Submitted to Statistical Science

Sparse regression: Scalable algorithms and empirical performance Appendices

Dimitris Bertsimas Jean Pauphilet and Bart Van Parys

Operations Research Center, MIT

APPENDIX A: EXTENSION OF THE SUB-GRADIENT ALGORITHM AND IMPLEMENTATION

From a theoretical point of view, the boolean relaxation of the sparse learning problem is often tight, especially in the presence of randomness or noise in the data, so that feature selection can be expressed as a saddle point problem with continuous variables only. This min-max formulation is easier to solve numerically than the original mixed-integer optimization problem (3) because the original combinatorial structure vanishes. Our proposed algorithm benefits from both tightness of the Boolean relaxation and integrality of optimal solutions. In this section, we present the `Julia` package `SubsetSelection` which competes with the `glmnet` implementation of Lasso in terms of computational time, while returning statistically more relevant features, in terms of accuracy but more significantly in terms of false discovery rate. With `SubsetSelection`, we hope to bring to the community an easy-to-use and generic feature selection tool, which addresses deficiencies of ℓ_1-penalization but scales just as well to high-dimensional data sets.

A.1 Cardinality-constrained formulation

In this paper, we have proposed a sub-gradient algorithm for solving the following boolean relaxation

$$\min_{s\in[0,1]^p\,:\,\mathbf{e}^\top s\leqslant k}\ \max_{\alpha\in\mathbb{R}^n}\ f(\alpha,s),$$

where

$$f(\alpha,s) := -\sum_{i=1}^{n}\hat{\ell}(y_i,\alpha_i) - \frac{\gamma}{2}\sum_{j=1}^{p} s_j\alpha^\top X_j X_j^\top\alpha$$

is a linear function in s and concave function in α. The function f depends on the loss function ℓ through its Fenchel conjugate $\hat{\ell}$. In this paper, we mainly focused on OLS and logistic loss but the same methodology could be applied to any convex loss function. Indeed, the package `SubsetSelection` supports all loss functions presented in Table A.1.

At each iteration, the algorithm updates the variable α by performing one step of projected gradient ascent with step size δ, and updates the support s by minimizing $f(\alpha,s)$ with respect to s, α being fixed. Since s satisfies $s\in[0,1]^p,\ s^\top\mathbf{e}\leqslant k$, and

TABLE A.1
Supported loss functions ℓ and their corresponding Fenchel conjugates $\hat{\ell}$ as defined in Theorem 1. The observed data $y \in \mathbb{R}$ for regression and $y \in \{-1, 1\}$ for classification. By convention, $\hat{\ell}$ equals $+\infty$ outside of its domain. The binary entropy function is denoted as $H(x) := -x \log x - (1-x)\log(1-x)$.

Method	Loss $\ell(y, u)$	Fenchel conjugate $\hat{\ell}(y, \alpha)$
Logistic loss	$\log\left(1 + e^{-yu}\right)$	$-H(-y\alpha)$ for $y\alpha \in [-1, 0]$
1-norm SVM - Hinge loss	$\max(0, 1-yu)$	$y\alpha$ for $y\alpha \in [-1, 0]$
2-norm SVM	$\frac{1}{2}\max(0, 1-yu)^2$	$\frac{1}{2}\alpha^2 + y\alpha$ for $y\alpha \leqslant 0$
Least Square Regression	$\frac{1}{2}(y-u)^2$	$\frac{1}{2}\alpha^2 + y\alpha$
1-norm SVR	$(\lvert y-u\rvert - \varepsilon)_+$	$y\alpha + \varepsilon\lvert\alpha\rvert$ for $\lvert\alpha\rvert \leqslant 1$
2-norm SVR	$\frac{1}{2}(\lvert y-u\rvert - \varepsilon)_+^2$	$\frac{1}{2}\alpha^2 + y\alpha + \varepsilon\lvert\alpha\rvert$

f is linear in s, this partial minimization boils down to sorting the components of $(-\alpha^\top X_j X_j^\top \alpha)_{j=1,\dots,p}$ and selecting the k smallest. Pseudo-code is given in Algorithm 2.2.

A.2 Scalability

As experiments in Sections 3 and 4 demonstrated, our proposed algorithm and implementation provides an excellent approximation for the solution of the discrete optimization problem (3), while terminating in times comparable with coordinate descent for Lasso estimators for low values of γ. Table A.2 reports some computational time of `SubsetSelection` for data sets with various values of n, p and k. The algorithm scales to data sets with $(n, p) = (10^5, 10^5)$s or $(10^4, 10^6)$s within a few minutes. More comparison on computational time are given in Appendix B.1.

TABLE A.2
Computational time of SS with $T_{max} = 200$ for data sets with large values of n and p, $\gamma = 2\,p/k/\max_i \|x_i\|^2/n$. Due to the dimensionality of the data, computations where performed on 1 CPU with 250GB of memory. We provide the average computational time (and the standard deviation) over 10 experiments.

Loss function ℓ	n	p	k	time (in s)
Least Squares	10,000	100,000	100	12.90 (0.45)
Least Squares	50,000	100,000	100	28.45 (1.83)
Least Squares	10,000	500,000	100	33.00 (1.86)
Least Squares	10,000	500,000	500	43.00 (0.54)
Hinge Loss	10,000	100,000	100	37.26 (0.14)
Hinge Loss	50,000	100,000	100	160.73 (0.28)
Hinge Loss	10,000	500,000	100	157.09 (1.18)
Hinge Loss	10,000	500,000	500	59.74 (0.08)

A.3 Extension to cardinality-penalized formulation

Our proposed approach naturally extends to cardinality-penalized estimators as well, where the 0-pseudonorm is added as a penalization term instead of an explicit constraint. Let us consider the ℓ_2-regularized optimization problem

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^n \ell(y_i, w^\top x_i) + \frac{1}{2\gamma}\|w\|_2^2 + \lambda\|w\|_0, \tag{A.1}$$

which corresponds to the estimator (1) in the unregularized limit $\gamma \to +\infty$. Similarly introducing a binary variable s encoding the support of w, we get that the previous problem (A.1) is equivalent to

$$\min_{s\in\{0,1\}^p} \max_{\alpha\in\mathbb{R}^n} f(\alpha, s) + \lambda \mathbf{e}^\top s. \tag{A.2}$$

The new saddle point function $s \mapsto f(\alpha, s) + \lambda \mathbf{e}^\top s$ is still linear in s and concave in α. As before, its boolean relaxation

$$\min_{s\in[0,1]^p} \max_{\alpha\in\mathbb{R}^n} f(\alpha, s) + \lambda \mathbf{e}^\top s \tag{A.3}$$

is tight if the minimizer of $f(\alpha, s) + \lambda \mathbf{e}^\top s$ with respect to s is unique. More precisely, we prove an almost verbatim analogue of Theorem 2.

THEOREM A.1. *The boolean relaxation (A.3) is tight if there exists a saddle point $(\bar{\alpha}, \bar{s})$ such that the vector $(\lambda - \frac{\gamma}{2}\bar{\alpha}^\top X_j X_j^\top \bar{\alpha})_{j=1,\dots,p}$ has non-zero entries.*

PROOF. The saddle-point problem (A.3) is a continuous convex/concave minimax problem and Slater's condition is satisfied so strong duality holds. Therefore, any saddle-point $(\bar{\alpha}, \bar{s})$ must satisfy

$$\bar{\alpha} \in \arg\max_{\alpha\in\mathbb{R}^n} f(\alpha, \bar{s}) + \lambda \mathbf{e}^\top s, \quad \bar{s} \in \arg\min_{s\in\in[0,1]^p} f(\bar{\alpha}, s) + \lambda \mathbf{e}^\top s.$$

If there exists $\bar{\alpha}$ such that $(\lambda - \frac{\gamma}{2}\bar{\alpha}^\top X_j X_j^\top \bar{\alpha})_{j=1,\dots,p}$ has non-zero entries, then there is a unique $\bar{s} \in \arg\min_{s\in\in[0,1]^p} f(\bar{\alpha}, s) + \lambda \mathbf{e}^\top s$. In particular, this minimizer is binary and the relaxation is tight. □

This theoretical result suggests that the Lagrangian relaxation (A.3) can provide a good approximation for the combinatorial problem (A.1) in many cases. The same sub-gradient strategy as the one described in Algorithm 2.2 can be used to solve (A.3), with a slightly different partial minimization step: Now, minimizing $f(\alpha, s) + \lambda s^\top \mathbf{e}$ with respect $s \in [0,1]^p$ for a fixed α boils down to computing the components of $(\lambda - \gamma/2\alpha^\top X_j X_j^\top \alpha)_{j=1,\dots,p}$ and selecting the negative ones, which requires $O(np)$ operations. This strategy is also implemented in the package `SubsetSelection`.

APPENDIX B: NUMERICAL EXPERIMENTS FOR REGRESSION - SUPPLEMENTARY MATERIAL

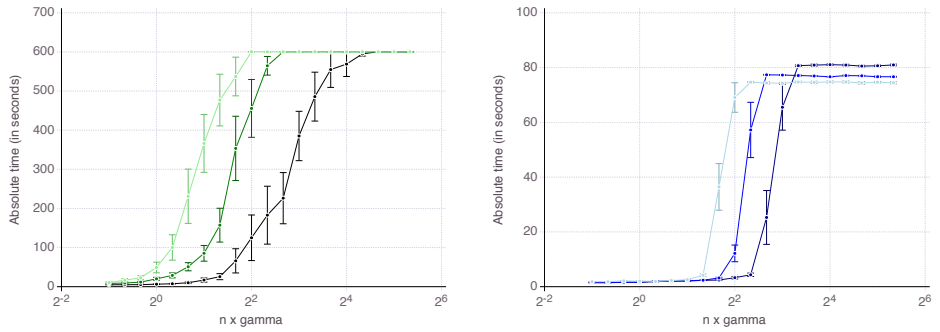
In this section, we provide additional material for the simulations conducted in Section 3 on regression examples.

B.1 Synthetic data satisfying mutual incoherence condition

We first consider the case where the design matrix $X \sim \mathcal{N}(0, \Sigma)$, with Σ a Toeplitz matrix. In particular, Σ satisfies the so-called mutual incoherence condition required by Lasso estimators to be statistically consistent. In this setting, we provide more details about the computational time comparison between algorithms for sparse regression presented in Sections 3.

B.1.1 Impact of the hyper-parameters k and γ The discrete convex optimization formulation (3) and its Boolean relaxation (5) involve two hyper-parameters: the ridge penalty γ and the sparsity level k.

Intuition suggests that computational time would increase with γ. Indeed, when $\gamma \to 0$, $w^\star = 0$ is an obvious optimal solution, while for $\gamma \to +\infty$ the problem can become ill-conditioned. We generate 10 problems with $p = 5,000$, $k_{true} = 50$, $SNR = 1$ and $\rho = .5$ and various sample sizes n, fix $k = k_{true}$ and report absolute computational time as $n \times \gamma$ increases in Figure B.1 (p. 4). For small values of γ, both methods terminate extremely fast - within 10 seconds for CIO and in less than 2 seconds for SS. As γ increases, computational time sharply increases. For CIO, we capped computational time to 600 seconds. For SS, we limited the number of iterations to $T_{max} = 200$. Regarding the sparsity k, the size of the feasible space



(a) CIO (b) SS

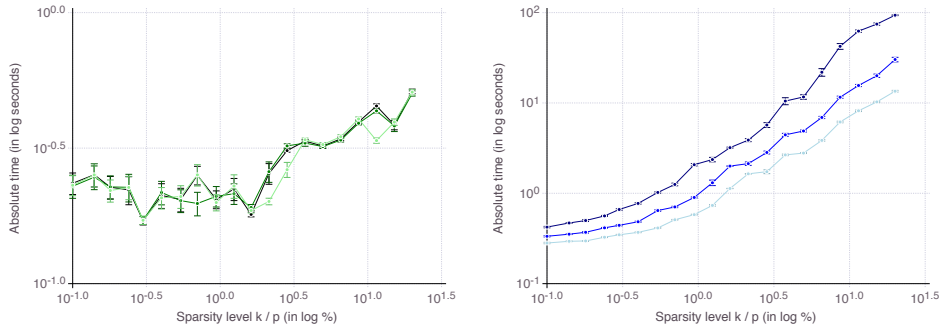
Fig B.1: Absolute computational time as $n \times \gamma$ increases, for CIO (left panel), SS (right panel), with OLS loss. We fixed $p = 5,000$, $k_{true} = 50$, $SNR = 1$, $\rho = .5$ and $k = k_{true}$, and averaged results over 10 data sets. We report results for $n = 500$, $n = 1,000$ and $n = 2,000$ (from light to dark)

$\{s \in \{0,1\}^p \ : \ s^\top \mathbf{e} \leqslant k\}$ grows as p^k. Empirically, we observe (Figure B.2 p. 5) that computational time increases at most polynomially with k.

B.1.2 Impact of the signal-to-noise ratio, sample size n and problem size p As more signal becomes available, the feature selection problem should become easier and computational time should decrease. Indeed, in Figure B.3 (p. 6), we observe that low SNR generally increases computational time for all methods (left panel). The correlation parameter ρ (right panel), however, does not seem to have a strong impact on computational time. In our opinion, with $SNR = 1$, the effect of correlation on computational time is second order compared to the impact of noise.

Figure B.4 (p. 7) represents computational for increasing p, n/p being fixed and for increasing n/p, p being fixed. As shown, all methods scale similarly with p (almost linearly), while CIO and SS are less sensitive to n/p than their competitors.

B.1.3 High dimensional and high noise regime In the high-noise regime, we presented experiments with $p = 2,000$ for n ranging from 500 to 10,000, illustrating both $p > n$ and $n < p$ regimes. In such regimes, we can already see on Figure 1 (p. 15) how poorly accurate and comparable all methods can be when $p > n$,



(a) CIO (b) SS

Fig B.2: Absolute computational time as k increases from 5 to 1,000, for CIO (left panel), SS (right panel), with OLS loss. We fixed $p = 5,000$, $k_{true} = 50$, $SNR = 1$, $\rho = .5$ and $n = 1,000$, and averaged results over 10 data sets. We report results for $\gamma = 2^i \gamma_0$ with $i = 0, 2, 4$ (from light to dark) and $\gamma_0 = \dfrac{p}{n\, k_{true}\, \max_i \|x_i\|^2}$.

which gives us little hope for the high-dimensional regime $p \gg n$. Indeed, Figure B.5 (p.8) displays the accuracy of all methods for n ranging from 50 to 2000 and confirms our results.

B.2 Synthetic data *not* satisfying mutual incoherence condition

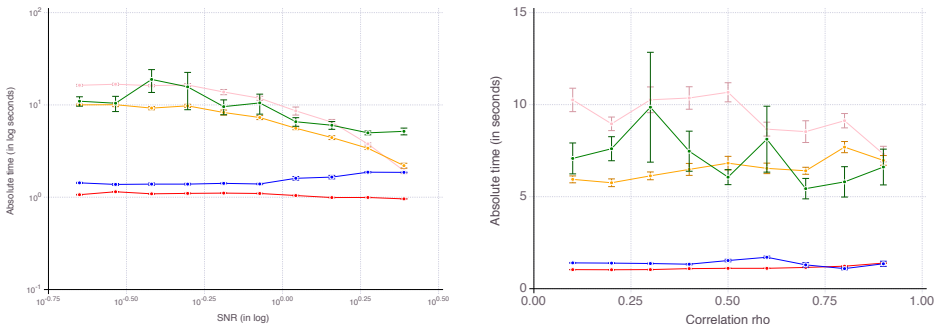
We now consider a covariance matrix Σ, which does not satisfy mutual incoherence, as proved in [1]. We consider three regimes of noise summarized in Table B.1 (p. 5).

TABLE B.1
Regimes of noise (SNR) considered in our experiments on regression when MIC does not hold

Low noise	Medium noise	High noise
$SNR = 6$	$SNR = 1$	$SNR = 0.05$
$p = 20,000$, $k = 100$	$p = 10,000$, $k = 50$	$p = 2,000$, $k = 10$

B.2.1 Feature selection with a given support size Figure B.6 on page 9 reports relative compared to `glmnet` (left panel) and absolute (right panel) computational time in log scale. As for the case where mutual incoherence is satisfied, all methods terminates within a 10-100 factor with respect to `glmnet`.

B.2.2 Feature selection with cross-validated support size We compare all methods when k_{true} is no longer given and needs to be cross-validated from the data itself. Figure B.7 on page 10 reports the results of the cross-validation procedure for increasing n. In terms of accuracy (left panel), all four methods are relatively equivalent and demonstrate a clear convergence: $A \to 1$ as $n \to \infty$. On false detection rate however (right panel), behaviors vary among methods. Cardinality-constrained estimators achieve the lowest false detection rate ($0 - 30\%$), followed by MCP ($10 - 60\%$), SCAD ($20 - 70\%$) and then ENet (c.80%). In case of ENet, this behavior was expected, for ℓ_1-estimators are provably inconsistent, so that FDR must be positive when $A = 1$.



(a) SNR, for $\rho = .5$ (b) Correlation ρ, for $SNR = 1$

Fig B.3: Absolute computational time as signal-to-noise or correlation increases, for CIO (in green), SS (in blue with $T_{max} = 200$), ENet (in red), MCP (in orange), SCAD (in pink) with OLS loss. We fixed $p = 5,000$, $k_{true} = 50$, and $n = 1,000$, and averaged results over 10 data sets. We report results of CIO and SS with $k = k_{true}$ and $\gamma = \dfrac{p}{2n\, k_{true}\, \max_i \|x_i\|^2}$.

B.3 Real-world design matrix X

In this section, we consider a real-world design matrix X and generated synthetic noisy signals Y for 10 levels of noise. Figure B.8 (p. 11) represents the out-of-sample MSE of all five methods as SNR increases. As mentioned, the difference in MSE between methods is far less acute than the difference observed in terms of accuracy and false detection.

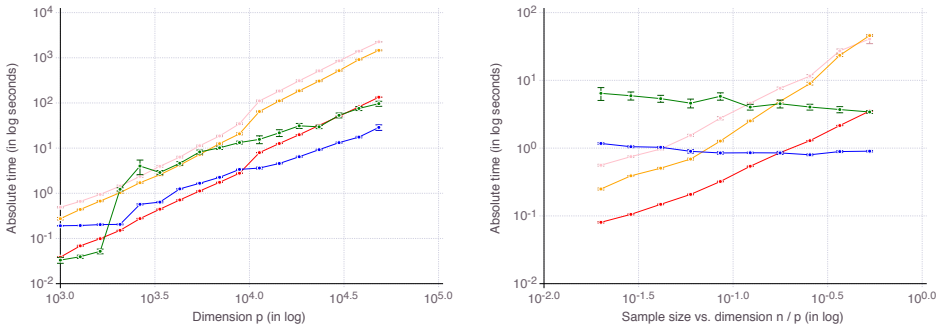
APPENDIX C: NUMERICAL EXPERIMENTS FOR CLASSIFICATION - SUPPLEMENTARY MATERIAL

C.1 Synthetic data satisfying mutual incoherence condition

C.1.1 Feature selection with a given support size We first consider the case when the cardinality k of the support to be returned is given and equal to the true sparsity k_{true} for all methods.

As shown on Figure C.1 (p. 12), all methods converge in terms of accuracy. That is their ability to select correct features as measured by A smoothly converges to 1 with an increasing number of observations $n \to \infty$. Compared to regression, the difference in accuracy between methods is much narrower. MCP now slightly dominates all methods, including CIO and SS. The suboptimality gap between the discrete optimization method and its Boolean relaxation appears to be much smaller as well and the two methods perform almost identically.

Figure C.2 on page 13 reports relative computational time compared to `glmnet` in log scale. It should be kept in mind that we restricted the cutting-plane algorithm to a 180-second time limit and the sub-gradient algorithm to $T_{max} = 200$ iterations. `glmnet` is still the fastest method in general, but it should be emphasized that other methods terminate in times at most two orders of magnitude larger, which is often an affordable price to pay in practice. Combined with results in accuracy from Figure C.1, such an observation speaks in favor of a wider use of cardinality-constrained or non-convex formulations in data analysis practice. As previously



(a) p, for $n/p = 1/5$ (b) Sample size n/p, for $p = 5,000$

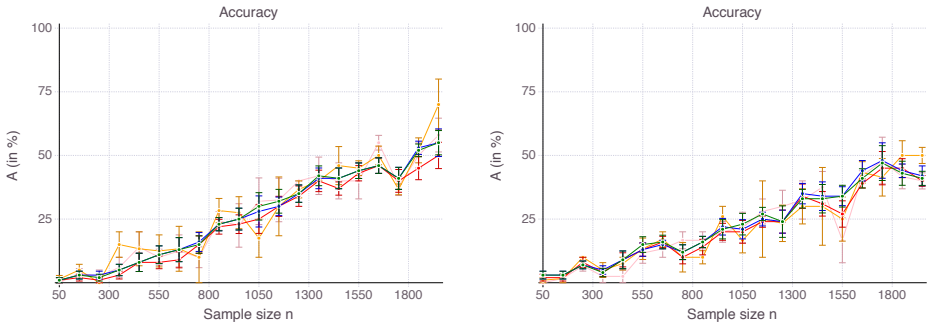
Fig B.4: Absolute computational time as dimension p or sample size n increases, for CIO (in green), SS (in blue with $T_{max} = 200$), ENet (in red), MCP (in orange), SCAD (in pink) with OLS loss. We fixed $p = 5,000$, $k_{true} = 50$ and averaged results over 10 data sets. We report results of CIO and SS with $k = k_{true}$ and $\gamma = \dfrac{p}{2n\, k_{true}\, \max_i \|x_i\|^2}$.

mentioned, for the sub-gradient algorithm, using an additional stopping criterion would drastically cut computational time (by a factor 2 at least) but would also deteriorate the quality of the solution significantly for such classification problems.

Figure C.3 (p. 14) represents the out-of-sample error $1 - AUC$ for all five methods, as n increases, for the six noise/correlation settings of interest. There is a clear connection between performance in terms of accuracy and in terms of predictive power, with CIO performing the best. Still, better predictive power does not necessarily imply that the features selected are more accurate. As we have seen for instance, MCP often demonstrates the highest accuracy, yet not the highest AUC.

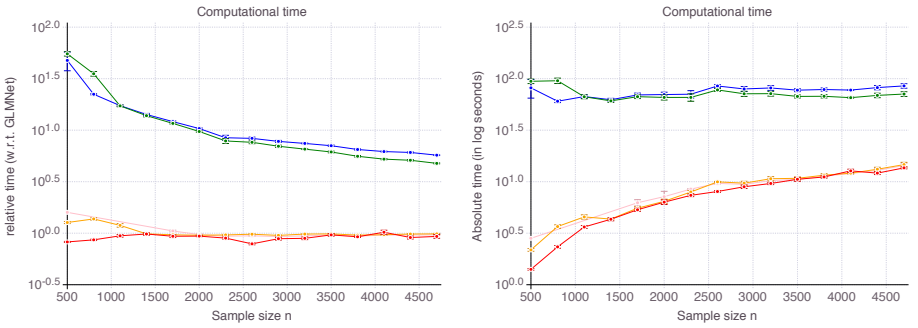
REFERENCES

[1] Loh, P. L. and Wainwright, M. J. (2017). Support recovery without incoherence: A case for nonconvex regularization. *The Annals of Statistics* **45** 2455–2482.

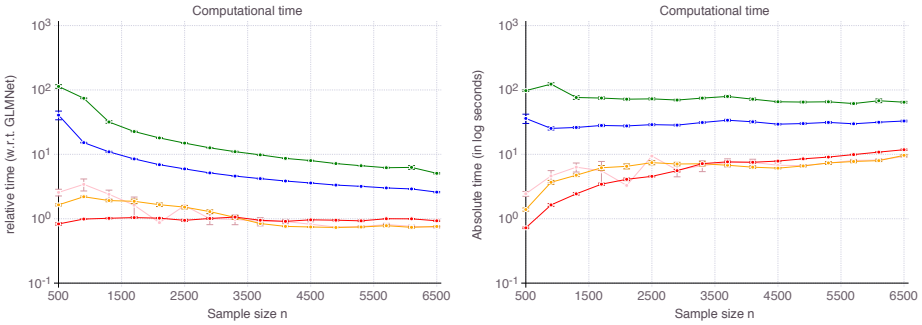


(a) High noise, low correlation (b) High noise, high correlation

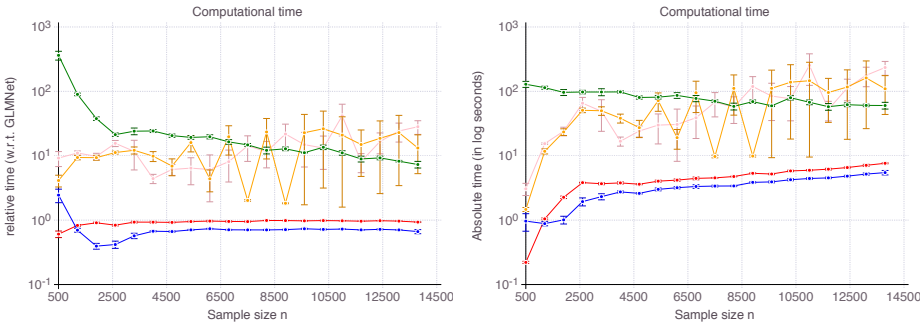
Fig B.5: Accuracy as n increases, for the CIO (in green), SS (in blue with $T_{max} = 200$), ENet (in red), MCP (in orange), SCAD (in pink) with OLS loss, under the mutual incoherence condition. We average results over 10 data sets.



(a) Low noise

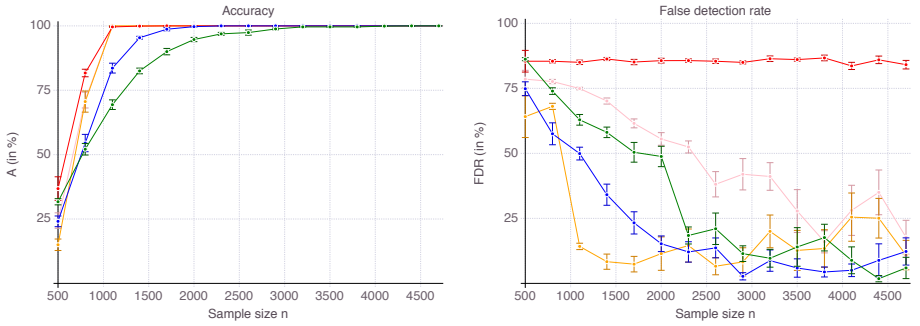


(b) Medium noise

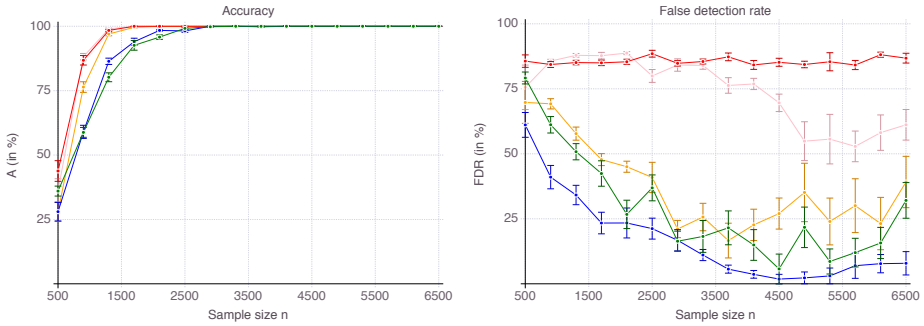


(c) High noise

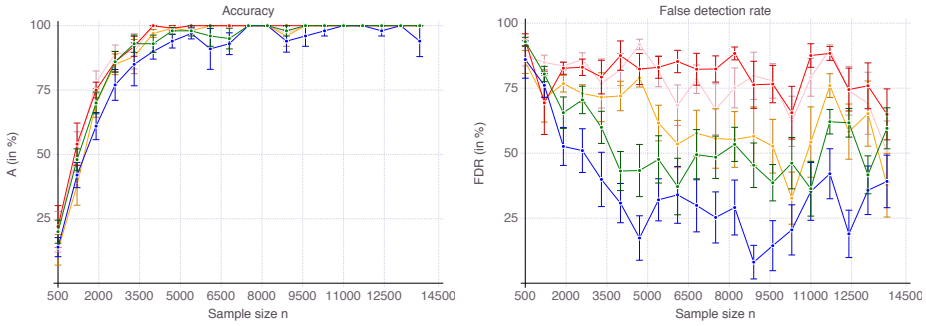
Fig B.6: Relative (left panel) and absolute (right panel) computational timeas n increases, for CIO (in green), SS (in blue with $T_{max} = 200$), ENet (in red), MCP (in orange), SCAD (in pink) with OLS loss, when mutual incoherence does not hold. We average results over 10 data sets.



(a) Low noise



(b) Medium noise



(c) High noise

Fig B.7: Accuracy A (left panel) and false detection rate FDR (right panel) as n increases, for the CIO (in green), SS (in blue with $T_{max} = 150$), ENet (in red), MCP (in orange), SCAD (in pink) with OLS loss, when mutual incoherence does not hold. We average results over 10 data sets.

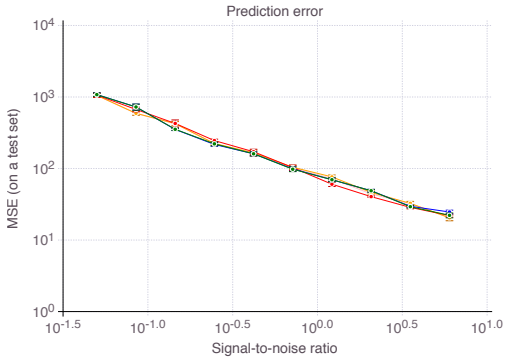
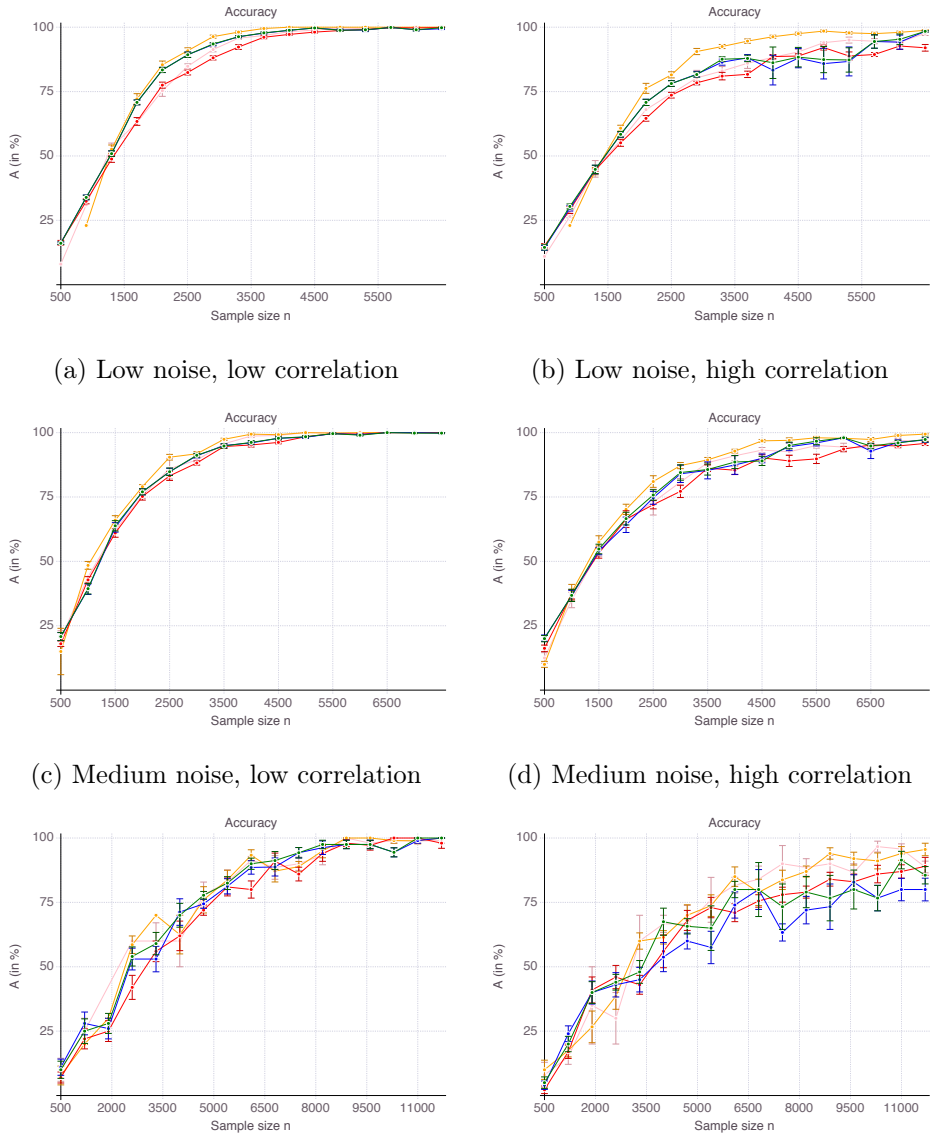


Fig B.8: Out-of-sample MSE as SNR increases, for the CIO (in green), SS (in blue with $T_{max} = 150$), ENet (in red), MCP (in orange), SCAD (in pink) with OLS loss. We average results over 10 data sets with $SNR = 0.05, ..., 6$, $k_{true} = 50$ and real-world design matrix X.



(a) Low noise, low correlation

(b) Low noise, high correlation

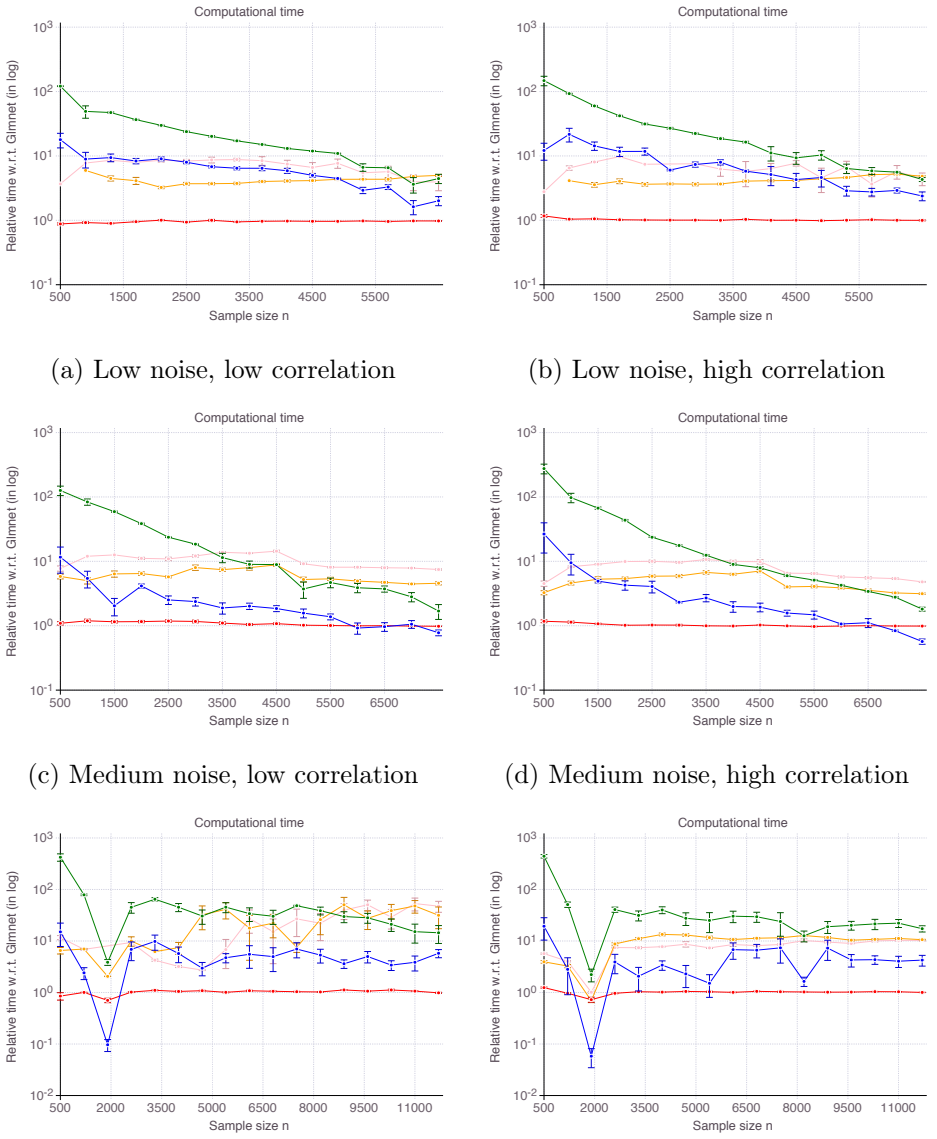
(c) Medium noise, low correlation

(d) Medium noise, high correlation

(e) High noise, low correlation

(f) High noise, high correlation

Fig C.1: Accuracy as n increases, for the CIO (in green), SS (in blue with $T_{max} = 200$) with Hinge loss , ENet (in red), MCP (in orange), SCAD (in pink) with logistic loss, under the mutual incoherence condition. We average results over 10 data sets.



(a) Low noise, low correlation

(b) Low noise, high correlation

(c) Medium noise, low correlation

(d) Medium noise, high correlation

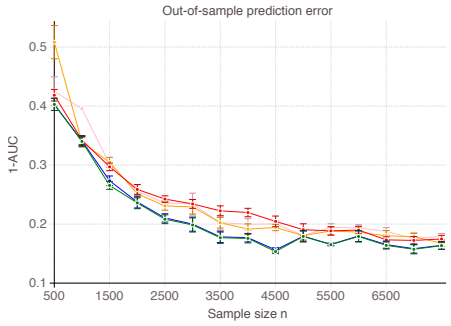
(e) High noise, low correlation

(f) High noise, high correlation

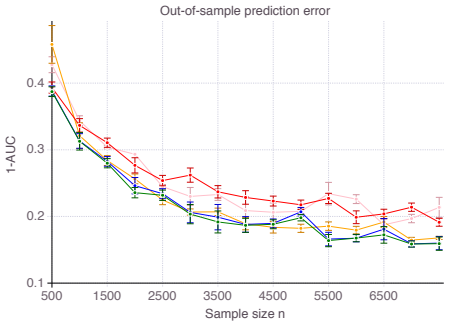
Fig C.2: Computational time relative to Lasso with glmnet as n increases, for CIO (in green), SS (in blue with $T_{max} = 200$) with Hinge loss, ENet (in red), MCP (in orange), SCAD (in pink) with logistic loss, under the mutual incoherence condition. We average results over 10 data sets.

(a) Low noise, low correlation

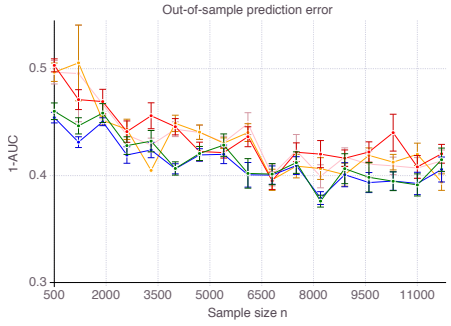
(b) Low noise, high correlation



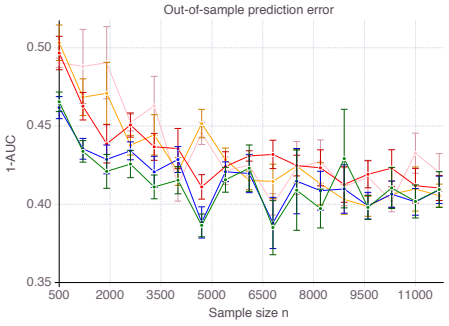
(c) Medium noise, low correlation



(d) Medium noise, high correlation



(e) High noise, low correlation



(f) High noise, high correlation

Fig C.3: Out-of-sample $1 - AUC$ as n increases, for the CIO (in green), SS (in blue with $T_{max} = 200$) with Hinge loss, ENet (in red), MCP (in orange), SCAD (in pink) with logistic loss, under the mutual incoherence condition. We average results over 10 data sets.