The Hessian Screening Rule

NeurIPS 2022

Johan Larsson Jonas Wallin

Department of Statistics, Lund University

October 20, 2022

The Lasso

A type of penalized regression, represented by the following convex optimization problem:

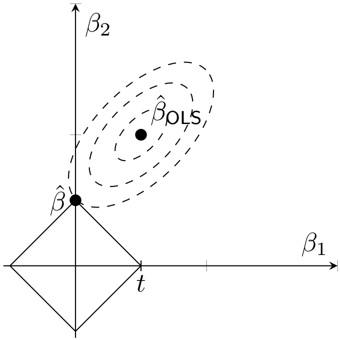
$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}} \{f(\beta) + \lambda \|\beta\|_1.\}$$

where $f(\beta)$ is smooth and convex.

$f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$ leads to the ordinary lasso.

λ is a hyperparameter that controls the level of **penalization**.

$\hat{\beta}(\lambda)$ is the solution to this problem for a given λ.



The Lasso Path

Solving the lasso for $\lambda \in [0, \lambda_{\text{max}})$, with

$$\lambda_{\text{max}} := \max\left\{\lambda \in \mathbb{R}^+ \mid \hat{\beta}(\lambda) = 0\right\},$$

traces the set of all solutions for the lasso.

The lasso path is **piece-wise linear** with breaks wherever the active set changes.

The active set:

$$\{i : |\beta_i| \neq 0\}.$$

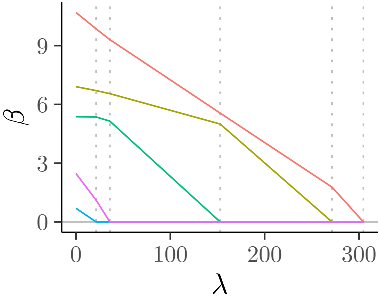


Figure 1: The lasso path for an example of the ordinary lasso

Picking λ

The Problem
Typically don't know the optimal value for λ. To tackle this, we use cross-validation to tune for λ.

Picking λ

The Problem

Typically don't know the optimal value for λ. To tackle this, we use cross-validation to tune for λ.

Grid Search

For $p \gg n$, the standard procedure is to create a grid of λs and solve the lasso numerically.

Picking λ

The Problem
Typically don't know the optimal value for λ. To tackle this, we use cross-validation to tune for λ.

Grid Search
For $p \gg n$, the standard procedure is to create a grid of λs and solve the lasso numerically.

But this is computationally demanding when p is large.

Screening Rules

Feature Screening

Motivation

Many solutions along the regularization path are **sparse**, especially if $p \gg n$ since the number of active features cannot exceed $\min(n, p)$.

Feature Screening

Motivation

Many solutions along the regularization path are **sparse**, especially if $p \gg n$ since the number of active features cannot exceed $\min(n, p)$.

Basic Idea

Say that we are at step k on the lasso path and are about to solve for step $k + 1$.

Intuitively, information at k should tell us something about which features are going to be active at step $k + 1$.

The idea is to use this information to **discard** a subset of the features and fit the model to a smaller set of features—the screened set.

The Gradient Perspective of the Path

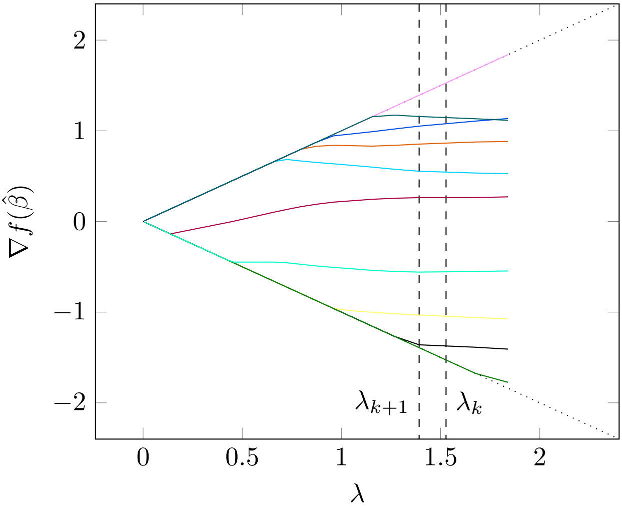


Figure 2: The gradient vector along the lasso path

Screening Rules Seen As Gradient Estimates

Let $c(\lambda) \coloneqq -\nabla f(\beta(\lambda))$ be the so-called **correlation** vector.

$\mathbf{0} \in \nabla f(\beta) + \lambda\partial$ suggests a simple template for a screening rule:

1. Replace c with an estimate $\tilde{c}$.

2. If $|\tilde{c}_j| < \lambda$, discard feature j.

If $\tilde{c}$ is accurate and not too conservative, we have a useful rule.

The Hessian Screening Rule

The Ordinary Lasso

We now focus on the ordinary lasso, ℓ_1-regularized least squares:

$$f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$$

and

$$\nabla f(\beta) = X^T(X\beta - y).$$

The Ordinary Lasso

We now focus on the ordinary lasso, ℓ_1-regularized least squares:

$$f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$$

and

$$\nabla f(\beta) = X^T(X\beta - y).$$

It turns out that we can express the solution as a function of λ:

$$\hat{\beta}(\lambda) = \left({X_{\mathcal{A}}}^T X_{\mathcal{A}}\right)^{-1}\left({X_{\mathcal{A}}}^T y - \lambda\,\mathrm{sign}(\hat{\beta}_{\mathcal{A}})\right).$$

The Ordinary Lasso

We now focus on the ordinary lasso, ℓ_1-regularized least squares:

$$f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$$

and

$$\nabla f(\beta) = X^T(X\beta - y).$$

It turns out that we can express the solution as a function of λ:

$$\hat{\beta}(\lambda) = \left(X_{\mathcal{A}}{}^T X_{\mathcal{A}}\right)^{-1}\left(X_{\mathcal{A}}{}^T y - \lambda \operatorname{sign}(\hat{\beta}_{\mathcal{A}})\right).$$

This expression holds for an interval $[\lambda_k, \lambda_{k+1}]$ in which no changes occur in the active set, which means we can retrieve any solution in this range via

$$\hat{\beta}(\lambda_{k+1})_{\mathcal{A}} = \hat{\beta}(\lambda_k)_{\mathcal{A}} - (\lambda_k - \lambda_{k+1})\left(X_{\mathcal{A}}{}^T X_{\mathcal{A}}\right)^{-1} \operatorname{sign}\left(\hat{\beta}(\lambda_k)_{\mathcal{A}}\right).$$

The Hessian Screening Rule

Take this expression and stick it into the gradient at step $k+1$:

$$\begin{aligned}\tilde{c}^H(\lambda_{k+1}) &= -\nabla f\big(\hat{\beta}(\lambda_{k+1})_{\mathcal{A}}\big) \\ &= c(\lambda_k) + (\lambda_{k+1}-\lambda_k) X^T X_{\mathcal{A}}\big({X_{\mathcal{A}}}^T X_{\mathcal{A}}\big)^{-1} \operatorname{sign}\big(\hat{\beta}(\lambda_k)_{\mathcal{A}}\big),\end{aligned}$$

which is the basic form of our screening rule: **The Hessian Screening Rule**.

Note that this is an exact expression for the correlation vector (negative gradient) at step $k+1$ if the activate set has remained unchanged.

The Hessian Screening Rule is a heuristic (un-safe) rule, so it needs safe-guarding in order to avoid discarding active features.

The Hessian and Strong Screening Rules

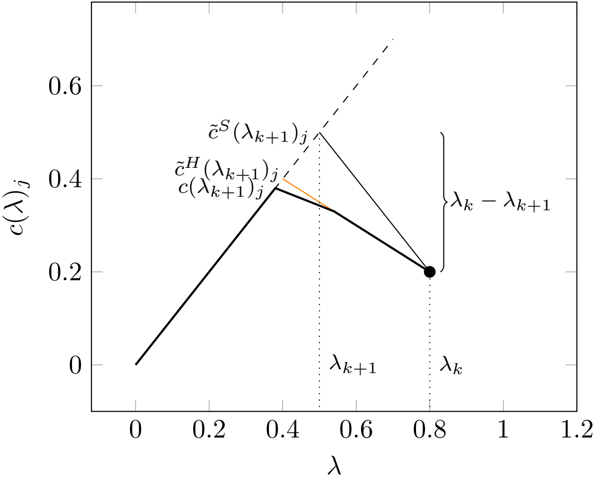


Figure 3: Conceptual comparison of screening rules

Results

Setup

- Rows of the feature matrix i.i.d. from $\mathcal{N}(0, \Sigma)$
- Response generated from $\mathcal{N}(X\beta, \sigma^2 I)$ with $\sigma^2 = \beta^T \Sigma \beta / \text{SNR}$
- s non-zero coefficients, equally spaced throughout the coefficient vector

Scenario 1 (Low-Dimensional)

$n = 10\,000$, $p = 100$, $s = 5$, and $\text{SNR} = 1$

Scenario 2 (High-Dimensional)

$n = 400$, $p = 40\,000$, $s = 20$, and $\text{SNR} = 2$

Code is located at
`github.com/jolars/HessianScreening`

Effectiveness

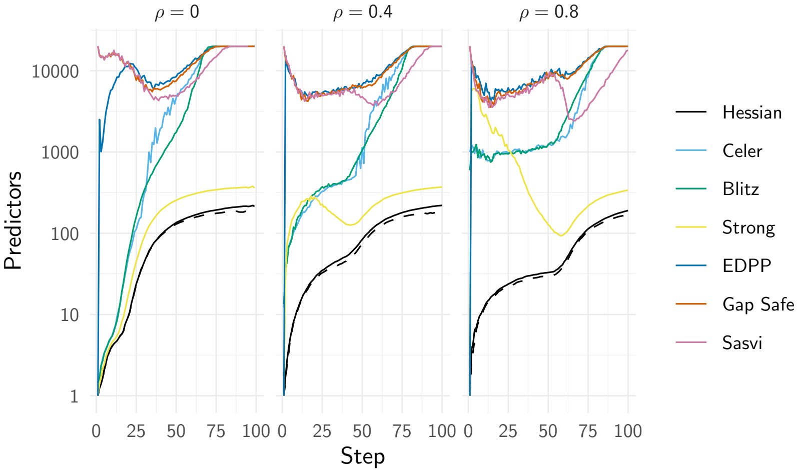


Figure 4: Number of features screened when fitting a lasso path for ℓ_1-regularized least-squares to a design with varying correlation (ρ), $n = 200$, and $p = 20000$. The actual number of active features at each step across iterations is given as a dashed line.

Simulated Data

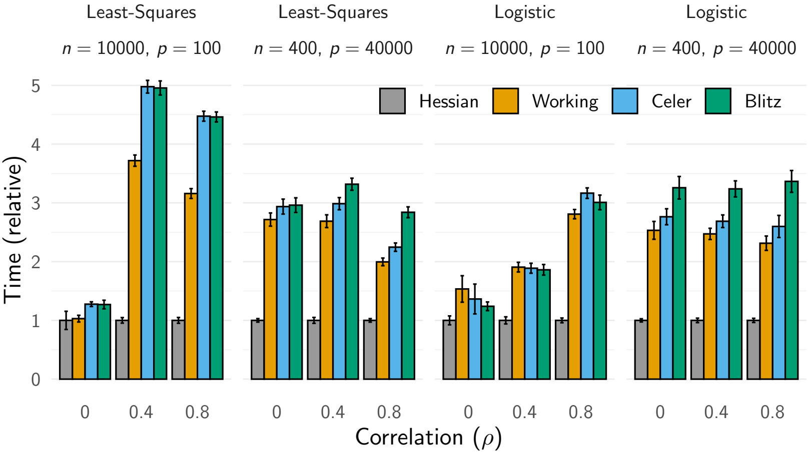


Figure 5: Time to fit a full regularization path for ℓ_1-regularized least-squares and logistic regression to a design with n observations, p features, and pairwise correlation between features of ρ. Time is relative to the minimal value for each group.

Discussion

- Simple, intuitive, idea
- Performs well in our examples
- Handles the highly-correlated case very well
- Works for arbitrary loss functions that are twice differentiable
- Works for other penalty functions too (SLOPE, MCP, SCAD, Elastic Net)