Ensembles of Lasso Screening Rules

Seunghak Lee, Nico Görnitz, Eric P. Xing, David Heckerman, and Christoph Lippert

Abstract—In order to solve large-scale lasso problems, screening algorithms have been developed that discard features with zero coefficients based on a computationally efficient screening rule. Most existing screening rules were developed from a spherical constraint and half-space constraints on a dual optimal solution. However, existing rules admit at most two half-space constraints due to the computational cost incurred by the half-spaces, even though additional constraints may be useful to discard more features. In this paper, we present AdaScreen, an adaptive lasso screening rule ensemble, which allows to combine any one sphere with multiple half-space constraints on a dual optimal solution. Thanks to geometrical considerations that lead to a simple closed form solution for AdaScreen, we can incorporate multiple half-space constraints at small computational cost. In our experiments, we show that AdaScreen with multiple half-space constraints simultaneously improves screening performance and speeds up lasso solvers.

Index Terms—Lasso, screening rule, ensemble

1 Introduction

In modern applications of machine learning, data sets with a large number of features are common. Examples include human genomes [3], gene expression measurements [6], spam data [28], and social media data [13]. Given such data, one of the fundamental challenges in machine learning is feature selection, that is, given input features (e.g., genetic variants) and an output variable to predict (e.g., disease status), select a subset of the input features that is relevant to the output. Among the many feature selection techniques, the lasso [22] has been very successful at solving the feature selection problem under the assumption that most features are irrelevant to the output [22]. However, solving the lasso problems on very high dimensional data (e.g., billions of features) still remains a computational challenge. In other words, either it takes too long to solve such big lasso problems, or the input data are too large to store in memory.

To address the computational problem, researchers have developed screening algorithms that can be used as a preprocessing step to efficiently discard features irrelevant to the output from the lasso model. The key idea is that after the screening step the input feature set is reduced, and we can use any lasso solver to solve the lasso problem on the reduced set. If the screening can be performed efficiently, and the resulting reduced feature set is small enough, we can obtain a lasso solution efficiently. Existing screening rules include the safe feature elimination (SAFE) rule [7], the sphere tests [31], the strong rule [23], the dome test [26], [29], efficient dual polytope projection (EDPP) rule [25], the two hyperplane test [27], the Sasvi rule [16], safe rules for the lasso [11], and GAP Safe screening rules for sparse multi-task and multi-class models [19]. Except for the strong rule, all screening rules are safe (i.e., features discarded by screening are guaranteed to be excluded in an optimal lasso solution). Recently, screening rules for graphical lasso [18] and L1 logistic regression have been developed [24]; further, Bonnefoy et al. [1] proposed a dynamic screening, where features are discarded in every iteration of an optimization algorithm if a certain condition is met. In this paper, we limit ourselves to lasso screening algorithms used before applying the optimization algorithms. A summary of existing lasso screening rules is shown in Table 1.

We note that the screening algorithms described in this paper are different from sure-screening algorithms [8], [9], [10], which, instead of discarding features with zero coefficients in a global optimal solution of the lasso problem, aims at discarding features irrelevant to the output (features uninvolved in the true model). Also, consistency of the lasso model [33] is beyond the scope of this paper.

While screening techniques have been successful in enabling us to solve big lasso problems efficiently, they become ineffective in discarding irrelevant features as lasso optimal solutions become non-sparse. The reason for this degradation is that most screening methods rely on finding a region containing the optimal solution that is as small as possible. For a non-sparse optimal solution such a region typically is difficult to estimate, resulting in low screening efficiency. One can incorporate multiple half-space constraints into a screening rule to mitigate the problem, but for existing screening algorithms the advantage of this is substantially reduced due the increased amount of computation needed to evaluate the screening rule.

In this paper, we present an adaptive screening rule ensemble for the lasso, referred to as AdaScreen. We first propose an adaptive screening rule ensemble that can

- *S. Lee and C. Lippert are with Human Longevity Inc., Mountain View, CA 94041. E-mail: {leeseunghak, christoph.a.lippert}@gmail.com.*
- *E. P. Xing is with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. E-mail: epxing@cs.cmu.edu.*
- *N. Görnitz is with the Machine Learning Group, Department of Software Engineering and Theoretical Computer Science, Berlin Institute of Technology, Berlin 10578, Germany. E-mail: nico.goernitz@tu-berlin.de.*
- *D. Heckerman is with Microsoft Research, Los Angeles, CA 90024. E-mail: heckerma@microsoft.com.*

Manuscript received 21 Apr. 2016; revised 5 Feb. 2017; accepted 13 Sept. 2017. Date of publication 24 Nov. 2017; date of current version 1 Nov. 2018. (Corresponding author: Seunghak Lee.)
Recommended for acceptance by E. B. Sudderth.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2017.2765321

TABLE 1
Summary of Lasso Screening Algorithms with Their Sphere and Half-Space Constraints (**Seq.** Column Shows Whether Sequential Screening Is Supported or Not)

Algorithm	Sphere Const. (center)	Sphere Const. (radius)	Half-space Const.	Reference	Seq.
Sasvi	$\frac{1}{2}\{\frac{\mathbf{y}}{\lambda}+\boldsymbol{\theta}^*(\lambda_0)\}$	$\frac{1}{2}\|\frac{\mathbf{X}\boldsymbol{\beta}^*(\lambda_0)}{\lambda_0}+(\frac{\mathbf{y}}{\lambda}-\frac{\mathbf{y}}{\lambda_0})\|_2$	$\langle\boldsymbol{\theta}^*(\lambda_0)-\frac{\mathbf{y}}{\lambda_0},\boldsymbol{\theta}(\lambda)-\boldsymbol{\theta}^*(\lambda_0)\rangle\geq 0$	[16]	Yes
Two hyperplane test	adaptive	adaptive	adaptive up to two half-space const.	[30]	Yes
EDPP	$\boldsymbol{\theta}^*(\lambda_0)+\frac{1}{2}\mathbf{v}_2^{\perp}(\lambda,\lambda_0)$	$\frac{1}{2}\|\mathbf{v}_2^{\perp}(\lambda,\lambda_0)\|_2$ (see Sec. 2.1)	None	[25]	Yes
DOME	$\boldsymbol{\theta}^*(\lambda_{max})$	$\|\mathbf{y}\|_2(\frac{1}{\lambda}-\frac{1}{\lambda_{max}})$	$\mathbf{x}_*^T\boldsymbol{\theta}(\lambda)\leq 1, \mathbf{x}_*=argmax_{\mathbf{x}\in\{\pm\mathbf{x}_j\}}\mathbf{x}^T\mathbf{y}$	[26], [29]	Yes
Strong rule	$\boldsymbol{\theta}^*(\lambda_0)$	$2(1-\frac{\lambda}{\lambda_0})$	None	[23]	Yes
Sphere test	$\boldsymbol{\theta}^*(\lambda_{max})-(\frac{\lambda_{max}}{\lambda}-1)\mathbf{x}_*$	$\sqrt{\frac{\|\mathbf{y}\|_2^2}{\lambda_{max}^2}-1}(\frac{\lambda_{max}}{\lambda}-1)$	$\mathbf{x}_*^T\boldsymbol{\theta}(\lambda)\leq 1, \mathbf{x}_*=argmax_{\mathbf{x}\in\{\pm\mathbf{x}_j\}}\mathbf{x}^T\mathbf{y}$	[31]	No
SAFE rule	$\boldsymbol{\theta}^*(\lambda_{max})$	$\|\mathbf{y}\|_2(\frac{1}{\lambda}-\frac{1}{\lambda_{max}})$	None	[7]	No
AdaScreen	adaptive	adaptive	adaptive up to K half-space const.	ours	Yes

include any one sphere and multiple half-space constraints on a dual optimal solution (it is adaptive in the sense that any constraints can be chosen). Then we derive a closed-form solution for AdaScreen, which allows us to efficiently evaluate multiple half-space constraints. The contributions of this paper are as follows: (1) We provide an adaptive screening rule ensemble with an efficient closed-form solution; and (2) we provide several instances of AdaScreen with different choices of constraints on a dual optimal solution. We note that AdaScreen offers a framework to combine multiple screening rules, rather than a novel screening rule. In our experiments, we confirm that AdaScreen can incorporate any sphere and any multiple half-space constraints, each resulting in novel screening rules. Furthermore, across a range of datasets we show that AdaScreen with multiple half-spaces can reject significantly more features with zero coefficients than existing screening rules. Note that AdaScreen is the first screening framework that allows us to improve screening performance, taking advantage of multiple half-space constraints. While in principle any half-space constraints could be used to achieve a computational speed-up, AdaScreen leads to safe screening iff all constraints are derived from safe screening rules. We further experimentally confirmed that the proposed screening rules are safe.

Notations. In the following, upper case letters denote matrices, boldface lowercase letters denote vectors, and italic lowercase letters denote scalars. Subscript denotes column index of matrices or element index of vectors. A superscript asterisk (*) denotes an optimal solution.

2 Background: Lasso Screening

The lasso [22] is defined as a linear regression from a set of J feature vectors $\{\mathbf{x}_j\}_{j=1}^J$ to a target vector $\mathbf{y}\in\mathbb{R}^N$, where N is the sample size, with a squared loss and L_1 norm penalty

$$\min_{\boldsymbol{\beta}}\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\beta}\|_1, \tag{1}$$

where $\mathbf{X}\in\mathbb{R}^{N\times J}$ is a design matrix, where each column represents a feature, $\boldsymbol{\beta}\in\mathbb{R}^J$ is a regression coefficient vector, and the regularization parameter λ controls the sparsity of $\boldsymbol{\beta}$. Lasso screening rules aim at discarding features with zero coefficients at a global optimal solution by computing simple, computationally inexpensive tests for each feature.

We can obtain a screening rule from a Lagrangian of the lasso problem in (1) [25], given by

$$\mathcal{L}(\boldsymbol{\beta},\theta,\mathbf{z})=\frac{1}{2}\|\mathbf{z}\|_2^2+\lambda\|\boldsymbol{\beta}\|_1+\lambda\theta^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}-\mathbf{z}), \tag{2}$$

where θ is a dual variable. By introducing an auxiliary variable $\mathbf{z}=\mathbf{y}-\mathbf{X}\boldsymbol{\beta}$, we can obtain (2) from (1) using the method of Lagrange multipliers [2]. Taking a subgradient of (2) with respect to β_j, we get $\mathbf{x}_j^T\theta^*=s_j$, where s_j is a subgradient of the L1 norm, defined by: if $\beta_j\neq 0$, $s_j=\mathrm{sign}(\beta_j)$; otherwise, $s_j\in[-1,1]$. Based on this, we can derive a screening rule

$$\beta_j^*(\lambda)=0 \text{ if } \left|\mathbf{x}_j^T\boldsymbol{\theta}^*(\lambda)\right|<1. \tag{3}$$

However, use of (3) is impractical because $\boldsymbol{\theta}^*(\lambda)$ is unknown. Instead, we estimate the region where $\boldsymbol{\theta}^*(\lambda)$ exists, denoted by $\boldsymbol{\theta}^*(\lambda)\in\boldsymbol{\Theta}$, and then, use the following rule to identify zero coefficients

$$\beta_j^*(\lambda)=0 \text{ if } \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\left|\mathbf{x}_j^T\boldsymbol{\theta}\right|<1. \tag{4}$$

From (4), we can see that screening rules can be developed by finding $\boldsymbol{\Theta}$ and a solution for $\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}|\mathbf{x}_j^T\boldsymbol{\theta}|$. Furthermore, screening efficiency (i.e., the proportion of rejected features to all features) increases as the size of $\boldsymbol{\Theta}$ decreases, making the left-hand side in (4) smaller. Screening time increases as the time to solve $\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}|\mathbf{x}_j^T\boldsymbol{\theta}|$ increases; thus, a closed-form solution is desirable.

2.1 Screening via Dual Polytope Projection

So far, we have shown how screening rules are developed in general. In this section, we review dual polytope projection (DPP) screening [25]. Screening rules are derived from a dual form of the lasso (See [25])

$$\sup_{\boldsymbol{\theta}}\left\{\frac{1}{2}\|\mathbf{y}\|_2^2-\frac{\lambda^2}{2}\left\|\boldsymbol{\theta}-\frac{\mathbf{y}}{\lambda}\right\|_2^2:\left|\mathbf{x}_j^T\boldsymbol{\theta}\right|\leq 1,\forall j\right\}, \tag{5}$$

where $\boldsymbol{\theta}$ is the dual variable. The optimal parameters in the primal (1) and the dual (5) are related via $\boldsymbol{\theta}^*(\lambda)=\frac{\mathbf{y}-\mathbf{X}\boldsymbol{\beta}^*(\lambda)}{\lambda}$ [25], where we have made the dependence on the regularization parameter explicit. Furthermore, in (5), a dual optimal solution $\boldsymbol{\theta}^*$ can be found by projecting $\frac{\mathbf{y}}{\lambda}$ onto

the constraints $C = \{|\mathbf{x}_j^T\boldsymbol{\theta}| \leq 1, j = 1, \ldots, J\}$. In other words, $\boldsymbol{\theta}^*$ is the vector closest to $\frac{\mathbf{y}}{\lambda}$, which satisfies C because it minimizes the objective function in (5) while satisfying all the constraints. We represent this projection as $P_C(\frac{\mathbf{y}}{\lambda})$, where the projection operator is defined as

$$P_C(\mathbf{w}) = \underset{\boldsymbol{\theta}\in C}{argmin}\|\boldsymbol{\theta} - \mathbf{w}\|_2.$$

In DPP, the $\boldsymbol{\Theta}$ in (4) is obtained using the "nonexpansiveness" property of the projection operator $P_C(\cdot)$ [17]

$$\|P_C(\mathbf{w}_2) - P_C(\mathbf{w}_1)\|_2 \leq \|\mathbf{w}_2 - \mathbf{w}_1\|_2, \ \forall \mathbf{w}_1, \mathbf{w}_2. \tag{6}$$

Using (6), a range of $\boldsymbol{\theta}^*(\lambda) \in \boldsymbol{\Theta}$ can be estimated by

$$\begin{aligned}\|\boldsymbol{\theta}^*(\lambda) - \boldsymbol{\theta}^*(\lambda_0)\|_2 &= \left\| P_C\left(\frac{\mathbf{y}}{\lambda}\right) - P_C\left(\frac{\mathbf{y}}{\lambda_0}\right)\right\|_2 \\ &\leq \left|\frac{1}{\lambda} - \frac{1}{\lambda_0}\right| \|\mathbf{y}\|_2,\end{aligned} \tag{7}$$

where we used the fact that $\boldsymbol{\theta}^*(\lambda) = P_C(\frac{\mathbf{y}}{\lambda})$ and $\boldsymbol{\theta}^*(\lambda_0) = P_C(\frac{\mathbf{y}}{\lambda_0})$. Let us now assume that $\lambda < \lambda_0$ and $\boldsymbol{\theta}^*(\lambda_0)$ are given, which shall be discussed in Section 2.2.

This provides us a region that contains $\boldsymbol{\theta}^*(\lambda)$, that is, $\boldsymbol{\Theta} = B(\mathbf{o}, \rho)$, which represents a sphere centered at $\mathbf{o}$ with radius ρ, where $\mathbf{o} = \boldsymbol{\theta}^*(\lambda_0)$ and $\rho = \left|\frac{1}{\lambda} - \frac{1}{\lambda_0}\right| \|\mathbf{y}\|_2$ from (7). Further, using the spherical region $\boldsymbol{\Theta} = B(\mathbf{o}, \rho)$, $\boldsymbol{\theta}^*(\lambda)$ can be represented as

$$\boldsymbol{\theta}^*(\lambda) = \mathbf{o} + \mathbf{v}, \tag{8}$$

where $\|\mathbf{v}\|_2 \leq \rho$. Using (3) and (8), we can derive a basic DPP lasso screening rule as follows:

$$\left|\mathbf{x}_j^T\boldsymbol{\theta}^*(\lambda)\right| \leq \left|\mathbf{x}_j^T\boldsymbol{\theta}^*(\lambda_0)\right| + \left|\mathbf{x}_j^T\mathbf{v}\right| < 1, \text{ and} \tag{9}$$

$$\left|\mathbf{x}_j^T\boldsymbol{\theta}^*(\lambda_0)\right| < 1 - \|\mathbf{x}_j\|_2\rho, \tag{10}$$

where we used $|\mathbf{x}_j^T\mathbf{v}| \leq \|\mathbf{x}_j\|_2\|\mathbf{v}\|_2 \leq \|\mathbf{x}_j\|_2\rho$ by the Cauchy-Schwarz inequality.

Different screening rules use differently estimated regions for $\boldsymbol{\Theta}$. Based on DPP, Wang et al. [25] developed enhanced DPP (EDPP) that achieves a smaller $\boldsymbol{\Theta}$ than that of DPP via projections of rays and the "firmly nonexpansiveness property" of the projection operator $P_C(\cdot)$. Specifically, EDPP uses a spherical region $\boldsymbol{\Theta} = B(\mathbf{o}, \rho)$, where

$$\mathbf{o} = \boldsymbol{\theta}^*(\lambda_0) + \frac{1}{2}\mathbf{v}_2^{\perp}(\lambda, \lambda_0) \ \& \ \rho = \frac{1}{2}\left\|\mathbf{v}_2^{\perp}(\lambda, \lambda_0)\right\|_2, \tag{11}$$

where $\mathbf{v}_2^{\perp}(\lambda, \lambda_0) = \mathbf{v}_2(\lambda, \lambda_0) - \frac{\langle \mathbf{v}_1(\lambda_0), \mathbf{v}_2(\lambda, \lambda_0)\rangle}{\|\mathbf{v}_1(\lambda_0)\|_2^2}\mathbf{v}_1(\lambda_0)$, and $\mathbf{v}_1(\lambda_0)$ and $\mathbf{v}_2(\lambda, \lambda_0)$ are defined by

$$\mathbf{v}_1(\lambda_0) = \begin{cases} \frac{\mathbf{y}}{\lambda_0} - \boldsymbol{\theta}^*(\lambda_0), & \text{if } \lambda_0 \in (0, \lambda_{max}) \\ \text{sign}(\mathbf{x}_*^T\mathbf{y})\mathbf{x}_*, & \text{if } \lambda_0 = \lambda_{max} \end{cases} \tag{12}$$

$$\mathbf{v}_2(\lambda, \lambda_0) = \frac{\mathbf{y}}{\lambda} - \boldsymbol{\theta}^*(\lambda_0), \tag{13}$$

where $\mathbf{x}_* = \underset{\mathbf{x}_j}{argmax}\,|\mathbf{x}_j^T\mathbf{y}|$. In comparison to DPP, EDPP uses smaller ρ, yielding more efficient screening. We refer readers to [25] for the details of EDPP.

2.2 One-Shot and Sequential Screening

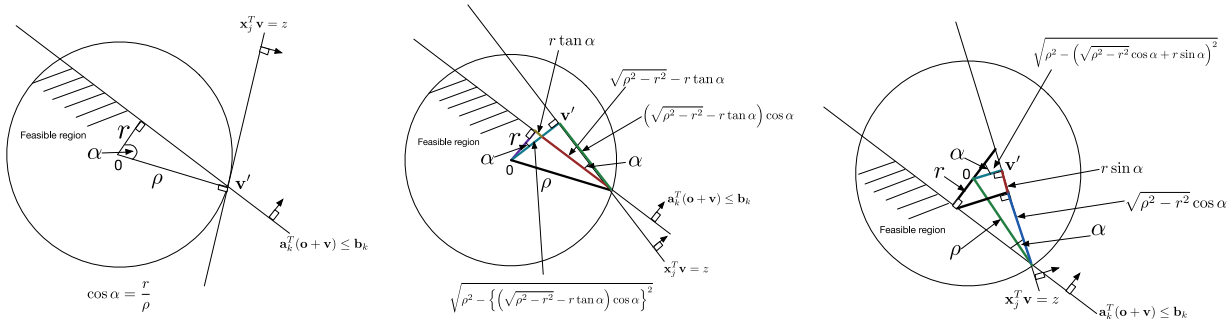
Here we describe how screening rules are used in two different scenarios: the first is one-shot screening, and the second is sequential screening. One-shot screening is for when we want to solve lasso problems for a single λ parameter. In sequential screening, we want to solve lasso problems for a given sequence of λ parameters in descending order, $\boldsymbol{\Lambda} = \{\lambda_1, \ldots, \lambda_T\}$. Typically, $\lambda_1 = \lambda_{max}$, where λ_{max} is the smallest λ that sets $\boldsymbol{\beta}^*(\lambda_{max}) = \mathbf{0}$. In this setting, we discard features based on a screening rule for λ_t given $\boldsymbol{\beta}_{t-1}^*$ $(t \geq 2)$ and then run a lasso solver on the unscreened features, resulting in $\boldsymbol{\beta}_t^*$. Subsequently, based on $\boldsymbol{\beta}_t^*$, we keep iterating screening and the lasso solver until the lasso solution for desired λ is achieved. It is called sequential screening due to its sequential nature. Throughout this paper, we assume a sequential screening setting; thus we denote λ_0 by the previous λ parameter (i.e., $\lambda_0 \equiv \lambda_{t-1}$) and λ by the current one (i.e., $\lambda \equiv \lambda_t$). Note that one-shot screening can be easily obtained by setting $\lambda_0 \equiv \lambda_{max}$.

It is well known that screening efficiency degrades as the gap between λ and λ_0 increases [25]. Thus, when desired λ is small, sequential screening is more appropriate than the one-shot version because it may include intermediate parameters between λ_{max} and λ. However, even sequential screening becomes more ineffective as the lasso solution becomes dense. This phenomenon can be explained as follows. Suppose that we solve lasso problems with a sequence of geometrically spaced λ parameters, that is, $\lambda = \alpha\lambda_0$, where $0 < \alpha < 1$. Then for both DPP and EDPP, as λ_0 decreases, ρ increases. For DPP, $\rho = \|\mathbf{y}\|\left|\frac{1}{\lambda} - \frac{1}{\lambda_0}\right| = \|\mathbf{y}\|\frac{1-\alpha}{\alpha}\left|\frac{1}{\lambda_0}\right|$; for EDPP, $\rho = \|\mathbf{y}\|\left|\frac{1}{\lambda} - \frac{r}{\lambda_0}\right| + \boldsymbol{\theta}^*(r-1) = \|\mathbf{y}\|\left|\frac{1-\alpha r}{\alpha}\right|\frac{1}{\lambda_0} + \boldsymbol{\theta}^*(r-1)$, where $r = \frac{\langle \mathbf{v}_1(\lambda_0), \mathbf{v}_2(\lambda, \lambda_0)\rangle}{\|\mathbf{v}_1(\lambda_0)\|_2^2}$. Obviously, as the sphere radius of ρ increases, the screening efficiency degrades. The same argument can be applied to linearly or logarithmically spaced λ sequences. Note that other screening rules have the same issue because they also rely on the spherical region containing $\boldsymbol{\theta}^*(\lambda)$ with the radius of ρ (see Table 1).

3 AdaScreen: Adaptive Lasso Screening Rule Ensemble

In this section, to improve the screening efficiency for small λ parameters, we present AdaScreen, an adaptive form of lasso screening that can take any one sphere constraint, and any multiple half-space constraints on $\boldsymbol{\theta}^*(\lambda)$. Note that AdaScreen can also be viewed as a general lasso screening framework, which can be instantiated with any sphere and half-space constraints; the whole space of AdaScreen remains to be explored.

Recall that lasso screening rules are derived based on the estimation of the region $\boldsymbol{\Theta}$ that includes a dual optimal solution $\boldsymbol{\theta}^*(\lambda)$. Given a sphere constraint $\|\boldsymbol{\theta}^*(\lambda) - \mathbf{o}\| \leq \rho$, one can represent $\boldsymbol{\theta}^*(\lambda) = \mathbf{o} + \mathbf{v}$, where $\mathbf{o}$ is a fixed center of a sphere, and $\mathbf{v}$ is a free vector with $\|\mathbf{v}\| \leq \rho$. Furthermore, K half-space constraints on $\boldsymbol{\theta}^*(\lambda)$ are represented by $\mathbf{a}_k^T\boldsymbol{\theta}^*(\lambda) \leq b_k$ or $\mathbf{a}_k^T(\mathbf{o} + \mathbf{v}) \leq b_k, k = 1, \ldots K$. Now we propose AdaScreen



(a) Case 1 (b) Case 2-1 (c) Case 2-2

Fig. 1. Graphical illustration for three different cases, where z is maximized in $\mathbf{x}_j^T\mathbf{v}=z$. (a): the case where $\mathbf{x}_j^T\mathbf{v}=z$ is tangent to the sphere at its contact, and meets the half-space constraint; (b): the case where $b_k-\mathbf{a}_k^T\mathbf{o}\ge 0$, and the hyperplane $\mathbf{x}_j^T\mathbf{v}=z$ meets the intersection between the sphere and the half-space constraint; (c): the case where $b_k-\mathbf{a}_k^T\mathbf{o}<0$, and the hyperplane $\mathbf{x}_j^T\mathbf{v}=z$ meets the intersection between the sphere and the half-space constraint.

$$\beta_j^*(\lambda)=0 \text{ if } \max(S_j^+,S_j^-)<1, \tag{14}$$

where S_j^+ and S_j^- are defined by

$$\begin{aligned} S_j^+ &\equiv \mathbf{x}_j^T\mathbf{o}+\sup_{\mathbf{v}:\|\mathbf{v}\|_2\le\rho,\ \{\mathbf{a}_k^T(\mathbf{o}+\mathbf{v})\le b_k\}\in\mathcal{D}} \mathbf{x}_j^T\mathbf{v}\,, \quad \text{and} \\ S_j^- &\equiv -\mathbf{x}_j^T\mathbf{o}+\sup_{\mathbf{v}:\|\mathbf{v}\|_2\le\rho,\ \{\mathbf{a}_k^T(\mathbf{o}+\mathbf{v})\le b_k\}\in\mathcal{D}} -\mathbf{x}_j^T\mathbf{v}\,. \end{aligned} \tag{15}$$

Here $\mathcal{D}$ is a set of user-defined half-space constraints on $\boldsymbol{\theta}^*(\lambda)$. One can easily see that AdaScreen is derived as follows:

$$\begin{aligned} \left|\mathbf{x}_j^T\boldsymbol{\theta}^*(\lambda)\right|<1 &\Leftrightarrow \sup_{\mathbf{v}:\|\mathbf{v}\|_2\le\rho,\ \{\mathbf{a}_k^T(\mathbf{o}+\mathbf{v})\}\le b_k\in\mathcal{D}} \left|\mathbf{x}_j^T(\mathbf{o}+\mathbf{v})\right|<1 \\ &\Leftrightarrow \sup_{\mathbf{v}:\|\mathbf{v}\|_2\le\rho,\ \{\mathbf{a}_k^T(\mathbf{o}+\mathbf{v})\}\le b_k\in\mathcal{D}} \max\left\{\mathbf{x}_j^T(\mathbf{o}+\mathbf{v}),-\mathbf{x}_j^T(\mathbf{o}+\mathbf{v})\right\}<1 \\ &\Leftrightarrow \max\left(S_j^+,S_j^-\right)<1. \end{aligned}$$

Note that S_j^+ and S_j^- can be solved in the same way except that we replace $\mathbf{x}_j$ by $-\mathbf{x}_j$. Thus, without loss of generality, we restrict the presentation to a closed-form solution for S_j^+ only.

AdaScreen has the following unique properties: first, one can incorporate any sphere or half-space constraints into (15), resulting in a new lasso screening rule. Second, AdaScreen directly estimates $\sup\mathbf{x}_j^T\mathbf{v}$ (or $\sup-\mathbf{x}_j^T\mathbf{v}$) for achieving a better bound without resorting to only the Cauchy Schwarz inequality. Both help us achieve a better screening rule. The former makes $\boldsymbol{\Theta}$ smaller by allowing us to add more constraints; the latter gives us a smaller feasible region for a half-space constraint considering the angle between $\mathbf{x}_j$ and $\mathbf{v}$. We start with a closed-form solution for AdaScreen with a single half-space constraint in $\mathcal{D}$ and a single sphere constraint. Then, we will extend it to the case for multiple half-space constraints, followed by the discussion on the choice of the constraints. AdaScreen is summarized in Algorithm 2.

3.1 Closed-form Solution with One Sphere and One Half-Space Constraint

To efficiently compute S_j^+, we first derive a closed-form solution given one sphere and one half-space constraint. In fact, Sasvi [16] derived a closed-form solution for the same problem when one sphere and one half-space come from variational inequality constraints. Here we seek a different closed-form solution based on geometry, which satisfies the following requirements. First, we need a general solution that allows any sphere and any half-space constraints; second, we need a solution that admits additional half-space constraints with negligible computational overhead.

In S_j^+, $\mathbf{x}_j^T\mathbf{o}$ is a constant, thus solving S_j^+ boils down to optimizing the following problem:

$$\begin{aligned} \max_{\mathbf{v}} \quad & \mathbf{x}_j^T\mathbf{v} \\ \text{subject to} \quad & \mathbf{a}_k^T\mathbf{o}+\mathbf{a}_k^T\mathbf{v}\le b_k, \\ & \|\mathbf{v}\|_2\le\rho. \end{aligned} \tag{16}$$

Here we assume that (16) includes a non-empty feasible region. A sphere constraint can be provided by any existing screening methods such as EDPP, and the candidate half-space constraints shall be discussed in Section 3.3.

One can view (16) as the problem of finding a hyperplane $\mathbf{x}_j^T\mathbf{v}=z$ with the maximum of z such that two constraints in (16) are satisfied. Since $\mathbf{x}_j^T\mathbf{v}$ is linear in $\mathbf{v}$, its maximum can be obtained in the following two extreme cases: (1) $\mathbf{x}_j^T\mathbf{v}=z$ is tangent to the sphere $\|\mathbf{v}\|_2\le\rho$ at the contact; (2) $\mathbf{x}_j^T\mathbf{v}=z$ meets the intersection between the half-space and the sphere. Below, we show that for all cases, a solution is given by the following closed-form

$$\sup_{\mathbf{v}:\|\mathbf{v}\|\le\rho,\ \mathbf{a}_k^T\mathbf{o}+\mathbf{a}_k^T\mathbf{v}\le b_k} \mathbf{x}_j^T\mathbf{v}=\left\|\mathbf{x}_j\right\|_2\rho', \tag{17}$$

where ρ' is a constant, determined by each case.

We define $\cos\alpha=\frac{\mathbf{x}_j^T\mathbf{a}_k}{\left\|\mathbf{x}_j\right\|_2\|\mathbf{a}_k\|_2}$, where α is the angle between $\mathbf{x}_j$ and $\mathbf{a}_k$, and $r=\frac{\left|b_k-\mathbf{a}_k^T\mathbf{o}\right|}{\|\mathbf{a}_k\|_2}$, i.e., the distance between $\mathbf{a}_k^T\mathbf{o}+\mathbf{a}_k^T\mathbf{v}=b_k$ and the origin. Hereafter, let us consider the case when $0\le\alpha\le\pi$ (if $\pi<\alpha\le 2\pi$, the same derivation can be applied by setting $\alpha\leftarrow 2\pi-\alpha$).

In the first case, $\mathbf{x}_j^T\mathbf{v}=z$ meets the sphere constraint at its boundary, and thus we disregard the half-space constraint. Using Cauchy Schwarz inequality, the maximum of $\mathbf{x}_j^T\mathbf{v}$ is given by $\sup_{\mathbf{v},\|\mathbf{v}\|\le\rho}\mathbf{x}_j^T\mathbf{v}=\|\mathbf{x}_j^T\|\rho$. Thus, $\rho'=\rho$. The geometry in Fig. 1a shows that if $\cos\alpha\le\frac{r}{\rho}$, we use this case. Note that if $\frac{\pi}{2}<\alpha\le\pi$, $\cos\alpha<0\Rightarrow\cos\alpha\le\frac{r}{\rho}$; thus, for the next case, the range of α should be $0\le\alpha\le\frac{\pi}{2}$.

In the second case, $\mathbf{x}_j^T\mathbf{v}=z$ meets the intersection between the half-space and the sphere constraint. We define $\sin\alpha=\sqrt{1-\cos^2\alpha}$, which is non-negative because $0\le\alpha\le\frac{\pi}{2}$ for this case. Let us consider the following two sub-cases.

(1) *Sub-case with* $b_k - \mathbf{a}_k^T\mathbf{o} \geq 0$. The geometry of this case is depicted in Fig. 1b, where $\mathbf{v}'$ is the vector on the hyperplane $\mathbf{x}_j^T\mathbf{v} = z$ with the direction $\frac{\mathbf{x}_j}{\|\mathbf{x}_j\|_2}$ and the length ρ', given by

$$\rho' = \sqrt{\rho^2 - \left\{\left(\sqrt{\rho^2 - r^2} - r\tan\alpha\right)\cos\alpha\right\}^2} = \sqrt{\rho^2 - \left(\sqrt{\rho^2 - r^2}\cos\alpha - r\sin\alpha\right)^2}. \quad (18)$$

Plugging $\mathbf{v}' = \rho'\frac{\mathbf{x}_j}{\|\mathbf{x}_j\|_2}$ into $\mathbf{x}_j^T\mathbf{v}'$, we get the maximum of the objective in (17).

(2) *Sub-case with* $b_k - \mathbf{a}_k^T\mathbf{o} < 0$.The geometry of this case is depicted in Fig. 1c, where $\mathbf{v}'$ is the vector on the hyperplane $\mathbf{x}_j^T\mathbf{v} = z$ with the direction $\frac{\mathbf{x}_j}{\|\mathbf{x}_j\|_2}$ and the length ρ', given by

$$\rho' = \sqrt{\rho^2 - \left(\sqrt{\rho^2 - r^2}\cos\alpha + r\sin\alpha\right)^2}. \quad (19)$$

The maximum z is obtained by plugging ρ' into (17).

Summarizing the above results, we have the following condition for ρ'

$$\rho' = \begin{cases} \rho & \text{if } \cos\alpha \leq \frac{r}{\rho} \\ \sqrt{\rho^2 - \left(\sqrt{\rho^2 - r^2}\cos\alpha - r\sin\alpha\right)^2} & \text{if } \cos\alpha > \frac{r}{\rho} \text{ and } b_k - \mathbf{a}_k^T\mathbf{o} \geq 0 \\ \sqrt{\rho^2 - \left(\sqrt{\rho^2 - r^2}\cos\alpha + r\sin\alpha\right)^2} & \text{if } \cos\alpha > \frac{r}{\rho} \text{ and } b_k - \mathbf{a}_k^T\mathbf{o} < 0. \end{cases} \quad (20)$$

3.2 AdaScreen with Multiple Half-Space Constraints

Solving (16), we showed how to compute the maximum of $\mathbf{x}_j^T\mathbf{v}$ with one sphere and one half-space constraint. However, we want to use multiple half-space constraints, denoted by $\mathcal{D}$. Here we show that an upper bound on $\mathbf{x}_j^T\mathbf{v}$ with multiple half-space constraints can be found by solving (16) multiple times ($|\mathcal{D}|$ times) and taking the minimum among them, as follows:

$$\sup_{\mathbf{v}:\|\mathbf{v}\|\leq\rho,\,\{\mathbf{a}_k^T(\mathbf{o}+\mathbf{v})\leq b_k\}\in\mathcal{D}} \mathbf{x}_j^T\mathbf{v} \quad \leq \quad \sup_{\mathbf{v}:\|\mathbf{v}\|\leq\rho,\,\mathbf{a}_k^T(\mathbf{o}+\mathbf{v})\leq b_k\in\mathcal{D}} \mathbf{x}_j^T\mathbf{v}, \forall\, k \quad (21)$$

$$\Leftrightarrow \sup_{\mathbf{v}:\|\mathbf{v}\|\leq\rho,\,\{\mathbf{a}_k^T(\mathbf{o}+\mathbf{v})\leq b_k\}\in\mathcal{D}\}} \mathbf{x}_j^T\mathbf{v} \quad \leq \min_{k\in\{1,\ldots,K\}} \left(\sup_{\mathbf{v}\,:\,\|\mathbf{v}\|_2\leq\,\rho,\; \mathbf{a}_k^T(\mathbf{o}+\mathbf{v})\,\leq\, b_k\,\in\,\mathcal{D}} \mathbf{x}_j^T\mathbf{v} \right). \quad (22)$$

In Algorithm 1, we summarize a closed-form solution for S_j^+ with a sphere and multiple half-space constraints. We note that AdaScreen is safe given a sphere and half-space constraints from safe screening rules, as shown below.

Proposition 1. *AdaScreen never discards non-zeros in* $\boldsymbol{\beta}^*(\lambda)$*, given sphere and half-space constraints obtained from safe screening rules.*

Proof 1. Suppose that we are given sphere and half-space constraints from safe screening rules. From (14), a safe lasso screening rule is $\beta_j^*(\lambda) = 0$ if $\max(S_j^+, S_j^-) < 1$, where $S_j^+ = \mathbf{x}_j^T\mathbf{o} + \sup_{\mathbf{v}:\|\mathbf{v}\|_2\leq\rho,\,\{\mathbf{a}_k^T(\mathbf{o}+\mathbf{v})\leq b_k\}\in\mathcal{D}} \mathbf{x}_j^T\mathbf{v}$. Here, we show that AdaScreen computes an upper bound on S_j^+, leading to a safe screening rule. S_j^- can be shown in a similar fashion. Since the first term $\mathbf{x}_j^T\mathbf{o}$ is a constant, let us consider the second term. When a single half-space constraint is given, AdaScreen takes a closed-form solution for the second term, given by (17); when multiple half-space constraints are given, AdaScreen takes an upper bound on S_j^+, given by (22). Therefore, AdaScreen takes an upper bound on $\max(S_j^+, S_j^-)$, and it is a safe lasso screening rule. □

Algorithm 1. A Closed-Form Solution for AdaScreen

1: Input: $q_j, \rho, \{\mathbf{a}_k^T\boldsymbol{\theta}^*(\lambda_t) \leq b_k, h_k, r_k, \cos\alpha_{jk}, \sin\alpha_{jk}\}_{k=1}^K$
2: Output: S_j
3: Find ${\rho'}_{jk}$ based on the conditions in (20), $\forall k = 1, \ldots, K$
4: ${\rho''}_j \leftarrow \min_k {\rho'}_{jk}$
5: $S_j \leftarrow q_j + \|\mathbf{x}_j\|_2 {\rho''}_j$

Algorithm 2. Sequential Screening with AdaScreen

Input: $\mathbf{X}, \mathbf{y}, \Lambda = \{\lambda_1(=\lambda_{max}), \lambda_2, \ldots, \lambda_T\}$, $\lambda_{t-1} > \lambda_t$, for $t \geq 2$, $\mathcal{H}_{global} = \{\mathbf{a}_k^T\boldsymbol{\theta}^*(\lambda_t) \leq b_k\}_{k=1}^L$
Output: $\boldsymbol{\beta}^*(\lambda_2), \ldots, \boldsymbol{\beta}^*(\lambda_T)$
$\cos\alpha_{jk} = \frac{\mathbf{x}_j^T\mathbf{a}_k}{\|\mathbf{x}_j\|_2\|\mathbf{a}_k\|_2}, \forall j, k = 1, \ldots, L$
$\sin\alpha_{jk} = \sqrt{1 - \cos^2\alpha_{jk}}, \forall j, k = 1, \ldots, L$
$\boldsymbol{\theta}^*(\lambda_1) = \frac{\mathbf{y}}{\lambda_1}$
for $t = 2$ to T **do**
  Find local half-spaces: $\mathcal{H}_{local} = \{\mathbf{a}_k^T\boldsymbol{\theta}^*(\lambda_t) \leq b_k\}_{k=L+1}^K$ (see section 3.3)
  $\cos\alpha_{jk} = \frac{\mathbf{x}_j^T\mathbf{a}_k}{\|\mathbf{x}_j\|_2\|\mathbf{a}_k\|_2}, \forall j, k = (L+1), \ldots, K$
  $\sin\alpha_{jk} = \sqrt{1 - \cos^2\alpha_{jk}}, \forall j, k = (L+1), \ldots, K$
  Given $\boldsymbol{\theta}^*(\lambda_{t-1})$ and λ_t, find a sphere constraint, represented by $\mathbf{o}$ and ρ
  $q_j = \mathbf{x}_j^T\mathbf{o}, \forall j$
  $h_k = \mathbf{a}_k^T\mathbf{o}, \forall k$
  $r_k = \frac{|b_k - h_k|}{\|\mathbf{a}_k\|_2}, \forall k$
  for $j = 1$ to J **do**
    $S_j^+ \leftarrow$ Algo1$(q_j, \rho, \{\mathbf{a}_k^T\boldsymbol{\theta}^* \leq b_k, h_k, r_k, \cos\alpha_{jk}, \sin\alpha_{jk}\})$
    $S_j^- \leftarrow$ Algo1$(-q_j, \rho, \{\mathbf{a}_k^T\boldsymbol{\theta}^* \leq b_k, h_k, r_k, -\cos\alpha_{jk}, \sin\alpha_{jk}\})$
  end for
  $U = \{j : S_j^+ \geq 1 \text{ or } S_j^- \geq 1, \forall j\}$
  Obtain $\boldsymbol{\beta}^*(\lambda_t)$, solving lasso with U
  $\boldsymbol{\theta}^*(\lambda_t) = \frac{\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*(\lambda_t)}{\lambda_t}$
end for

3.3 Selection of Half-Space Constraints

So far, we presented how to solve AdaScreen, assuming that one sphere and K half-space constraints are given by users. Here we discuss how to select half-space constraints that may increase the screening efficiency.

The following set of half-space constraints can be useful to increase the screening efficiency:

$$\left\{\left|\mathbf{x}_k^T(\mathbf{o} + \mathbf{v})\right| \leq 1 : \beta_k^*(\lambda_0) \neq 0, \forall k\right\}, \quad (23)$$

$$\left(\frac{\mathbf{y}}{\lambda_0} - \boldsymbol{\theta}^*(\lambda_0)\right)^T (\mathbf{o} + \mathbf{v}) \leq \left(\frac{\mathbf{y}}{\lambda_0} - \boldsymbol{\theta}^*(\lambda_0)\right)^T \boldsymbol{\theta}^*(\lambda_0). \quad (24)$$

The set of constraints in (23) is originated from dual lasso constraints; (24) stems from variational inequality [16], proven to be useful. We propose that useful global and half-space constraints can be selected in the pool of constraints in (23) and (24).

Here we claim that the half-spaces in (23) can be useful for screening. To achieve a small $\rho' < \rho$ in (20), we need $\cos\alpha > \frac{r}{\rho}$, and thus small r is desirable. Let us denote $\boldsymbol{\theta}^*(\lambda) = \mathbf{o} + \mathbf{v}^*$ ($\mathbf{v}^*$ is a fixed unknown vector). By dual lasso constraints, $\left|\mathbf{x}_k^T(\mathbf{o} + \mathbf{v}^*)\right| \leq 1$, $\forall k$. Thus, $\mathbf{x}_k^T(\mathbf{o} + \mathbf{v}^*) \leq 1$ and $-\mathbf{x}_k^T(\mathbf{o} + \mathbf{v}^*) \leq 1$, $\forall k$, that lead to the following: $\max(\frac{\mathbf{x}_k^T\mathbf{v}^*}{\|\mathbf{x}_k\|}, \frac{-\mathbf{x}_k^T\mathbf{v}^*}{\|\mathbf{x}_k\|}) \leq r$, where equality holds when $\beta_k^*(\lambda) \neq 0$. Thus, for each feature, r is smallest when $\beta_k^*(\lambda) \neq 0$. Furthermore, Proposition 2 shows that $\{\beta_k^*(\lambda)\}$ can never be discarded by an independent screening rule if $\{\beta_k^*(\lambda_0) \neq 0\}$; thus, (23) give us nontrivial constraints.

Proposition 2. *Suppose that we are given λ_0 and λ ($\lambda_0 > \lambda$). For all $j = 1, \ldots, J$, if $\beta_j^*(\lambda_0) \neq 0$, then $\mathbf{x}_j$ can never be discarded by any independent screening rules with λ.*

Proof 2. We assume that $\beta_k^*(\lambda_0) \neq 0$, that is,

$$\left|\mathbf{x}_k^T\boldsymbol{\theta}^*(\lambda_0)\right| = 1. \quad (25)$$

Suppose that there exists an independent screening rule which can discard $\mathbf{x}_k$ at λ, that is

$$\left|\mathbf{x}_k^T\boldsymbol{\theta}^*(\lambda)\right| < 1. \quad (26)$$

Note that a screening rule estimates the region of $\boldsymbol{\theta}^*(\lambda)$ based on $\boldsymbol{\theta}^*(\lambda_0)$ as follows: $\boldsymbol{\theta}^*(\lambda) = \boldsymbol{\theta}^*(\lambda_0) + \mathbf{v}$, $\|\mathbf{v}\| \leq \rho$. By plugging it into (26), we get

$$\sup_{\mathbf{v}:\|\mathbf{v}\|\leq\rho} \left|\mathbf{x}_k^T\boldsymbol{\theta}^*(\lambda_0) + \mathbf{x}_k^T\mathbf{v}\right| < 1 \quad (27)$$

$$\Rightarrow \left|\mathbf{x}_k^T\boldsymbol{\theta}^*(\lambda_0)\right| < 1. \quad (28)$$

However, (28) contradicts to the assumption (25), and thus the proof is completed. □

For half-space constraints, one may use (24) by setting $\lambda_0 = \lambda_{max}$ and/or a subset of constraints in (23) using the following procedure. Following the λ sequence, we start solving a lasso problem using screening with only one sphere constraint (lasso can be solved efficiently with large λ). When the number of nonzero coefficients is larger than a user-defined threshold at λ_t, we generate half-space constraints based on (23). From λ_{t+1}, we use the half-space constraints for screening.

We note that combination of (23) and (24) generates a feasible set for $\boldsymbol{\theta}^*(\lambda)$ because the former is a feasible set for $\boldsymbol{\theta}^*(\lambda)$ by the dual form of the lasso (5) and it has been proven that the latter is a feasible set for $\boldsymbol{\theta}^*(\lambda)$ [16]. Therefore, AdaScreen is safe if combination of (23) and (24) and a sphere constraint from a safe screening rule are used.

3.4 Analysis of Screening Time Complexity

The half-space constraints can be divided into global and local constraints. The global half-space constraints (H_{global}) refer to the half-space constraints which are used for all features and all λ parameters. The local half-space constraints (H_{local}) are generated for each λ, but applicable to all features.

Here we analyze the screening time complexity under two different scenarios: AdaScreen with (1) K global half-space constraints, and (2) K local half-space constraints. Let us first consider the screening complexity of AdaScreen when K global half-spaces are used. In such a case, the time complexity of AdaScreen is $\mathcal{O}(JN|\Lambda| + JNK + JK|\Lambda|)$, where $\mathcal{O}(JN|\Lambda|)$ stems from a sphere constraint, and $\mathcal{O}(JNK + JK|\Lambda|)$ stems from K half-space constraints. Thus, if $K << |\Lambda|$ and $K << N$, K half-space constraints barely affects the screening speed. When using K local half-space constraints, the time complexity of AdaScreen is $\mathcal{O}(JNK|\Lambda|)$. Therefore, approximately, use of local half-space constraints is K times more expensive than use of global half-space constraints. For sparse data, depending on screening rules, the time complexity can be reduced. For example, for each feature vector, given DPP and a local half-space in Eq. (23), the time complexity of AdaScreen is $O(\max(N', Q')K|\Lambda|)$, where N' is the number of nonzeros in the feature vector of interest and Q' is the number of nonzeros in the normal vector of the local half-space constraint. In Algorithm 2, we summarize AdaScreen with one sphere constraint, L global half-space constraints, and $K - L$ local half-space constraints. In practice, a small number of local half-space constraints (e.g., 5 half-spaces) is sufficient to significantly benefit from the constraints.

4 Experiments

We systematically evaluate the screening performance in simulations and several real world datasets. To get a broad comparison of existing screening rules, we chose DPP [25], DOME [29], SAFE [7], strong rule [23], EDPP [25], and Sasvi [16], covering both classical as well as recent methods. For the classical ones including DOME and SAFE rules, we used one-shot screening, while for the others, we used a sequential screening setting. We settled for three instances of AdaScreen that incorporate constraints from the advanced techniques EDPP and Sasvi. Each instance uses EDPP sphere constraint with (a) Sasvi half-space in (24) as global half-space constraint (EDPP+Sasvi); (b) 100 half-spaces in (23) as local half-space constraints (EDPP + DualLasso(100)); and (c) all half-space constraints in (a) and (b) (EDPP+Sasvi+DualLasso(100)).

We evaluated screening performances based on the feature rejection ratio, defined by

$$\frac{\#\text{ of discarded features by screening}}{\#\text{ of features with true zero coefficients}},$$

and the runtime used to solve the lasso problems over a sequence of λ parameters. Throughout the experiments, we use the standard coordinate descent algorithm [12], implemented in scikit-learn [20], with tolerance of 10^{-4}. We use a geometrically spaced sequence of 65 λ values and step length 0.9, starting from λ_{max} to $\sim 10^{-3}\lambda_{max}$. Experiments

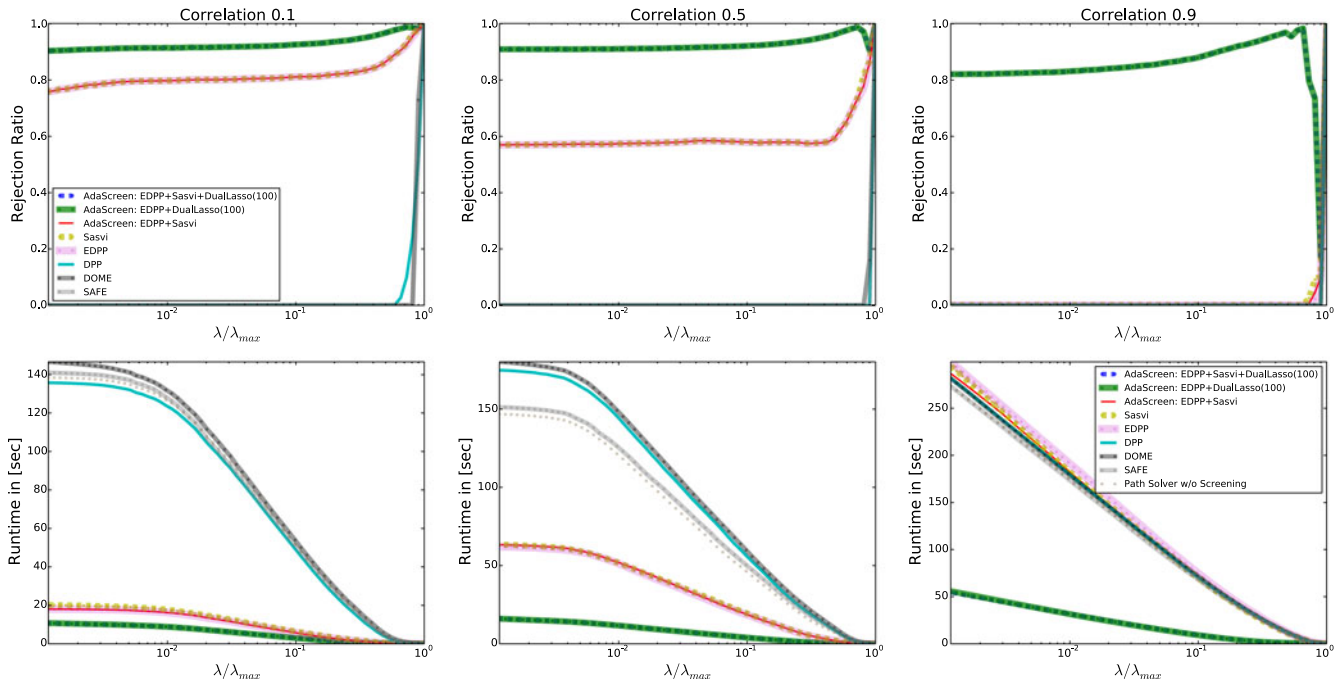


Fig. 2. Rejection ratio and runtime of various screening methods on simulated datasets with the feature correlations of 0.1 (first column), 0.5 (second column), and 0.9 (third column) (see text for details) given a sequence of λ parameters. Two AdaScreen instances with 100 dual lasso half-space constraints outperformed the other methods in terms of both rejection ratio and runtime.

were run on a single machine with 48 2.1 GHz AMD cores and 384 GB RAM.

4.1 Simulation Study

We first investigate screening performance over different degrees of feature correlations in the simulated data. Note that as more features are correlated with each other, the lasso takes longer to converge. This can be seen by inspection of the coordinate descent update rule [12]: $\beta_j^{(t)} \leftarrow S(\mathbf{x}_j^T\mathbf{y} - \sum_{k\neq j}\mathbf{x}_j^T\mathbf{x}_k\beta_k^{(t-1)}, \lambda)$, where $S(\beta_j, \lambda) \equiv \text{sign}(\beta)(|\beta| - \lambda)$. If $\mathbf{x}_j^T\mathbf{x}_k = 0, \forall j \neq k$ (i.e., all features are completely independent), the update rule becomes $\beta_j^{(t)} \leftarrow S(\mathbf{x}_j^T\mathbf{y}, \lambda)$, and then the optimal j-th coefficient can be found by a single update. When features are highly correlated, multiple iterations are needed to converge the lasso objective because the update of one coefficient affects optimal values of other coefficients. We generated simulation data with 250 samples and 10,000 dimensions as follows: $\mathbf{x}_1 \leftarrow \mathbf{r}$ and $\mathbf{x}_j \leftarrow c\mathbf{x}_{j-1} + (1-c)\mathbf{r}$ for $j \geq 2$, where $\mathbf{r} \in \mathbb{R}^{250}$ is a random vector drawn from a uniform distribution $\text{unif}(0, 1)$, and c represents the degree of feature correlations in the simulated data.

4.1.1 Effects of Feature Correlations on Screening

Fig. 2 shows the rejection ratio and runtime of the screening methods on simulated datasets with the degree of feature correlation (c) set to 0.1, 0.5 and 0.9. Here we did not have the case $c = 0$ (all features are independent from each other) because it is a trivial case for lasso optimization and existing sequential screening methods (e.g., EDPP) and AdaScreen perform equally well. In all settings, AdaScreen with local half-space constraints achieved better rejection ratio and runtime than all the other methods. Notably, the performance gap between AdaScreen with local half-spaces and others was more substantial when the degree of feature correlations was high (e.g., c = 0.5 & c = 0.9); further, AdaScreen's rejection ratio was barely affected by the degree of feature correlations. It demonstrates that when features are highly correlated, local half-spaces effectively reduce the region $\boldsymbol{\Theta}$, resulting in high rejection ratio. We also observed that recent methods (AdaScreen, Sasvi, EDPP) dramatically outperformed the classical methods (DPP, DOME, SAFE). In Fig. 2 (second and third column), AdaScreen's rejection ratio dropped when $\lambda = 0.9\lambda_{max}$ decreased from λ_{max} because there exist no local half-spaces to be used (i.e., (23) is the empty set because $\beta_k^*(\lambda_{max}) = 0, \forall k$); after that, the rejection ratio increased taking advantage of local half-spaces.

4.1.2 Effects of Local Half-Space Constraints on Screening

Fig. 3 (third column) shows the impact of increasing the number of local half-space constraints in (23), in terms of the rejection ratio. As expected, the more half-space constraints are taken into account, the higher the rejection ratios. Interestingly, AdaScreen with five local half-spaces showed substantially better rejection ratio than AdaScreen with one local half-space, and the benefits were quickly saturated when we allowed ≥ 5 local half-space constraints.

4.1.3 Comparison between AdaScreen and Trivial Combination of Screening Rules (BagScreen)

Here we show that AdaScreen provides a non-trivial way of combining multiple constraints for screening. To this end, let us compare AdaScreen with BagScreen. BagScreen simply tests multiple screening rules, and then discards the j-th feature if any of them discards it. To ensure fair comparison, for AdaScreen, we used the EDPP spherical constraint, and Sasvi and DOME half-spaces as global half-space constraints; for BagScreen, we combined EDPP, Sasvi, and DOME rules. Fig. 3 (first column) shows the rejection ratios

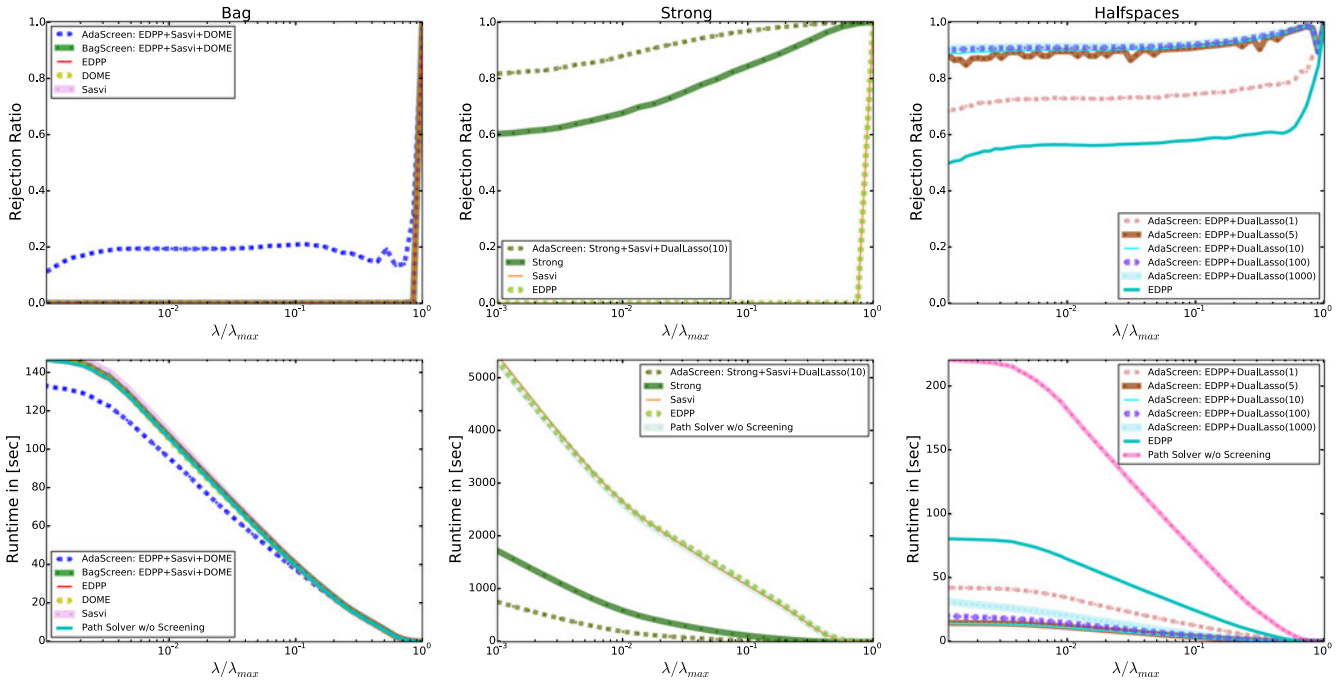


Fig. 3. Rejection ratio (first row) and runtime (second row) for comparison between AdaScreen and BagScreen (first column); comparison between AdaScreen and strong rule (second column); and demonstration of the effects of local half-spaces on screening (third column) on simulated datasets with the feature correlation of 0.5 given a sequence of λ parameters.

by AdaScreen and BagScreen given the simulated data with feature correlation $c = 0.5$ and step length 0.85 for the geometric sequence. AdaScreen maintained > 0.1 rejection ratios throughout all λs, but BagScreen discarded no features for most λs. It is because AdaScreen generated an efficient screening rule with a sphere and multiple half-space constraints, while BagScreen is a set of weaker screening rules with a sphere and a half-space constraint. Here high rejection ratios were unachievable by AdaScreen because we did not use local half-spaces. The experimental results clearly demonstrate that AdaScreen combines multiple constraints in a non-trivial way, resulting in high screening performance.

4.1.4 Comparison between AdaScreen and Strong Rule

The strong rule has been developed as a screening algorithm for lasso-type problems such as lasso and L1 logistic regression. It is developed based on an assumption that given λ and λ_0, correlation between $\mathbf{x}_j$ and the residual (i.e., $\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*)$) does not change more than $|\lambda - \lambda_0|$. Therefore, the strong rule is not safe, meaning that it may discard features whose coefficients are nonzero in a global optimal solution. In Fig. 3 (second column) we compare the rejection ratios by strong rule, AdaScreen, Sasvi, and EDPP in the simulated data with the feature correlation $c = 0.5$ for 1,000 samples; we used step length 0.75 for the geometric sequence. Notably, AdaScreen substantially outperformed the strong rule, and the strong rule was significantly better than Sasvi and EDPP. This result shows that with multiple half-spaces, we can obtain safe screening rules that are even superior to the strong rule.

4.2 Experiments on Real Datasets

We performed experiments on real world datasets including PEMS [4], [15], Alzheimer's disease [32] and PIE [21] datasets. The PEMS input data contains 440 samples (daily records) and 138672 features (963 sensors by 144 time-stamps) describing the occupancy rate of multiple car lanes in San Francisco; the PEMS output data consists of the day of the week. The input from the Alzheimer's disease dataset contains 540 samples (270 disease and 270 healthy individuals) and 511997 features (genetic variants); its output data contains the expression levels of a randomly selected gene. The PIE dataset (11554 examples and 1024 features) is a face recognition dataset that contains 11554 gray face images of 68 people under various conditions and expressions. For the output of PIE dataset, we randomly chose one feature in PIE images, and then generated input data by concatenating all features except the selected feature for output. We note that PEMS and Alzheimer's disease are large-scale datasets for a single machine setting; for example, it took more than 3 hours for coordinate descent lasso solver implemented in scikit-learn with the DPP screening rule to solve the lasso problem on the full Alzheimer's disease dataset.

4.2.1 Screening Efficiency and Runtimes against Baseline Competitors

Fig. 4 shows our main result on real-world data sets, namely rejection ratio and runtime by AdaScreen, Sasvi, EDPP, DPP, DOME, and SAFE rules on PEMS, PIE, and Alzheimer's disease datasets. For PEMS dataset, AdaScreen with 100 local half-space constraints maintained very high rejection ratios (> 0.9) throughout all λ parameters, and it also showed significantly better runtime than all the other methods. For Alzheimer's disease and PIE datasets, AdaScreen with local half-spaces also showed the best screening performance for all λ parameters. These results confirm that multiple local half-spaces employed by AdaScreen are truly useful to improve the rejection ratio at a low computational cost; as a result, we achieved a speedup in runtime compared to the screening rules without local half-space constraints.

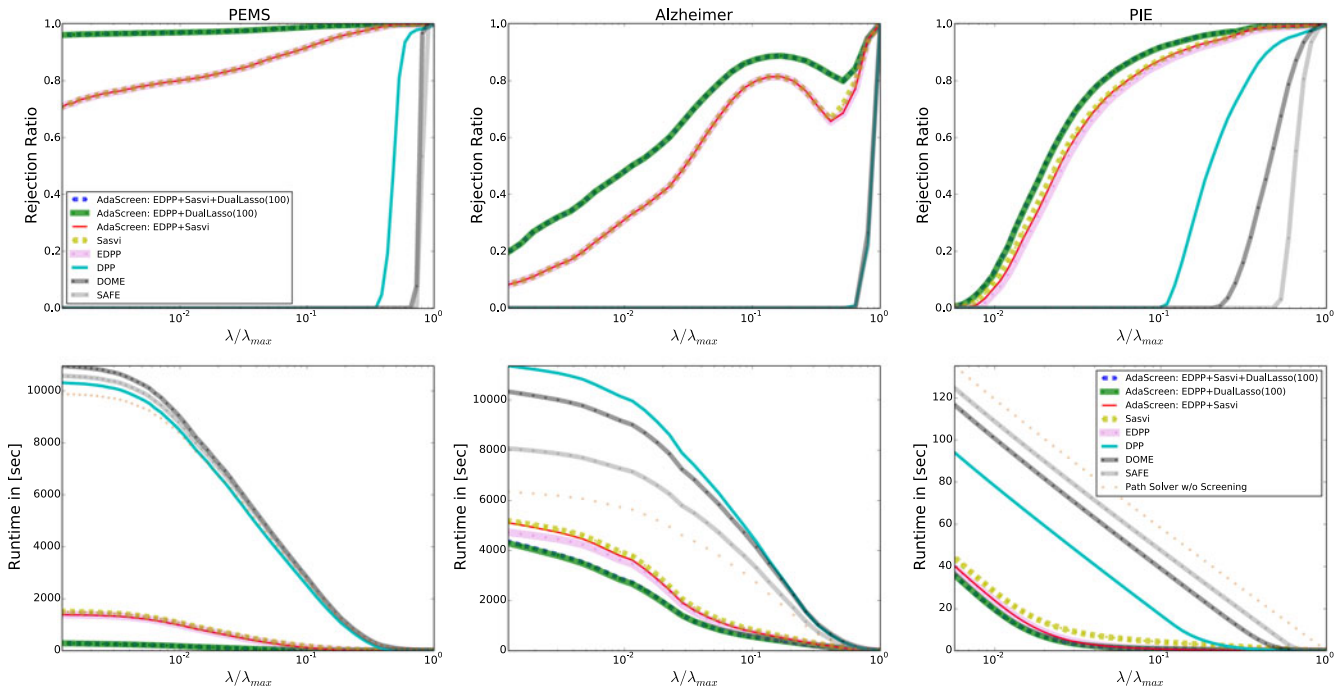


Fig. 4. Rejection ratio and runtime on PEMS (first column), Alzheimer's disease (second column), and PIE image (third column) datasets by three instances of AdaScreen, Sasvi, EDPP, DPP, DOME, and SAFE rules given a sequence of λ parameters.

4.2.2 Speed-Up Comparison over Path-Solver without AdaScreen

So far, experiments showed screening rejection ratios and accumulated runtime behavior. While runtime behavior is a very informative measure to know exactly *how much time it takes to get to this* λ, it obfuscates the driving reason on where the optimization benefits the most from screening, i.e. where the difference in computation time is the highest. Fig. 5 shows the expected accumulated speedup when comparing the path-solver with screening against the path-solver without screening. As can be seen in the figures, all three experiments gain the most in the beginning of the λ sequence. This is partially due to high screening ratios as well as longer distances between consecutive λ's where the path solver needs more iterations to reach a sufficient optimum. Experiments have been repeated 10 times and mean speedup values are reported.

4.2.3 Accuracy Assessment and Impact of Regularization

We are interested in solving a specific problem and hence, interested in reaching a specific λ for which the problem is solved sufficiently accurate. To assess the accuracy of lasso with λs tested in our experiments, we divided the data into training (80 percent) and test sets (remaining 20 percent). Fig. 6 shows the accuracy on the test sets in mean squared error (MSE) for the three real-world datasets PIE, PEMS, and Alzheimer. The experiment was repeated 10 times and mean accuracies are reported. For PEMS and PIE, accuracies seem to reach a plateau for decreasing λ, as more and more features get activated. For Alzheimer on the other hand, a minimum error is reached for a low number of features early on in the λ sequence. It demonstrates that the range of λ parameters tested includes practically useful λ. We also note that in applications for feature selection [14], lasso solution is useful even when the test error is not minimal because the goal is to find a small feature set associated with outputs.

4.2.4 Solver Comparison

Fig. 7 shows the speedup performance on PIE dataset comparing various path-solver using AdaScreen against the corresponding path-solver without screening. We chose 5 distinct solvers demonstrating the usefulness of our approach among different types of optimization techniques.

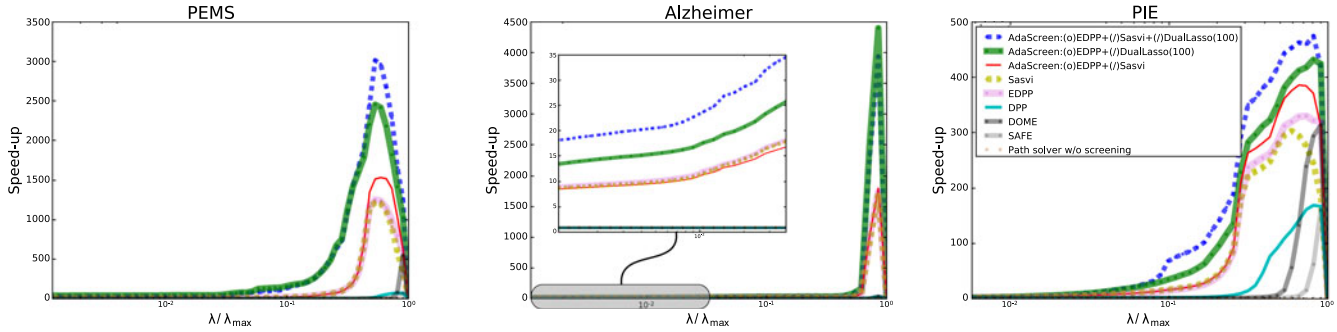


Fig. 5. Speed-up comparison of a variety of screening algorithms on PEMS (left), Alzheimer's disease (center), and PIE image (right) dataset.

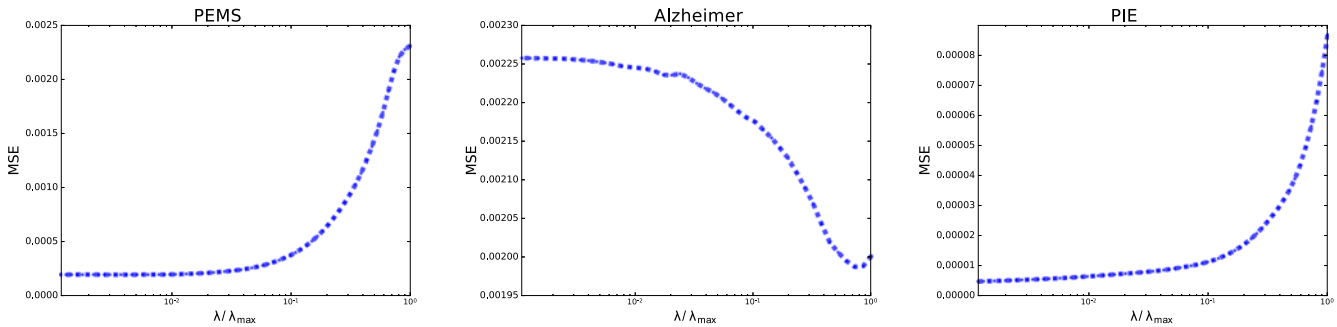


Fig. 6. Accuracy in mean squared error (MSE) for varying regularization parameter λ on PEMS (left), Alzheimer's disease (center), and PIE image (right) dataset.

Among those is our standard method used throughout the experiments, a coordinate descent solver (implementation used from scikit-learn), a LARS solver, a coordinate descent solver with active set selection (based on scikit-learn coordinate descent solver), proximal gradient descent solver, and accelerated proximal gradient descent solver. We repeated the experiments 10 times and report the means of the measured speedup. Fig. 7 demonstrates that AdaScreen is able to achieve significant speed-up ($> 100\times$) for a wide range of λs with various solvers with different optimization algorithms.

5 Software & Practical Considerations

Previous sections dealt with the principal properties of AdaScreen for ensemble screening. In this section, we will introduce our Python software package which is easily extendable for other solvers, screening rules, and settings.

5.1 Implementation

We developed our screening framework in the Python programming language and it can be freely downloaded at http://nicococo.github.io/AdaScreen/ or conveniently downloaded and installed automatically, using the Python *pip* command. An UML (Unified Modeling Language) diagram of our implementation is shown in Fig. 8. It is designed to efficiently implement various screening rules without changing the lasso path solver (e.g., scikit-learn lasso solver [20], cf. Table 2). Even though different screening rules require different constraints and equations, they all share common data structures; thus, we wrap all of them into a single framework. An advantage of this approach is that the lasso path solver needs to interact with only one abstract class for screening rules.

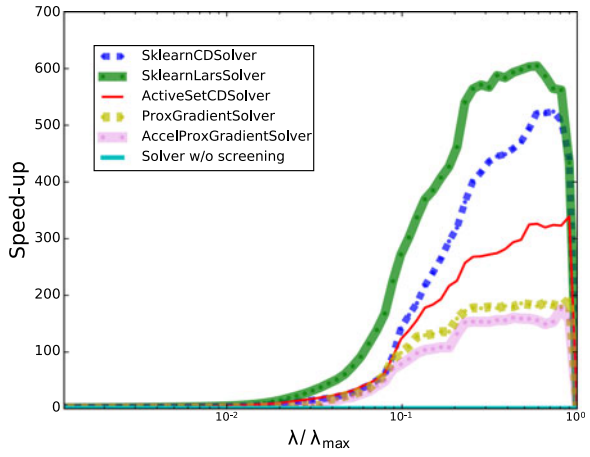


Fig. 7. The speed-up for various solvers using AdaScreen when compared against the corresponding solver without screening.

Algorithm 3. Lasso Path Solver

1: Input: $\mathbf{X}, \mathbf{y}, \{\lambda_1(=\lambda_{max}), \lambda_2, \ldots, \lambda_T\}, \lambda_{t-1} > \lambda_t$, for $t \geq 2$
2: Output: $\boldsymbol{\beta}^*(\lambda_2), \ldots, \boldsymbol{\beta}^*(\lambda_T)$
3: $\alpha_j = \lambda_1, \forall j$ and $\boldsymbol{\theta}^*(\lambda_1) = \frac{\mathbf{y}}{\lambda_1}$
4: **for** $t = 2$ to T **do**
5: $\quad \alpha_j \leftarrow$ given $\boldsymbol{\theta}^*(\lambda_{t-1})$, $\beta_j^*(\lambda_t) = 0$ based on a screening rule, $\forall j$
6: $\quad U \leftarrow \{j : \alpha_j > \lambda_t, \forall j\}$
7: $\quad$ Obtain $\boldsymbol{\beta}^*(\lambda_t)$, solving lasso with U
8: $\quad \boldsymbol{\theta}^*(\lambda_t) = \frac{\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*(\lambda_t)}{\lambda_t}$
9: **end for**

To systematically manage data structures involved in screening, we divide them into Globals and Locals. Globals refer to variables that do not change over the lambda path (e.g., the inputs X, y, λ_{max}). In contrast, Locals refer to variables that change over the lambda path (e.g., the previous regularization parameter λ_0 or the solution $\boldsymbol{\beta}^*(\lambda_0)$ from the previous λ_0).

Furthermore, we designed our screening framework in such a way that all screening rules can be derived from the abstract base class. Screening rules with a single sphere constraint can be implemented by overloading the getSphere function. For more advanced rules, corresponding functions need to be overloaded. For example, to implement AdaScreen with EDPP sphere constraint and Sasvi local half-space constraint, we first instantiate EDPP, Sasvi, and AdaScreen. Then in AdaScreen, we simply call setSphereRule(EDPP) and addHalfspace(Sasvi).

6 Discussions

We presented an adaptive lasso screening rule ensemble, AdaScreen, which can include any sphere and multiple half-space constraints. AdaScreen takes advantage of multiple half-spaces based on a simple, computationally efficient closed-form solution. We experimentally validated that AdaScreen benefits from multiple half-spaces and compared the rejection ratio and the runtime performance against a set of state-of-the-art competitors as well as a naïve implementation of the screening ensemble (BagScreen). Further, we provide a Python software package, which includes various screening methods, solvers, and settings.

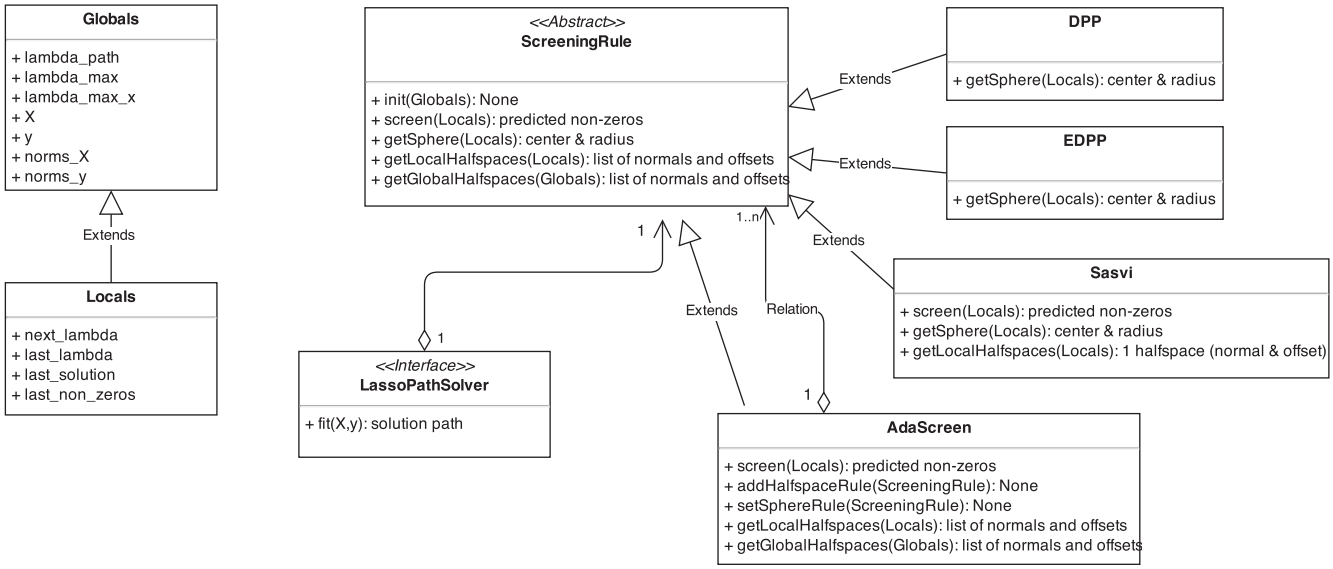


Fig. 8. UML diagram of our screening implementation: The modular design allows us to easily implement screening rules by integrating any sphere and any multiple half-space constraints.

Some datasets consist of a large number of categorical or binary features with only few entries set to non-zero values. In those cases, it would suffice to use single sphere constraint based screening rules (e.g. EDPP). Naturally, those datasets only maintain a weak correlation structure with features or examples being orthogonal (i.e. $\langle x_i, x_j \rangle = 0,\ i \neq j$) frequently. Fig. 2 shows that the gain of using AdaScreen with multiple halfspace constraints decreases when the correlation structure is less prominent.

It is worthwhile to mention that we considered lasso problems with a sequence of λ parameters. However, we often solve a lasso problem with a fixed λ. To this end, data adaptive sequential screening (DASS) has been developed whose key idea is to choose a feedback-controlled sequence of λ parameters to reach λ, which attempts to increase screening efficiency across a λ sequence. A promising direction of future research for AdaScreen would be to use a λ sequence suggested by DASS, rather than a fixed one.

Furthermore, one interesting direction of research is to develop distributed AdaScreen with a parallel lasso algorithm to solve very large problems. For such a parallel screening, MapReduce [5] would be an appropriate framework because screening rules are embarrassingly parallel. Furthermore, extensions of AdaScreen that deal with different loss functions such as logistic loss or hinge loss would be an interesting research direction. We are also interested in incorporating AdaScreen in the lasso optimization procedure, along the lines of Bonnefoy et al. [1] to further improve rejection ratio and runtime.

TABLE 2
List of Screening Rules and Their Properties Implemented in Our Screening Software Package

Name	Sequential	One-shot	Safe	Strong	Ensemble
DPP	x	x	x	-	-
EDPP	x	x	x	-	-
Sasvi	x	x	x	-	-
DOME	x	x	x	-	-
ST3	x	x	x	-	-
SAFE	-	x	x	-	-
Seq-SAFE	x	x	x	-	-
Strong Rule	x	x	-	x	-
This Work					
AdaScreen	x	x	x	x	x

ACKNOWLEDGMENTS

Part of the work was done while SL, NG, and CL were with Microsoft Research, Los Angeles. NG was supported by BMBF ALICE II grant 01IB15001B, and EPX was supported by NIH R01GM114311 and NIH P30DA035778. Seunghak Lee, Nico Görnitz, and Christoph Lippert contributed equally to this work.

REFERENCES

[1] A. Bonnefoy, V. Emiya, L. Ralaivola, and R. Gribonval, "A dynamic screening test principle for the lasso," in *Proc. Eur. Signal Process. Conf.*, pp. 5121–5132, 2014.
[2] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U. K.: Cambridge Univ. Press, 2004.
[3] F. S. Collins, M. Morgan, and A. Patrinos, "The human genome project: Lessons from large-scale biology," *Sci.*, vol. 300, no. 5617, pp. 286–290, 2003.
[4] M. Cuturi, "Fast global alignment kernels," in *Proc. Int. Conf. Mach. Learn.*, pp. 929–936, 2011.
[5] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.
[6] A. L. Dixon, et al., "A genome-wide association study of global gene expression," *Nature Genetics*, vol. 39, no. 10, pp. 1202–1207, 2007.
[7] L. El Ghaoui, V. Viallon, and T. Rabbani, "Safe feature elimination in sparse supervised learning," *Pacific J. Optimization*, vol. 8, no. 4, pp. 667–698, 2012.
[8] J. Fan, Y. Feng, and R. Song, "Nonparametric independence screening in sparse ultra-high-dimensional additive models," *J. Amer. Statistical Assoc.*, vol. 106, no. 494, pp. 544–557, 2011.
[9] J. Fan and J. Lv, "Sure independence screening for ultrahigh dimensional feature space," *J. Roy. Statistical Soc.: Series B (Statistical Methodology)*, vol. 70, no. 5, pp. 849–911, 2008.
[10] J. Fan and R. Song, "Sure independence screening in generalized linear models with NP-dimensionality," *Ann. Statist.*, vol. 38, no. 6, pp. 3567–3604, 2010.
[11] O. Fercoq, A. Gramfort, and J. Salmon, "Mind the duality gap: Safer rules for the lasso," in *Proc. Int. Conf. Mach. Learn.*, pp. 333–342, 2015.

[12] J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani, "Pathwise coordinate optimization," *Ann. Appl. Statist.*, vol. 1, no. 2, pp. 302–332, 2007.
[13] K. Lang, "Newsweeder: Learning to filter netnews," in *Proc. 12th Int. Conf. Mach. Learn.*, vol. 10, pp. 331–339, 1995.
[14] S. Lee, S. Kong, and E. P. Xing, "A network-driven approach for genome-wide association mapping," *Bioinf.*, vol. 32, no. 12, pp. i164–i173, 2016.
[15] M. Lichman, "UCI machine learning repository," University of California, Irvine, School of Information and Computer Sciences, 2013, http://archive.ics.uci.edu/ml
[16] J. Liu, Z. Zhao, J. Wang, and J. Ye, "Safe screening with variational inequalities and its application to lasso," in *Proc. Int. Conf. Mach. Learn.*, pp. 1556–1571, 2014.
[17] D. G. Luenberger, *Optimization by Vector Space Methods*. Hoboken, NJ, USA: Wiley, 1969.
[18] R. Mazumder and T. Hastie, "Exact covariance thresholding into connected components for large-scale graphical lasso," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 781–794, 2012.
[19] E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon, "Gap safe screening rules for sparse multi-task and multi-class models," in *Proc. Advances Neural Inf. Process. Syst.*, pp. 811–819, 2015.
[20] F. Pedregosa, et al., "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[21] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression (PIE) database," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, pp. 46–51, 2002.
[22] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statistical Soc.. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.
[23] R. Tibshirani, et al., "Strong rules for discarding predictors in lasso-type problems," *J. Roy. Statistical Soc.: Series B (Statistical Methodology)*, vol. 74, no. 2, pp. 245–266, 2012.
[24] J. Wang, J. Zhou, J. Liu, P. Wonka, and J. Ye, "A safe screening rule for sparse logistic regression," in *Proc. Advances Neural Inf. Process. Syst.*, pp. 1053–1061, 2014.
[25] J. Wang, J. Zhou, P. Wonka, and J. Ye, "Lasso screening rules via dual polytope projection," in *Proc. Advances Neural Inf. Process. Syst.*, pp. 1070–1078, 2013.
[26] Y. Wang, Z. J. Xiang, and P. J. Ramadge, "Lasso screening with a small regularization parameter," in *IEEE Int. Conf. Acoust., Speech-Signal Process.*, pp. 3342–3346, 2013.
[27] Y. Wang, Z. J. Xiang, and P. J. Ramadge, "Tradeoffs in improved screening of lasso problems," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 3297–3301, 2013.
[28] S. Webb, J. Caverlee, and C. Pu, "Introducing the webb spam corpus: Using email spam to identify web spam automatically," in *Proc. Conf. Email Anti-Spam*, 2006, http://dblp.org/db/conf/ceas/ceas2006.html
[29] Z. J. Xiang and P. J. Ramadge, "Fast lasso screening tests based on correlations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, pp. 2137–2140, 2012.
[30] Z. J. Xiang, Y. Wang, and P. J. Ramadge, "Screening tests for lasso problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 1008–1027, 2016.
[31] Z. J. Xiang, H. Xu, and P. J. Ramadge, "Learning sparse representations of high dimensional data on large scale dictionaries," in *Proc. Advances Neural Inf. Process. Syst.*, pp. 900–908, 2011.
[32] B. Zhang, et al., "Integrated systems approach identifies genetic nodes and networks in late-onset Alzheimer's disease," *Cell*, vol. 153, no. 3, pp. 707–720, 2013.
[33] P. Zhao and B. Yu, "On model selection consistency of lasso," *J. Mach. Learn. Res.*, vol. 7, pp. 2541–2563, 2006.

Seunghak Lee received the PhD degree in computer science from Carnegie Mellon University, in 2015. He is a data scientist with Human Longevity, Inc (HLI) in Mountain View, California. His research interests include machine learning and computational biology with a focus on integrative approaches to the analysis of genetic and biomedical datasets, genome-wide association studies, distributed optimization, and large-scale machine learning algorithms and systems. Prior to HLI, he was a project scientist in Machine Learning Department, Carnegie Mellon University.

Nico Görnitz is a research associate in the Machine Learning Group, TU Berlin (Berlin Institute of Technology, Germany) headed by Klaus-Robert Müller. He is interested in machine learning in general and in one-class learning based anomaly detection for data with dependency structure, large-margin structured output learning and corresponding optimization techniques in specific. Applications that Nico has been working on cover computational biology, computer security, computational sustainability, brain-computer-interfaces, natural language processing, and porosity estimation for geosciences.

Eric P. Xing received the PhD degree in molecular biology from Rutgers University, and another PhD degree in computer science from UC Berkeley. He is a professor of machine learning in the School of Computer Science, Carnegie Mellon University, and the director of the CMU Center for Machine Learning and Health. His principal research interests lie in the development of machine learning and statistical methodology, especially for solving problems involving automated learning, reasoning, and decision-making in high-dimensional, multimodal, and dynamic possible worlds in social and biological systems. His current work involves, 1) foundations of statistical learning, including theory and algorithms for estimating time/space varying-coefficient models, sparse structured input/output models, and nonparametric Bayesian models; 2) framework for parallel machine learning on big data with big model in distributed systems or in the cloud; 3) computational and statistical analysis of gene regulation, genetic variation, and disease associations; and 4) application of machine learning in social networks, natural language processing, and computer vision. He is an associate editor of the *Annals of Applied Statistics (AOAS)*, the *Journal of American Statistical Association (JASA)*, the *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, the *PLoS Journal of Computational Biology*, and an action editor of the *Machine Learning Journal (MLJ)*, the *Journal of Machine Learning Research (JMLR)*. He is a member of the DARPA Information Science and Technology (ISAT) Advisory Group, and a program chair of ICML 2014.

David Heckerman received the PhD degree in 1990 and the MD degree in 1992 from Stanford University. He is a distinguished scientist and director of the Microsoft Genomics at Microsoft. In his current scientific work, he is developing machine-learning and statistical approaches for biological and medical applications including genomics (see https://github.com/microsoftgenomics) and HIV vaccine design. In his early work, he demonstrated the importance of probability theory in artificial intelligence, and developed methods to learn graphical models from data, including methods for causal discovery. At Microsoft, he has developed numerous applications including the junk-mail filters in Outlook, Exchange, and Hotmail, machine-learning tools in SQL Server and Commerce Server, handwriting recognition in the Tablet PC, text mining software in Sharepoint Portal Server, troubleshooters in Windows, and the Answer Wizard in Office. He is a fellow of the ACM and AAAI.

Christoph Lippert received the PhD degree in bioinformatics from University of Tübingen, Germany, in 2014. He is a data scientist with Human Longevity, Inc (HLI) in Mountain View, California. It is a focus of his research to provide statistical and computational methodology for advanced genetic analyses of heritable traits and diseases. His research spans the fields of machine learning, statistical genetics, and bioinformatics. Before joining HLI, he has held research positions at Microsoft Research, the Max Planck Institute (MPI) for Developmental Biology, the MPI for Intelligent Systems, and Siemens Corporate Technology