design is typically Ridge Regression. For example, in Fig. 6a for $s_0^B = 1$, Ridge Regression (yellow circle) has a 25% decrease in RMSE relative to the independence design (yellow cross), while all other methods show little change in RMSE. Ridge regression may benefit the most because it has a non-sparse solution and, due to the correlation between signals and non-signals in each block, the correlated designs are also, in a sense, non-sparse.

An L_2 penalty and AdaLasso still provide no substantive gains over Lasso As for the independence design, Ridge and HENet do not substantively outperform the other approaches for prediction in any of the scenarios considered here, and this is the case even though Ridge often benefits the most from correlation (see above). In “easier” scenarios, Ridge still performs notably worse than other approaches (e.g., Fig. 6a), but in “hard” scenarios with small r, Ridge can marginally outperform other methods. For example, for the “hard” semisynthetic data scenario in Fig. 6d where $r = 0.3$ and SNR = 2, Ridge has a 7% decrease in RMSE relative to Lasso when $s_0^B = 1$. HENet also performs marginally better than Lasso in these “scenarios”, but typically marginally worse than Ridge (HENet has a 3% decrease in RMSE relative to Lasso in the above example). Similar behavior is observed for “hard” synthetic data scenarios, and this is particularly noticeable for large correlated blocks (yellow line in Fig. 7g).

AdaLasso remains similar to or, for small SNR, worse than the other approaches (see blue lines in Fig. 6).

SCAD transition in performance SCAD again shows transition behavior, offering modest gains over other methods when r and SNR are large, and s_0^B is small, but becoming worse than Lasso, HENet and sometimes Ridge as scenario difficulty, s_0^B or ρ increases. For example, SCAD performs

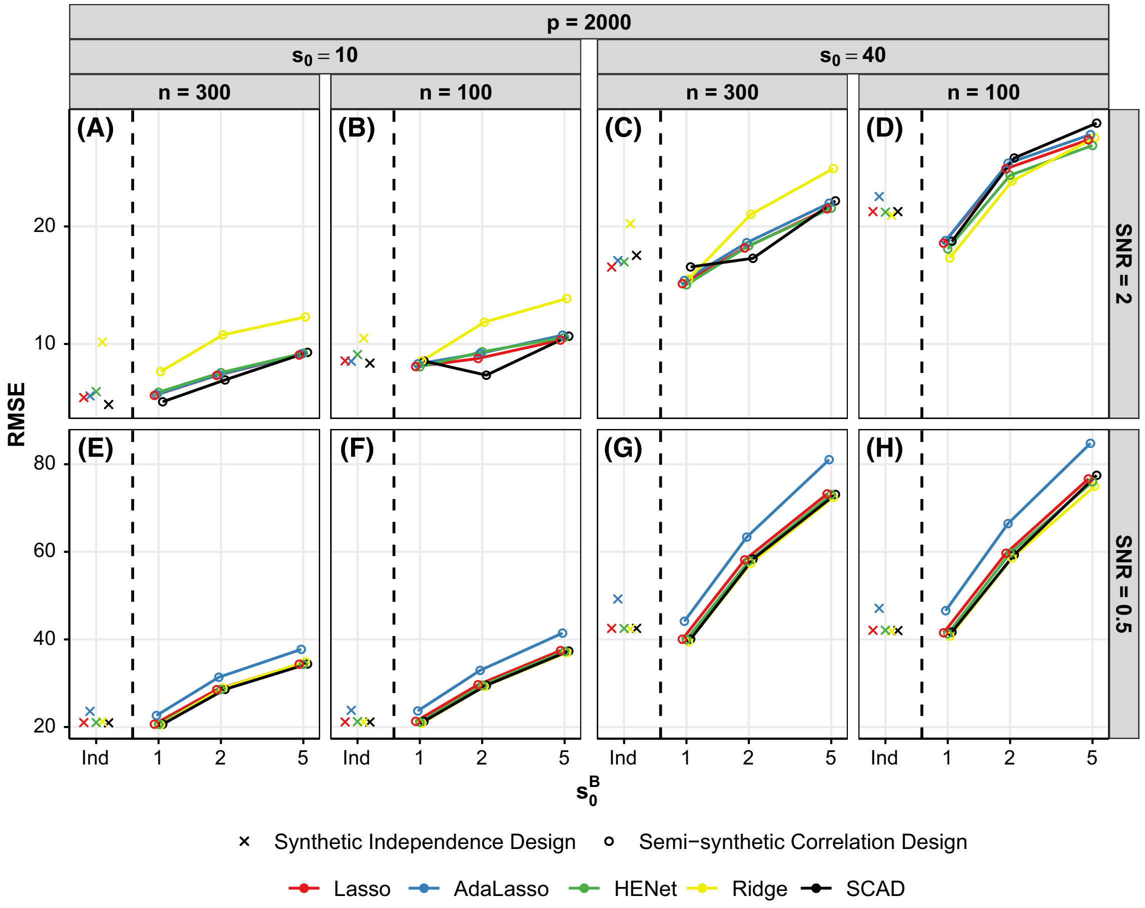


Fig. 6 Prediction performance (RMSE) versus s_0^B for a subset of semisynthetic “high”-correlation designs. Each panel represents a different combination of n, s_0 and SNR. All results shown are for $p = 2000$ (see Figure S11 for results with $p = 500$). For comparison, results for the corresponding independence design scenarios are also shown in each panel (“Ind”). Line color indicates method, x-axis is on a log scale and y-axis scales vary across rows

best when $r = 3.9$ and $s_0^B = 1$ (Fig. 6a), but worst when $r = 0.3$ and $s_0^B = 5$ (Fig. 6d).

3.3.3 Summary and recommendations

In settings with uncorrelated or very weakly correlated variables, predictive performance of methods relative to each other is mostly similar to that for ranking, so we make a similar recommendation: that is, use Lasso, or potentially SCAD if there is confidence that the scenario at hand is “easy”. The key difference from ranking is that we would not recommend AdaLasso because it can perform much worse than Lasso.

For more highly correlated settings, Lasso is mostly competitive and so can be considered a “safe” option. Ridge Regression may provide some small gains in “harder” scenarios, particularly for large correlated blocks, but can perform much worse than other approaches in “easier” settings. Therefore, HENet could be a good option here as it can still offer some gains over Lasso, but is not as sensitive to the scenario difficulty, remaining competitive where Ridge performs poorly. SCAD and AdaLasso may not be good options since they do not result in substantive benefits over Lasso or HENet and can both perform much worse than other methods in some scenarios.

3.4 Selection

3.4.1 Independence design: synthetic data

Figure 8 shows selection performance for a subset of independence design scenarios. See also Figures S13 and S14 where performance of pairs of methods is plotted against each other for all independence design scenarios.

Stability Selection or SCAD often best for PPV; trade-off between PPV and TPR All methods achieve optimal TPR=1 when r and SNR are sufficiently large, but can at the same time have substantial differences in terms of PPV (see, e.g., Fig. 8a; range of PPVs ≈ 0.1−0.6). SCAD typically offers

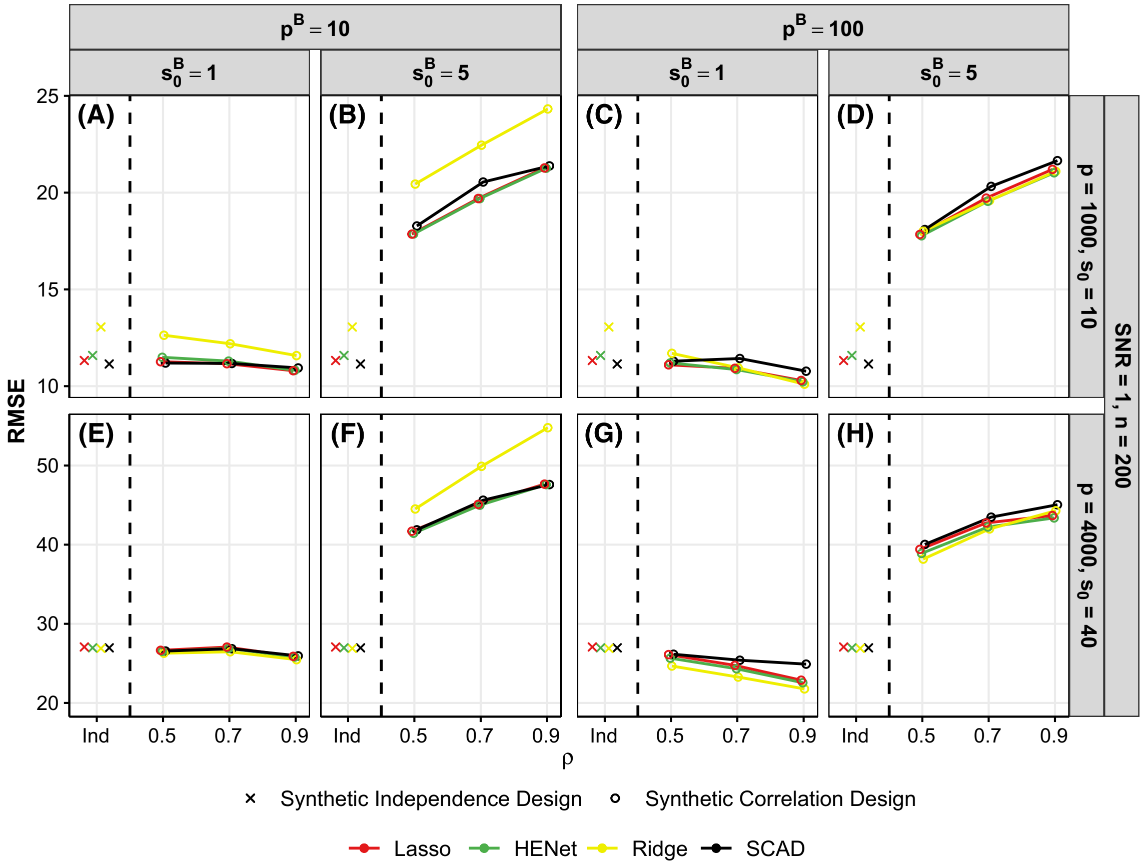


Fig. 7 Prediction performance (RMSE) versus ρ for a subset of synthetic pairwise correlation designs. Each row represents a different combination of p and s_0, while each column represents a different combination of p^B and s_0^B. All results shown are for SNR = 1 and $n = 200$ (see Figure S12 for SNR = 2 and 4). The top row has $(n, p, s_0) = (200, 1000, 10)$, giving $r = 2.9$, and the bottom row has $(n, p, s_0) = (200, 4000, 40)$, giving $r = 0.6$. For comparison, results for the corresponding independence design scenarios are also shown ("Ind"; these data points are identical across the panels in each row). Line color indicates method, and y-axis scales vary across rows

the best PPV in these "easiest" scenarios, followed by Stability Selection and AdaLasso.[3]

In scenarios where TPR is sub-optimal (small-to-moderate values of r or SNR), as could be expected, the relative performance of two methods typically follows the rule: If method A has a higher TPR than method B, then method A will have a lower PPV (see, e.g., Fig. 8d). For the majority of these scenarios, Stability Selection has the highest PPV and lowest TPR. SCAD performs similar to or better than Lasso, HENet and AdaLasso in terms of PPV, but similar or worse in terms of TPR (see, e.g., Figs. 8b–d, S13 and S14). Lasso, HENet and AdaLasso fail to obtain a PPV higher than 0.55 across all scenarios, contrasting with a maximum PPV greater than 0.8 for SCAD or Stability Selection. The range of PPVs across methods decreases as SNR decreases, and for SNR = 0.5, Stability Selection no longer has an advantage over the other approaches (Fig. 8e–h).

HENet and AdaLasso provide gains over Lasso for TPR There is a benefit of using an L_2 penalty or AdaLasso for TPR, but it comes at the cost of poorer false positive control. Across the majority of scenarios, HENet has small gains in TPR (of at most 0.1) over Lasso, but the converse is true for PPV (see, e.g., red and green lines in Fig. 8d). AdaLasso offers the highest TPR, particularly for small SNR where it provides large gains (of up to 0.35) over all the other approaches, but again its PPV suffers (see, e.g., blue lines in Fig. 8e).

3.4.2 Correlation designs

For selection performance in the "low"-correlation semisynthetic design, see Figures S15 and S16, where performance of pairs of methods is plotted against each other. Relative performance of methods agrees well with the synthetic independence design (Figs. S13 and S14).

Figure 9 shows selection performance for a subset of the "high"-correlation semisynthetic design scenarios with

[3] Note that this inferior performance of Stability Selection relative to SCAD could in part be due to the lack of false positive control in the implementation of Stability Selection used here.

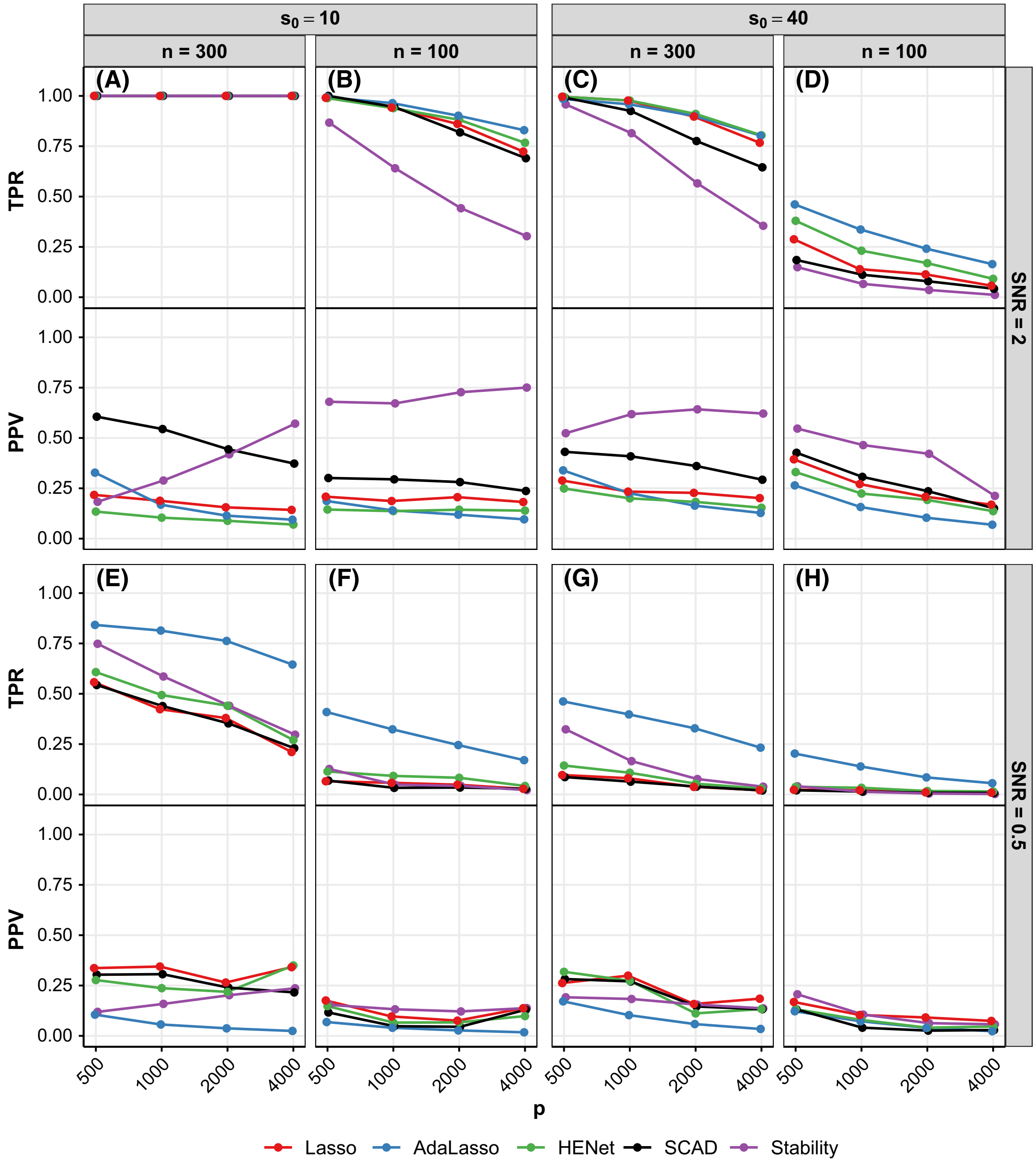


Fig. 8 Selection performance (TPR and PPV) versus p for a subset of synthetic independence design scenarios. Each panel shows TPR (top) and PPV (bottom) for a different combination of n, s_0 and SNR. Line color indicates method. See also Figures S13 and S14

$p = 2000$ (analogous results for $p = 500$ are shown in Figure S17), and Fig. 10 shows selection performance for a subset of pairwise correlation synthetic design scenarios with SNR = 1 (analogous results for SNR = 2 and 4 are shown in Figure S18).

Improved performance relative to the independence design for some scenarios The influence of correlation design parameters on selection performance is in line with that seen for ranking in Sect. 3.2.2. In particular, we find that the largest benefits from correlation relative to the independence design are again in "hard" scenarios (small SNR or r) with small blocks, strong correlation and several signals per block. For example, in Fig. 9d for semisynthetic data with $r = 0.3$, SNR = 2 and $s_0^B = 5$, all methods have increased TPR relative to the independence design, with the largest increase in TPR of 0.23 for HENet. Similarly, in Fig. 10f for synthetic data with $r = 0.6$, SNR = 1, $p^B = 10$, $s_0^B = 5$ and $\rho = 0.9$, the largest increase in TPR is 0.32, again for HENet. These increases in TPR do not necessarily come with decreases in PPV of corresponding magnitude; HENet has a similar PPV to the independence design.

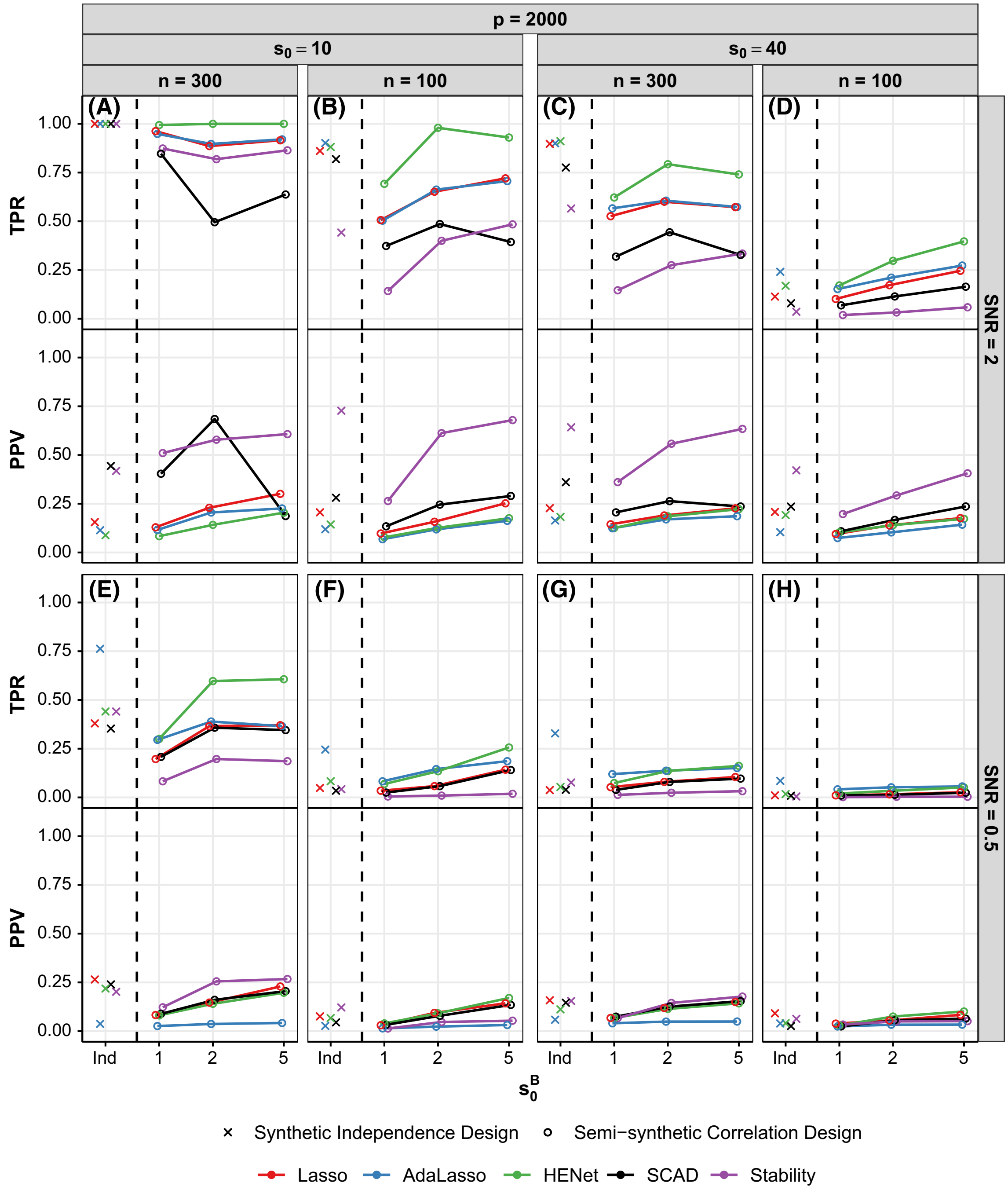


Fig. 9 Selection performance (TPR and PPV) versus s_0^B for a subset of semisynthetic "high"-correlation design scenarios. Each panel shows TPR (top) and PPV (bottom) for a different combination of n, s_0 and SNR. All results shown are for $p = 2000$ (see Figure S17 for results with $p = 500$). For comparison, results for the corresponding independence design scenarios are also shown in each panel ("Ind"). Line color indicates method and x-axis is on a log scale

HENet provides increased gains over Lasso for TPR, while also being competitive for PPV In the independence design, we found that HENet has small gains in TPR over Lasso, but has worse PPV. In the "hard", correlated scenarios described above where HENet benefits from correlation, HENet can give more substantial improvements in TPR over Lasso, while also remaining competitive in terms of PPV. In the semisynthetic data example from above (Fig. 9d), HENet has an increase in TPR of 0.15 relative to Lasso when $s_0^B = 5$; the corresponding increase for the independence design was 0.05. At the same time, PPV remains competitive at 0.17 for HENet and 0.18 for Lasso. This behavior is in line with Elas-

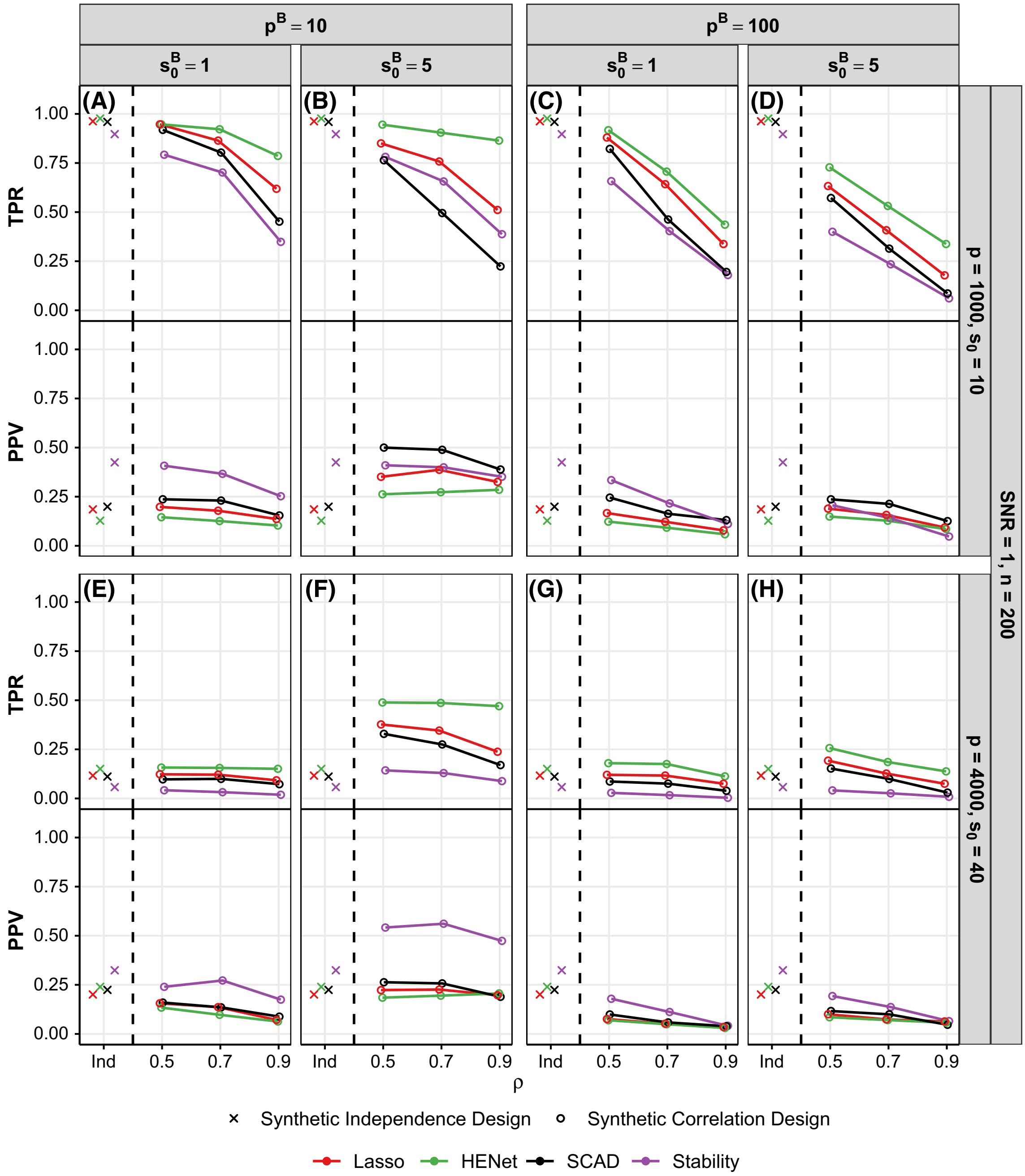


Fig. 10 Selection performance (TPR and PPV) versus ρ for a subset of synthetic pairwise correlation design scenarios. Each panel shows TPR (top) and PPV (bottom) for a different combination of p and s_0 (rows), and p^B and s_0^B (columns). All results shown are for SNR = 1 and $n = 200$ (see Figure S18 for SNR = 2 and 4). The top row has $(n, p, s_0) = (200, 1000, 10)$, giving $r = 2.9$, and the bottom row has $(n, p, s_0) = (200, 4000, 40)$, giving $r = 0.6$. For comparison, results for the corresponding independence design scenarios are also shown ("Ind"; these data points are identical across the panels in each row). Line color indicates method

tic Net enjoying the grouping effect property for correlated variables.

Stability Selection can be best for PPV, but is sensitive to correlation As for the independence design, Stability Selection typically performs best in terms of PPV, followed by SCAD, and they perform worse in terms of TPR (purple and black lines in Figs. 9 and 10). However, Stability Selection and SCAD are sensitive to correlation. For example, in the SNR = 0.5 semisynthetic data scenario with $s_0^B = 1$ shown in Fig. 9b, the substantial improvements in PPV provided by Stability Selection in the independence design are mostly lost. Also in line with the independence design, the advantage Stability Selection provides for PPV reduces as SNR decreases, with little to no advantage remaining for SNR = 0.5; here, all approaches have a similar performance, with AdaLasso typically performing worst (Fig. 9e–h).

AdaLasso no longer competitive for TPR AdaLasso offered the best performance for TPR in the independence design, but this is no longer the case as HENet has a similar or higher TPR, and AdaLasso is still not competitive for PPV (semisynthetic data; green and blue lines in Fig. 9).

3.4.3 Summary and recommendations

Since there is a trade-off between PPV and TPR, the best method to use depends on the aim. If the aim is primarily to have a low false positive rate, then Stability Selection is a good choice for both correlated and uncorrelated covariates, since it is likely to provide the best PPV. If the focus is more on maximizing the number of signals selected, then AdaLasso results in a TPR that dominates the other methods in most uncorrelated and very weakly correlated scenarios. However, it loses its advantage in more highly correlated designs, where HENet performs best. Lasso could be used to obtain a compromise between the two aims. If the scenario at hand is thought to be particularly “easy” with high r or SNR and covariates are uncorrelated or very weakly correlated, SCAD may provide the best PPV while retaining a competitive TPR.

4 Additional investigations

Below we extend the main simulations above in three directions. Section 4.1 investigates a synthetic data Toeplitz correlation design, Sect. 4.2 explores sensitivity of Stability Selection to its tuning parameters and Sect. 4.3 investigates the ability of methods to detect weak signals when coefficients are heterogeneous.

4.1 Toeplitz correlation design

We now consider method performance for synthetic data with a Toeplitz correlation design. This is as for pairwise correlation, but with covariates $\mathbf{x}_{j_1}$ and $\mathbf{x}_{j_2}$ within the same block having correlation $0.95^{|j_1-j_2|}$. We only consider block sizes of $p^B = 100$ that have two active variables per block, $s_0^B = 2$, with their positions, j_1' and j_2', within a block chosen such that $|j_1' - j_2'| = 7$, to give a correlation of $0.95^7 \approx 0.7$.

Figure 11 compares performance in the Toeplitz design against that in the corresponding pairwise correlation design ($\rho = 0.7$) for SNR $= 2$ and all possible combinations of n, p and s_0 (see Figs. S19 and S20 for SNR = 1 and SNR = 4, respectively). Performance is typically similar for the two designs or worse in the Toeplitz design. For prediction, Ridge Regression is most negatively affected by Toeplitz correlation, while SCAD is most affected for the other metrics.

On the one hand, the pairwise correlation design could be considered more difficult than the Toeplitz design because the average correlation between signals and non-signals (within a block) is higher for pairwise than for Toeplitz (0.7 vs. 0.19). However, on the other hand, the Toeplitz design could be considered more difficult because there are several non-signals that are more strongly correlated with the signals than the signals are with each other; for the pairwise correlation design, all signals and non-signals within a block are correlated with equal strength. The generally poorer performance observed for the Toeplitz design therefore suggests that having strongly correlated signals and non-signals is more detrimental than a higher average correlation.

Relative performance of methods in the Toeplitz design is generally consistent with that seen for the corresponding pairwise correlation design. For ranking, the impact of an L_2 penalty (relative to Lasso) is larger under the Toeplitz design than the pairwise design, with Ridge performing relatively well when SNR = 1, but poorly when SNR = 4.

4.2 Stability selection tuning parameters

Stability Selection has several tuning parameters: the subsample size $\tilde{n}$, an upper bound $\tilde{V}$ for $\mathbb{E}[V]$ (the expected number of false positives), and either a threshold π_{thr} on the selection probabilities or a set of regularization parameters to consider Λ (see Sect. 2.2). Making appropriate choices for these parameters is non-trivial. Here, we explore the effects of varying $\tilde{n}$, $\tilde{V}$ and π_{thr} on selection performance.

We simulated data (as described in Sect. 2.3) with SNR = 2, $n = 200$, $p = 1000$ and $s_0 = 10$ or 20 (giving $r = 2.90$ or 1.45 respectively) for the independence design, and the pairwise correlation design with $p^B = 10$, $s_0^B = 2$ and $\rho = 0.7$. We applied Stability Selection with all possible combinations of the following tuning parameter values: $\tilde{V} \in \{1, 5, 10, 15, 20\}$, $\pi_{\text{thr}} \in \{0.6, 0.9\}$ and $\tilde{n} = \lfloor n\gamma \rfloor$ where $\gamma \in \{0.4, 0.5, 0.6, 0.7\}$ is the subsample proportion.

Figure 12 shows that, in general, as $\tilde{V}$ or γ increases, or π_{thr} decreases, the number of selected variables increases, resulting in higher TPR, but lower PPV. An exception is for $s_0 = 20$, where, for the most conservative choices of the parameters ($\gamma = 0.4$, $\tilde{V} = 1$ and $\pi_{\text{thr}} = 0.9$), in addition to a very poor TPR, PPV is also low on average (see solid line, $\gamma = 0.4$ in Fig. 12g, h). Here, selection is too stringent and the majority of signals are missed. When the underlying model size is smaller ($s_0 = 10$), the most conservative parameter choices are again sub-optimal in terms of performance (Fig. 12e, f), but the same is also true for the least conservative choices ($\gamma = 0.7$,$\tilde{V} = 20$ and $\pi_{\text{thr}} = 0.6$; Fig. 12a, b). However, in the scenarios considered here, being too strin-

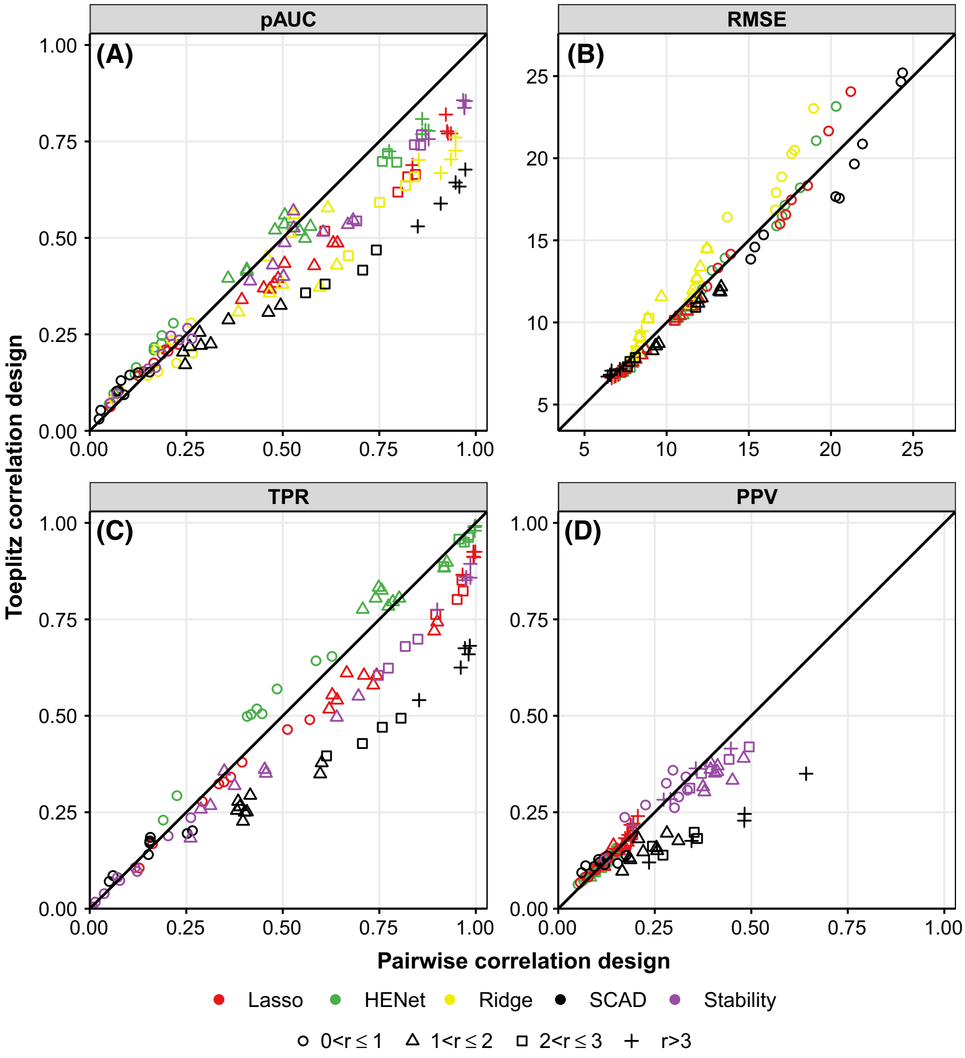


Fig. 11 Comparison between Toeplitz correlation and pairwise correlation designs for ranking, prediction and selection performance. Performance in the Toeplitz correlation design is plotted against performance in the corresponding pairwise correlation design with $\rho = 0.7, s_0^B = 2$ and $p^B = 100$. Each point corresponds to a method (indicated by color) and a single (n, p, s_0) triplet. (The resulting value of the rescaled sample size r is indicated by symbol.) Results shown are for SNR = 2 (see Figs. S19 and S20 for SNR = 1 and SNR = 4) and are averages over 64 replicates

gent seems to have a more deleterious effect on performance than being too lenient.

Results from the main simulations, where we set $\tilde{n} = \lfloor 0.632n \rfloor$, $\pi_{\text{thr}} = 0.6$ and had no explicit false positive control $\tilde{V}$ (i.e., the full range of regularization parameters Λ was considered; see Sect. 2.4), are indicated by crosses in Fig. 12a–d. Performance in the main simulations is most similar to that of the largest $\tilde{V}$ considered here ($\tilde{V} = 20$), but with better TPR and worse PPV (except for $s_0 = 10$ where TPR is already optimal and so there is only a decrease in PPV).

4.3 Heterogeneous coefficients

In the main simulations, all nonzero coefficients were assigned the same value. Here, we consider detection of signals with heterogeneous coefficients for three methods: Lasso, HENet and SCAD. We simulated data (for the independence design) as described in Sect. 2, except instead of s_0 active variables all having coefficient 3, half of them had coefficient β' and the other half had coefficient $c_\beta \beta'$ where $c_\beta \in [0, 1]$. We chose $\beta' = \sqrt{18/(1 + c_\beta{}^2)}$ such that with fixed SNR, $\mathbb{E}(\sigma^2)$ remains the same as in the homogeneous β's case. Note that $c_\beta = 1$ gives the main simulation setup with homogeneous coefficients. Informed by the main simulations, we set $n = 300$, $s_0 = 40$, $p = 4000$ and SNR = 2 or 4, guaranteeing that when nonzero coefficients all take the same value, we are in a relatively "easy" scenario where the majority of the signals can be detected.

Figure 13 shows that as c_β decreases, signals with smaller coefficients are less likely to be detected, resulting in a decrease in TPR. All methods fail to detect the very weak signals when $c_\beta = 0.1$ (i.e., only the stronger 50% of the signals are detected giving TPR≈0.5). Consistent with the main simulations, SCAD has better false positive control (higher PPV) than Lasso and Elastic Net when SNR is large, and this is especially the case when c_β is near 0.1 or 1 (contrast black

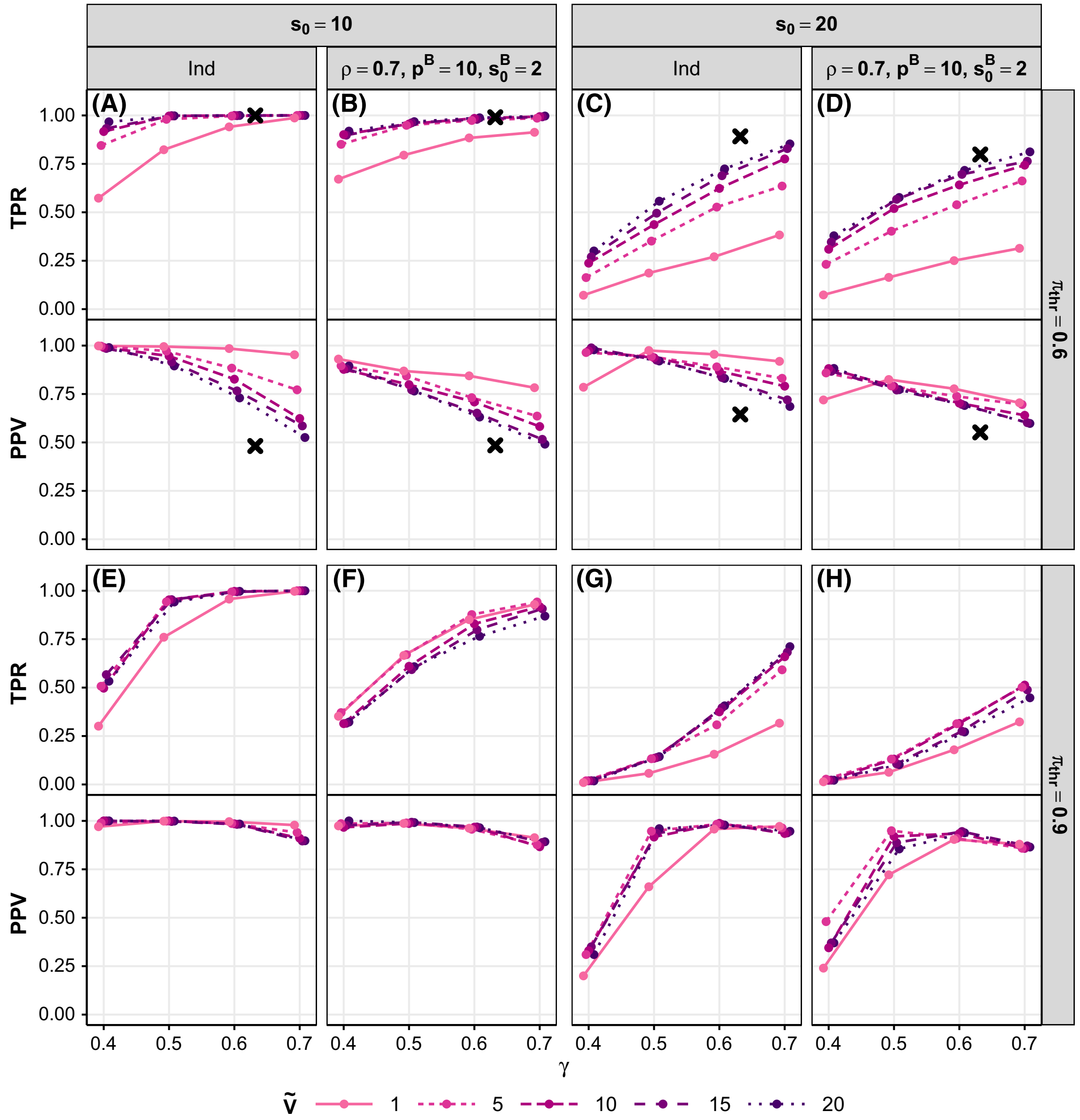


Fig. 12 Stability Selection tuning parameter sensitivity. Each panel shows TPR (top) and PPV (bottom) versus subsample proportion γ. Within each panel, line type indicates $\tilde{V}$, the upper bound for the expected number of false positives. Top row of panels corresponds to threshold probability $\pi_{\text{thr}} = 0.6$ and bottom row to $\pi_{\text{thr}} = 0.9$. Each column corresponds to a different simulation scenario: the synthetic independence design ("Ind") with $n = 200$, $p = 1000$, SNR = 2, and $s_0 = 10$ or 20, or the corresponding synthetic pairwise correlation design scenarios with $\rho = 0.7$, $p^B = 10$ and $s_0^B = 2$. Black crosses in the top row of panels show performance observed in the main simulations where $\pi_{\text{thr}} = 0.6$, $\gamma = 0.632$ and there was no explicit false positive control $\tilde{V}$. Results are averages over 100 replicates

line with red and green lines for PPV in Fig. 13b). The "U" shape of the SCAD PPV curve here is likely due to the fact that bias is largest when c_β is moderate, which leads to selection of more variables to compensate. (SCAD is known to be nearly unbiased for strong signals; for large c_β, all signals are relatively strong, while for small c_β the $s_0/2$ weaker signals have such a small influence that the underlying model is well approximated by a model with $s_0/2$ strong signals and no weak signals.) In contrast, Lasso and Elastic Net are biased estimators, so their PPVs are not as affected. SCAD also seems to have higher power to detect the weaker signals when SNR is large and c_β is moderate (see TPR in Fig. 13b). However, as observed in the main simulations, SCAD is more sensitive to SNR and so is less competitive in "harder" scenarios (SNR = 2; Fig. 13a). Lasso has higher PPV than HENet, and this is largely unaffected by changes in c_β. Differences in TPR between HENet and Lasso decrease as c_β decreases, until they both have a similar performance for $c_\beta = 0.1$. (Note that which method performs best depends on SNR.)

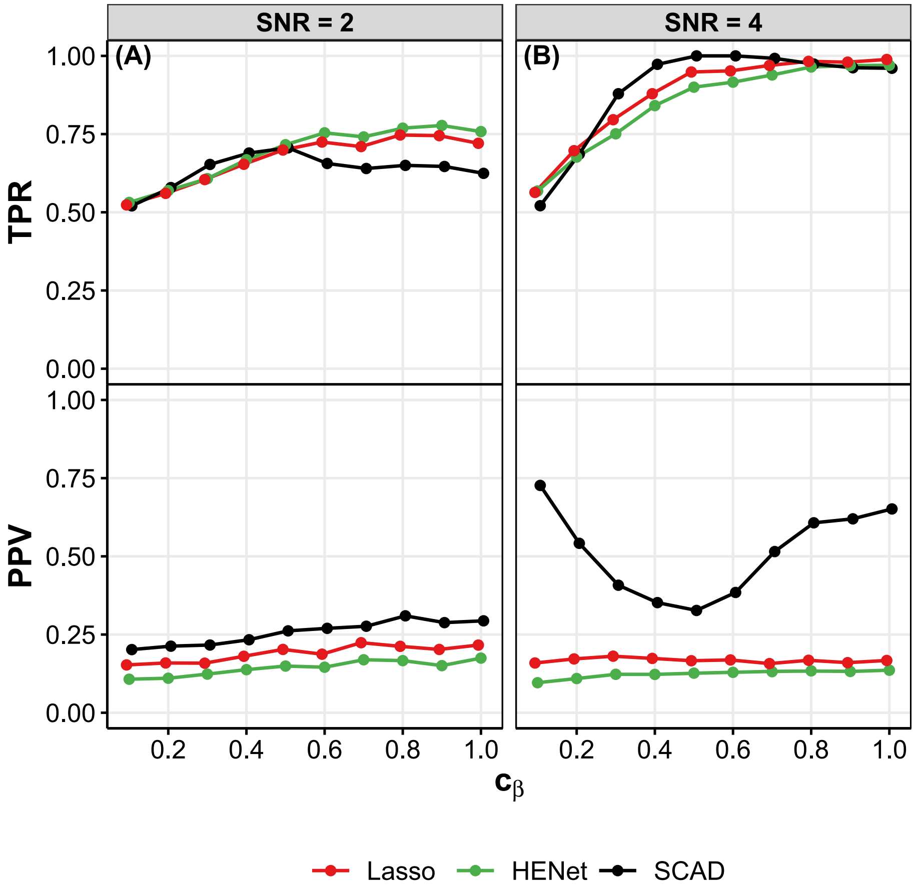


Fig. 13 Influence of heterogeneous regression coefficients on selection performance. TPR (solid lines) and PPV (dotted lines) are plotted against the coefficient scaling factor c_β for the independence design with $(n, p, s_0) = (300, 4000, 40)$ and SNR = 2 (**a**) or SNR = 4 (**b**). In the data-generating linear model, half of the signals have coefficient β' and the other half have coefficient $c_\beta \beta'$ (see text for details). Note that $c_\beta = 1$ gives the main simulation setup with homogeneous coefficients. Line color indicates method. Results are averages over 50 replicates

5 Discussion

Our results complement theory by shedding light on the finite-sample relative performance of methods. Many of our results do align with available theory. For instance, SCAD is known to have nearly unbiased estimates for coefficients that are large (relative to noise), explaining why it tends to have better selection performance in "easy" scenarios. However, some conditions of theoretical results (asymptotic or finite-sample) can be hard to verify in practice, and the results do not directly provide insight into the performance of a method relative to others, making it difficult to pick a suitable approach in any given finite-sample setting. Our results suggest that there is no one method which clearly dominates others in all scenarios, even in the relatively narrow set of possibilities considered here (e.g., we did not consider heavy-tailed noise, non-sparsity, non-block-type covariance etc.). Relative performance depends on many factors and also on the specific metric(s) of interest.

A challenge of translating results of our empirical study into practice is that not all of the factors will be known to a user in a given setting, specifically those that are related to the unknown signals (e.g., s_0). However, domain knowledge may provide some indication as to, for example, whether SNR is likely to be high or low, or as to the likely number of signals, which could then give an idea of the "difficulty" of the problem. Nevertheless, with the above caveats, we have been able to make some general observations that in turn have allowed some broad recommendations to be made (see Sects. 3.2.3, 3.3.3 and 3.4.3). These recommendations are primarily based on covariate correlation and focus on which approach is most likely to perform well across a broad range of scenarios. The synthetic independence design and semisynthetic "low"-correlation design resulted in similar method performance, so we have made a single, joint recommendation for uncorrelated and very weakly correlated scenarios, for each metric. For example, for ranking we have recommended Lasso or AdaLasso for uncorrelated or very weakly correlated covariates, and Ridge Regression when variables are more strongly correlated. We have also highlighted when a method may be a risky choice. For example, SCAD is double-edged, dominating in "easier" scenarios but deteriorating rapidly when conditions become difficult. Therefore, its high variability means that it should only be used when one is sure that the scenario at hand is very "easy". Six out of the seven approaches considered in our study have been recommended for at least one of the goals. (Further to the above, we recommended Stability Selection for PPV and Elastic Net for TPR in correlated settings.) The Dantzig Selector is not recommended in any setting, since it is usually similar or worse than Lasso and is more computationally expensive.

The overall average correlation between pairs of covariates is weak in all correlated designs (due to the block structure for synthetic data and reflecting the correlation in the real data set for the semisynthetic data). However, despite this weak average correlation, we have found that method performance in the synthetic pairwise correlation design and semisynthetic "high"-correlation design can still differ greatly to performance in the synthetic independence design (or semisynthetic "low"-correlation design). This is because an important factor for method performance is the magnitude of correlation between signals, or between signals and non-signals.

For prediction, where we mostly recommended Lasso, Ridge does particularly badly in many "easier" scenarios, but it is worth pointing out that most scenarios considered here were unfriendly to Ridge in the sense of being highly sparse, and with low overall correlation (across all predictors). In many areas such as biomedicine, signals can be weak and so SNR may be at the low end of the values considered here, or possibly even smaller. In such difficult settings, Ridge may be a good option and our results indeed suggest this, as the only scenarios where we saw any benefit of an L_2 penalty for prediction were those with small r and SNR.

We focused on simulations from the sparse linear model to better understand the variability of performance in a broadly favorable setting. Extending this systematic empirical approach to (the huge range of) less favorable settings, spanning many kinds of model mis-specification, could be illuminating, but experimental design would be nontrivial. As one example, we revisited a "low"-correlation scenario from the semisynthetic data analysis, but with a non-Gaussian error distribution. Figure S21 shows method performance for all metrics and provides details of data generation. Method performance deteriorates as non-normality increases. SCAD is the most affected and mirrors its previous behavior, with a transition in performance from best to worst as non-normality increases for ranking and prediction.

Our comparison focused on seven popular penalized linear regression methods, but there are of course many others that have been proposed, and some of these are also well known. For example, there are relatively well-known extensions of Lasso that have been proposed for data where covariates can be grouped (Group Lasso; Yuan and Lin 2006) or ordered (Fused Lasso; Tibshirani et al. 2005). While, for reasons of tractability, our comparison was restricted to seven methods, we make our simulation code and method performance data available, allowing users to add in other approaches of interest into the comparison without the need to regenerate the results for the seven methods considered here.

Choices of tuning parameters can be crucial. In line with known results, we saw that standard cross-validation often yielded overly large models for Lasso and Elastic Net. An interesting alternative is proposed in Lim and Yu (2016), where cross-validation is based on an estimation stability metric. Compared to traditional cross-validation, this approach significantly reduces the false positive rate while slightly sacrificing the true positive rate and achieves similar prediction but higher accuracy in parameter estimation. For Stability Selection, in Zou (2010) the author points out that there is no established lower bound for the expected number of true positives, and the tuning parameters π_{thr} and $\tilde{V}$ have significant influences on the true positive rate. They also found in their simulation study that the number of false positives is usually smaller than the specified $\tilde{V}$. This suggests that less stringent $\tilde{V}$ can help improve signal detection without sacrificing false positive control too much, thus providing a better balance between the two. This is reflected in our results in Sect. 4.2.

We explicitly defined the true model in terms of exact sparsity (i.e., some coefficients being precisely zero). Although this is the best studied case, in practice such a notion of sparsity may not be realistic and a more reasonable assumption may be that there are a few strong signals, several moderate signals and even more weak signals, but the majority of variables are irrelevant with small, but sometimes nonzero coefficients. In this case, since it may not be possible to find all relevant variables, a good method might be expected to detect all strong and moderate signals while removing the weaker ones. In this vein, Zhang and Huang (2008) consider the problem where weak signals exist outside the ideal model, such that their total signal strength is below a certain level. The authors prove that the Lasso estimate has model size of the correct order, and the selection bias is controlled by the weak signal coefficients and a threshold bias.

Due to the comprehensive nature of our simulation study, we focused on summarizing the predominant trends and relationships across the scenarios. There will always be some scenarios which are exceptions to these summaries, but this in itself motivates the need for extensive simulation studies. If a simulation study has limited scope, then the derived conclusions may not generalize beyond the few scenarios considered. So while such studies may be useful in exploring and understanding the properties of a method, they may have limited practical implications for an end user. In contrast, a large-scale simulation study, such as the one presented here, can reveal which approaches perform well across a broad range of scenarios. These approaches may then translate into being a good or "safe" choice for the user's setting. In addition, the study can offer some insight as to whether certain methods are best avoided, because they have high variability across scenarios in the study.

Acknowledgements This work was supported by the UK Medical Research Council (University Unit Programme Numbers MC_UU_00002/2 and MC_UU_00002/10).

Code and data availability All analysis was performed in R (R Core Team 2018). Scripts for generating the main simulation datasets, applying the regression methods, assessing performance and plotting results are available at https://github.com/fw307/high_dimensional_regression_comparison, together with performance metric data from the main simulation.

References

Bickel, P.J., Ritov, Y., Tsybakov, A.B.: Simultaneous analysis of Lasso and Dantzig selector. Ann. Stat. **37**, 1705–1732 (2009)

Bondell, H.D., Reich, B.J.: Consistent high-dimensional Bayesian variable selection via penalized credible regions. J. Am. Stat. Assoc. **107**, 1610–1624 (2012)

Breheny, P., Huang, J.: Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann. Appl. Stat. **5**, 232–253 (2011)

Bühlmann, P., van de Geer, S.: Statistics for High-Dimensional Data: Methods, Theory and Applications, 1st edn. Springer, Berlin (2011)

Bühlmann, P., Mandozzi, J.: High-dimensional variable screening and bias in subsequent inference, with an empirical comparison. Comput. Stat. **29**, 407–430 (2014)

Candes, E., Tao, T.: The Dantzig selector: statistical estimation when p is much larger than n. Ann. Stat. **35**, 2313–2351 (2007)

Celeux, G., Anbari, M.E., Marin, J.M., Robert, C.P.: Regularization in regression: comparing Bayesian and frequentist methods in a poorly informative situation. Bayesian Anal. **7**, 477–502 (2012)

Efron, B., Hastie, T., Tibshirani, R.: Discussion: The Dantzig selector: statistical estimation when p is much larger than n. Ann. Stat. **35**, 2358–2364 (2007)

Fan, J., Li, R.: Variable selection via nonconcave penalized likelihood and its oracle properties. J. Am. Stat. Assoc. **96**, 1348–1360 (2001)

Fan, J., Lv, J.: A selective overview of variable selection in high dimensional feature space. Stat. Sin. **20**, 101–148 (2010)

Fan, J., Peng, H., et al.: Nonconcave penalized likelihood with a diverging number of parameters. Ann. Stat. **32**, 928–961 (2004)

Friedman, J.H., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. **33**, 1–22 (2010)

Hastie, T., Tibshirani, R., Tibshirani, R.J.: Extended Comparisons of Best Subset Selection, Forward Stepwise Selection, and the Lasso (2017). arXiv preprint arXiv:1707.08692

Hoerl, A.E., Kennard, R.W.: Ridge regression: biased estimation for nonorthogonal problems. Technometrics **12**, 55–67 (1970)

James, G.M., Radchenko, P., Lv, J.: DASSO: connections between the Dantzig selector and Lasso. J. R. Stat. Soc. Ser. B **71**, 127–142 (2009)

Li, X., Zhao, T., Yuan, X., Liu, H.: The flare package for high dimensional linear regression and precision matrix estimation in R. J. Mach. Learn. Res. **16**, 553–557 (2015)

Lim, C., Yu, B.: Estimation stability with cross-validation (ESCV). J. Comput. Graph. Stat. **25**, 464–492 (2016)

Meinshausen, N., Bühlmann, P.: High-dimensional graphs and variable selection with the lasso. Ann. Stat. **34**, 1436–1462 (2006)

Meinshausen, N., Bühlmann, P.: Stability selection. J. R. Stat. Soc. Ser. B **72**, 417–473 (2010)

Meinshausen, N., Rocha, G., Yu, B.: Discussion: a tale of three cousins: Lasso, L2 Boosting and Dantzig. Ann. Stat. **35**, 2373–2384 (2007)

Perrakis, K., Mukherjee, S.: The Alzheimer's Disease Neuroimaging Initiative: Scalable Bayesian regression in high dimensions with multiple data sources. J. Comput. Graph. Stat. Adv. (2019). https://doi.org/10.1080/10618600.2019.1624294

R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2018)

Sill, M., Hielscher, T., Becker, N., Zucknick, M.: c060: extended inference with lasso and elastic-net regularized cox and generalized linear models. J. Stat. Softw. **62**, 1–22 (2014)

The Cancer Genome Atlas Research Network: Integrated genomic analyses of ovarian carcinoma. Nature **474**, 609–615 (2011)

Tibshirani, R.: Regression shrinkage and selection via the Lasso. J. R. Stat. Soc. **58**, 267–288 (1996)

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., Knight, K.: Sparsity and smoothness via the fused lasso. J. R. Stat. Soc. Ser. B **67**, 91–108 (2005)

Tucker, S.L., Gharpure, K., Herbrich, S.M., Unruh, A.K., Nick, A.M., Crane, E.K., Coleman, R.L., Guenthoer, J., Dalton, H.J., Wu, S.Y., Rupaimoole, R., Lopez-Berestein, G., Ozpolat, B., Ivan, C., Hu, W., Baggerly, K.A., Sood, A.K.: Molecular biomarkers of residual disease after surgical debulking of high-grade serous ovarian cancer. Clin. Cancer Res. **20**, 3280–3288 (2014)

Wainwright, M.J.: Sharp thresholds for high-dimensional and noisy sparsity recovery using L1-constrained quadratic programming (Lasso). IEEE Trans. Inf. Theory **55**, 2183–2202 (2009)

Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. J. R. Stat. Soc. Ser. B **68**, 49–67 (2006)

Zhang, C.H., Huang, J.: The sparsity and bias of the Lasso selection in high-dimensional linear regression. Ann. Stat. **36**, 1567–1594 (2008)

Zhao, P., Yu, B.: On model selection consistency of lasso. J. Mach. Learn. Res. **7**, 2541–2563 (2006)

Zou, H.: The adaptive lasso and its oracle properties. J. Am. Stat. Assoc. **101**, 3133–3164 (2006)

Zou, H.: Discussion of "Stability selection" by Nicolai Meinshausen and Peter Buhlmann. J. R. Stat. Soc. Ser. B **72**, 468 (2010)

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