Statistica Sinica **22** (2012), 1021-1040
doi:http://dx.doi.org/10.5705/ss.2010.023

ROBUST COMBINATION OF MODEL SELECTION METHODS FOR PREDICTION

Xiaoqiao Wei and Yuhong Yang

University of Minnesota

Abstract: One important goal of regression analysis is prediction. In recent years, the idea of combining different statistical methods has attracted an increasing attention. In this work, we propose a method, l_1-ARM (adaptive regression by mixing), to robustly combine model selection methods that performs well adaptively. In numerical work, we consider the LASSO, SCAD, and adaptive LASSO in representative scenarios, as well as in cases of randomly generated models. The l_1-ARM automatically performs like the best among them and consequently provides a better estimation/prediction in an overall sense, especially when outliers are likely to occur.

Key words and phrases: Adaptive LASSO, ARM, combining model selection methods, LASSO, SCAD.

1. Introduction

Model selection with a number of predictors has been an exciting research area. Methods have been proposed in recent years to conduct variable selection with computationally feasible algorithms, sometimes maintaining familiar statistical properties of traditional information criteria. These methods have been increasingly used, and numerous numerical results demonstrate their advantages in some settings. With multiple model selection tools available, a question a statistics user faces is: How should one select a model selection method for his/her data?

Obviously, we should not expect a single choice to perform best in different scenarios. Some insights have been offered in the literature on this issue. For example, sparsity of the underlying regression function in terms of the number of explanatory variables involved is regarded as a key feature that makes some methods perform better than others. Fan and Li (2001) pointed out that in terms of model error, the SCAD outperforms the LASSO (Tibshirani (1996)) when the model noise level is low, while the LASSO does better than the SCAD when the noise level is high. Zou (2006) observed that the LASSO outperforms the SCAD and adaptive LASSO in model error when the signal-noise-ratio (SNR) is small, while the SCAD and adaptive LASSO methods do better than the LASSO when

the SNR is large. However, in applications, with the model noise level and true regression function unknown, much more needs to be done both theoretically and through systematic numerical investigations before satisfactory conclusions can be reached to provide statistical characterizations of the data that determine the relative performance of the different methods. Intuitively, if one gets to know when to use which model selection method, there is an advantage if one considers a list of distinct model selection rules so that at least one of them is optimal or well-behaving for the unknown underlying data generating process (DGP).

For moderate or high dimensional regression problems, however, with a small or moderate sample, the task of identifying the best among several model selection methods is typically very difficult. There is a serious challenge to realize the potential advantage of sharing strengths of a number of model selection rules in a pool. For the goal of prediction or estimating the regression function (in contrast to identifying the important variables), as is the focus in our paper, one approach is to combine the model selection methods by a proper weighting of the predictions or estimates from them. If the combination leads to a performance similar or close to the best method in each scenario of the underlying DGP, the combined estimator or prediction can outperform all the candidate model selection methods in repeated applications across different scenarios of the DGP. This will be seen in our numerical results later.

Combining regression procedures has been studied and allows various interesting theoretical properties. Oracle inequalities show that properly combining arbitrary regression procedures leads to a risk close to the best among a target class of combinations of the candidate estimators/predictions plus a minimax-rate optimal "price of combining" that reflects the largeness of the class of allowed combinations. See Chen and Yang (2010) for a literature review. Successes of combining different predictions in applications have prompted more interest. For instance, in the well-known Netflix competition, an ensemble of different methods was employed by top teams (see, e.g., `http://www.netflixprize.com/leaderboard`).

The previous theoretically proven combining methods, in e.g., Yang (2001) and Catoni (2004), use quadratic-type loss in determining weights for the candidates and show that the combined regression estimator achieves the best performance offered by the candidates in an accumulated risk. The quadratic-type loss is also used in combining methods of Juditsky and Nemirovski (2000), Yang (2004), and Tsybakov (2003) for larger target classes of combinations.

The mathematically convenient quadratic loss for weighting regression estimators works very well under Gaussian noise. However, when the noise has a heavier tail, as commonly occurs in practice, a few outliers can destabilize the weights. A robust combination of estimates or predictions is thus sought.

In this paper, we propose a robust method, called l_1-ARM, to combine regression estimates/predictions from a list of model selection methods. Quadratic loss in the ARM (adaptive regression by mixing, see Yang (2001)) is replaced by absolute loss, and an oracle risk bound is presented that allows a screening step to be incorporated to remove poor model selection methods; this can be helpful when a large number of methods are considered. In our numerical work, we focus on combining the LASSO, SCAD (Fan and Li (2001)), and adaptive LASSO (Zou (2006)) in the linear regression setting. The results are highlighted as follows.

1. Several representative linear expressions with different degrees of sparsity and multiple noise levels are considered in comparing the performance of the model selection methods and l_1-ARM. The results show that the l_1-ARM performs like the best model selection method in the different scenarios.
2. The results show the advantage of the l_1-ARM over the original ARM: if the noise is Gaussian, they perform similarly; if the noise has a heavy tail, the l_1-ARM performs significantly better.
3. For randomly generated models, we find that the relative performances of the LASSO, SCAD, and adaptive LASSO depend on the sparsity of true regression function and the SNR.

The paper is organized as follows. In Section 2 we propose the l_1-ARM algorithm. In Section 3 we investigate the LASSO, SCAD, adaptive LASSO, and l_1-ARM via various simulation settings and data examples. In Section 4 we present a theoretical result for the l_1-ARM. We give concluding remarks in Section 5. The proof of the theorem of Section 4 is in the Appendix.

2. The Proposed Method

Consider a general regression problem $Y_i = f(\mathbf{x}_i) + \varepsilon_i$, $i = 1, \ldots, n$, $f(\cdot)$ the true regression function. For estimating $f(\cdot)$, a number of models are considered, for example, the collection of all the subset models with terms chosen from a list of predictors (with possible transformations and/or interaction terms). We focus on linear models in our numerical work, although our proposed method and the theoretical result are applicable more generally. We apply K model selection methods on the data: model selection method j yields an estimator $\widehat{f}_{j,n}(\mathbf{x})$. Denote the set of the K candidate methods by Γ.

2.1. The l_1-ARM algorithm

Yang (2001) proposed the ARM algorithm for combining a group of regression models. Yuan and Yang (2005) extended the ARM with model screening. In the ARM, the l_2-norm was used in the core step to apportion the weights to

each candidate. Under quadratic loss, if the underlying model generates outliers the weight of the best candidate model can easily be diluted and other models can unexpectedly obtain more weight. We propose the l_1-ARM in the hope that it performs similarly as the ARM when the noise is normally distributed and outperforms the ARM when the noise distribution has a heavy tail.

The l_1-ARM algorithm is as follows.

Step 1. Apply the model selection methods to the data to get their recommended models and regression estimates $\hat{f}_{j,n}(\mathbf{x})$.

Step 2. Split the data into two parts, $Z^{(1)} = (\mathbf{x}_i, Y_i)$, $1 \le i \le n/2$, and $Z^{(2)} = (\mathbf{x}_i, Y_i)$, $n/2 + 1 \le i \le n$.

Step 3. Based on $Z^{(1)}$, compute the mean absolute prediction error $\hat{d}_j = (2/n) \sum_1^{n/2} |Y_i - \hat{f}_{j,n}(\mathbf{x}_i)|$ for each candidate model j.

Step 4. For each model $j \in \Gamma$, predict Y_i by $\hat{f}_{j,n}(\mathbf{x}_i)$ for $Z^{(2)}$. Compute

$$D_j = \sum_{i=n/2+1}^{n} |Y_i - \hat{f}_{j,n}(\mathbf{x}_i)|.$$

Step 5. Compute the convex weight for model j as

$$W_j = \frac{\hat{d}_j^{-n/2} \exp(-\eta D_j / \hat{d}_j)}{\sum_{k \in \Gamma} \hat{d}_k^{-n/2} \exp(-\eta D_k / \hat{d}_k)}.$$

Step 6. Randomly permute the order of the data $N - 1$ times. Repeat Step 2 – Step 5 and let $W_{j,r}$ denote the weight of method j computed at the rth permutation for $0 \le r \le N - 1$. Let $\hat{W}_j = (1/N) \sum_{r=0}^{N-1} W_{j,r}$.

Step 7. Let

$$\hat{f}_n(\mathbf{x}) = \sum_{j \in \Gamma} \hat{W}_j \hat{f}_{j,n}(\mathbf{x})$$

be the final l_1-ARM estimate of the true regression function f. At a new $\mathbf{x}'$, the combined prediction is $\hat{Y} = \hat{f}_n(\mathbf{x}')$.

Remarks.

1. When one considers a large number of model selection methods, it may be helpful to combine a reduced candidate set rather than the full set Γ both for better accuracy and for saving computation cost. We address this issue later in Section 4.
2. For ensuring optimal rate of convergence, a fixed splitting ratio such as half/half works and, in our experience, typically works well for regression estimation/prediction.

3. Absolute error is used here so that outliers have less influence on the weights.
4. In Step 5, there is a tuning parameter η to control the degree of reliance of weighting on the predictive performance (note that when $\eta = 0$, the prediction errors D_j have no effect at all on weighting). In our numerical work, we set $\eta = 1$ and it worked well.

2.2. Combining SCAD, LASSO, and adaptive LASSO

In the numerical work of this paper, we focus on combining some recently proposed selection methods: the SCAD, LASSO, and adaptive LASSO (a-LASSO). We describe some details about applying them.

There are two tuning parameters a and λ in the SCAD. Following Fan and Li (2001), we set $a = 3.7$. As suggested by Zou (2006), we estimate the weight vector in the a-LASSO by $\hat{\mathbf{w}} = 1/|\hat{\boldsymbol{\beta}}|^\gamma$, where $\hat{\boldsymbol{\beta}}$ is the OLS estimator and γ is selected from $\{.5, 1, 2\}$ by fivefold cross validation. The two selection options of the tuning parameter λ of the three methods are as follows. Denote the full data set by D and the testing set by D_l, $l = 1, \ldots, 5$. For each λ and l, we get the estimate $\hat{\boldsymbol{\beta}}_\lambda^l$ using the training set $D - D_l$. Then the fivefold cross validation selection minimizes

$$CV_\lambda = \sum_{l=1}^{5} \sum_{(\mathbf{x}_i, Y_i) \in D_l} \left(Y_i - f(\mathbf{x}_i; \hat{\boldsymbol{\beta}}_\lambda^l) \right)^2.$$

The BIC selection minimizes

$$BIC_\lambda = \log \hat{\sigma}_\lambda^2 + \frac{df_\lambda \log(n)}{n},$$

where df_λ is the number of nonzero coefficients of the fitted model (see, Wang, Li, and Tsai (2007); Zou (2008)).

In numerical work, the LASSO and a-LASSO are computed by using the R package *lars*, where the optimal λ is chosen from 100 candidates along the entire solution path by using fivefold cross validation and BIC selections, respectively. The SCAD is computed by using the one-step SCAD program provided by Zou and Li (2008), where the optimal λ is chosen from 100 discretized values by using fivefold cross validation and BIC selections, respectively. Thus the SCAD in this work is a one-step SCAD.

The comparison of different estimators is done under regression estimation loss $E(f(\mathbf{x}) - \hat{f}(\mathbf{x}))^2$ via simulation, where the expectation is taken on the new observation $\mathbf{x}$ that has the same distribution as the data. For the empirical comparison usingl data sets, we consider predictive mean squared error as an objective measure.

Table 1. Performance comparison for the high sparsity case.

	Scad_{cv}	Scad_{bic}	Lasso_{cv}	$\text{Lasso}_{\text{bic}}$	$\text{Alasso}_{\text{cv}}$	$\text{Alasso}_{\text{bic}}$	Arm	l_1-Arm
$\sigma = 1$	0.25	0.22	0.60	0.53	0.44	0.37	0.34	0.35
$\sigma = 3$	0.43	0.36	0.61	0.53	0.49	0.39	0.39	0.41
$\sigma = 5$	0.58	0.54	0.60	0.51	0.55	0.55	0.49	0.49

3. Numerical Results

We focus on the performance of the selection and combining methods in linear regression: $Y = \mathbf{x}^T\boldsymbol{\beta} + \varepsilon$. Assume that $\mathbf{x}$ follows a multivariate normal distribution with zero mean and covariance matrix as defined, and that the random noise ε is iid $N(0, \sigma^2)$ or is a contaminated normal.

3.1. Some representative examples

Assume there are 12 predictor variables in $\mathbf{x}$ and that the covariance between x_k and x_l is $\rho^{|k-l|}$ with $\rho = 0.5$, $1 \le x_k, x_l \le 12$. The sample size n is set to be 50 and 100, and the performance of the competing methods is evaluated at 1,000 independently generated observations from the same distribution. We replicate each estimation process 100 times and, in each replication, we set $N = 100$ to calculate the combining weights. We consider the relative loss, the ratio of the loss of a competing method over that of the OLS estimate from the full model, and the median of the relative loss over the 100 replicates is reported as in Fan and Li (2001). Another measure, the mean of the relative loss, gives similar results, but the combining methods usually have smaller standard errors than the selection methods (not reported here due to space limitation), indicating that they are more robust. Since the results for n equal to 50 and 100 are similar, we present only the $n = 100$ case.

High sparsity. In this example, $\boldsymbol{\beta}$ is $(3, 1.5, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0)^T$, identical to a model investigated in Zou and Li (2008). The underlying model here is sparse (3 nonzero coefficients out of 12 potential predictors). In Table 1, the SCAD methods perform the best when the model noise level is low. However, when the model noise level is high, the $\text{LASSO}_{\text{BIC}}$ is more accurate than the SCAD and a-LASSO. Note that BIC consistently outperforms fivefold CV in this case. The two combining methods are automatically close to or outperform the best selection method.

To have an idea of the uncertainty of the reported median relative losses in the table, we repeated the whole process 100 times. The standard errors of the reported medians in Table 1 are between 0.003 and 0.006. In contrast, the computationally less costly bootstrapping of the 100 relative losses gives substantially larger standard error estimates (ranging between 0.021 and 0.035)

Table 2. Performance comparison for the low sparsity case.

	Scad_{cv}	Scad_{bic}	Lasso_{cv}	$\text{Lasso}_{\text{bic}}$	$\text{Alasso}_{\text{cv}}$	$\text{Alasso}_{\text{bic}}$	Arm	l_1-Arm
$\sigma = 1$	0.79	0.76	0.89	0.92	0.83	0.77	0.77	0.77
$\sigma = 3$	1.00	1.08	0.85	0.90	0.98	1.00	0.91	0.92
$\sigma = 5$	1.00	1.09	0.79	0.84	0.97	0.94	0.81	0.82

Table 3. Performance comparison for the non-sparsity case.

	Scad_{cv}	Scad_{bic}	Lasso_{cv}	$\text{Lasso}_{\text{bic}}$	$\text{Alasso}_{\text{cv}}$	$\text{Alasso}_{\text{bic}}$	Arm	l_1-Arm
$\sigma = 1$	1.00	1.10	1.00	1.00	0.98	0.98	1.00	1.00
$\sigma = 3$	1.00	1.08	0.80	0.86	1.01	1.04	0.86	0.86
$\sigma = 5$	0.90	1.06	0.63	1.18	0.72	0.93	0.69	0.68

and thus is not reliable for our problem. Due to high expense in simulating the median relative losses, henceforth, we report only the median relative losses as in Fan and Li (2001).

Low sparsity. In this example, $\boldsymbol{\beta}$ is $(3, 1.5, 0, 1.1, 2, 0, 0.9, 0.8, 0.6, 0, 0, 0)^T$, expanding the model complexity by adding more nonzero coefficients. In Table 2, the performance of the a-LASSO is comparable to or better than the SCAD. The LASSO performs relatively better than the others when the model noise level is moderate or high. It can be seen that fivefold CV starts to show its advantage in some settings. The two combining methods tend to be close to the best procedure, and they perform similarly.

Non-sparsity. In this example, $\boldsymbol{\beta}$ is $(0.5, 0.5, 0, 0.5, 0.5, 0.5, 0, 0.5, 0.5, 0.5, 0, 0.5)^T$. In Table 3, when the model noise level is low, all three selection methods perform similarly to the OLS estimator. When the model noise level is moderate, the SCAD and a-LASSO perform worse than the LASSO. When the model noise level is high, all three selection methods with fivefold CV do better or much better than the OLS estimator, but those with BIC give poor results, much worse than those from fivefold CV. The two combining methods automatically perform like the best procedure, and their performances are very close to each other.

Heavy-tailed noise case. We investigate the robustness of the selection and combining methods. As in Fan and Li (2001), let ε be mixed with 90% standard normal and 10% standard Cauchy distributions in the examples above. In Table 4, the selection methods show different rankings in the three examples: the SCAD does the best in the first example, the a-LASSO the best in the second, and the LASSO the best in the third. We observe that the strengths of the selection methods are weaker when the degree of the model sparsity is reduced. BIC favors sparse models, while fivefold CV favors non-sparse models with the existence of outliers. Unlike the previous examples, the two combining methods perform differently, and the l_1-ARM shows a significant improvement over the ARM.

Table 4. The robustness of the selection and combining methods.

	Scad_{cv}	Scad_{bic}	Lasso_{cv}	$\text{Lasso}_{\text{bic}}$	$\text{Alasso}_{\text{cv}}$	$\text{Alasso}_{\text{bic}}$	Arm	l_1-Arm
Example 1	0.32	0.32	0.54	0.56	0.40	0.42	0.39	0.32
Example 2	0.89	0.86	0.86	0.86	0.83	0.78	0.80	0.73
Example 3	1.00	1.05	0.89	0.96	0.99	1.01	0.99	0.93

Table 5. Performance comparison for highly correlated predictors

Aic	Bic	Scad_{cv}	Scad_{bic}	Lasso_{cv}	$\text{Lasso}_{\text{bic}}$	$\text{Alasso}_{\text{cv}}$	$\text{Alasso}_{\text{bic}}$	Arm	l_1-Arm
0.84	0.74	1.00	1.00	0.99	0.93	1.00	0.98	0.79	0.79

Highly correlated predictors. For establishing consistency and efficiency properties of the LASSO, a-LASSO, and the one-step SCAD, regularity conditions are used, one of which is that the design matrix behaves nicely (see, e.g., Zou (2006), Zhao and Yu (2006), Meinshausen and Bühlmann (2006), Zou and Li (2008), and Zhang and Huang (2008)). See Lv and Fan (2009) for discussion on differences of conditions for l_1 and concave penalties. In this example, we simulated a model in the opposite direction and compare the performance of the above methods plus forward selections by AIC and BIC. The coefficients of this model are those in the second example. The predictors $\mathbf{x}_i$, $i = 1, \ldots, n$, follow a multivariate normal distribution with mean $\mathbf{0}$ and a covariance matrix that is a random realization of a Wishart distribution with $df = 12$ and scale matrix the identity. The maximum eigenvalue of the covariance matrix was 46.032, while the minimum eigenvalue was 0.0002. The error ε followed a standard normal distribution. The other simulation settings remain the same as in the previous examples.

In Table 5, the SCAD and a-LASSO show no advantage compared to the OLS. For instance, the SCAD with BIC selection on average generates two zero coefficients for this model and only one of them is correct. The forward selections by AIC and BIC work more favorably than the other selection methods in this situation. As before, the performance of the combining methods is close to that of the best candidate.

3.2. Randomly generated models

Relative performance. The purpose of random model settings is to try to get an unbiased understanding on the competing methods. We randomly generated 100 models. The number of zero coefficients was uniformly distributed from 2 to 8, and their orders in the 12 potential predictors were also uniformly distributed. The nonzero coefficients were uniformly on [0, 3]. Other settings remain the same as in the previous examples. For each model, we calculated the mean of the relative losses. The median of the mean relative losses of the 100 models

Table 6. Performance comparison based on randomly generated models.

	Scad_{cv}	Scad_{bic}	Lasso_{cv}	$\text{Lasso}_{\text{bic}}$	$\text{Alasso}_{\text{cv}}$	$\text{Alasso}_{\text{bic}}$	Arm	l_1-Arm
$\sigma = 1$	0.81	0.85	0.92	0.97	0.87	0.77	0.79	0.79
$\sigma = 3$	0.93	0.99	0.89	0.95	0.97	0.85	0.81	0.82
$\sigma = 5$	1.00	1.12	0.86	0.97	0.96	0.92	0.83	0.83
heavy tail	0.78	0.84	0.85	0.93	0.84	0.74	0.77	0.69

is shown in Table 6. The a-LASSO with the BIC selection performed the best with $\sigma = 1, 3$, and the heavy-tail noise case. The LASSO with the fivefold CV selection performed the best with $\sigma = 5$. Interestingly fivefold CV did better than BIC for the SCAD and LASSO, while BIC did better than fivefold CV for the a-LASSO in the random model settings. The SCAD was close to the best under low or heavy-tail noise. The two combining methods performed well consistently. When the underlying model noise was normally distributed, they performed almost identically. However, when outliers were present, the l_1-ARM had the edge.

How do the SNR and sparsity affect the relative performances? In the simulation above, the number of zero coefficients was treated as a measure of the model sparsity. We estimated the variance of the mean function $\mathbf{x}^T\boldsymbol{\beta}$, V_s, with sample size equal to $1,100$ and then obtained the SNR of the model by taking the ratio of V_s/σ^2. To get some insight on how the SNR and sparsity are associated with the performance of the three selection methods, we regressed the mean relative losses of the random models on the SNR and sparsity. To save space, we only consider the case in which the tuning parameters were selected by BIC with $\sigma = 3$. The SNR of the 100 random models ranged from 0.94 to 13.2. For ease of notation, the subscript bic is omitted in Table 7. We also consider the ratios of the mean relative losses among the three selection methods as the response variables. Table 7 gives the coefficients of the SNR and sparsity of each model (the intercept is not presented here). For each model, we check the linear model assumptions by applying the diagnostic means (e.g., residual plots). It shows that all the simple linear models look proper. Note that all of these coefficients are significant at $\alpha = 0.01$ level and that the interaction effects of these two predictors in these models are not significant. Thus, in our setting, the SNR and sparsity have "additive" effects on the performance of these selection methods.

For the first three rows of Table 7, all coefficients of sparsity are negative, which indicates the three methods (relative to the full model) perform better with sparse models. The SNR coefficients of the SCAD and a-LASSO are negative, but that of the LASSO is positive, which indicates the LASSO does better when the SNR is small. For the second three rows of Table 7, all coefficients of the SNR

Table 7. The sparsity vs the signal-to-noise ratio.

	SNR	sparsity
Scad	-0.026	-0.131
Lasso	0.004	-0.035
Alasso	-0.008	-0.092
$\frac{\text{Scad}}{\text{Lasso}}$	-0.033	-0.104
$\frac{\text{Scad}}{\text{Alasso}}$	-0.020	-0.033
$\frac{\text{Alasso}}{\text{Lasso}}$	-0.013	-0.066

and sparsity are negative. It seems that for the large SNR and high sparsity cases the SCAD dominates the LASSO and a-LASSO while the a-LASSO dominates the LASSO. This suggests that the accuracy of the SCAD relative to LASSO and a-LASSO increases with higher SNR and sparsity. The same can be said of the a-LASSO relative to LASSO.

3.3. High-dimensional cases

Forty predictors. Consider high-dimensional cases with the number of the predictors at 40. Assume that the covariance between x_k and x_l is $\rho^{|k-l|}$ with $\rho = 0.75$, $1 \le x_k, x_l \le 40$. In Case 1, there are 5 nonzero coefficients, in Case 2, there are 10 nonzero coefficients, and in Case 3, there are 20 nonzero coefficients. All nonzero coefficients were uniform $[0, 3]$ and their orders were uniformly distributed in the model. We considered two scenarios of the model noise for each case: $\sigma = 2$, and having a heavy tail as in the previous examples. We repeated each case 50 times. To ease the computation burden, for each replication, we generated 50 random samples and, for each sample, we set $N = 50$ to get the combining weights. The other simulation settings remain as before. The median of the relative losses over the 50 replicates is presented in Table 8, where the two rows in each case correspond to the two model noises, respectively. For the different high-dimensional cases, we can see that the selection methods have different performances. The two combining methods are close to the best performance among the candidates when the model noise is normal. With heavy-tail noise, the l_1-ARM outperforms all the selection methods and the ARM.

Eighty predictors. A useful strategy for dealing with high-dimensional cases is to add a screening step, so only significant variables are included in the later modeling process. In this example, 80 variables have a distribution as in the above example. The first eight coefficients are $(2, 2, 2, 2, 2, 2, 2, 2)^T$; the remaining coefficients are zero. We applied Sure Independence Screening (Fan and Lv (2008)) to reduce the dimension from 80 to 40. Table 9 shows the median relative losses over 50 random samples of the competing methods. With the screening step,

Table 8. Performance comparison for the high-dimensional data.

	Scad_{cv}	Scad_{bic}	Lasso_{cv}	$\text{Lasso}_{\text{bic}}$	$\text{Alasso}_{\text{cv}}$	$\text{Alasso}_{\text{bic}}$	Arm	l_1-Arm
Case 1	0.22	0.22	0.33	0.30	0.28	0.28	0.22	0.22
	0.24	0.24	0.30	0.28	0.25	0.25	0.25	0.18
Case 2	0.42	0.41	0.52	0.42	0.57	0.42	0.37	0.37
	0.40	0.41	0.46	0.39	0.46	0.36	0.39	0.32
Case 3	0.77	0.75	0.75	0.62	1.35	0.68	0.62	0.62
	0.73	0.73	0.67	0.60	1.11	0.63	0.64	0.55

Table 9. Performance comparison for the high-dimensional data with screening.

	Scad_{cv}	Scad_{bic}	Lasso_{cv}	$\text{Lasso}_{\text{bic}}$	$\text{Alasso}_{\text{cv}}$	$\text{Alasso}_{\text{bic}}$	Arms	l_1-Arms
$\sigma = 1$	0.14	0.13	0.47	0.29	0.22	0.26	0.16	0.15
$\sigma = 3$	0.43	0.47	0.31	0.29	0.39	0.38	0.33	0.32
heavy tail	0.32	0.33	0.33	0.31	0.29	0.33	0.29	0.22

Table 10. Performance comparison for the high-dimensional data ($p > n$).

	Scad	Lasso	Mcp	Sica	Arm	l_1-Arm
$\sigma = 1$	0.16	0.33	0.11	0.31	0.15	0.15
$\sigma = 3$	3.25	2.85	3.41	1.48	1.43	1.43
heavy tail	0.55	0.69	0.50	0.43	0.38	0.37

the selection methods had different finite sample performances. The combining methods performed as if they knew which selection method was best for a specific situation. When there were outliers, the l_1-ARMS showed a clear advantage compared to the selection methods and the ARMS.

Case with $p > n$. Consider a high-dimensional case with $n = 100$ and $p = 300$. Assume that the covariance between x_k and x_l is $\rho^{|k-l|}$ with $\rho = 0.5$, $1 \le x_k, x_l \le p$. The true coefficients are zero except for the first $(3, 1.5, 0, 1.1, 2, 0, 0.9, 0.8, 0.6)^T$. The addaptive LASSO is not applicable for this situation. Instead, we consider SCAD, LASSO, SICA (Lv and Fan (2009)), and MCP (Zhang (2007)), which all can handle p > n cases. We applied the selection methods with a R package 'EZPATH' (Yang and Zou (2010), `http://www.stat.umn.edu/~yi`) and used the default non-convex penalty parameters in the R package for SCAD, MCP, and SICA. The tuning parameter of each method was selected by fivefold CV. Table 10 shows the performances of the selection and combining methods. For the $p > n$ case, the relative performances of the selection methods depend on the noise situation, and no single choice works consistently well. In contrast, the combining methods are always among the best. When there are outliers, they significantly outperform the selection methods.

Table 11. Results for Data Example 1.

Scad_{cv}	Scad_{bic}	Lasso_{cv}	$\text{Lasso}_{\text{bic}}$	$\text{Alasso}_{\text{cv}}$	$\text{Alasso}_{\text{bic}}$	Arm	l_1-Arm
0.96	0.99	0.85	0.85	0.79	0.82	0.86	0.82

Table 12. Results for Data Example 2.

	Scad_{cv}	Scad_{bic}	Lasso_{cv}	$\text{Lasso}_{\text{bic}}$	$\text{Alasso}_{\text{cv}}$	$\text{Alasso}_{\text{bic}}$	Arms	l_1-Arms
$n = 60$	0.77	0.82	0.91	0.84	0.91	0.93	0.90	0.77
$n = 120$	0.93	0.93	0.96	0.94	0.95	0.95	0.95	0.90

3.4. Data examples

Data Example 1. This data set is from the Berkeley Guidance Study (Weisberg (1985, pp.55-57)). There are ten predictors, and the response variable is a measurement of fatness for 32 girls at age 18. The training sample size was 26 and the competing methods were evaluated at the remaining 6 observations. This process was repeated 100 times with random data splittings. The medians of the relative prediction MSE, i.e., the ratio of the MSE of these methods over that of the OLS estimator, are shown in Table 11. The a-LASSO with the fivefold CV selection performed the best, and the l_1-ARM performed similarly.

Data Example 2. This data set is taken from Johnson (1996). It originally contained 17 predictors and 252 observations. We took 16 predictors, removing a body density variable that varied narrowly, and 251 observations since the 42nd observation was apparently incorrect. The training sample size n was 60 and 120, respectively. The medians of the relative prediction MSE are represented in Table 12. When $n = 60$, the SCAD with the fivefold CV selection performed best among the selection methods, while the l_1-ARM performed similarly. When $n = 120$, the two SCAD selections outperformed the other selections by a small margin. The l_1-ARM even improved over the SCAD methods. Note that when the training sample size was increased, the OLS method performed better so that the relative advantage of the selection methods decreased. In these two examples, the l_1-ARM outperformed the ARM.

4. Theory

Differently from some previous work on consistency of model selection, our theoretical focus is on prediction risk. Assume that $(\mathbf{x}, Y)$, $(\mathbf{x}_i, Y_i)$, $1 \le i \le n$, are independent and identically distributed. Let $\varepsilon = Y - f(\mathbf{x})$, where f is the regression function (conditional mean of Y given $\mathbf{x}$). To derive the theoretical result, we study a somewhat different version of the algorithm in Section 2. The differences are: a screening step is allowed to reduce the candidate model selection methods to be combined, saving computational cost and/or improving prediction accuracy; model selection methods are applied to part of the data to

come up with the models to be combined, as this is mathematically tractable for theoretical investigation in terms of independence; weights are sequentially averaged. We recommend the earlier algorithm (with screening if so desired) in practice. The discrepancies between the practical and theoretical algorithms are due to practical considerations and mathematical tractability, and they do not have fundamental inconsistencies. As seen in Section 3, the practical algorithm worked well in simulations and data examples.

We first split the data into $Z^{(1)}$ and $Z^{(2)}$. We obtain $\hat{\boldsymbol{\beta}}_j$ and $\hat{d}_j$ on $Z^{(1)}$ and D_j on $Z^{(2)}$, respectively, for each candidate model j. Let $\hat{f}_j$ be the estimate of the regression function based on $Z^{(1)}$ from method j, $1 \le j \le K$. Consider a screening procedure that is applied on $Z^{(1)}$ to get a reduced list of candidate model selection methods, denoted by Γ_s, with size K_s. Note that K_s is allowed to be random (depending on $Z^{(1)}$). For $i = n/2 + 1$, let $W_{j,i} = 1/K_s$ for $j \in \Gamma_s$ and for $n/2 + 1 \le i \le n$, let

$$W_{j,i} = \frac{(\hat{d}_j)^{-(i-n/2-1)} \exp(-\eta \sum_{l=n/2+1}^{i-1} |Y_l - \hat{f}_j(\mathbf{x}_l)|/\hat{d}_j)}{\sum_{k\in\Gamma_s} (\hat{d}_k)^{-(i-n/2-1)} \exp(-\eta \sum_{l=n/2+1}^{i-1} |Y_l - \hat{f}_k(\mathbf{x}_l)|/\hat{d}_k)}.$$

Define

$$\tilde{W}_j = \frac{1}{n/2} \sum_{i=n/2+1}^{n} W_{j,i},$$

and let

$$\tilde{f}(\mathbf{x}) = \sum_{j\in\Gamma_s} \tilde{W}_j \hat{f}_j(\mathbf{x})$$

be the combined estimator.

Condition 1: The true regression function $f(\cdot)$ is bounded by $\frac{A}{2}$ in absolute value for some positive constant A, and the estimators $\hat{f}_j$, $j \in \Gamma$, are clipped accordingly.

Condition 2: The conditional variance $E\left(\varepsilon^2|\mathbf{x}\right)$ is uniformly upper bounded by some positive constant B^2 with probability 1.

Condition 3: There exist a constant $t_0 > 0$ and a monotone function $0 < H(t) < \infty$ on $[-t_0, t_0]$ such that, for $-t_0 \le t \le t_0$, $E\left(\exp(t|\varepsilon|)|\mathbf{x}\right) \le H(t)$ with probability 1.

Condition 4: Zero is a median of the distribution of the error ε conditional on $\mathbf{x}$.

For simplicity, we take $\hat{d}_{j,i} = 1$. Let j_* be the minimizer of the risk $E|Y - \hat{f}_j|$ in Γ.

Theorem 1. *Assume Conditions* 1−3 *hold.*

1. *When the tuning parameter* η *is chosen small enough,*

$$E|Y-\tilde{f}(\mathbf{x})| \le (A+B)\,P(j_* \notin \Gamma_s) + E|Y-\hat{f}_{j_*}(\mathbf{x})| + CE\left(\sqrt{\frac{\log(K_s)}{n}}\right),$$

   where C *is a positive constant that depends on* $t_0, A,$ *and* B*.*

2. *If, in addition, Condition* 4 *holds,*

$$E|Y-\tilde{f}(\mathbf{x})| \le AP(j_* \notin \Gamma_s) + E|Y-\hat{f}_{j_*}(\mathbf{x})| + CE\left(\sqrt{\frac{\log(K_s)}{n}}\right).$$

Remarks.

1. The same risk bound holds (due to convexity) when data are randomly split multiple times and the resulting estimates $\tilde{f}(\mathbf{x})$ are averaged.
2. The tuning parameter η is chosen to be of order $\sqrt{\log K_s/n}$.
3. In the initial theories on combining regression estimators, quadratic loss is used (see, Yang (2001) and Catoni (2004)). The tools there do not work for absolute error. The idea used here is to add a small multiple of the quadratic loss to the absolute loss so that the total loss has a quadratic behavior and is still close to the absolute loss.
4. When no screening is done, the risk bounds becomes $E|Y-\tilde{f}(\mathbf{x})| \le \inf_j(E|Y-\hat{f}_j(\mathbf{x})|) + C\sqrt{\log(K)/n}$. The theorem shows that the combined estimator behaves like the best $\hat{f}_j$ adaptively, up to an additive penalty term of order $\sqrt{\log(K)/n}$ that cannot be generally improved.
5. For screening of model selection methods, besides approaches such as Fan and Lv (2008), cross validation can also be used, e.g., to keep the top m methods for a pre-determined integer m. With a proper data splitting ratio, as shown in Yang (2007), under some conditions, and for any choice of $m \ge 1$, $P(j_* \notin \Gamma_s) \to 0$. If one method is taken as a reference and the methods that perform much worse are removed, the exclusion probability of j_* can be exponentially small.
6. The improvement of the risk bound under Condition 4 can be important when $E\left(\varepsilon^2|\mathbf{x}\right)$ is large.
7. The risk bounds in Theorem 1 do not require consistency or other optimality properties of the candidate model selection methods. However, those properties are certainly not irrelevant. For instance, in the now popular setting that the true model has dimension $m^* = m_n^*$ increasing in n, if a mode selection method among a bounded number of candidates is consistent for the DGP

then, under some favorable conditions, its pointwise squared L_2 risk can be exactly at the optimal rate $m_n^*\sigma^2/n$. If the screening method is successful so that $P(j_* \notin \Gamma_s)$ is smaller in order than $\sqrt{m_n^*\sigma^2/n}$, then Theorem 1 implies that $E|Y - \tilde{f}(\mathbf{x})| = E(|\varepsilon|) + O\left(\sqrt{m_n^*\sigma^2/n}\right)$. Since for symmetric random noise ε, $E(|\varepsilon|)$ is the smallest prediction error under absolute error loss, such a risk bound means that the combined prediction is of order $\sqrt{m_n^*\sigma^2/n}$ from the ideal.

5. Concluding Remarks

Exciting new model selection methods have been derived from various perspectives. One may naturally consider a number of such methods so that there is a better chance that the best one works well for the data at hand.

For the goal of regression function estimation or prediction, the LASSO, SCAD, and a-LASSO behave the best in different scenarios in terms of the model sparsity and model noise level. For tuning parameter selection, the use of BIC is not always better than fivefold CV and, in fact, it is sometimes much worse. In applications, when the sample size is not large relative to the number of predictors, it is difficult to determine which selection method should be used and how to choose the tuning parameter.

We propose robust adaptive regression by mixing, l_1-ARM, to aggregate the predictive strengths of the different selection methods so as to perform as if one knew which selection method is best for each scenario in advance.

Both the theory and the numerical work support that, for estimating the regression function or prediction, the l_1-ARM performs as well as the best candidate method in the individual scenarios. When various scenarios are considered, the l_1-ARM shows its predictive advantage over the individual model selection methods. A contribution of the l_1-ARM is that it is more robust than the ARM when the underlying model tends to generate outliers. Furthermore, when there are no outliers, it does not lose much efficiency compared to the ARM.

Finally, it must be pointed out that the l_1-ARM loses model interpretability, and it does not perform variable selection as the important model selection methods do.

6. Appendix: Proof of Theorem 1

Let $L(u) = |u|$, and $h(u) = \exp(-\eta L(u))$. Let $n_1 = n_2 = n/2$. Define $Q^{n_2} = \sum_{j\in\Gamma_s}(1/K_s)\Pi_{i=n_1+1}^{n} h(Y_i - \hat{f}_j(\mathbf{x}_i))$, where $\hat{f}_j(\mathbf{x}_i)$ is the prediction from the j^{th} method at time i. Then

$$-\log(Q^{n_2}) \le \log(K_s) + \eta \sum_{i=n_1+1}^{n} L(Y_i - \hat{f}_j(\mathbf{x}_i)).$$

On the other hand,

$$\begin{aligned}Q^{n_2} = &\sum_{j\in\Gamma_s}\frac{1}{K_s}h(Y_{n_1+1}-\hat{f}_j(\mathbf{x}_{n_1+1}))\\ &\times\frac{\sum_{j\in\Gamma_s}h(Y_{n_1+1}-\hat{f}_j(\mathbf{x}_{n_1+1}))h(Y_{n_1+2}-\hat{f}_j(\mathbf{x}_{n_1+2}))}{\sum_{j\in\Gamma_s}h(Y_{n_1+1}-\hat{f}_j(\mathbf{x}_{n_1+1}))}\\ &\times\cdots\times\frac{\sum_{j\in\Gamma_s}\Pi_{i=n_1+1}^{n}h(Y_i-\hat{f}_j(\mathbf{x}_i))}{\sum_{j\in\Gamma_s}\Pi_{i=n_1+1}^{n-1}h(Y_i-\hat{f}_j(\mathbf{x}_i))}.\end{aligned}$$

That is, $Q^{n_2}=\Pi_{i=n_1+1}^{n}\sum_{j\in\Gamma_s}W_{j,i}h(Y_i-\hat{f}_j(\mathbf{x}_i))$. Accordingly, $-\log(Q^{n_2})$ $=-\sum_{i=n_1+1}^{n}\log\left(\sum_{j\in\Gamma_s}W_{j,i}h(Y_i-\hat{f}_j(\mathbf{x}_i))\right)=-\sum_{i=n_1+1}^{n}\log(E^Jh(Y_i-\hat{f}_J(\mathbf{x}_i)))$, where E^J denotes the expectation with respect to J under the distribution $P(J=j)=W_{j,i}$, given each i for $j\in\Gamma_s$. Define $V=L(Y_i-\hat{f}_J(\mathbf{x}_i))-E^JL(Y_i-\hat{f}_J(\mathbf{x}_i))$, and $\bar{f}_i(\mathbf{x})=\sum_{j\in\Gamma_s}W_{j,i}\hat{f}_j(\mathbf{x})$. Then

$$\begin{aligned}E^JV^2&=E^J(L(Y_i-\hat{f}_J(\mathbf{x}_i))-E^JL(Y_i-\hat{f}_J(\mathbf{x}_i)))^2\\&\le E^J(L(Y_i-\hat{f}_J(\mathbf{x}_i))-L(Y_i-E^J\hat{f}_J(\mathbf{x}_i)))^2\\&\le E^J(\hat{f}_J(\mathbf{x}_i)-E^J\hat{f}_J(\mathbf{x}_i))^2\\&=E^J(\hat{f}_J(\mathbf{x}_i)-\bar{f}_i(\mathbf{x}_i))^2.\end{aligned}$$

By Lemma 3.6.1 of Catoni (2004, p.85), we get

$$\log(E^Jh(Y_i-\hat{f}_J(\mathbf{x}_i)))\le-\eta E^JL(Y_i-\hat{f}_J(\mathbf{x}_i))+I,$$

where $I\le(\eta^2/2)\exp(\eta(|L(Y_i-\hat{f}_J(\mathbf{x}_i))|+\sup_{j\ge1}|L(Y_i-\hat{f}_J(\mathbf{x}_i))|))E^JV^2$. Observe that

$$\begin{aligned}I\le&\frac{\eta^2}{2}\exp(2\eta\sup_{j\ge1}|L(Y_i-\hat{f}_J(\mathbf{x}_i))|)E^JV^2\\\le&\frac{\eta^2}{2}\exp(2\eta\sup_{j\ge1}|Y_i-\hat{f}_J(\mathbf{x}_i)|)E^JV^2\\\le&\frac{\eta^2}{2}\exp(2\eta(|Y_i-f(\mathbf{x}_i)|+\sup_{j\ge1}|f(\mathbf{x}_i)-\hat{f}_J(\mathbf{x}_i)|))E^JV^2\\\le&\frac{\eta^2}{2}e^{2\eta A}\exp(2\eta|\varepsilon_i|)E^JV^2\\\le&\frac{\eta^2}{2}e^{2\eta A}\exp(2\eta|\varepsilon_i|)E^J(\hat{f}_J(\mathbf{x}_i)-\bar{f}_i(\mathbf{x}_i))^2,\end{aligned}$$

where the second to last inequality holds because of Condition 1. Under Condition 3, taking expectation with respect to the randomness of the errors ε_i and $\mathbf{x}_i$

for $n_1 + 1 \le i \le n$ conditional on the first n_1 observations, we have that when $2\eta \le t_0$,

$$E_{n_1}(I) \le \frac{\eta^2}{2} e^{2\eta A} H(2\eta) E_{n_1}(E^J(\hat{f}_J(\mathbf{x}_i) - \bar{f}_i(\mathbf{x}_i))^2).$$

For absolute error loss in l_1-ARM, our approach is to add a small multiple of the quadratic loss so that the total loss is still close to the absolute loss and the tools of Catoni (2004) can be modified to work for our problem. Take $L_s(u) = L(u) + au^2$, $a > 0$. Let $b_0 = Y_i - \bar{f}_i(\mathbf{x}_i)$ and $b = Y_i - \hat{f}_J(\mathbf{x}_i)$. Then, as in Shan and Yang (2009),

$$\begin{aligned} &L_s(b) - (2ab_0 + 1_{b_0 \ge 0} - 1_{b_0 < 0})(b - b_0) - L_s(b_0) \\ &\quad = a(b - b_0)^2 + b(1_{b \ge 0} - 1_{b < 0} + 1_{b_0 < 0} - 1_{b_0 \ge 0}) \\ &\quad = a(b - b_0)^2 + \begin{cases} 0 & \text{if } b, b_0 \ge 0 \\ 0 & \text{if } b, b_0 < 0 \\ 2b & \text{if } b \ge 0 \text{ and } b_0 < 0 \\ -2b & \text{if } b < 0 \text{ and } b_0 \ge 0 \end{cases} \\ &\quad \ge a(b - b_0)^2. \end{aligned}$$

Since $E^J(b - b_0) = 0$, we have $E^J L_s(b) - L_s(b_0) \ge aE^J(b - b_0)^2$, namely,

$$E^J L_s(Y_i - \hat{f}_J(\mathbf{x}_i)) - L_s(Y_i - \bar{f}_i(\mathbf{x}_i)) \ge aE^J(\hat{f}_J(\mathbf{x}_i) - \bar{f}_i(\mathbf{x}_i))^2.$$

Thus,

$$E_{n_1}(E^J L_s(Y_i - \hat{f}_J(\mathbf{x}_i)) - L_s(Y_i - \bar{f}_i(\mathbf{x}_i))) \ge aE_{n_1}(E^J(\hat{f}_J(\mathbf{x}_i) - \bar{f}_i(\mathbf{x}_i))^2).$$

Then we have

$$E_{n_1}(I) \le \frac{a^{-1}\eta^2}{2} e^{2\eta A} H(2\eta) E_{n_1}(E^J L_s(Y_i - \hat{f}_J(\mathbf{x}_i)) - L_s(Y_i - \bar{f}_i(\mathbf{x}_i))).$$

Let $\eta/2 \ge (a^{-1}\eta^2/2)e^{2\eta A} H(2\eta)$. Then we have

$$E_{n_1}(I) \le \frac{\eta}{2} E_{n_1}(E^J L_s(Y_i - \hat{f}_J(\mathbf{x}_i)) - L_s(Y_i - \bar{f}_i(\mathbf{x}_i))).$$

Recall there are two constraints on η: $2\eta \le t_0$ and $\eta/2 \ge (a^{-1}\eta^2/2)e^{2\eta A} H(2\eta)$. When we choose a and η so that $\eta \le t_0/2$ and $a \ge \eta e^{2\eta A} H(2\eta)$, the constraints are met. Let $a_\eta = \eta e^{t_0 A} H(t_0)$. With such a choice of η and a_η,

$$\begin{aligned} &E_{n_1}(\log E^J \exp(-\eta L(Y_i - \hat{f}_J(\mathbf{x}_i)))) \\ &\quad \le -\eta E_{n_1}(L(Y_i - \bar{f}_i(\mathbf{x}_i))) + \eta E_{n_1}(L(Y_i - \bar{f}_i(\mathbf{x}_i)) - E^J L(Y_i - \hat{f}_J(\mathbf{x}_i))) \\ &\qquad + \frac{\eta}{2} E_{n_1}(E^J L_s(Y_i - \hat{f}_J(\mathbf{x}_i)) - L_s(Y_i - \bar{f}_i(\mathbf{x}_i))) \end{aligned}$$

$$\begin{aligned}
&\le -\eta E_{n_1}(L(Y_i-\bar{f}_i(\mathbf{x}_i))) - \frac{\eta}{2}E_{n_1}(E^J L(Y_i-\hat{f}_J(\mathbf{x}_i)) - L(Y_i-\bar{f}_i(\mathbf{x}_i))) \\
&\quad + \frac{a_\eta \eta}{2}E_{n_1}(E^J(Y_i-\hat{f}_J(\mathbf{x}_i))^2) - \frac{a_\eta\eta}{2}E_{n_1}((Y_i-\hat{f}_J(\mathbf{x}_i))^2) \\
&\le -\eta E_{n_1}(L(Y_i-\bar{f}_i(\mathbf{x}_i))) + \frac{a_\eta\eta(A^2+B^2)}{2}.
\end{aligned}$$

The last inequality holds because of the convexity of L and Conditions 1 and 2. Assume $j \in \Gamma_s$ for the moment. Then we have

$$\sum_{i=n_1+1}^{n} E_{n_1}L(Y_i-\bar{f}_i(\mathbf{x}_i)) \le \frac{\log(K_s)}{\eta} + \frac{a_\eta(A^2+B^2)n_2}{2} + \sum_{i=n_1+1}^{n} E_{n_1}L(Y_i-\hat{f}_j(\mathbf{x}_i)).$$

Under the iid assumption on the data and $n_2 = n/2$, we have

$$\frac{1}{n_2}\sum_{i=n_1+1}^{n} E_{n_1}L(Y-\bar{f}_i(\mathbf{x})) \le E_{n_1}L(Y-\hat{f}_j(\mathbf{x})) + \frac{2\log(K_s)}{\eta n} + \frac{a_\eta(A^2+B^2)}{2}.$$

With an optimal choice of η, $\eta' = \sqrt{(4\log(K_s))/(ne^{t_0 A}H(t_0)(A^2+B^2))}$,

$$\frac{1}{n_2}\sum_{i=n_1+1}^{n} E_{n_1}L(Y-\bar{f}_i(\mathbf{x})) \le E_{n_1}L(Y-\hat{f}_j(\mathbf{x})) + C\sqrt{\frac{\log(K_s)}{n}},$$

where C is a positive constant depending on t_0, A, and B. By convexity of L, together with that $\tilde{f} = (1/n_2)\sum_{i=n_1+1}^{n}\bar{f}_i$, we have for each $j \in \Gamma_s$,

$$E_{n_1}|Y-\tilde{f}(\mathbf{x})| \le E_{n_1}|Y-\hat{f}_j(\mathbf{x})| + C\sqrt{\frac{\log(K_s)}{n}}.$$

Therefore, if $j^* \in \Gamma_s$,

$$E_{n_1}|Y-\tilde{f}(\mathbf{x})| \le E_{n_1}|Y-\hat{f}_{j^*}(\mathbf{x})| + C\sqrt{\frac{\log(K_s)}{n}}.$$

When $j^* \notin \Gamma_s$, $E_{n_1}|Y-\tilde{f}(\mathbf{x})| \le E_{n_1}|\varepsilon| + E_{n_1}|f(\mathbf{x})-\tilde{f}(\mathbf{x})| \le B + A$. Thus

$$E|Y-\tilde{f}(\mathbf{x})| \le (B+A)\,P\,(j^* \in \Gamma_s^c) + E|Y-\hat{f}_{j^*}(\mathbf{x})| + CE\left(\sqrt{\frac{\log(K_s)}{n}}\right).$$

This risk bound can be improved if the error ε has a distribution with median 0 given $\mathbf{x}$, as shown below. From before, if $j^* \in \Gamma_s$,

$$E_{n_1}|Y-\tilde{f}(\mathbf{x})| - E_{n_1}|\varepsilon| \le E_{n_1}|Y-\hat{f}_{j^*}(\mathbf{x})| - E_{n_1}|\varepsilon| + C\sqrt{\frac{\log(K_s)}{n}},$$

and if $j^* \notin \Gamma_s$, $E_{n_1}|Y - \tilde{f}(\mathbf{x})| - E_{n_1}|\varepsilon| \le E_{n_1}|f(\mathbf{x}) - \tilde{f}(\mathbf{x})| \le A$. Then,

$$\begin{aligned} &E_{n_1}|Y - \tilde{f}(\mathbf{x})| - E_{n_1}|\varepsilon| \\ &\quad \le AI_{j^* \in \Gamma_s^c} + \left(E_{n_1}|Y - \hat{f}_j(\mathbf{x})| - E_{n_1}|\varepsilon| + C\sqrt{\frac{\log(K_s)}{n}} \right) I_{j^* \in \Gamma_s}, \end{aligned}$$

where I denotes the indictor function. If the error distribution (given $\mathbf{x}$) has median zero, we must have $E_{n_1}|Y - \tilde{f}(\mathbf{x})| - E_{n_1}|\varepsilon| \ge 0$ with probability 1 and thus

$$E_{n_1}|Y - \tilde{f}(\mathbf{x})| - E_{n_1}|\varepsilon| \le AI_{j^* \in \Gamma_s^c} + E_{n_1}|Y - \hat{f}_j(\mathbf{x})| - E_{n_1}|\varepsilon| + C\sqrt{\frac{\log(K_s)}{n}}.$$

That is,

$$E_{n_1}|Y - \tilde{f}(\mathbf{x})| \le AI_{j^* \in \Gamma_s^c} + E_{n_1}|Y - \hat{f}_{j^*}(\mathbf{x})| + C\sqrt{\frac{\log(K_s)}{n}}.$$

Thus,

$$E|Y - \tilde{f}(\mathbf{x})| \le AP(j^* \notin \Gamma_s) + E|Y - \hat{f}_{j^*}(\mathbf{x})| + CE\left(\sqrt{\frac{\log(K_s)}{n}} \right).$$

When there is no screening step, $\Gamma_s = \Gamma$ and $K_s = K$. Then we have

$$E|Y - \tilde{f}(\mathbf{x})| \le E|Y - \hat{f}_{j_*}(\mathbf{x})| + C\sqrt{\frac{\log(K)}{n}}.$$

This completes the proof.

Acknowledgement

This work is partially supported by NSF grant DMS-0706850. We greatly appreciate receiving computer packages of Jinchi Lv and of Yi Yang and Hui Zou. We thank an associate editor and three referees for their very detailed and helpful comments on improving our work.

References

Catoni, O. (2004). *Statistical Learning Theory and Stochastic Optimization.* Springer, New York.

Chen, L. and Yang, Y. (2010). *Combining Statistical Procedures.* World Scientific, Singapore.

Fan, J. and Li, R. (2001). Variable selection via nonconave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.* **96**, 1348-1360.

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space (with discussion). *J. Roy. Statist. Soc. Ser. B* **70**, 849-911.

Johnson, R. W. (1996). Fitting percentage of body fat to simple body measurements. *J. Statist. Education* **4**.

Juditsky, A. and Nemirovski, A. (2000). Functional aggregation for nonparametric estimation. *Ann. Statist.* **28**, 681-712.

Lv, J. and Fan, Y. (2009). A unified approach to model selection and sparse recovery using regularized least squares. *Ann. Statist.* **37**, 3498-3528.

Meinshausen, N. and Bühlmann, P. (2006). High dimensional graphs and variable selection with the Lasso. *Ann. Statist.* **34**, 1436-1462.

Shan, K. and Yang, Y. (2009). Combining regression quantile estimators. *Statist. Sinica* **19**, 1171-1191.

Tibshirani, R. J. (1996). Regression shrinkage and selection via the Lasso. *J. Roy. Statist. Soc. Ser. B* **58**, 267-288.

Tsybakov, A. B. (2003). Optimal rates of aggregation. *Learning Theory and Kernel Machines, Lecture Notes in Artificial Intelligence* 2777, 303–313. Springer, Heidelberg.

Wang, H., Li, R. and Tsai, C. L. (2007). Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika* **94**, 553-568.

Weisberg, S. (1985). *Applied Linear Regression*. Wiley, New York.

Yang, Y. (2001). Adaptive regression by mixing. *J. Amer. Statist. Assoc.* **96**, 574-588.

Yang, Y. (2004). Aggregating regression procedures to improve performance. *Bernoulli* **10**, 25-47.

Yang, Y. (2007). Consistency of cross validation for comparing regression procedures. *Ann. Statist.* **35**, 2450-2473.

Yuan, Z. and Yang, Y. (2005). Combining linear regression models: When and how? *J. Amer. Statist. Assoc.* **100**, 1202-1214.

Zhang, C. (2007). Penalized linear unbiased selection. Technical Report, Dept. Statistics, Rutgers Univ.

Zhang, C. and Huang, J. (2008). The sparsity and bias of the lasso selection in high-Dimensional linear regression. *Ann. Statist.* **36**, 1567-1594.

Zhao, P. and Yu, B. (2006). On model selection consistency of Lasso. *J. Machine Learning Research* **7**, 2541-2563.

Zou, H. (2006). The adaptive lasso and its oracle properties. *J. Amer. Statist. Assoc.* **101**, 1418-1429.

Zou, H. (2008). On the "degrees of freedom" of the LASSO. *Ann. Statist.* **35**, 2173-2192.

Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *Ann. Statist.* **36**, 1509-1533.

School of Statistics, University of Minnesota, 313 Ford Hall, 224 Church Street S.E., Minneapolis, MN 55455, USA.

E-mail: xiaoqiao@stat.umn.edu

School of Statistics, University of Minnesota, 313 Ford Hall, 224 Church Street S.E., Minneapolis, MN 55455, USA.

E-mail: yyang@stat.umn.edu

(Received February 2010; accepted July 2011)