Computational Statistics (2024) 39:1241–1280
https://doi.org/10.1007/s00180-023-01342-8

ORIGINAL PAPER

Predictive stability criteria for penalty selection in linear models

Dean Dustin[1] · Bertrand Clarke[1] · Jennifer Clarke[1]

Received: 26 August 2022 / Accepted: 25 February 2023 / Published online: 16 March 2023

Abstract

Choosing a shrinkage method can be done by selecting a penalty from a list of prespecified penalties or by constructing a penalty based on the data. If a list of penalties for a class of linear models is given, we introduce a predictive stability criterion based on data perturbation to select a shrinkage method from the list. Simulation studies show that our predictive method identifies shrinkage methods that usually agree with existing literature and help explain heuristically when a given shrinkage method can be expected to perform well. If the preference is to construct a penalty customized for a given problem, then we propose a technique based on genetic algorithms, again using a predictive criterion. We find that, in general, a custom penalty never performs worse than any commonly used penalties and there are cases the custom penalty reduces to a recognizable penalty. Since penalty selection is mathematically equivalent to prior selection, our method also constructs priors. Our methodology allows us to observe that the oracle property typically holds for penalties that satisfy basic regularity conditions and therefore is not restrictive enough to play a direct role in penalty selection. In addition, our methodology, can be immediately applied to real data problems, and permits us to take model mis-specification into account.

Keywords Prediction · Penalized regression · Shrinkage · Oracle property · Penalty selection · Prior selection · Genetic algorithm · Evolutionary computation

✉ Bertrand Clarke
bclarke3@unl.edu

Dean Dustin
ddustin8@huskers.unl.edu

Jennifer Clarke
jclarke3@unl.edu

[1] Department of Statistics, University of Nebraska, Lincoln, 340 Hardin Hall North, PO Box 830963, Lincoln, NE 68583-0963, USA

1 Shrinkage and prediction

In the context of linear models, inference problems in which the number of parameters p is bigger than the sample size n, i.e., with $p > n$, are ill-posed and require some form of regularization to be solved. The earliest form of this is called Tikhonov regularization and was used initially for matrix inversion. In Statistics, ridge regression (RR) is probably the first occurrence of Tikhonov regularization, see Hoerl (1962). By the early 1990s, L^2 regularization was common in neural networks contexts, see Sjöburg and Ljung (1992), not only to ensure that a solution existed but also to reduce variance. An important step forward was replacing the L^2 penalty with an L^1 penalty, see Tibshirani (1996). This shrinkage method is called the least absolute shrinkage and selection operator (LASSO). It provides a form of regularization that does variable selection as well as variance reduction while ensuring solutions exist. The elastic net (EN), Zou and Hastie (2005), was introduced as a compromise between RR and LASSO. It uses both an L^1 penalty and an L^2 penalty; RR and LASSO are special cases of EN. Another insight was the concept of an 'oracle property' (OP) first proved for a penalty called the smoothly clipped absolute deviation (SCAD) in the context of linear models; see Fan and Li (2001). The OP meant that, as $n \to \infty$, the parameter estimates from the SCAD penalty behaves as if the correct regression parameters were known, i.e., the parameter estimates either were consistent, asymptotically normal, and efficient or went to zero according to whether the variables they multiplied were or were not in the true model.

Over the last two decades, numerous shrinkage methods have been proposed and studied as individual methods, representing individual penalties or priors, for the purposes of parameter inference. Chief amongst these is Wang et al. (2020a) who focused exclusively on the "high sparsity" case; see also Bühlmann and Mandozzi (2014). High sparsity describes the situation where the number of true parameters, p_0, is small relative to p. Formally high sparsity means $\frac{p_0}{p} = o(1)$. Within this case, Wang et al. (2020a) generated over 2300 specific scenarios and examined how well various shrinkage methods performed in three senses, including root mean square predictive error (RMSPE) on a hold out set. RMSPE is the closest of their criteria to the predictive stability evaluation we advocate here. The benefit of RMSPE over the other performance metrics is that it does not require knowledge of the true model. In short, their main recommendations under RMSPE were to use LASSO, or potentially SCAD, and to avoid using adaptive LASSO in "easy" scenarios. For "harder" scenarios they recommend LASSO, RR, or EN. Our closest results below recommend RR and EN in "easy" scenarios[1] but we note LASSO is nearly as good. For "harder" scenarios, e.g., tridiagonal covariance structure and 90% sparsity, we identify RR and EN again with LASSO nearly as good. Overall, we find reasonable agreement for sufficiently comparable scenarios. Differences in recommendations can be explained partially by the fact that Wang et al. (2020a) uses RR to estimate adaptive weights and we use $\sqrt{n}$-consistent estimators. In addition, we are using

[1] See Table A1 in Appendix A, the line labeled (Ind, Li).

empirical predictive stability only, and this differs mathematically and conceptually from RMSPE. Throughout this paper we have emphasized giving the clearest recommendations possible and and usage of our techniques for data analysis rather than simply exploring the performance of shrinkage methods via simulations.

More recent work examining the performance of shrinkage techniques includes Hastie et al. (2020) who argue in favor of their "relaxed LASSO" that combines LASSO with OLS estimates and in some scenarios performs well compared to best subsets regression. In Celeux et al. (2012), the authors noted that although purely Bayesian methods appear to be more parsimonious than shrinkage methods in terms of variable selection, there is little difference predictively between them. They also state that shrinkage methods can be expected to perform better than purely Bayesian methods predictively because shrinkage methods usually minimize cross-validation predictive error in their implementation. Most recently, Wang et al. (2020b) compares variable selection techniques such as shrinkage methods in an economic setting with various dependence structures. Unlike earlier work, they, too, adopt a predictive criterion. Then they argue that shrinkage methods generally perform better than other methods such as factor analysis, which is commonly used in econometrics.

Like other authors, our goal is to choose a penalty (or prior) from a class of penalties (or priors). One way to do this is to consider a list of shrinkage methods, find a method for comparing them, and choose the best. Alternatively, one can take a 'build-your-own' approach, construct a shrinkage method from part of the data and use it on the rest of the data. Here, we provide both a comparison of 'off the shelf' methods and a build-your-own technique (based on genetic algorithms) under a predictive optimality criterion. The build-your-own shrinkage methods never perform worse that the off-the-shelf methods, but may return an off-the shelf method as optimal. We give an example of each case.

More formally, write a linear model (LM) of the form

$$Y = X\beta + \epsilon$$

where $Y = (Y_1, \dots, Y_n)^T$, X is an $n \times p$ design matrix, $\beta = (\beta_1, \dots, \beta_p)^T$ is a p-dimensional parameter, and $\epsilon = (\epsilon_1, \dots, \epsilon_n)^T$ is random error $\epsilon_i \sim N(0, \sigma^2)$ for some $\sigma > 0$ and assume we have a data set

$$\mathcal{D}_n = \{(y_i, x_i) \mid i = 1, \dots, n\}.$$

A nonadaptive shrinkage method gives parameter estimates

$$\hat{\beta} = \arg\min_{\lambda,\beta} \sum_{i=1}^{n} L_1(y_i - x_i\beta) + \lambda \sum_{j=1}^{p} L_2(\beta_j) \tag{1.1}$$

where L_1 and L_2 are loss functions, x_i is the i-th row of X, and λ is the decay parameter. The term shrinkage arises from the fact that as $\lambda \to \infty$, each $\beta_j \to 0$. Sometimes the first term on the right hand side of (1.1) is replaced by a log-likelihood; for the normal likelihood, L_1 corresponds to squared error. In addition, taking $L_2(\beta_j) = \beta_j^2$ error leads to RR.

In (1.1) all coordinates of β are penalized by a single factor λ. By contrast, adaptive shrinkage methods allow for different shrinkages on the coordinates of β. An adaptive shrinkage method typically gives estimates of the form

$$\hat{\beta} = \arg\min_{\lambda, w^p, \beta} \sum_{i=1}^{n} L_1(y_i - x_i\beta) + \lambda \sum_{j=1}^{p} w_j L_2(\beta_j) \tag{1.2}$$

where $w^p = (w_1, \dots, w_p)^T$ and the w_j's are weights on the individual β_j's. Again, the first term on the right hand side of (1.2) may be replaced by a log-likelihood. Also, the dependence on w_j in the second term may be more complicated; we have represented the adaptivity of the constraint to the data as multiplicative in the penalty term for simplicity, but more general forms of adaptivity are possible. For instance, we have chosen $w_jL_2(\beta_j)$, but one could also construct an adaptive penalty of the form $L_2(w_j, \beta_j)$. We regard the SCAD and minimax concave penalties (MCP) (Zhang (2010)) as adaptive because they are data driven even though they only introduce one extra non-multiplicative parameter (and have the OP). Thus adaptivity is not simply parameter counting. Moreover, EN introduces two parameters and is not adaptive while the adaptive EN (AEN) introduces p parameters has the OP; see Zou and Zhang (2009).

As written, expression (1.2) introduces p new parameters, the w_j's, that must be estimated. Two main estimation techniques have been proposed to obtain the $\hat{w}_j$'s. One, due to Zou (2006) (see also Wang et al. 2007), is to set $\hat{w}_j = \frac{1}{|\hat{\beta}_j|^\gamma}$ where $\hat{\beta}_j$ is any $\sqrt{n}$-consistent estimator of β_j, e.g., from SCAD or the ordinary least squares (OLS) estimator when n is large enough, and choose γ by a cross-validation criterion. The justification for this choice is given in remark 2 of Zou (2006)—as $n \to \infty$, the estimated weights for the zero coefficient β_j's tend to infinity and the weights for non-zero coefficient β_j's tend to a constant. This allows for asymptotically unbiased estimates. Another method, due to Qian and Yang (2013), sets $\hat{w}_j = \frac{SE(\hat{\beta}_{j,OLS})^\gamma}{|\hat{\beta}_{j,OLS}|^\gamma}$. This method seems to work well when there is high collinearity; see Qian and Yang (2013). In practice, for both methods, $\gamma = 1$ is used to avoid extra computation. Here, we have exclusively used the Zou (2006) method since it does not require $n > p$ and has a nice interpretation: As $\hat{\beta}_j \to 0$, $\hat{w}_j \to \infty$ forcing $\beta_j = 0$ in (1.2).

Formally, the OP has two components, consistency in variable selection, and asymptotic normality of the estimates. Write $\beta = (\beta_1, \beta_2)$ where β_2 represents the zero components of β. Under various conditions, $\sqrt{n}$-consistent local minimizers $\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2)$ from certain shrinkage criteria (such as SCAD) satisfy the following two properties:

1. $P(\hat{\beta}_2 = 0) \to 1$, and
2. $\sqrt{n}J(\beta_1, 0)(\hat{\beta}_1 - \beta_1) \xrightarrow{D} N(0, J(\beta_1, 0))$

where J denotes the Fisher information matrix.

Roughly, shrinkage methods segregate into those that are nonadaptive, i.e., introduce exactly one 'decay' parameter and usually do not satisfy the OP, and those that are adaptive, introduce two or more decay parameters, and often satisfy the OP. We will see that, contrary to initial impressions, the oracle property is not rare.

We note that (1.1) corresponds to a joint density ρ on the data and β. Indeed, exponentiating gives that $\hat{\beta}$ corresponds to the mode of the posterior

$$\rho(\beta \mid Y^n) \propto e^{-\lambda \sum_{j=1}^{p} L_2(\beta_j)} e^{-\sum_{i=1}^{n} L_1(y_i - x_i^T \beta)}$$

in which L_2 defines a prior on β with hyperparameter λ. A similar manipulation can be applied to (1.2). This means that penalty selection is mathematically equivalent to prior selection. However, the equivalence is only mathematical because the class of reasonable penalties is a proper subset of the class of reasonable priors. In particular, many reasonable penalties are convex—our method in Sect. 5.1 assumes convexity, for instance—but priors do not have to be log-convex. Hastie et al. (2020) suggests that non-convex penalties may be preferable for variable selection, and that convex penalties can be seen primarily as variance reduction techniques, although they can still be effective in variable selection.

Even though shrinkage methods were originally introduced as a way to solve the $n < p$ problem, the OP uses $n \to \infty$. This is partially ameliorated by results that give analogs of the OP where p increases much faster than n does; see Fan and Lv (2013). However, the cost of amerlioration is often artificial conditions on the parameter space and/or design matrix. Moreover, many of the original examples given to verify that shrinkage methods were effective actually had $n > p$ and took $p = 8$, see Fan and Li (2001), Zou (2006), and Wang et al. (2007).

Our comparison is in terms of predictive stability and accuracy of model selection. That is, after finding $\hat{\beta}$ for a given shrinkage method we define the predictor

$$\hat{Y}(x) = x^T \hat{\beta} \tag{1.3}$$

for $Y(x)$ at some new value x. Then we evaluate how well $\hat{Y}$ predicts when the data are perturbed. We perturb the data using the technique of Luo et al. (2006). The idea is to add $N(0, \tau^2)$ noise to the y_i's in $\mathcal{D}$ and then use part of the data to form a predictor and the rest of the data to evaluate the predictor. We do this by generating 'instability' curves, basically L^2 predictive errors as a function of τ. A good predictor will have smooth instability curves with low values that increase slowly with τ. We also use more conventional accuracy measures for variable selection similar to the measures used in Wang et al. (2020a) for simulations and semi-synthetic data settings. We argue that pairing the two provides an assessment that captures analogues of both variance (from the instability curves) and bias (from the accuracy measures). We regard instability as more important than variable selection because even if a variable is incorrectly included its contribution may be small if its coefficient is near zero and if it is incorrectly excluded the bias should show up in the instability curve.

Since our overall approach is predictive and requires no knowledge of a true model to implement, it is a technique for data analysis suitable for real data sets not just for simulation studies. In particular, it allows us to consider the effects of model mis-specification as in Sect. 4. This contrasts the applicability of Wang et al. (2020a) as the authors note in Sect. 5 that their simulation studies do not provide a method for choosing a shrinkage method when the true model is unknown.

The main contributions of this paper are:

1. The use of predictive stability as a general model selection technique. Here, we have used it to choose among many shrinkage methods. Predicgtive stabnility as a criterion verifiably works as it should in simulations, and can be directly used in real data settings when the true model is unknown; see Sects. 3 and 4.
2. We observe that the OP is more common than most practitioners seem to realize; see Sect. 2. So, we needn't limit ourselves to the specific shrinkage methods that have been studied. Hence when $n > p$, we search a general class of shrinkage methods that have the OP. However, when $n < p$ we enlarge the class of shrinkage methods to include some that do not have the OP. This larger set may give us stronger optimality; see Sect. 5.1. In either case, we generate penalties (or priors) that are optimal for a given data set.
3. The use of Genetic Algorithms (GAs) with part of the data to form a prior that we then apply predictively. By optimizing over the prior in this way we are effectively optimizing over the penalty and hence choosing the optimal shrinkage method. This optimum may or may not coincide with an established shrinkage method but will still have the OP. Thus, we have chosen our shrinkage method to be predictively optimal for our data. We verify in examples that this is predictively better than simply using all the data to make predictions.

The structure of this paper is as follows. Section 2 gives general conditions for the OP to hold for a wide range of adaptive penalties. We present our comparisons of existing shrinkage methods in Sect. 3 and confirm out method agrees with existing literature when the true model is known. We illustrate the use of our method for a real data set in Sect. 4. In Sect. 5, we present our GA optimization verifying that the theory in Sect. 2 holds and the result of the GA is optimal given the data. We summarize our overall findings and intuition in Sect. 6.

2 Theoretical results

Our first two results are oracle properties for penalized log-likelihoods and empirical risk settings under relatively standard regularity conditions. Our second pair of results are oracle properties where we have relocated the penalties on log-likelihoods and empirical risks at $\sqrt{n}$-consistent estimators.

Our proofs for these four results are motivated by the techniques in Fan and Li (2001) and Wang et al. (2007). We assume an adaptive setting and allow different penalty functions on different parameters. This is different from Fan and Lv (2013) who did not treat different penalty functions on different parameters or adaptivity. Their main result concerned the asymptotic equivalence of penalized methods and permitted p to increase with n. Like other papers that allow p to increase with n, some of the conditions appear artificial. For example, aside from truncations of the parameter space, one must assume that there is a sequence of explanatory variables such that if one of the early variables is correct and accidentally not included it can be reconstructed from later explanatory variables in the sequence, thereby sacrificing identifiability. Our result, like many others, assumes either fixed p or p increasing so slowly with n that the required convergences hold.

2.1 General penalized log likelihood

The linear model can be written as

$$Y_i = x_i^T \beta + \epsilon_i, \quad i = 1, \ldots, n, \tag{2.1}$$

Assume that $\beta_j \neq 0$ for $j \leq p_0$ and $\beta_j = 0$ for $p_0 < j \leq p$ for some $p_0 \geq 0$. Without loss of generality, we write the true vector of regression coefficients as $\beta = (\beta_1, \beta_2)' = (\beta_1, 0)'$. Here, we regard the $\underline{x_i'}s$ as deterministic. When needed we write $\beta \in \Omega \subset \mathbb{R}^p$ where Ω is open and $\Omega = (\overline{\Omega})^0$.

Let (x_i, Y_i) for $i = 1, \ldots, n$ each have density $p(Y_i|x_i, \beta)$ (with respect to a fixed dominating measure) such that the six regularity conditions stated below are satisfied. Let $L(\beta|x^n)$ be the log-likelihood function of the observations $(x_1, Y_1), \ldots, (x_n, Y_n)$ and denote the penalized log-likelihood objective function as

$$Q(\beta) = L(\beta|x^n) + n \sum_{i=1}^{n} \lambda_j f_j(\beta_j).$$

In Sect. 5.2.2, we use the extra generality of allowing different f_j's for different β_j's. Here we write x^n to mean $x_1, \ldots, x_n$ for ease of notation. Recall $\lambda_j = w_j \lambda$ and define

$$\lambda^*_{\max} = \max\{w_j\lambda \,:\, j = 1, \ldots, p_0\},$$
$$\lambda_{\min} = \min\{w_j\lambda \,:\, j = p_0 + 1, \ldots, p\}.$$

When estimating the $\lambda^*_{\max}$ and $\lambda_{\min}$ we must consider the ordering of these two quantites. Note that as the sample size increases, $\hat{\lambda}^*_{\max} < \hat{\lambda}_{\min}$; see Remark 2 in Zou (2006). To help see why, note that the max and min are over different sets.

Now we state six regularity conditions required for our first result.

Condition 1 Each Fisher information matrix

$$J(\beta|x_i) = -E\left(\frac{\partial^2}{\partial^2 \beta} \ln \; p(Y_i|x_i, \beta) \right)$$

exists and is positive semi-definite uniformly in i. In addition, $\exists B > 0$ so that $BI_{p_0 \times p_0} \geq I(\beta|x_i) \geq \frac{1}{B} I_{p_0 \times p_0}$ uniformly in i where $I_{p_0 \times p_0}$ is the $p_0 \times p_0$ Fisher information matrix for the non-zero β_j's.

Condition 2 Assume there exists an $\epsilon > 0$ so that for $\eta > 0$ small enough,

$$E\left[\sup_{\beta\in\mathcal{B}(\beta_0,\eta)}\left|\frac{\partial^2}{\partial\beta_j\partial\beta_\ell}\,L(\beta|x_i)\right|^{1+\epsilon}\right]<\infty,$$

where $\mathcal{B}(\beta_0,\eta)$ is the Euclidean ball centered at β_0 with radius $\eta>0$. Also, assume the log likelihood has a convergent second order Taylor expansion. That is, for all $j,\ell=1,\dots,p$, we have $\forall\beta\in\Omega$ and $\forall x_i$ that, as $\eta\to 0$,

$$E\left[\sup_{\beta\in\mathcal{B}(\beta_0,\eta)}\left|\frac{\partial^2}{\partial\beta_j\partial\beta_\ell}\,L(\beta|x_i)-J_{j,\ell}(\beta_0|x_i)\right|\right]\to 0.$$

Condition 3 For any ϵ, $\exists N$ such that $\forall n\geq N$, $\exists I(\beta|x^\infty)$ positive semi-definite so that

$$\sup_\beta\left|\frac{1}{n}\sum_{i=1}^{n}J(\beta|x_i)-J(\beta|x^\infty)\right|<\epsilon.$$

We write $J_1(\beta_1)=J_1(\beta_1|x^\infty)$ to mean the information matrix for β_1 only.

Condition 4 There exists an increasing sequence of compact sets $C=C_n$ in the parameter space and constants $M=M_n\in\mathbb{R}^+$ such that for all n, $\sup_{\beta_j\in C_n}|f_j'(\beta_j)|\leq M_n$. That is, the first derivative of the penalty term is uniformly bounded on compact sets.

Condition 5 The penalty function satisfies $f_j(0)=0$ and $f_j(\beta_j)>0$ for $\beta_j\neq 0$.

Condition 6 The penalty function, f_j, is uniformly Taylor expandable when $\beta_{j0}\neq 0$. That is, for $h\in\mathbb{R}$ and $\beta_{j0}\neq 0$, $f_j(\beta_{j0}+h)-f_j(\beta_{j0})=f_j'(\beta_{j0})h+o_j(1)$ uniformly in j i.e., $\sup_j o_j(1)\to 0$.

Remark Condition 6 requires f_j to be differentiable at every point except 0. This condition is reasonable because of the Condition 5. Since $f_j(0)=0$, we do not need to worry about the derivative at 0. We discuss the consequences of allowing $f_j'(0)=0$ in Sect. 2.3.

Our main result generalizes the class of oracle procedures to a penalized log likelihood with an arbitrary penalty function that satisfies mild conditions. By using arbitrary penalty functions, our result shows that the class of methods that have the OP contains an infinite dimensional vector space of functions. Our proof technique is modified from Fan and Li (2001) and Wang et al. (2007).

Theorem 2.1 (Oracle Property) *Assume Conditions* 1–6 *hold. Suppose* $\sqrt{n}\lambda^*_{\max}\to 0$ *and* $\sqrt{n}\lambda_{\min}\to\infty$ *Then, the estimator* $\hat\beta=(\hat\beta_1,\hat\beta_2)^T$ *satisfies*

1. $P(\hat\beta_2=0)\to 1$, *and*

2. $\sqrt{n}(J_1(\beta_1|x^\infty))^{\frac{1}{2}}(\hat{\beta}_1 - \beta_1) \xrightarrow{D} \mathcal{N}(0, I_{p_0})$

where I_{p_0} is the $p_0 \times p_0$ identity matrix.

For proof see Appendix B1.

Therefore, we can construct an oracle procedure by choosing any $f_j(\beta_j)$'s that satisfy Conditions 4 and 5. We can choose a single penalty function $f(\beta_j) = f_j(\beta_j)$ for all j, or we can choose an individual f_j for each β_j.Thus, we treat different β_j's differently and retain the OP.

2.2 Penalized empirical risk

We now extend the OP results for penalized likelihoods to penalized empirical risks. This is a more general setting because we allow a larger class of loss functions on the data.

Consider the same regression scenario as Sect. 2.1 and a distance $d(y_i - x_i^T\beta)$. Let

$$R(\beta|x^n) = \frac{1}{n}\sum_{i=1}^{n} d(y_i - x_i^T\beta)$$

be the empirical risk of the observations $(x_1, Y_1), \dots, (x_n, Y_n)$ and denote the penalized empirical risk objective function by

$$Q(\beta) = R(\beta|x^n) + n\sum_{j=1}^{p} \lambda_j f_j(\beta_j).$$

We introduce one more condition for empirical risks.

Condition 7 Let

$$J^*(\beta|x_i) = -E\left(\frac{\partial^2}{\partial^2\beta}R(\beta|x_i)\right).$$

The empirical risk R satisfies for some $\epsilon > 0$

$$E\left[\sup_{\beta\in\mathcal{B}(\beta_0,\eta)}\left|\frac{\partial^2}{\partial\beta_j\partial\beta_\ell}\ R(\beta|x_i)\right|^{1+\epsilon}\right] < \infty,$$

and has a convergent second order Taylor expansion. That is, for all $j, \ell = 1, \dots, p$ we have that $\forall\beta \in \Omega$ and as $\eta \to 0$,

$$E\left[\sup_{\beta\in\mathcal{B}(\beta_0,\eta)}\left|\frac{\partial^2}{\partial\beta_j\partial\beta_\ell}\frac{1}{n}\sum_{i=1}^{n} R(\beta|x_i) - J^*(\beta|x^\infty)_{j,\ell}\right|\right] \to 0$$

and $\exists B > 0$ so that $BI_{p_0 \times p_0} \geq J(\beta|x_i) \geq \frac{1}{B} I_{p_0 \times p_0}$ uniformly in i. Hence, $J^*(\beta|x^\infty)$ is positive semi-definite and we abbreviate it to $J^*(\beta)$.

We see that $J^*(\beta) = J^*(\beta|x^\infty)$ is an analog to the Fisher information matrix but for an empirical risk rather than a log-likelihood.

The distance function we use in the empirical risk must satisfy some regularity conditions as well. Namely, we must use an even distance function with a unique minimum at 0. This ensures the expectation is 0, as shown in the following lemma.

Lemma 1 *Let $d(u)$ be an even distance function with a unique minimum at* 0. *If u comes from some distribution with pdf $f_U(u)$ that is symmetric about zero with support* $[-a, a]$*for $a \in \mathbb{R}$, then $E_U[d'(u)] = 0$.*

Proof Since $d(u)$ is an even function, $d'(u)$ is an odd function. By definition,

$$E_U(d'(u)) = \int_{-a}^{a} d'(u) f_U(u) du.$$

Since $f_U(u)$ is symmetric, $f_U(u) = f_U(-u)$, i.e., $f_U(u)$ is even. Let $g(u) = d'(u)f_U(u)$. Then $g(u)$ is an odd function, and we have $\int_{-a}^{a} g(u)du = 0$, so $E_U(d'(u)) = \lim_{a \to 0} \int_{-a}^{a} g(u)du = 0$. □

Note that Lemma 1 is true for any even distance function $d(u)$, and this is useful for us because we focus on the distance function $d(y_i - x_i^T \beta)$ which is even due to the symmetry of $(y_i - x_i^T \beta)$.

Next we present a result analogous to Theorem 2.1 but we replace the log-likelihood with an empirical risk. We see that under mild regularity conditions, the OP holds for penalized empirical risks.

Theorem 2.2 (Oracle Property) *Assume Conditions* 4–7 *and Lemma* 2 *in Appendix B*2 *hold and suppose $\sqrt{n}\lambda^*_{\max} \to 0$ and $\sqrt{n}\lambda_{\min} \to \infty$. Then the estimator $\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2)^T$ satisfies*

1. $P(\hat{\beta}_2 = 0) \to 1$, *and*
2. $\sqrt{n}(J_1^*(\beta_1|x^\infty))^{\frac{1}{2}}(\hat{\beta}_1 - \beta_1) \xrightarrow{D} \mathcal{N}(0, I_{p_0})$,

where I_{p_0} is the $p_0 \times p_0$ identity matrix.

For proof see Appendix B2.

Taken together, Theorems 2.1 and 2.2 give us insight into how large the class of oracle procedures is. Previously it seemed oracle procedures were rare, isolated choices of priors. Now, even though we have not characterized the class of all oracle procedures, we can see that the conditions for a procedure to have the oracle property are quite general, allowing a large range of likelihoods, distances, and penalties (or priors).

Given the fact that many shrinkage methods are equivalent asymptotically, providing a method for choosing a shrinkage method in finite sample problems is essential. Hence, since there are infinitely many oracle procedures, two key questions arise. For a given data set, which shrinkage method should we use? Also, when it is desirable to use a method without the OP? We answer both these questions using simulations. We propose choosing a 'best' shrinkage method (which may or may not satisfy the OP) by optimizing a stability criterion over both a class of penalty functions and a class of distance functions. Allowing $f_j(\beta_j)$ to be different for each parameter, as long as they are uniformly Taylor expandable and have similar properties, is powerful because we can choose variable-dependent penalties. This is a more general sense of adaptivity than each β_j merely having its own shrinkage parameter λ_j.

In Sect. 3, we study popular shrinkage methods, many of which satisfy the conditions of our Theorems. For instance, ALASSO and AEN satisfy the conditions for Theorem 2.1. Other penalties such as SCAD and MCP also have the OP, but they are not special cases of our results, emphasizing the fact that the class of methods that have the OP is large.

2.3 Parameter specific locations for the penalty

Let $\hat{\beta}^*$ be a $\sqrt{n}$-consistent estimator of β. To take advantage of the fact that shrinkage methods can set $\hat{\beta}_j$'s to zero, it is natural to choose $\hat{\beta}^*$ to be from a specific shrinkage method such as SCAD that only requires the estimation of one extra parameter. Adaptive methods such as ALASSO, AEN, etc., are also viable. The idea is to use the $\hat{\beta}_j^*$'s in $\hat{\beta}^*$ to adjust the location of the penalty function (in the James-Stein sense). Using the data multiple times in this manner is done regularly in shrinkage methods.

We state our extensions to Theorems 2.1 and 2.2 as corollaries since we assume the same hypotheses. For penalized log likelihoods with location shifted penalties we have the following analog to Theorem 2.1.

Corollary 2.1 *Redefine the objective function in Sect. 2.1 to be*

$$Q(\beta) = L(\beta|x^n) + n\sum_{j=1}^{p} \lambda_j f_j(\beta_j - \hat{\beta}_j^*).$$

Then, under the same conditions as in Theorem 2.1, the estimator $\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2)^T$ that minimizes $Q(\beta)$ has the OP, i.e.,

1. $P(\hat{\beta}_2 = 0) \rightarrow 1$, *and*
2. $\sqrt{n}J_1(\beta_1|x^\infty)^{\frac{1}{2}}(\hat{\beta}_1 - \beta_1) \rightarrow \mathcal{N}(0, I_{p_0})$.

Proof The proof of Corollary 2.1 follows directly from the proof of Theorem 2.1. Indeed, if for each true $\beta_j = 0$, then for large n and for $p_0 + 1 \leq j \leq p$ we have

$P(\hat{\beta}_2^* = 0) \to 1$. This is a direct result of using a consistent shrinkage estimator for β_j^*. So, in the proof of Lemma 2 in appendix B2, the inequalities at the end are asymptotically unchanged. Then, since $\hat{\beta}_j^*$ only appears in the penalty, not the likelihood, and in the proof of Theorem 2.1 the penalty only has to be controlled in the last term of (B8) in appendix B1, to get the asymptotic normality it is enough for $\sqrt{n}\lambda_j \to 0$, as guaranteed by the hypotheses. □

For penalized empirical risks with location shifted penalties we have the following analog to Theorem 2.2.

Corollary 2.2 *Redefine the objective function in Sect. 2.2 to be*

$$Q(\beta) = R(\beta|x^n) + n \sum_{j=1}^{p} \lambda_j f_j(\beta_j - \hat{\beta}_j^*).$$

Then, under the same conditions as Theorem 2.2, the estimator $\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2)^T$ that minimizes $Q(\beta)$ has the OP, i.e.,

1. $P(\hat{\beta}_2 = 0) \to 1$, *and*
2. $\sqrt{n} J_1^*(\beta_1|x^\infty)^{\frac{1}{2}}(\hat{\beta}_1 - \beta_1) \to \mathcal{N}(0, I_{p_0})$.

Corollary 2.2 follows from Theorem 2.2 the same way that Corollary 2.1 follows from Theorem 2.1.

The methods motivated by these corollaries continue to allow shrinkage via the w_j's as well as 'James-Stein' type shrinkage (i.e. shifting the penalty to be located around a $\sqrt{n}$-consistent estimator of its true value). We are introducing another p hyper-parameters, and for this reason we only recommend this 'double shrinkage' approach when n is not too much smaller than p (preferably $n > p$). For these cases, we use $\hat{w}_j = 1/|\hat{\beta}_{j,OLS}|$ when $n > p$ and $w_j = 1$ for all j when $n < p$ because our simulations show adaptive methods perform poorly in this case. Taken together, double shrinkage lets us set coefficients to zero from the OP on the $\hat{\beta}_j^*$'s and from the OP on the β_j's. Moreover, when $n > p$ the estimated weights $\hat{w}_j$ give tighter intervals around β_j's that have smaller $|\hat{\beta}_{j,OLS}|$'s. Thus, in practice, we tend to get penalties/priors that are centered around zero when they should be and not centered around zero when they shouldn't be.

All of the theory to this point requires the derivative of the penalty f_j' not be defined at 0. If we allow the derivative of the penalty to be 0 at 0, i.e. $f_j'(0) = 0$, then we get similar, but weaker results. Namely, we have

1. $\hat{\beta}_2 \xrightarrow{p} 0$, and
2. $\sqrt{n} J_1^*(\beta_1|x^\infty)^{\frac{1}{2}}(\hat{\beta}_1 - \beta_1) \to \mathcal{N}(0, I_{p_0})$.

This follows from the same proof as in Theorem 2.1, but without assuming Lemma 1 in Appendix B.1. That is,we can allow the penalty to be differentiable everywhere and still get the convergence in distribution and thus in probability analog.

The main difference from this and our main results presented here is that differentiable penalties are unable to set estimates exactly to zero, but asymptotically, they still converge to 0 for the parameters that are indeed 0. While this is a weaker mode of convergence, the fact that a convergence result holds for differentiable penalties allows us to search for an optimal penalty from a list of differentiable and non-differentiable penalties. This is seen in Sect. 5.1.

3 Computational comparisons

Every shrinkage method for linear models generates a predictor of the form (1.3),i.e.,

$$\hat{Y}(x_{n+1}) = x_{n+1}^T \hat{\beta}$$

where the estimate $\hat{\beta}$ of β is a function of the data-driven estimates of λ and the w_j 's. It is well-known that many shrinkage methods (LASSO, EN, etc) zero-out coefficients β_j and thus do variable selection as well as estimation—specifically penalties that have a corner at 0; see the discussion in Sect. 6. Here, we look only at the instability of predictive error and the accuracy of variable selection.

Following Luo et al. (2006) we add random $N(0, \tau^2)$ noise to the y_i's in $\mathcal{D}_n$ and denote the partition of the perturbed data by

$$\mathcal{D}_n(\tau) = \mathcal{D}_{train}(\tau) \cup \mathcal{D}_{test}(\tau).$$

For any predictor $\hat{Y}$, we define its instability to be

$$S(\hat{Y})_\tau = \sqrt{\frac{1}{n_{test}} \sum_{i \in \mathcal{D}_{test}(\tau)} (y_i - \hat{Y}_\tau(x_i))^2}$$

where $\hat{Y}_\tau$ means we have formed a predictor using $\mathcal{D}_{train}(\tau)$. In the computations we present in this section we used $\tau_k = k, k = 1, \ldots, 10$, to generate instability curves of the form $(k, S(\hat{Y})_{\tau_k})$, and looked for patterns.

Intuitively, perturbing the Y's by adding normal noise should only increase $S(\hat{Y})_{\tau_k}$, i.e., the instability curves should increase with τ. Of course, we prefer instability curves that are small—less instability upon perturbation suggests a better predictor. However, if an instability curve decreases with τ then perturbation of Y is making the predictor more stable. We take this to mean the predictor is discredited for some reason. We suggest this behavior arises when the predictor has omitted or included terms incorrectly or has poorly chosen coefficients. We seek predictors with instability curves that are lower than the instability curves of competing predictors and smoothly increase slowly with τ.

Our basic computational procedure is as follows. Fix a number K of values of τ to form the points on the instability curve and a (large) number L for the number of iterations to be averaged. For each $k = 0, 1, \ldots, K$, let $\ell = 1, \ldots, L$. Note, we allow $k = 0$ to be the first iteration, meaning no extra noise is added to the data. This first iteration corresponds to standard RMPSE. Thus, our method compares the shrinkage methods predictive performance as part of the instability evaluation.

Instability curves for a given predictor can generically be formed by the following steps.

1. For each ℓ, randomly split $\mathcal{D}_n$ in to $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$.
2. Perturb the y-values in $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ using $N(0, \tau_k^2)$ noise. Call the results $\mathcal{D}_{train,\tau_k}$ and $\mathcal{D}_{test,\tau_k}$, respectively.
3. Using $\mathcal{D}_{train,\tau_k}$ form the competing predictors denoted $\hat{Y}_{\tau_k}$.
4. Using $\mathcal{D}_{test,\tau_k}$ obtain $S(\hat{Y}_{\tau_k})$ for each predictor.
5. Let $S(\hat{Y}_{\tau_k,\ell})$ be the ℓ-th value of $S(\hat{Y}_{\tau_k})$.
6. For each predictor and k, find the sample mean:

$$S_{\tau_k}(\hat{Y}) = \frac{1}{L}\sum_{\ell=1}^{L} S(\hat{Y}_{\tau_k,\ell}).$$

7. Plot $S_{\tau_k}(\hat{Y})$ as a function of τ_k for each predictor.

In this section all simulated data come from

$$Y = X\beta + \epsilon$$

where the rows $X_i = x_i$ of X, for $i = 1, \ldots, n$, are either $MVN_p(0, M)$ or are IID $\sim t_3$ (t_3 is a t distribution on 3 degrees of freedom) to see the effect of heavier tails. In the $MVN_p(0, M)$ case, we consider various choices of variance matrix M.

The vector $\beta = (\beta_0, 0)$ has values $\beta_0 \in \mathbb{R}^{p_0}$ drawn from IID $N_{p_0}(4, 1)$ with a $1 \times p - p_0$ vector of zeros appended to allow for sparsity.

Thus we have, $\dim(Y) = n$, $\dim(X) = n \times p$, and $\dim(\epsilon) = n$. We use two choices for the distribution of ϵ_i, $N(0,\ 1)$ and t_3, to represent light and heavy tails in the error, respectively. We are concerned mainly with the case $p < n$, but include cases $p > n$ for completeness.

We compared predictors from seven different shrinkage methods as well as a full linear model. Four of the shrinkage methods have the OP, namely, ALASSO (Zou 2006), AEN (Zou and Zhang 2009),

SCAD, and MCP (Zhang 2010). The remaining three methods, RR, LASSO, and EN, do not have the OP.

Below we list our choices for estimating the adaptive weights as well how we split the data into a training set and testing set.

As noted in Sect. 1, we follow Zou (2006) for the adaptive methods by choosing $\hat{w}_j = 1/|\hat{\beta}_{j,OLS}|$ for $p < n$. When we do not have enough data to implement OLS, i.e. $p > n$, we used $\hat{w}_j = 1/|\hat{\beta}_{j,SCAD}|$ because SCAD is a $\sqrt{n}$-consistent estimator (although this is not necessarily important since $n < p$. Note for $\hat{\beta}_{j,SCAD} = 0$, we set $\hat{w}_j = 500$ for computational reasons since we cannot divide by 0.

We examined four settings of p relative to n, while considering three different sparsity levels. We set $p = 100$ and considered three sparsity levels, 10%, 50%, and 90%, which corresponds to $p_0 = 90, 50, 10$, repsectively. We consider four sample sizes $n = 40, 75, 150, 500$. For each n we let $L = 1000$ for the instability computations. That is we averaged over 1000 datasets to get an instability value for each perturbation level. We used a training data set to form the predictor and a testing data set to evaluate its performance. Formally we have

$$\mathcal{D}_n = \mathcal{D}_{train} \cup \mathcal{D}_{test}.$$

We reserve 75% of the data for training and the remaining 25% for testing.

Producing the instability curves for the seven shrinkage methods required two packages in R. For LASSO, RR, EN, ALASSO, AEN, we used the glmnet package (see Friedman et al. 2010) in RStudio Ver. 1.2.5033. To implement SCAD and MCP we used the ncvreg package. Using both of these packages, we implement the same k-fold cross validation to estimate the shrinkage parameter λ.

The next three subsections present our simulation results for the four sample sizes and three sparsity levels. We provide a summary with our recommendations in Appendix A. We included this section in the appendix because we want to emphasize that our method can be used for real data. These simulations are only meant to show that our method behaves as it should in settings when the true model is known.

3.1 Sample size $n = 40$

Our first example uses $n = 40$ that is small compared to $p = 100$. Here, we examine variable selection performance and predictive performance for the three sparsity levels.

We use two assessments for this. First, we generate instability curves to evaluate predictive performance. Then we also look 'inside' the predictor to see which variables were included correctly.

As noted in Sect. 1, we think of instability curves as more important because they reflect 'variability' and bias.

Figure 1 shows that there is overall more instability with heavy tails, and as sparsity increases the methods become more stable for both light and heavy tails. Further, EN is the top performing method for both light and heavy tailed cases. As sparsity increases, LASSO becomes better and is closer to EN. This is due to the fact that LASSO is only able to retain n variables at most, and when there is little sparsity LASSO does not have enough data to retain all the true variables. Since EN is a trade off between RR and LASSO, it is able to retain more than n variables.

The adaptive methods do not perform as well as some of the non-adaptive methods in this case. This is partially due to having to estimate the extra parameters.

In all curves, note that that the $\tau = 0$ point on an instability curve is the RMPSE. That is, the actual predictive error of the predictor with no perturbations. So sudden increases (jumps) or decreases (falls) indicate model instability that should be interpreted in the context of the problem.

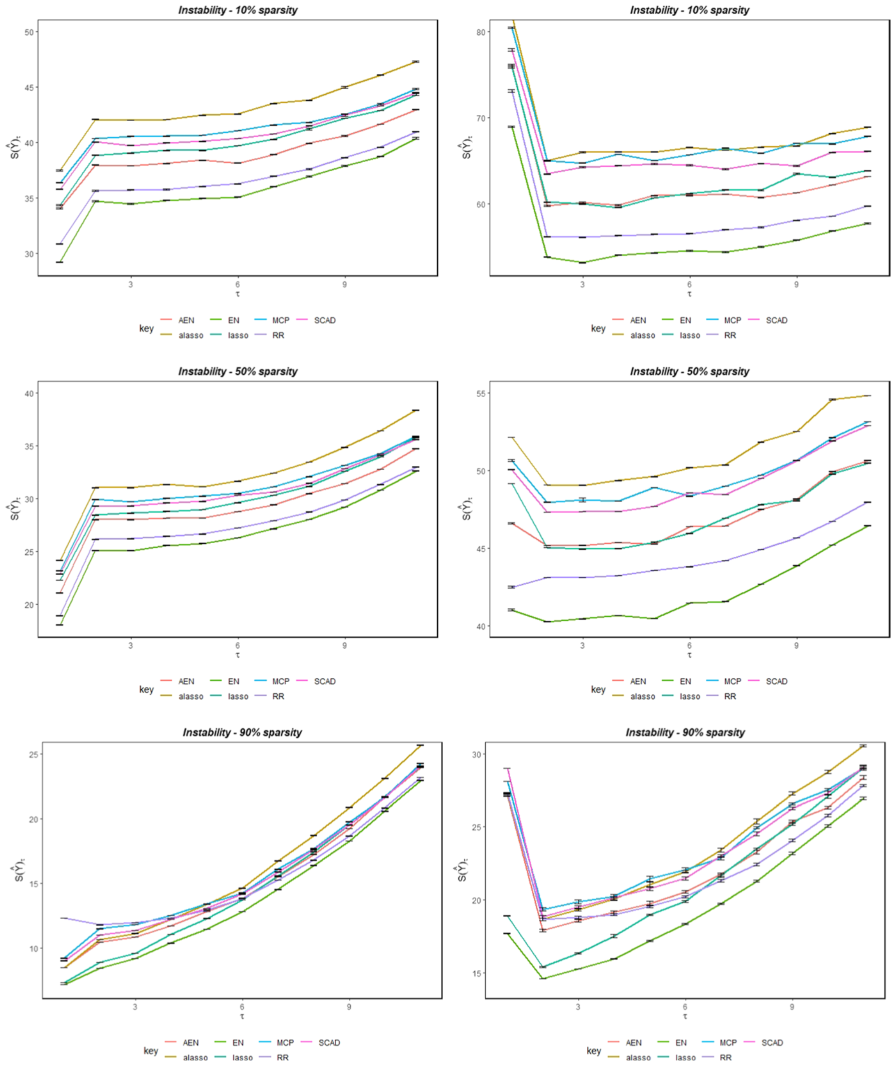

Fig. 1 Instability curves for $n = 40$. From top to bottom the sparsity increases from 10%, to 50%, to 90%. The left column is for $\epsilon_i \sim N(0, 1)$ and right column is $\epsilon_i \sim t_3$

For the heavy tailed cases, almost all the instability curves decrease markedly as the perturbation increases suggesting they have chosen poor initial models. This is likely due to the small sample size coupled with the heavy tails, making it difficult for the assumptions of the model to be met in the data. Also, it is seen that there are some jumps in the instability curves. These are usually when the initial noise is added. We suggest these jumps indicate the shrinkage method is unstable in terms of choosing a good predictor. While there may be a lack of stability, we still cannot immediately discredit models for these initial jumps. Since we are choosing from a

Table 1 Variable selection performance for $n = 40$

Sparsity		.10			.50			.90		
		Tot	TP	FP	Tot	TP	FP	Tot	TP	FP
Li	LASSO	.05	.05	.12	.06	.06	.07	.16	.59	.11
Li	ALASSO	.05	.05	.12	.05	.05	.03	.07	.44	. 03
Li	EN	.36	.37	.42	.52	.53	.53	.18	.60	.13
Li	AEN	.32	.32	.37	.45	.45	. 46	.07	.45	.04
Li	SCAD	.05	.12	.12	.08	.07	.08	.11	.53	.06
Li	MCP	.02	.11	.11	.05	.05	.03	.08	.45	.04
H	LASSO	.02	.03	.12	.04	.05	.05	.09	.40	.07
H	ALASSO	.03	.03	.12	.03	.03	.04	.06	.25	.04
H	EN	.28	.28	.35	.55	.55	.54	.14	.43	.11
H	AEN	.28	.28	.34	.52	.52	.52	.09	.28	.07
H	SCAD	.04	.05	.13	.03	.04	.05	.05	.21	.05
H	MCP	.02	.03	.12	.02	.02	.04	.04	.17	.03

list of shrinkage methods, we would still choose the most stable among the list, even if they are all relatively unstable.

Next, we consider a more classical approach for comparing the predictors by looking at the variable selection performance of each. Table 1 compares each of the methods for both light (Li) and heavy (H) tails based on the total percentage of variables selected on average (Tot; ideally equal to 1 minus the sparsity level), the true positive rate (TP; ideally 100%) and the false positive rate (FP; ideally zero). In our view, the predictive performance (as in the stability curves) is more important, but when the goal is to have a parsimonious model, that is predictively just as good as another model, we can use variable selection performance to choose between comparable shrinkage methods.

From Table 1, it is seen that none of the methods perform well. The values in the Tot columns for low and medium sparsity are often much too low which leads to high values in the FP columns, trivially. Further the TP columns are all too low to consider any of them good. EN and AEN appear to be better than the rest, albeit still not good, in that they retain more or close to the true number of non-zero parameters, on average. This leads to higher TP values for EN and AEN, but also higher FP values. In this scenario, its hard to choose a good method from the more classical approach (variable selection tables), which makes the newer predictive approach (instability curves) more usable and hence more important. Table 1 agrees with Fig. 1 in that variable selection is generally worse for heavier tails.

As a final point about the classical approach versus the predictive stability approach we emphasize that the variable selection tables only describe how well the variables were chosen, and not how well the coefficients of the chosen variables estimate their corresponding parameters. Predictive stability encompasses both selection and estimation in that the methods that estimate the parameters better naturally form better predictors which can be seen in the curves.

We also considered two cases where the x_j's have a nontrivial dependence structure. Specifically, we set $X \sim MVN_{100}(0, M)$ where M is tridiagonal or Toeplitz, both with light tailed errors. We do not show the plots for these cases, however we include them in the recommendations we give in Appendix A.

As in the independence case, the methods performed generally better as sparsity increased. For figures analogous to those in Fig. 1 but for tridiagonal variance matrices, we found that for low and medium sparsity, EN was always the best method. For high sparsity we got the same results as in the upper right panel of Fig. 1. When the variance matrix of X was Toeplitz, we found that EN performed best for all sparsity levels.

In terms of variable selection as indicated in Table 1, we found that the results for the tridiagonal and Toeplitz were similar to the independence case across all methods and sparsity levels except for EN in the Toeplitz case where EN performed noticeably better than the other methods across all sparsity levels. In addition, LASSO does well in the high sparsity case. We suggest that being able to choose what the penalty looks like as in EN gives some advantage over the other methods. We return to this point in Sect. 5.

3.2 Sample size $n = 75$

Next we consider a second example where $p > n$. Here we still have fewer observations than explanatory variables, but n is much closer to p than in Sect. 3.1. We see in Fig. 2 that if $n \approx p$, shrinkage methods can result in good predictive performance in high sparsity cases.

Examining the instability plots in Fig. 2, we see that as in Fig. 1, the methods become more stable as sparsity increases (top to bottom) and less stable as the tails become heavier (left to right).

Similar to the $n = 40$ case, EN remains the top performing method for 10% and 50% sparsity. However, for 90% sparsity we observe SCAD and MCP outperforming the other methods for both light and heavy tailed cases. The $n = 40$ and $n = 75$ cases are qualitatively similar apart from $n = 75$ being slightly more stable, especially for the the light tail, high sparsity case.

Table 2 shows that, as with Table 1, no method performs variable selection well for low to medium sparsity. For high sparsity, all methods were very much improved. However, LASSO and EN were the worst—the only methods not having the OP. The other methods, ALASSO, AEN, SCAD, and MCP, have the OP and perform roughly equally well. As a generality, the methods performed better for light tailed than for heavy tailed distributions. As before, the better performing methods in the instability curves (EN, RR, LASSO) tended to perform worse in terms of variable selection. The differences in variable selection and predictive stability performance can be attributed to better parameter estimation for EN, RR, and LASSO. Namely when a method included more variables than necessary, the incorrectly included variables may have coefficient estimates close to zero.

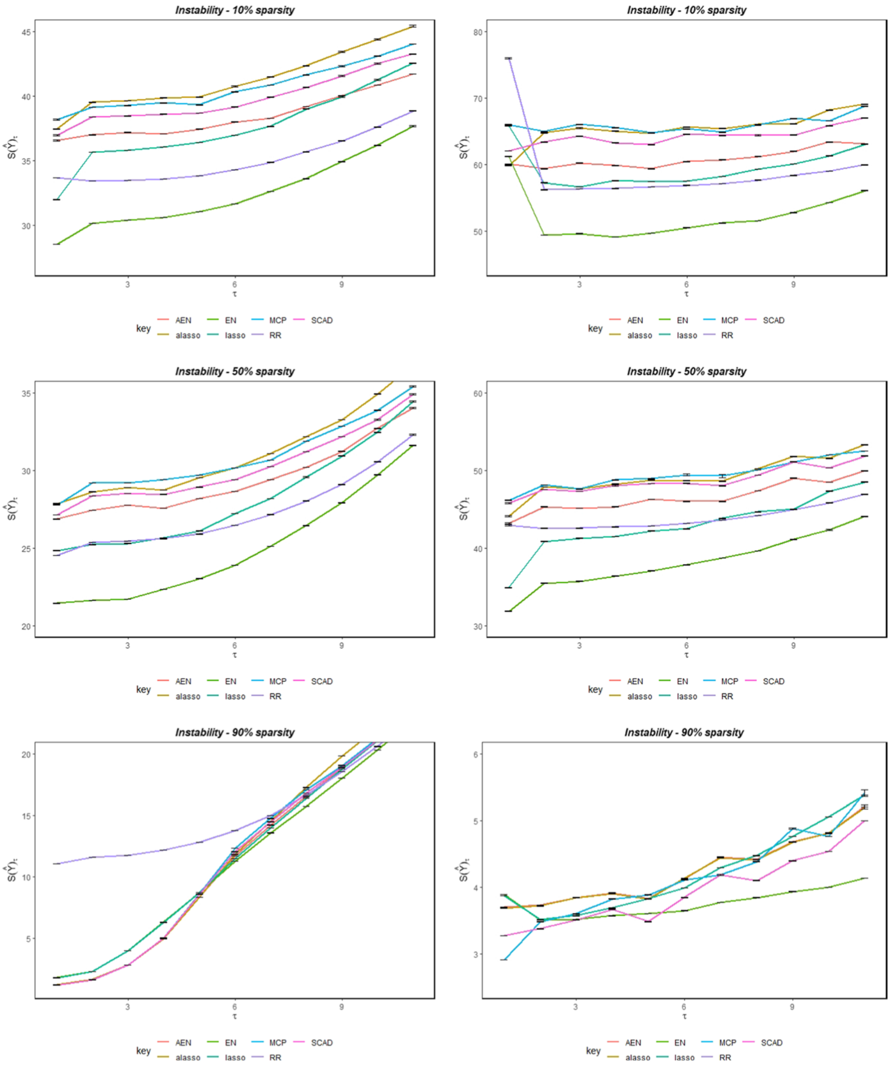

Fig. 2 Instability curves for $n = 75$. From top to bottom the sparsity increases from 10%, to 50%, to 90%. The left column is for $\epsilon_i \sim N(0, 1)$ and right column is $\epsilon_i \sim t_3$

It is seen that the curves in Fig. 2 have fewer sudden increases and decreases to the right of zero than the curve in Fig. 1. In fact,it is the upper two panels that show most instability perhaps due to the higher sparsity level. Overall, the comparison indicates that more data (unsurprisingly) generally provides more stability.

When we included dependence via tridiagonal matrices, we found results similar to the independent case. Namely, the instability curves show that EN performed best at all sparsity levels, roughly tying with most other methods for high sparsity. The

Table 2 Variable selection performance for $n = 75$

Sparsity		.10			.50			.90		
		Tot	TP	FP	Tot	TP	FP	Tot	TP	FP
Li	LASSO	.20	.21	.19	.17	.23	.13	.21	.91	.14
Li	ALASSO	.10	.11	.12	.08	.12	.08	.10	.90	.01
Li	EN	.43	.44	.37	.50	.53	.43	.22	.91	.14
Li	AEN	.16	.16	.17	.20	.24	.18	.10	.90	.01
Li	SCAD	.12	.13	.13	.12	.16	.09	.10	.90	.01
Li	MCP	.07	.07	.12	.06	.09	.05	.10	.90	.01
H	LASSO	.12	.12	.13	.20	.30	.12	.22	.90	.13
H	ALASSO	.04	.04	.11	.07	.11	.04	.10	.87	.01
H	EN	.24	.25	.21	.31	.41	.22	.22	.90	.15
H	AEN	.08	.08	.14	.10	.15	.08	.10	.87	.01
H	SCAD	.09	.10	.13	.12	.18	.08	.10	.88	.02
H	MCP	.06	.07	.12	.08	.12	.05	.10	.89	.01

key difference was that RR tended to perform poorly overall. When we generated the data using the Toeplitz matrices, for low and medium sparsity, EN was the best method and for high sparsity all methods except RR worked well relatively.

In terms of variable selection, the tridiagonal case was similar to the independence case but slightly better. This was surprising and difficult to interpret. In the Toeplitz case, EN, SCAD and MCP are noticeably better than the other methods for low and medium sparsity. For high sparsity, LASSO also performs well. This is the same as the Toeplitz case with $n = 40$. Thus, overall, the results for dependence cases with $n = 75$ are very close to the corresponding results for $n = 40$.

3.3 Sample size $n = 150$

Now we examine a case where $n > p$, making it qualitatively different from the earlier two subsections. Here we implement LM's as well as the same shrinkage methods. Note that the "penalty" associated with LM's is a constant and corresponds to a uniform prior.

Figure 3 parallels Figs. 1 and 2, but includes LM.

For low sparsity, light tails LM, SCAD and MCP are initially the best and indistinguishable from each other. The other methods are more stable as more noise is added, but the noticeably worse initial performance suggest LM, SCAD, or MCP is preferred here. For the heavy tailed case LM is the clear top performing model initially, but its curve increases faster than for some other penalties

For medium sparsity, LM becomes slightly worse than all the other methods except for RR. Here we also begin to see several methods performing roughly equally well: SCAD, MCP, EN, AEN, and ALASSO are all initially roughly equal (their instability curves and associated predictive intervals overlap initially). That said, EN, RR, and LASSO's instability curves increase the slowest, indicating they

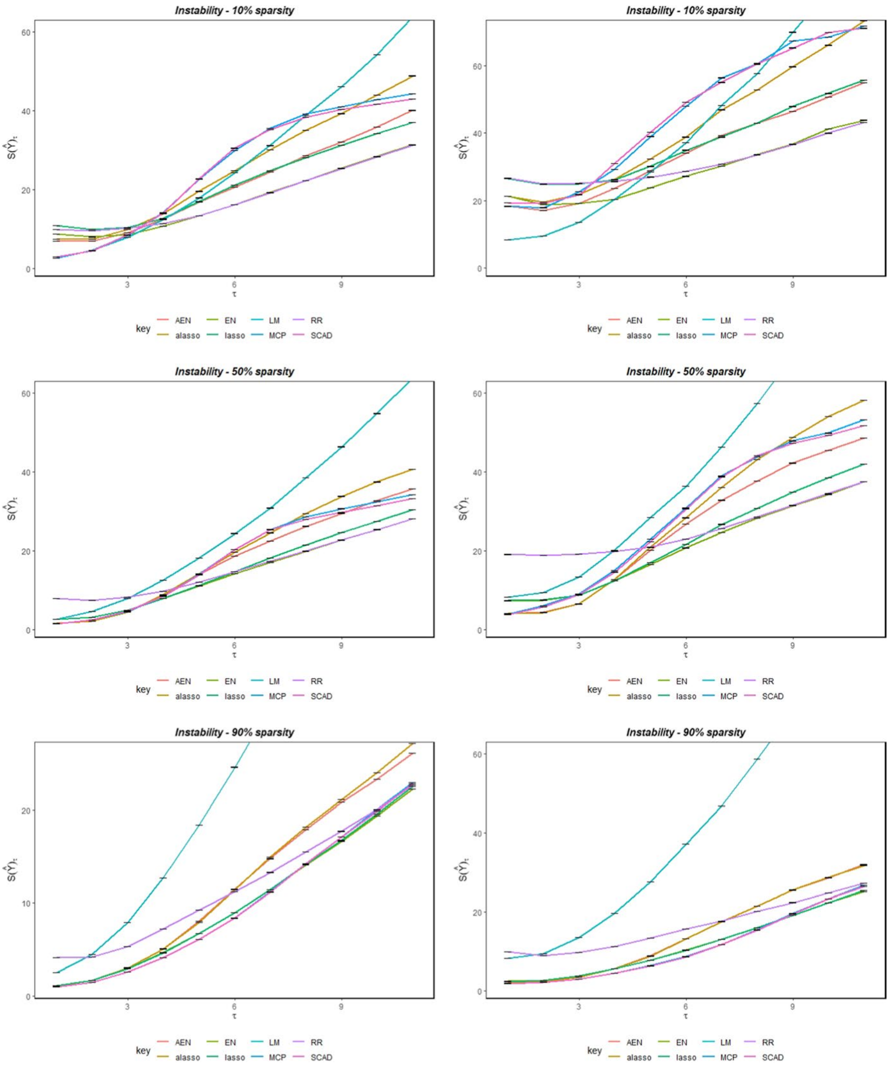

Fig. 3 Instability curves for $n = 150$. From top to bottom the sparsity increases from 10%, to 50%, to 90%. The left column is for $\epsilon_i \sim N(0, 1)$ and right column is $\epsilon_i \sim t_3$

are stable relative to the other methods. Taken together, we find that the adaptive methods appear to be better initially, but the non-adaptive methods are not much worse initially and overall more stable since they do not need to estimate as many parameters.

Finally for high sparsity, RR and LM are discredited, but the other methods are all roughly the same with SCAD and MCP being slightly less stable than EN, AEN, LASSO and ALASSO.

Table 3 Variable selection performance for $n = 150$

Sparsity		.10			.50			.90		
		Tot	TP	FP	Tot	TP	FP	Tot	TP	FP
Li	LASSO	.94	.97	.68	.71	.99	.42	.25	1	.17
Li	ALASSO	.89	.99	.10	.50	.98	.02	.14	.92	.05
Li	EN	.98	.99	.84	.71	.99	.43	.27	1	.19
Li	AEN	.92	.99	.33	.50	.98	.02	.14	.92	.06
Li	SCAD	.97	.99	.73	.62	.98	.25	.16	.95	.07
Li	MCP	.97	1	.74	.60	.98	.23	.13	.94	.04
H	LASSO	.83	.86	.57	.72	.99	.46	.21	.92	.13
H	ALASSO	.83	.91	.12	.51	.98	.04	.11	.90	.02
H	EN	.94	.96	.75	.73	.99	.47	.21	.92	.13
H	AEN	.89	.96	.22	.51	.98	.04	.11	.90	.02
H	SCAD	.89	.93	.53	.64	.99	.28	.11	.90	.02
H	MCP	.89	.93	.53	.63	.99	.28	.10	.90	.02

A key difference between the $n = 150$ case and the $n = 75, 40$ cases is that the adaptive methods are initially performing better than the nonadaptive methods. For instance, AEN and ALASSO are performing better for small perturbations than EN or LASSO, respectively. When the perturbations are too high, it makes sense that the non-adaptive version of a penalty will perform better that their adaptive versions because they are less affected by the noise; they use fewer estimators. One can argue that the perturbation level at which the curves for non-adaptive penalties and their adaptive versions cross represents the largest reasonable perturbation that should be considered for that penalty. Moreover, the OP is not a determining factor for performance: Some methods with the OP perform well and some do not. Some methods that do not have the OP perform better than other methods that do.

For light tails and low sparsity, Table 3 shows that ALASSO and AEN are the generally the best methods in terms of variable selection. For medium sparsity they remain noticeably better than the other. For high sparsity, ALASSO, AEN, SCAD, and MCP all perform similarly. For heavy tails, the results are roughly the same.

Compared to Table 2 we see that all methods improved in variable selection, which is not surprising, but that the adaptive methods improved more. This is true for the instability curves as well. LM's and RR are not included in Table 3 because they don't do variable selection.

Comparing the conclusions from Fig 3 and Table 3, we see that the methods with the OP perform roughly the same, and there are no obvious contradictions in variable selection and predictive performance.

Again, we considered two dependence cases with light tails, the tridiagonal and the Toeplitz. For the tridiagonal case, the instability curves and the variable selection table are qualitatively the same as for the independence case. For the Toeplitz case, LM is the best method in general.

Overall, variable selection in this case is worse than in the independence case. This is virtually the opposite of Toeplitz in the $n < p$ case where variable selection

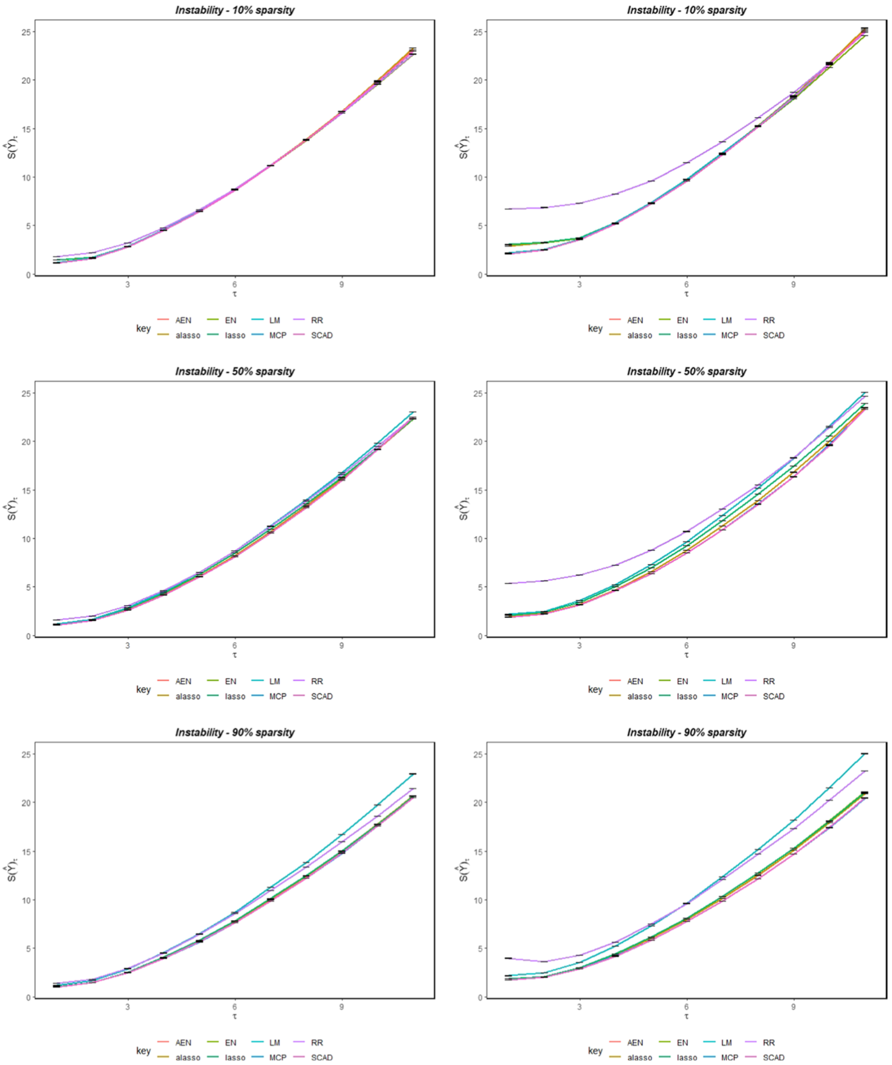

Fig. 4 Instability curves for $n = 500$. From top to bottom the sparsity increases from 10%, to 50%, to 90%. The left column is for $\epsilon_i \sim N(0, 1)$ and right column has $\epsilon_i \sim t_3$

was improved. Typically, our intuition resides in the $n > p$ setting, so this is more in line with intuition.

3.4 Sample size $n = 500$

For completeness we also considered the case $n = 500$ to observe the limiting behavior of the methods. Figure 4 and Table 4 have the same general properties as the earlier figures and tables. Namely, as sparsity increases instability decreases.

Table 4 Variable selection performance for $n = 500$

Sparsity		.10			.50			.90		
		Tot	TP	FP	Tot	TP	FP	Tot	TP	FP
Li	LASSO	.93	1	.38	.72	.98	.46	.14	.93	.05
Li	ALASSO	.90	.99	.10	.50	.98	.02	.10	.90	.01
Li	EN	.94	1	.42	.72	.98	.46	.14	.93	.05
Li	AEN	.90	.99	.10	.50	.98	.02	.10	.90	.01
Li	SCAD	.90	.99	.13	.51	.98	.05	.11	.90	.02
Li	MCP	.90	.99	.13	.51	.98	.04	.10	.90	.01
H	LASSO	.93	.99	.38	.65	.98	.31	.14	.90	.06
H	ALASSO	.90	.99	.10	.50	.98	.02	.10	.90	.01
H	EN	.93	.99	.42	.64	.98	.33	.15	.90	.07
H	AEN	.90	.99	.10	.50	.98	.02	.10	.90	.01
H	SCAD	.90	.99	.13	.50	.98	.03	.10	.90	.02
H	MCP	.90	.99	.13	.50	.98	.02	.10	.90	.01

The methods improve as sparsity increases although the improvement is not as dramatic as in the smaller sample cases. In the heavy tailed cases, as before there is more variability.

Many of the methods at this point are indistinguishable via Fig. 4 or Table 4. So for descriptive purposes it is easier to identify the methods that perform poorly rather than the ones that perform well. From the left column in Fig. 4, the only relatively poor method is RR for light tails. Even so, RR performs well, but it's instability curve is slightly above the upper bound of the confidence intervals from the other methods.

In the right column in Fig. 4, RR is clearly the worst in all cases. For low sparsity LM SCAD and MCP are best. For medium sparsity, RR is is the only method that can be discredited. For high sparsity the worst performers are RR and LM's. Note that LASSO and EN still perform well even though they don't have the OP, reiterating the fact that having the OP should not be the main driver in choosing a shrinkage method. That said, EN is a generalization of LASSO and under stronger conditions, LASSO has some consistency properties, see Zhao and Yu (2006). Thus, the good performance of LASSO and EN is not surprising.

Table 4 shows that at $n = 500$, most methods are performing variable selection quite well. In fact, almost all the methods used that have the OP are nearly perfect, on average, in performing variable selection. Even the methods that look worse in terms of variable selection (LASSO, EN) predict well because they almost always retain all the important variables (TP $\geq$ 93). In general, we start to see consistency properties take effect, although not perfectly yet. Hence, we get generally agreement in the instability curves and the variables selection table for $n = 500$.

For the dependence cases, the tridiagonal covariance matrix resulted in the qualitatively the same results as the independence case. The Toeplitz case resulted in LM always performing the best.

3.5 Increasing *p* and *n*

In this subsection, we verify that the conclusions from the earlier subsections in this section remain valid for a larger range of p and n. So, we consider two new settings, namely $p = 200$ and $p = 500$, and continue to use the same sparsity settings as before. For $p = 200$, we set $n = 80, 150, 300$ and 1000. This allows use to observe the effect of larger n and p while keeping the same ratio n/p as in our earlier simulations. For ease of presentation, we only consider the independent observations case; we omit the low sparsity (10%) cases; and, we omit the low sample size ($n = 80$) cases. However, we note the results are consistent with the earlier subsections. For $p = 500$, we use the same sample sizes but have a smaller n/p ratio.

Figure 5 shows the stability curves for $p = 200$. It is seen that EN tends to peform best although the degree of outperformance decreases as n and/or the sparsity level increases. Also, as n or the sparsity level increases, the curves shift lower and become more similar. This is consistent with Tables 1A,2, 3 and 4A, in the Appendix. Overall there is a bigger separation among curves compared to earlier simulations, but the same qualitative patterns are seen.

Next we consider $p = 500$ for increasing values of n to observe the effect of a larger model space on our stability criterion. Figure 6 shows the stability curves for the competing shrinkage methods. We make the same key observations as before. Namely, as n increases or as sparsity increases the methods become more stable. It is also seen that some of the curves decrease as τ increases from zero; this is like the $p = 100$, $n = 40$ case from Sect. 3.1 and shows that for small sample sizes model uncertainty dominates. Otherwise, we see separation of the curves and that EN remains the best choice if only by a tiny margin. Again, this corroborates our earlier findings.

4 Corroboration on real data

As a test of our predictive methodology, we examine the predictive instability of the shrinkage methods on the data set Superconductivity presented in Hamidieh (2018). This data set has 81 explanatory variables of a physical or chemical nature to explain a response Y representing temperature measurements (in degrees K) for when a compound begins to exhibit superconductivity. Initial data analyses suggested the data were sparse, but it was unclear how sparse. Hamidieh (2018) suggests a sparsity level of about 90% and our techniques here confirm this in the sense that we find, if a penalized linear model that performs variable selection is fit, around 90% of the coefficients will be zero and this will be nearly best possible from a predictive standpoint. In addition, Hamidieh (2018) implicitly used light tails in the error term, ϵ, and did not comment on the distribution of the explanatory variables apart from effectively taking them as independent and not requiring any special treatment to account for spread. Accordingly, we treated these as coming from a light tailed distribution.

Furthermore, Hamidieh (2018) identified 20 variables of potential importance. Of those 20, we suggest only seven of them are important because the variable

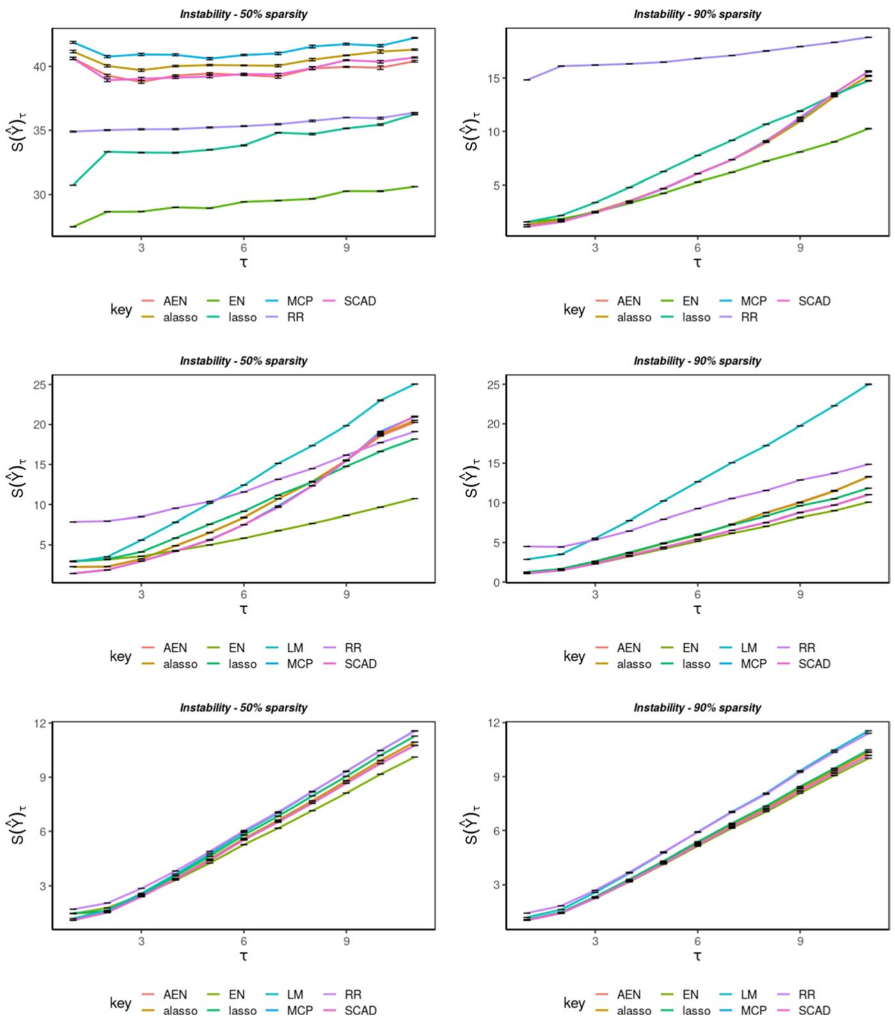

Fig. 5 Instability curves for $p = 200$. From top to bottom n increases from 150 to 300 to 1000. The left column is for 50% sparsity and right column shows 90% sparsity

importance factors decreased suddenly at the eighth most important variable. This gives 7/81 < 10%, confirming this case corresponds to the high sparsity setting. Histograms of the residuals from the full LM suggest this falls into the light tail case as well. Thus, we compare our computed results in this section to the recommendations for the light tailed high sparsity cases treated in Appendix A.

In fact, the full Superconductivity data set had $n = 21,263$, so Hamidieh (2018) was able to use a standard (unpenalized) LM as a 'benchmark model' and then improve on it by developing an XGBoosting model—a boosted, penalized tree model in which the penalty was carefully constructed to be appropriate for trees.

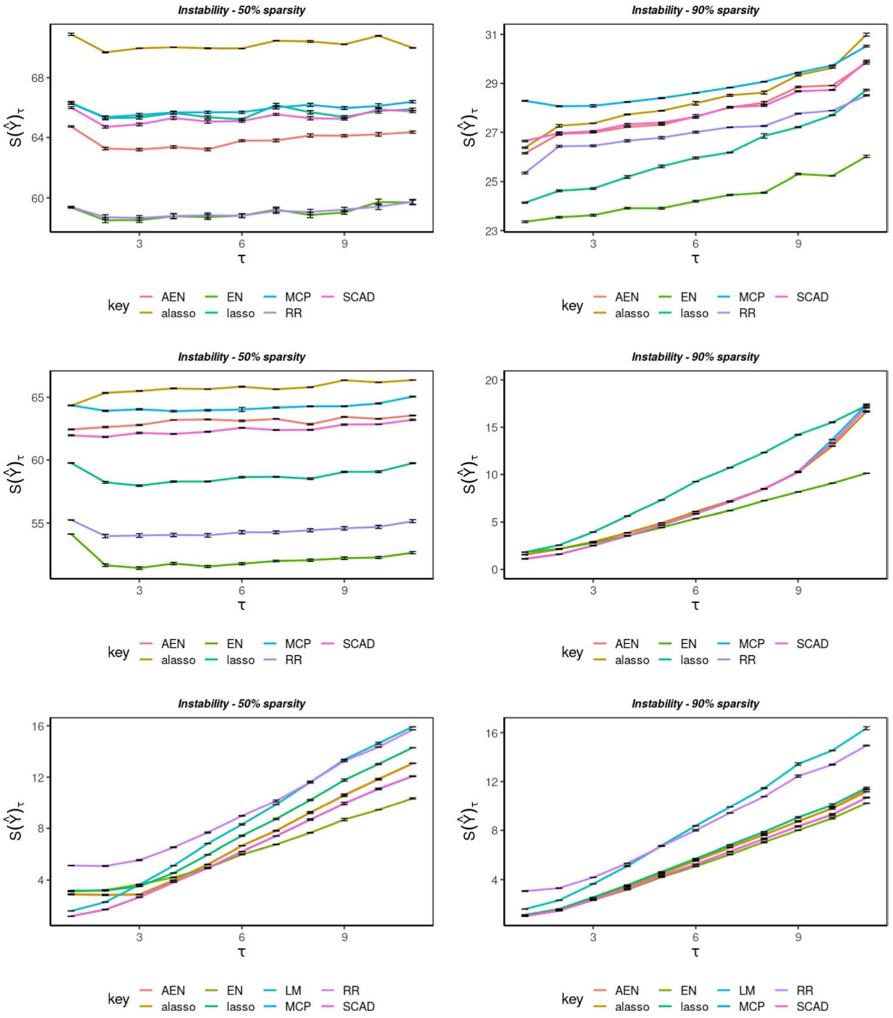

Fig. 6 Instability curves for $p = 500$. From top to bottom n increases from 150 to 300 to 1000. The left column is for 50% sparsity and right column shows 90% sparsity

Here, as is common in practice, especially where a more justifiable methodology is infeasible, we have used LM's for their interpretability. Also, when $n << p$, XGBoosting often does not perform well. So, it may sometimes be reasonable to use shrinkage techniques in mis-specified model situations with small sample sizes.

Since Superconductivity is so much larger than the data sets used in our simulations, we drew 40, 75, 150, and 500 data points at random to match the sample sized used in our simulations. We note that many data sets are much smaller than Superconductivity so our example here is intended to be suggestive for them, too.

We repeated the analyses presented in Sect. 3 for the independent cases with light tails but replaced the simulated data with the randomly chosen subsets of

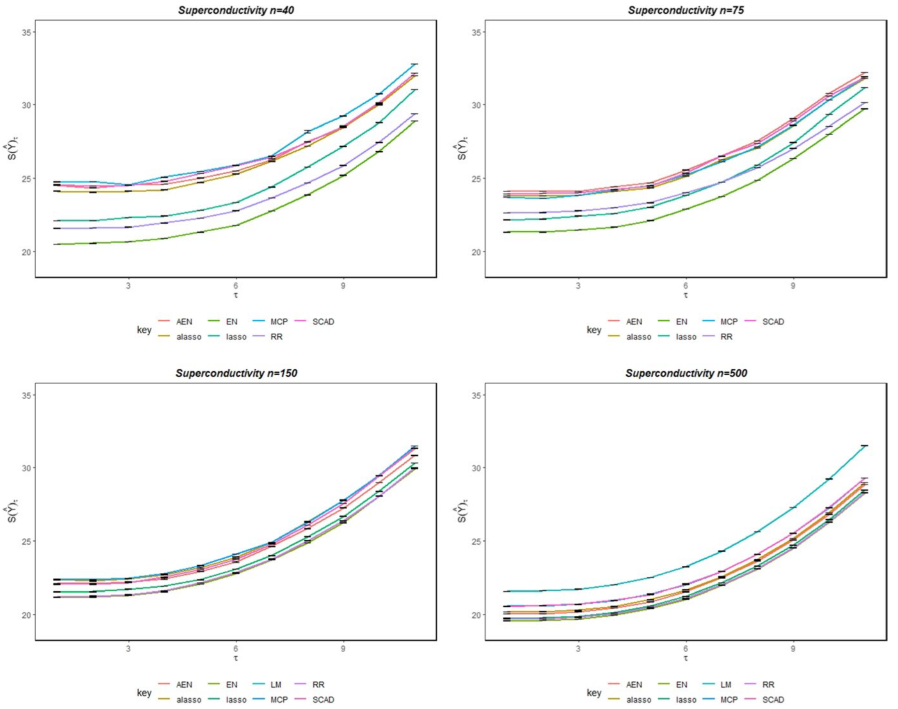

Fig. 7 Instability curves for the Superconductivity data for $n = 40, 75$ (top) and $n = 150, 500$ (bottom)

Superconductivity. We were able to generate instability curves but not the variable selection accuracy tables because the true model is unknown. We highlight this point because our predictive instability methodology is always usable, even when the true model is unknown, as is the case here.

The instability curves for Superconductivity are given in Fig. 7.

For each sample size, we compare the best methods from Fig. 7 to the corresponding recommendations in Appendix A. For $n = 40$, the upper left panel in Fig. 7 shows that EN gives the lowest predictive error and is the most stable. This is the same as recommended in Table A1 in Appendix A for sparsity.9 and light, independent tails. For $n = 75$, the upper right panel in Fig. 7 shows EN is again the best performing method, followed LASSO which improved to be better than RR when increasing the sample size from 40 to 75. Table A2 in appendix A shows only that RR should not be used with light, independent tails. So, again we see agreement even if the recommendations are not specific.

By contrast, for $n = 150$, the lower left panel in Fig. 7 shows that EN and RR are the top performing methods. Table A3 in appendix A indicates that RR and LM's are to be avoided (for light independent tails). So, the good performance of RR disagrees our recommendations. Finally, for $n = 500$, the lower right panel in Fig. 7 shows that five methods form a cluster of the best of the 3 methods. The cluster of top methods is LASSO, EN, and RR. In this case LM is noticeably worse than the rest of the shrinkage methods. The recommendation from

Table A4 in Appendix A is not to use RR or LM's. Again, we have a disagreement on the use of RR.

We explain these findings by model mis-specification. First, the true model is almost certainly not a LM. Indeed, Hamidieh (2018) proposes a model based on trees. The agreement between our recommendations and the data analysis for small values of n probably means that the sample size is too small to detect the difference between the true model and a LM. However, when the sample size increases, the model mis-specification matters. RR normally performs well for non-sparse cases but here is performing well when the true model is sparse.

We conjecture this occurs because when the true data generator is a sparse non-linear model, a non-sparse linear model may approximate the data generator better than a sparse linear model. As a simple example, consider the space of all functions on a domain that have convergent Fourier series expansions. Within this space the model $Y = \sin(x) + \epsilon$ is nonlinear and sparse. However, within the space of analytic functions $\sin(x)$ is approximated arbitrarily well by taking enough terms in its Taylor expansion. That is, it is linear but not sparse. A more complex example in keeping with the Superconductivity data, is to imagine representing a single true tree model with a single linear model. The linear model would have to have many terms to approximate a tree; even one with relatively few nodes. That is, a large enough LM might provide a good approximation. A further point is that even though both LM and RR retain all of the explanatory variables, RR performs better because the regularization provides variance reduction.

One limitation to this argument is that if there are many explanatory variables, the terms in the linear model may be collinear. With large enough sample size or small enough model bias, this is not a problem. However, if these conditions fail, linear models may be discredited as an adequate summary for the data. In such cases, techniques such as neural nets and projection pursuit (that do not suffer the curse of dimensionality) may be necessary as a way to control model bias under constrained sample sizes. Even so, lack of data may still be a problem.

5 Optimizing over the shrinkage method

Previous sections used existing well-studied shrinkage methods, but the results of Sect. 2 show that there are infinitely many other penalties that could be used to get shrinkage methods with the OP. Recalling that penalties are special cases of priors, it is clear that the choices of shrinkage methods used in Sect. 3 are limited. Here we propose that, rather than choosing a shrinkage method from a list, one should find a prior by optimizing a predictive optimality criterion using an adaptive search technique such as a genetic algorithm (GA). The idea is that we use part of the data in a GA optimization to find an optimal prior/penalty and then treat that data dependent prior/penalty as a prior on the rest of the data to make predictions. The fact that the result of our optimization is data dependent makes it look like a posterior.

We use the results in Sect. 2.3 to allow for data-dependent shifts in parameter locations. The main benefit of shifting the location of the penalty is that it reduces prior-data conflict. That way, when we find an optimal penalty in the next subsection, it will correspond to putting priors on the β_j's that have more of their mass close to the true values of the parameters. If the location shift is not used in the penalty, our method below still can be used but is not as effective, especially in small samples. Then we present our GA methodology and verify that we get better predictive performance than off the shelf methods.

5.1 Using GAs to find a shrinkage method

First, we define our initial class of penalty functions. Cors. 2.1 and 2.2 imply we can use any penalty function within a very large class, as long as the regularity conditions are met. Here we represent $f_j(\beta_j)$ using finitely many polynomials. That is, with mild abuse of notation, we set

$$f_j(\beta_j) = \sum_{k=1}^{6} \alpha_k |\beta_j - \hat{\beta}_j^*|^k. \tag{5.1}$$

Obviously, we would get a better approximation to an optimal penalty if we used more terms but for present purposes sixth order polynomials turned out to be sufficient. Our initial population of penalty functions is generated from (5.1) by selecting M values of $\alpha = (\alpha_1, \ldots, \alpha_6)$ IID from a Unif[0,10], say $\alpha_m = (\alpha_{1,m}, \ldots, \alpha_{6,m})$ for $m = 1, \ldots, M$. The GA will update this initial population denoted $A^0 = \{\alpha_1^0, \ldots, \alpha_M^0\}$ of size M over F iterations to a final population $A^F = \{\alpha_1^F, \ldots, \alpha_M^F\}$ also of size M in which we expect essentially all members to be the same. (Givens and Hoeting 2013 p. 75 states that the algorithm often stops when there is little diversity in the population, as we detected.)

We start by showing how the typical iteration from A^0 to A^1 proceeds. Assume we have data $\mathcal{D} = \mathcal{D}_n = \{(y_i, x_i) | i = 1, \ldots, n\}$ and $\dim(x_i) = p$ and the empirical risk

$$R(\beta|\mathcal{D}_n) = \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2.$$

In view of Corollary 2.2 we seek

$$\hat{\beta}_{\alpha_m^0} = \arg \inf_{\beta, \lambda, w^p} \left(\frac{1}{n} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} w_j f_{j,m}^0(\beta_j) \right) \tag{5.2}$$

where

$$f_{j,m}^0(\beta_j) = \sum_{k=1}^{6} \alpha_{m,k}^0 |\beta_j - \hat{\beta}_j^*|^k$$

for each $\alpha_m^0 \in A^0$. We find suitable values of the decay parameter $\lambda \in \mathbb{R}^+$ and the $\hat{\beta}_j^*$'s based on the data as described shortly. We will use two versions of (5.2) depending on the relative sizes of n and p. Specifically, if $p \geq n$ or not too much smaller than n, we set all $\hat{w}_j = 1$ and all $\hat{\beta}_j^* = 0$. If $p < n$, we set $\hat{w}_j = 1/|\hat{\beta}_{j,OLS}|$ as noted in Sect. 2.3. We make this choice because when $n < p$ typically our asymptotic results do not apply. In the case that $p \geq n$ (5.2) reduces to

$$\hat{\beta}_{\alpha_m^0} = \arg\inf_{\beta} \left(\frac{1}{n} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \sum_{k=1}^{6} \alpha_{m,k}^0 |\beta_j|^k \right). \tag{5.3}$$

To solve (5.2) or (5.3), we randomly split the data to estimate the various parameters. We begin by writing $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test}$. We reserve $\mathcal{D}_{test}$ for comparing predictors after the entire GA process is completed. Next we split the training data

$$\mathcal{D}_{train} = \mathcal{D}_{train,\lambda} \cup \mathcal{D}_{train,\beta} \cup \mathcal{D}_{train,\alpha}.$$

We use $\mathcal{D}_{train,\lambda}$ to find λ and $\mathcal{D}_{train,\beta}$ to find the $\hat{\beta}_j^*$'s and the $\hat{w}_j$'s ($n > p$). Since $\hat{\beta}_{\alpha_m^0}$ depends on λ we begin by searching over a list of values Λ equally spaced from

$$\lambda_{max} = \left(\frac{1}{n_{train,\beta}} \right) \max |Y_{train,\beta}^T X_{train,\beta}|$$

to $\lambda_{min} = \gamma \lambda_{max}$ for some $0 < \gamma < 1$. (Here, $n_{train,\beta} = \#\mathcal{D}_{train,\beta}$ with corresponding data indicated by $Y_{train,\beta}$ and $X_{train,\beta}$.) For each fixed α_m^0 and each choice of $\lambda \in \Lambda$, we find $\hat{\beta}_{\alpha_m^0,\lambda}$ from $\mathcal{D}_{train,\beta}$ and choose the $\hat{\lambda}_m^0$ that minimizes $R(\hat{\beta}_{\alpha_m^0,\lambda}|\mathcal{D}_{train,\lambda})$.

We find $\hat{\beta}_{\alpha_m^0}$ for each m in (5.2) by sub-gradient descent since α_m^0, $\lambda = \hat{\lambda}_m^0$, $w_j = \hat{w}_j$ and $\hat{\beta}_j^*$ can be taken as given. (The $\hat{w}_j$ and $\hat{\beta}_j^*$ should also have sub- and super-scripts m and 0; we omit these for convenience.) Recall, the sub-gradient descent algorithm allows for us to have points of non-differentiability in the penalty (e.g., a corner as in L or SCAD), and in cases where the penalty is differentiable, the sub-gradient is uniquely defined by the gradient. Note that the objective function is constructed to be convex, so we are sure to find a minimum. We initialize the gradient descent algorithm at the LASSO solution for $n > p$ and at the RR solution for $n < p$.

Now define the fitness function for the GA to be

$$f = \sum_{i \in \mathcal{D}_{train,\alpha}} (y_i - x_i^T \hat{\beta}_{\alpha_m^0, \hat{\lambda}_m^0})^2. \tag{5.4}$$

We evaluate the fitness for each α_m^0 in A^0. Note that for each α_m^0 for $m = 1, \ldots, M$ we get a single best choice for $\hat{\lambda}_m^0$ and $\hat{\beta}_{\alpha_m^0,\hat{\lambda}_m^0}$ and hence a single fitness value. However, it is possible for different α_m^0's to give exactly the same f-value because its possible $\hat{\beta}_{\alpha_i^0,\hat{\lambda}_m^0} = \hat{\beta}_{\alpha_j^0,\hat{\lambda}_m^0}$ for some $i \neq j$. Although this would appear to happen with probability zero, it is observed on a regular basis. This arises because different but similar penalties may lead to the same solution and because computing only has limited precision.

Next, by elitism we select off the top 20% of members of A^0. We fill in the 'missing' 80% by applying crossover and mutation to the bottom 80% of fitness values to obtain a new generation of size M from the algorithm to go into the second iteration.

Crossing means switching some entries of a genome α'_m with entries from another $\alpha^\dagger_m$ to generate a 'new genome'. This is done at random keeping only the 'child' until the population size M is achieved. Mutation means adding a perturbation to all members of α (here a random number between the user specified maximum and minimum values for each component in α). Mutation does not change the size of the population, only the specific genomes already in it. In this way we get a new population A^1 to which we can apply the same procedure. Then, we can iterate to get A^2, A^3 and so on until A^F contains little diversity.

To see that this is the typical behavior of this sort of GA, we use the framework of Rudolph (1996). First, it is easy to see that, as we have set it up here, the GA is a Markov process. That is, the probabilistic behavior in moving from time t to time $t + 1$ depends only on the state at time t. Moreover, this Markov process is homogeneous in the sense that the transition from time step to time step is the same for any two adjacent time steps. Note that the Markov process is 'discrete time' and has a discrete population (leading to distinct crosses) but the mutation is continuous because of the uniform distribution. Thus, there is no transition 'matrix'. Instead, there is a transition kernel, $K(x, S)$, where x is a population member at time t and S is a set of possible states to which x may be transformed and K is independent of t. In fact, $K(\cdot, \cdot)$ can be partitioned into a K_m and K_c, a mutation and crossover kernel. The crossover kernel is a transition matrix since crossover is discrete. The mutation kernel includes the continuous mutation phase based on the uniform distribution. So, let x be any state at time t and suppose an optimum f^* exists and the Markov process has state space E. Then, there will be elements of E arbitrarily close to f^*. Let $b(x_t)$ be the best fitness value within the t-th population and let $d(x) = b(x) - f^*$. As long as the population is large enough, $B_\epsilon = \{d(x) < \epsilon\}$ will have nonzero probability for $\epsilon > 0$ and hence $K_m(x, B_\epsilon)$ will be bounded away from zero. Now, given that we have used elitism, Theorem 2 in Rudolph (1996) applies to give convergence of the GA to the global minimum of f within the class of priors that satisfy the conditions of Cors. 2.1 and 2.2.

The behavior of the GA—as opposed to the behavior of the subgradient descent used to estimate the parameters—depends on M, the elements of A^0, the size of F, the choice of f, the number of generations, the form of elitism and mutation, and the data. Being based on Markov processes, convergence is not the question, rather convergence *rate* is. However, it is difficult to provide guidance on how to choose any of these optimally in general. If we fix minimal predictive error as our criterion (as opposed to running time) and fix a method for choosing the α's (such as used here) then as M increases, the error can only decrease, assuming the other factors are held constant. Of course, if the # generations increases, the predictive error can only decrease albeit at the cost of longer running time. It is unclear what happens if the elitism and mutation rate change. This may be a setting where the 'No Free Lunch' theorem applies i.e., any solution that is optimized for one setting will perform poorly in another setting to compensate. For instance, on our system, in Sect. 5.2.1, the running time was about two hours whereas in Sect. 5.3, running times were typically over nine hours. Thus, each setting should be addressed indvidually.

Indeed, a pragmatic check on the behavior of a GA would be to run it with different initial populations to see if the GA outputs approximately the same minimum.

To ensure convergence of the GA one should set a large population as well as a large number of generations. We comment that in the sub-gradient descent phase of our procedure, we have limited ourselves to convex objective functions. For more general results we would have to ensure convergence of the gradient based optimization to ensure convergence of the GA-based optimization. We implemented our GA computations using genalg, see Willighagen and Ballings (2015).

Our intuition tells us that this method will be beneficial in low to medium sparsity cases, as well as heavy dependence or non-asymptotic cases. For large sample cases, the OP will take over and methods having the OP are equivalent. Thus a GA can do no better. Further, in the smaller sample cases with high sparsity we observed an effect similar to the OP in that all methods seemed to achieve 'asymptotic' performance. In this setting using the GA approach will likely not provide much benefit. For low to medium sparsity cases, there is more variability between the methods and thus, optimizing to find a best penalty is reasonable. We focus our GA simulations on this setting.

Also using only a standard basis expansion may not allow us to approximate some penalties well. Thus, there may be scenarios where one should use a more general expansion in order to approximate a wider class of penalties. This idea needs to be explored further. Further, one could also choose to optimize the points at which to shift the locations, $\hat{\beta}_j^*$, rather than fixing those point before performing the optimization. In principle, this should provide a more optimal solution. However, we have chosen not to perform this extra step in the optimization for two reasons. The first being this obvious issue of computation time. This extra step could require much more time for the optimal solution to be found. Second, our goal is to find the optimal penalty, i.e. the shape of f_j which does not depend on the location.

5.2 Simulations

Here we present two simulations, one for $p > n$ and one for $n > p$ to show how implementing the GA performs relative to other shrinkage methods in a predictive setting. We simulate IID observations from

$$Y = X\beta + \epsilon$$

where $X \sim MVN_p(0, I_{100})$, $\epsilon = (\epsilon_1, \dots, \epsilon_n)^T$ with $\epsilon_i \sim N(0,1)$, and $\beta = (\beta_1, \beta_2)^T$ and we set $p = 100$, as before. We assume 50% sparsity, so the dimension of both β_1 and β_2 are 50. We take $\beta_1 \sim MVN_{50}(4, I_{50})$ and set $\beta_2 = 0$. We consider $n = 40$ and $n = 150$ and we split the data as described in Sect. 5.1.

The GA will find an optimal penalty as defined by an optimal vector $\alpha_{opt} = (\alpha_{1,opt}, \dots, \alpha_{6,opt})^T$. The entries $\alpha_{j,opt} = \alpha_{j,opt}(\mathcal{D}_{train,\alpha})$ so we are treating the penalty as a hyperparameter in the prior that would be mathematically equivalent to it. The difference from actually estimating a hyperparameter comes from the fact we are only using $\mathcal{D}_{train,\alpha} \subsetneqq \mathcal{D}_{train}$. Given the penalty, we have a potentially new shrinkage method, dependent on a proper subset of $\mathcal{D}_{train}$. So, we can form a posterior using the prior determined from the penalty given the rest of the data. This posterior can be used to generate predictions for $\mathcal{D}_{test}$ that can be compared with the predictions from the other shrinkage methods used in Sect. 3.

Table 5 MSPE for our new GA method and seven other methods

GA	LASSO	RR	EN	AEN	ALASSO	SCAD	MCP
20.26	45.89	22.75	22.75	31.75	33.00	31.31	30.65

5.2.1 GA example $n = 40$

The first example we show is for the case where $n < p$. Here we split the data so that

$$\#(\mathcal{D}_{train}) = \#(\mathcal{D}_{train,\lambda} \cup \mathcal{D}_{train,\beta} \cup \mathcal{D}_{train,\alpha}) = 36$$

with corresponding sample sizes (2, 30, 4) and $\#(\mathcal{D}_{test}) = 4$. All other methods we compare use all of the training data to find estimates of β and λ. Note for comparisons with other methods, those that use glmnet and ncvreg use all 36 observations in the training data to form the predictor. Thus, we ensured that each method used all the training data, providing a fair comparison.

Interestingly, but perhaps not surprisingly, we find

$$\hat{\alpha} = (0, 1, 0, 0, 0, 0)^T$$

which corresponds exactly to RR. This is consistent with the methods that performed the best in the analogous cases in Sect. 3.1. This suggests that when we have few data points relative to explanatory variables, we do not have enough information to obtain an informative prior (in terms of its location and variance) so we default to the prior that makes us retain all the explanatory variables.

The predictive errors for $\#(\mathcal{D}_{test})$ are given in Table 5. We comment that because GA's require a lot of computing time, we have not averaged over many data sets to get the prediction errors reported in this table. However, we believe we have used a large enough population and large enough number of generations that our results are accurate.

This example illustrates that by optimizing over the choice of penalties, we are not guaranteed to find a penalty that is different from an established method (although we argue this is the typical case). The guarantee is only that we will find an optimal penalty for prediction and it is no surprise if there are settings where a well known technique is optimal. The novelty in our GA approach is that it can be used in any linear regression problem and, if properly implemented, will always give the best predictions.

5.2.2 GA example $n = 150$

Now we provide an example for the case where n is slightly larger than p. Splitting the data in this scenario is delicate because we must keep more than 100 observations in $\mathcal{D}_{train,\beta}$ to ensure $n > p$. Accordingly, we set

$$\#(\mathcal{D}_{train}) = \#(\mathcal{D}_{train,\lambda} \cup \mathcal{D}_{train,\beta} \cup \mathcal{D}_{train,\alpha}) = 135$$

Table 6 MSPE for our new GA method and for eight other methods

GA	LM	LASSO	RR	EN	AEN	ALASSO	SCAD	MCP
4.620	5.287	4.750	7.195	4.898	4.907	4.667	4.623	4.705

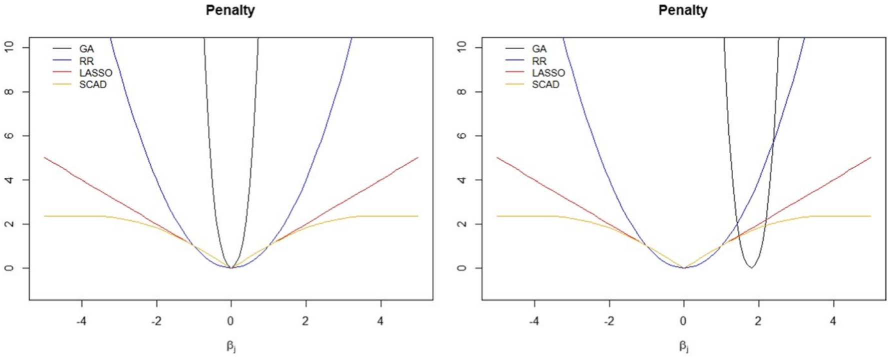

Fig. 8 GA optimal penalty versus standard penalties for zero (left) and nonzero β_j's (right)

with corresponding sample sizes of (9, 113, 13) respectively, and $\#(\mathcal{D}_{test}) = 15$. Parallel to our methodology in Sect. 5.2.1, the methods implemented using glmnet and ncvreg used all 135 observations in the training data to form the predictor.

The GA approach resulted in the chosen penalty being defined by

$$\hat{\alpha} = (1, 0.3, 7, 1, 2.4, 0)^T.$$

The associated prediction error on $\mathcal{D}_{test}$ for each method is given in Table 6. We observe the penalty selected through GA achieves the best predictive error among all methods considered. As in Sect. 5.2.1, we comment that because GA's require a lot of computing time, we have not averaged over many data sets to get the prediction errors reported in this table. However, we believe we have used a large enough population and large enough number of generations that our results are accurate.

Since we found a new (and better) penalty, we have graphed it in Fig. 8 for two β_j's—one where $\beta_j = 0$ (left) and one where $\beta_j \neq 0$ (right). This shows the utility of allowing different f_j's for different β_j's. Since half the parameter values are zero, half are non-zero, and the penalty term depends on the index of the parameter, we find different penalties on different parameters. In both plots, we compare the penalty found from the GA procedure against to the other common penalties RR, LASSO and SCAD. It is obvious that the GA method described in Sect. 2.3 gives two sorts of f_j's. The training data forces the f_j's corresponding to $\beta_j = 0$ to concentrate at zero and forces the f_j's that correspond to nonzero β_j's to concentrate away from zero. This explains the improvement in prediction error seen in Table 6.

Table 7 The first two columns show the best penalties chosen by GA's and by the, stability criterion for the Superconductivity data

n	GA	Stability	Simulation results
75	EN	EN	Not RR
150	RR	RR/EN	Not: RR/LM
500	RR	RR/EN/L	Any
5000	LM	Any	Any

The third column shows the results from simulations with high-sparsity linear models for the given sample size

To end this section, note that our simulations only shows proof of concept; the priors we found here may not be genuinely optimal for prediction because we have not run the GA for many generations with a large population size. Thus we cannot assume the GA has converged. In fact, in both cases here ($n = 40, 150$) we only ran a single generation of the GA and we only used a population size of $M = 150$. However, because of the elitism operation, running the GA longer can never result in a worse predictor and our results show that it can be relatively easy to find a penalty that is better for prediction than established penalties—even if they are not optimal within the class of all penalties with the OP.

5.3 Superconductivity real data example

Here, we revisit the superconductivity data and use a GA to choose the shrinkage method. We run the GA for four different sample sizes namely for $n = 75, 150, 500$ and 5000, randomly sampling n observations from the super conductivity data. Then we use the GA to find an optimal penalty as described in Sect. 5.1. The results are in Table 7. The columns show the optimal penalties chosen by the GA, by the stability criterion applied to the superconductivity data, and the best penalty found from the simulations shown in Sect. 3; $n = 5000$ was done separately.

First, note that the GA and stability columns never disagree. The only lack of agreement is that the GA's always give a unique result whereas the stability criterion allows for multiples methods to perform nearly equally well.

Second, for $n = 150$, the simulation results contradict the results for GA's and stability. Obviously, this is a sample size large enough for the model mis-specification to have a substantial impact: Here, with $p = 81$, n is large enough that the methods detect the difference between the tree-based models found optimal in Hamidieh (2018) and the linear models on which the simulations were based. This is to be expected whenever mis-specification is an imnportant factor. Note that for $n = 75$ there is not enough data to detect mis-specification whereas for $n = 500, 5000$, we are effectively in the asymptotic case. That is, all methods perform as well as the model mis-specification allows. In particular, since Bayesian posteriors are always consistent for the point in model space closest to the true model, all the techniques show the prior washing out making all methods essentially equivalent.

Overall, this reinforces the point that our methods—GA's and stability—are fit for purpose in that they are data driven. They respond to the specifics of the data set because they rely on predictive criteria.

6 Discussion

This paper assumes a predictive stability perspective and within that context shows several results that may be a bit unexpected. First, the OP is not rare; it is actually rather common. A proof for a general result requires little more than what most would regard as regularity conditions. Second, for small n and large p, shrinkage methods did not perform very well even if they have the OP and the true model has reasonable sparsity. Third, on the other hand, if optimal or near optimal penalties are used they give shrinkage methods that often perform noticeably better than the established ones. Our findings indicate that methods having the OP do not perform particularly well for $n < p$ and for $n > p$ the OP is no guarantee that they perform better than methods not having the OP. Fourth, our results suggest there may be a sort of 'OP in terms of increasing sparsity' rather than increasing sample size. However, this intuition needs to be developed because the limit of 'perfect' sparsity gives the trivial model.

Even though the OP is important, it is not always clear how important it is or when it is important in cases where sample sizes are finite. We still think it's better to have the OP than not if only because it gives consistency, asymptotic normality and efficiency. This is especially the case with high sparsity and large n relative to p, but in these cases other methods often perform comparably. When n is small compared to p, the OP is not a useful property, and thus the adaptive methods that have the OP do not perform well. A possible explanation for this is a poor bias variance trade off when $p > n$. It does not seem to be a good idea to use methods that require estimating w_j for each β_j: For $p > n$ we have not seen any example where the adaptive penalty gives better results than its nonadaptive version.

Since the OP requires $n \rightarrow \infty$ whereas n often must be taken as truly finite, we introduce the notion of instability of predictions as a criterion for selecting a penalty or prior. Comparing instability curves is a finite sample check for good predictive performance. For instance, with high sparsity using a linear model by itself is often unstable. In general, quantifying the variability of variable selection when p is large is difficult, so defining instability in terms of the prediction errors seems reasonable. Furthermore, predictive error alone, without the introduction of perturbations in the data, may lead to choosing a method that is less stable than another method. This is seen in the instability curves when one curve crosses over another, appearing to be good at first (with no added noise), but deteriorating quickly with small amounts of added noise. See for example Fig. 2, where in the top left panel, LASSO becomes worse and its curves crosses over others. Further, using instability curves to select a shrinkage method is more robust than simply looking predictive error alone because the instability curves are able to detect both variability and bias. Hence, if a method has small bias initially, but large variability, it may not be preferred to a method with slightly higher bias initially, but much less variability.

Our simulation studies show that as a generality, shrinkage methods tend to perform better in terms of variable selection, and thus prediction, as sparsity increases as well as when n increases. In fact, our simulations showed that regardless of n, as the sparsity increased, the methods seemed to perform roughly equally well. For instance, recall the $n = 75$ simulations in Sect. 3.2. At 90% sparsity, we observed what appeared to be asymptotic convergence of the method with the OP. Of course this situation is not asymptotic as $n < p$, but an increase in sparsity is associated with an increase in efficiency of the methods.

We have presented a methodology that can often be used in practice, i.e. when the true model is not known. Like others who have compared the performance of shrinkage methods, our recommendation may not apply because we cannot identify which scenario (sparsity, signal to noise ratio, dependence structures, etc.) the true model represents. However, we can always generate instability curves. Our real data example in Sect. 4 verifies that our approach is usable and allows us to choose what we deem the most appropriate off the shelf shrinkage method for the given data set.

Since there are infinitely many such choices for penalties that have the OP, we take the subjectivity out of penalty (or prior) selection by using a GA to find an optimal penalty/prior for prediction. When $n > p$ we use the GA approach to find a predictively optimal penalty that has the OP or other asymptotic properties related to the OP for continuous penalties. When $n << p$, we do not search for methods with the OP because we do not benefit from the known asymptotic results. Thus, we search over the class of penalties that are non-adaptive and do not require estimation of many hyper-parameters. In principle, as long as we let the GA run long enough to converge, this approach can never do worse that simply choosing a standard shrinkage method. In fact, we have examples where the GA approach does better than others; when the GA approach selects the best among standard methods, we can infer the standard method was the right choice.

Another way to look at the procedure in Sect. 5 is that when we find a penalty/prior based on the data we are producing an approximation to a predictively optimal posterior given the training data that can then be used with the log-likelihood. Thus, the predictive improvement comes from the efficiency of the way the posterior uses the data with an optimal prior.

We close with another heuristic that seems to be borne out by our results. Namely, we associate corners and other points of non-differentiability in the penalty with setting parameter values equal to zero in finite samples. Recall that minimizing

$$\sum_{i=1}^{n} L(y_i - x_i^T \beta) + n\lambda \sum_{j=1}^{p} w_j f_j(\beta_j)$$

is equivalent to minimizing

$$Q = \sum_{i=1}^{n} L(y_i - x_i^T \beta)$$

subject to the constraint $\sum_{j=1}^{p} w_j f_j(\beta_j) \leq R \in \mathbb{R}$ where R typically decreases as λ increases. Denote the constraint region by

$$D = \left\{ \sum_{j=1}^{p} w_j f_j(\beta_j) \leq R \right\}.$$

Since D is closed and compact, the Krein-Milman theorem (see Royden and FitzPatrick 2010) gives that D is the closed convex hull of its extreme points, i.e., $D = CCH(D_{ext})$. For reasonable choices of f_j, D is defined by the intersection of regions of the form

$$U_k(\beta_1, \dots, \beta_p) \leq \alpha_k, k = 1, \dots, p,$$

where the U_k's are defined from the f_j's and w_j's. When our goal is optimizing Q over D, it is often the case that the optima occur at extreme points of D. When the extreme points of D are on the coordinate axes we will find at least some of β_j's are zero. Indeed, if Q is convex and continuous on an open set containing D, then Q often attains its minimum over D on a 'face' of D and the exact point where the optima occur may lie at the intersections of some or all of the $U_k = \alpha_k$; this defines a subset of the extreme points of D. This is well-established for the case of linear optimization with linear constraints. Indeed, if Q is minimized for at least one extreme point of D that lies on a coordinate axis then at least some β_j's will be set to zero. This means that any locally convex penalty with a 'corner' on a coordinate axis will perform nontrivial variable selection if R is small enough. We conjecture a converse to this statement will hold, too.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00180-023-01342-8.

Acknowledgements This work was completed utilizing the Holland Computing Center at the University of Nebraska, which receives support from the Nebraska Research Initiative.

Funding The authors were supported by the Agricultural Research Division in the Institute of Agriculture and Natural Resources, University of Nebraska-Lincoln, and the University of Nebraska Program of Excellence in Computational Sciences.

References

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

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

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

Fan J, Lv J (2013) Asymptotic equivalence of regularization methods in thresholded parameter spaces. J Am Stat Assoc 108:1044–1061

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

Givens G, Hoeting J (2013) Computational statistics, 2nd edn. Wiley, Hoboken

Hamidieh K (2018) A data-driven statistical model for predicting the critical temperature of a superconductor. J Am Stat Assoc 96:1348–1360

Hastie T, Tibshirani R, Tibshirani RJ (2020) Best subset, forward stepwise or lasso? analysis and recommendations based on extensive comparisons. Stat Sci 35(4):579–5920

Hoerl A (1962) Application of ridge analysis to regression problems. Chem Eng Prog 58:54–59

Luo X, Stefanski L, Boos D (2006) Variable selection via concave penalized likelihood and its oracle properties. Technometrics 48:165–175

Qian W, Yang Y (2013) Model selection via standard error adjusted adaptive lasso. Ann Inst Stat Math 65:295–318

Royden H, FitzPatrick P (2010) Real analysis, 4th edn. Prentice-Jall, Hoboken

Rudolph G (1996) Convergence of evolutionary algorithms in general search spaces. In: Proceedings of IEEE international conference on evolutionary computation, pp 50–54. https://doi.org/10.1109/ICEC.1996.542332

Sjöburg J, Ljung L (1992) Overtraining, regularization, and searching for minimum in neural networks. In: Proceedings of the 4th IFAC symposium on adaptive systems in control and signal processing, pp 73–78

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

Wang H, Li G, Giang G (2007) Robust regression shrinkage and consistent variable selection through the LAD-Lasso. J Bus Econ Stat 25:347–355

Wang W, Mukherjee S, Richardson S et al (2020) High-dimensional regression in practice: an empirical study of finite-sample prediction, variable selection and ranking. Stat Comput 30:697–719

Wang Z, Zhu Z, Yu C (2020b) Variable selection in macroeconomic forecasting with many predictors. Submitted arXiv:2007.10160

Willighagen E, Ballings M (2015) genalg: R based genetic algorithm. https://CRAN.R-project.org/package=genalg, R package version 0.2.0

Zhang CH (2010) Nearly unbiased variable selection under minimax concave penalty. Ann Stat 38:894–942

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

Zou H (2006) The adaptive lasso and its oracle properties. J Am Stat Assoc 101:1418–1429

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

Zou H, Zhang HH (2009) On the adaptive elastic-net with a diverging number of parameters. Ann Stat 37:1733–1751

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.