The Hessian Screening Rule and Adaptive Paths for the Lasso

Statistics Seminar at the Department of Mathematical Statistics, Chalmers/Gothenburg University

Johan Larsson

Department of Statistics, Lund University

September 21, 2021

Overview

Preliminaries

The Hessian Screening Rule

Adaptive Lasso Path (Work in Progress)

Preliminaries

The Lasso

The lasso (Tibshirani 1996) is a type of penalized regression, represented by the following convex optimization problem:

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

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 hyper-parameter that controls the level of **penalization**.

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

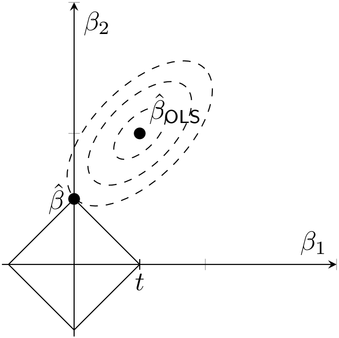

Figure 1: Level-curves and constraints for two-variable OLS

The Lasso Path

varying $\lambda \in [0, \infty)$ traces the set of all solutions for the lasso—the (exact) lasso path.

In practice, we start at the all-sparse solution, $\hat{\beta}(\lambda_{\mathsf{max}}) = 0$ and finish at some fraction of λ_{max} where the model is almost saturated.

The Lasso Path

varying $\lambda \in [0, \infty)$ traces the set of all solutions for the lasso—the (exact) lasso path.

In practice, we start at the all-sparse solution, $\hat{\beta}(\lambda_{\text{max}}) = 0$ and finish at some fraction of λ_{max} where the model is almost saturated.

This set of solutions is piece-wise linear with breaks occurring whenever the active set, the support set of the non-zero regression coefficients corresponding to $\hat{\beta}(\lambda)$, changes.

The size of the active set cannot exceed n, the number of observations.

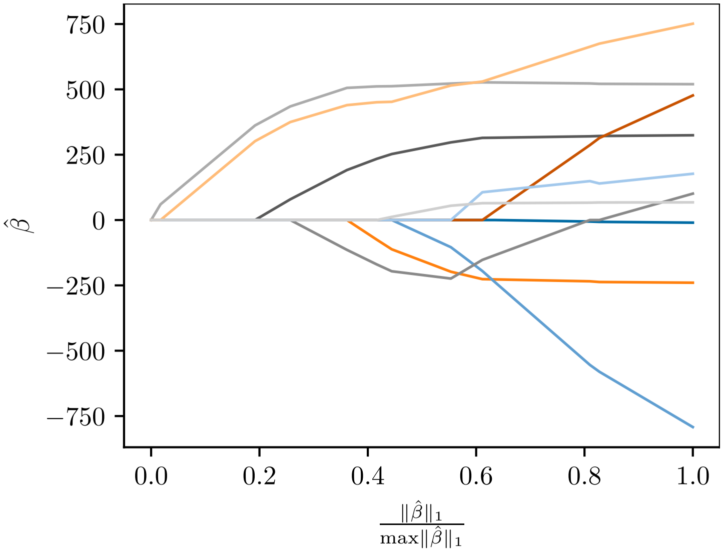

Figure 2: The lasso path for an example of the standard lasso

Picking λ

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

Picking λ

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

Grid Search
In our problem domain ($p \gg n$), the standard procedure is to create a grid of λs and fit the lasso iteratively.

Picking λ

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

Grid Search
In our problem domain ($p \gg n$), the standard procedure is to create a grid of λs and fit the lasso iteratively.

This can, however, get computationally demanding when the number of predictors, p is large. In our target domain, it may not even be possible to fit the data set into memory.

Screening Rules

Predictor Screening Rules

Motivation

Many of the solution vectors, $\hat{\beta}$, along the regularization path will be **sparse**, which means some predictors (columns) in X will be **inactive**, especially if $p \gg n$.

Predictor Screening Rules

Motivation
Many of the solution vectors, $\hat{\beta}$, along the regularization path will be **sparse**, which means some predictors (columns) in X will be **inactive**, especially if $p \gg n$.

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

Intuitively, the information at step k should give us some information about which predictors are going to be non-zero at step $k+1$.

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

Types of Screening Rules

Safe Rules
Provides a certificate that discarded predictors will be inactive at the optimum.

Types of Screening Rules

Safe Rules

Provides a certificate that discarded predictors will be inactive at the optimum.

Heuristic (Un-Safe) Rules

May result in **violations**: discarding predictors that actually will be active. Requires post-optimization checks of optimality conditions.

Checking the optimality conditions requires recomputing the full gradient, which is costly.

Optimality Conditions

β is a solution to the lasso problem if it satisfies the stationarity criterion

$$\mathbf{0} \in \nabla f(\beta) + \lambda \partial$$

where ∂ is the subdifferential of $\|\beta\|_1$, defined as

$$\partial_j \in \begin{cases} \{\text{sign}(\beta_j)\} & \text{if } \beta_j \neq 0, \\ [-1, 1] & \text{otherwise.} \end{cases}$$

Optimality Conditions

β is a solution to the lasso problem if it satisfies the stationarity criterion

$$\mathbf{0} \in \nabla f(\beta) + \lambda \partial$$

where ∂ is the subdifferential of $\|\beta\|_1$, defined as

$$\partial_j \in \begin{cases} \{\text{sign}(\beta_j)\} & \text{if } \beta_j \neq 0, \\ [-1, 1] & \text{otherwise.} \end{cases}$$

This means that

$$|\nabla f(\beta)_j| < \lambda \implies \hat{\beta}_j = 0.$$

Of course, we don't know $\nabla f(\beta)$ prior to solving the problem.

The Gradient Perspective of the Path

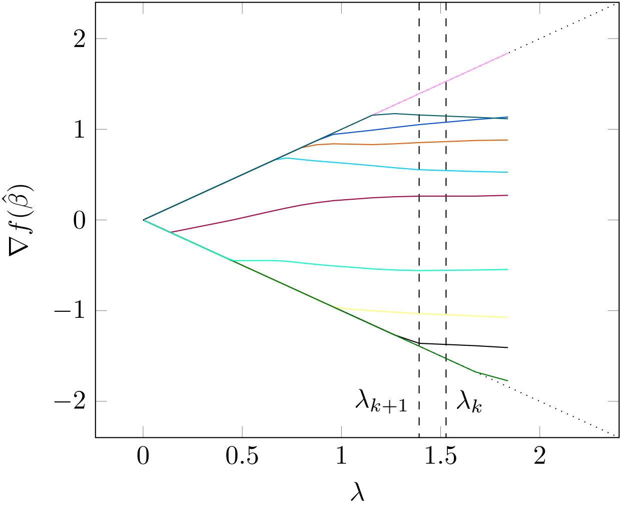

Figure 3: The gradient vector along the lasso path

Screening Rules as a Gradient Estimate

From now on, let $c(\lambda) := -\nabla f(\beta(\lambda))$ be the so-called **correlation** vector.

Now note that the stationarity criterion, $\mathbf{0} \in \nabla f(\beta) + \lambda\partial$, suggests a simple template for a screening rule: substitute the true gradient (correlation) vector in the optimality condition with some estimate. If this estimate is $\tilde{c}$, simply input this into the optimality condition, that is, check if

$$|\tilde{c}_j| < \lambda,$$

and discard any predictors that fail this test.

If the gradient estimate is accurate enough and not too conservative, we probably have a useful rule.

The Strong Rule

The Strong Rule gradient estimate is

$$\tilde{c}^S(\lambda_{k+1}) = \underbrace{c(\lambda_k)}_{\text{previous gradient}} + \underbrace{(\lambda_k - \lambda_{k+1})\,\mathrm{sign}(c(\lambda_k))}_{\text{unit slope bound}}.$$

simple idea: assume that the slope of the gradient is bounded by one (the unit slope bound)

discovered by (Tibshirani et al. 2012)

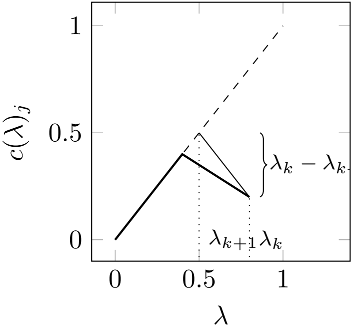

Figure 4: The unit slope bound

The Working Set Algorithm

The Strong Rule turns out to be very conservative when predictors are correlated.

It turns out that a better alternative is to use the predictors that have **ever been active** as a screened set and then

1. fit the lasso on the predictors in the screened set,
2. check the optimality conditions in the strong set, and then
3. check the optimality conditions for all predictors.

Whenever we encounter violations (in step 2 or 3), go back to step 1 and repeat.

This method is called the **working set strategy** and is used in many popular implementations, such as `glmnet` (Friedman, Hastie, and Tibshirani 2010)

The Hessian Screening Rule

The Ordinary Lasso

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

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

and

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

The Ordinary Lasso

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

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

and

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

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

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

The Ordinary Lasso

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

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

and

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

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

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

This function is piece-wise linear 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

$$\hat{\beta}(\lambda_{k+1})_{\mathcal{A}} = \hat{\beta}(\lambda_k)_{\mathcal{A}} - (\lambda_k - \lambda_{k+1})\big(X_{\mathcal{A}}{}^T X_{\mathcal{A}}\big)^{-1}\,\mathrm{sign}\,\big(\hat{\beta}(\lambda_k)_{\mathcal{A}}\big).$$

The Hessian Screening Rule

Now, we 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: it may result in violations.

The Hessian and Strong Screening Rules

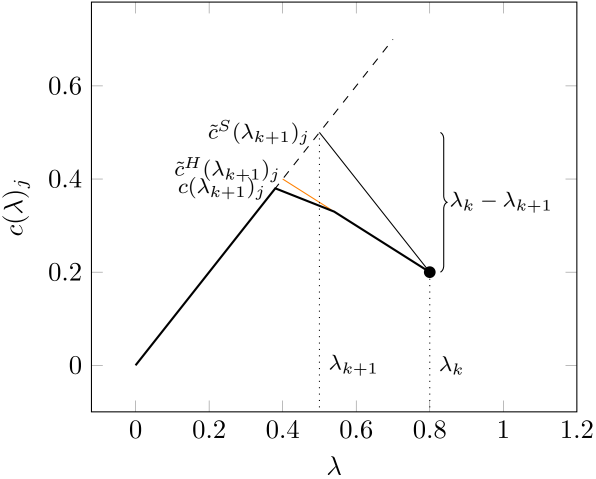

Figure 5: Conceptual comparison of screening rules

Tweaks

Avoiding Expensive Inner Products

The expression

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

involves an expensive inner product with the full design matrix. Instead, we replace X with the columns indexed by the strong rule.

Upwards Shift

We need some upwards bias on the estimate or else risk excessive numbers of violations. We add a fraction γ of the unit bound[1]

[1] We've set $\gamma = 0.01$ in our simulations, which has worked very well.

Warm Starts

The availability of the Hessian (and its inverse) enables a much improved warm start:

$$\hat{\beta}(\lambda_{k+1})_{\mathcal{A}} \coloneqq \hat{\beta}(\lambda_k)_{\mathcal{A}} + (\lambda_k - \lambda_{k+1})\big({X_{\mathcal{A}}}^T X_{\mathcal{A}}\big)^{-1} \operatorname{sign}\big(\hat{\beta}(\lambda_k)_{\mathcal{A}}\big)$$

Warm Starts

The availability of the Hessian (and its inverse) enables a much improved warm start:

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

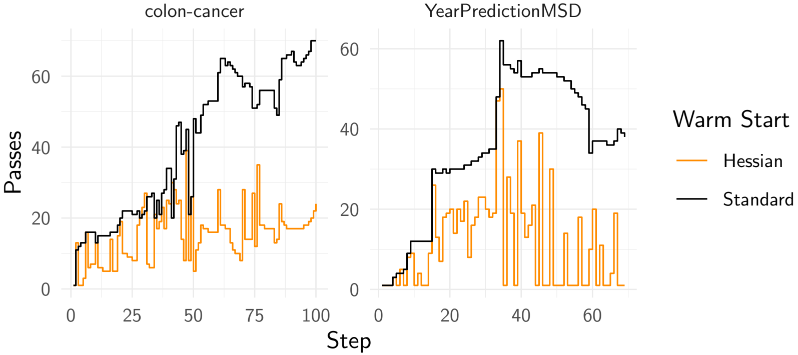

Figure 6: Number of passes of coordinate descent for two datasets using either Hessian warm starts or standard warm starts.

Updating the Hessian

A potential hurdle with the screening rule is of course computing the Hessian and its inverse, which is an $\mathcal{O}(|\mathcal{A}|^3 + |\mathcal{A}|^2 n)$ operation if executed naively.

Fortunately, we can compute these operations efficiently by sweeping columns of the Hessian (and inverse) in our out as we proceed along the path, with numerical complexity

- $\mathcal{O}(|\mathcal{D}|^2 n + n|\mathcal{D}||\mathcal{E}| + |\mathcal{D}^2||\mathcal{E}| + |\mathcal{D}|^3)$ when augmenting the Hessian and
- $\mathcal{O}(|\mathcal{C}|^3 + |\mathcal{C}|^2|\mathcal{E}| + |\mathcal{C}||\mathcal{E}|^2)$ when reducing it,

where $\mathcal{C} = \mathcal{A}_{k-1} \setminus \mathcal{A}_k$, $\mathcal{D} = \mathcal{A}_k \setminus \mathcal{A}_{k-1}$, and $\mathcal{E} = \mathcal{A}_k \cap \mathcal{A}_{k-1}$.

General Loss Functions

The rule can be extended to many other loss functions in the family of generalized linear models.

We skip the details here, but note the following:

- The gradient estimate is a natural extension of that in the ordinary lasso case, but now involves a matrix of weights—for logistic regression a diagonal matrix.
- The lasso path is no longer piece-wise linear, which means that our estimate is never exact.
- There is no longer any cheap way to update the Hessian. In our implementation, we decide heuristically whether to approximate it with an upper bound or compute it fully.

Results

Setup

The setup is as follows:

- We draw the rows of the predictor matrix i.i.d. from $\mathcal{N}(0, \Sigma)$
- The response is generated from $\mathcal{N}(X\beta, \sigma^2 I)$ with $\sigma^2 = \beta^T \Sigma \beta/\text{SNR}$ where SNR is the signal-to-noise ratio.
- We set s coefficients, equally spaced throughout the coefficient vector, to 1 and the rest to zero.

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`

Simulated Data

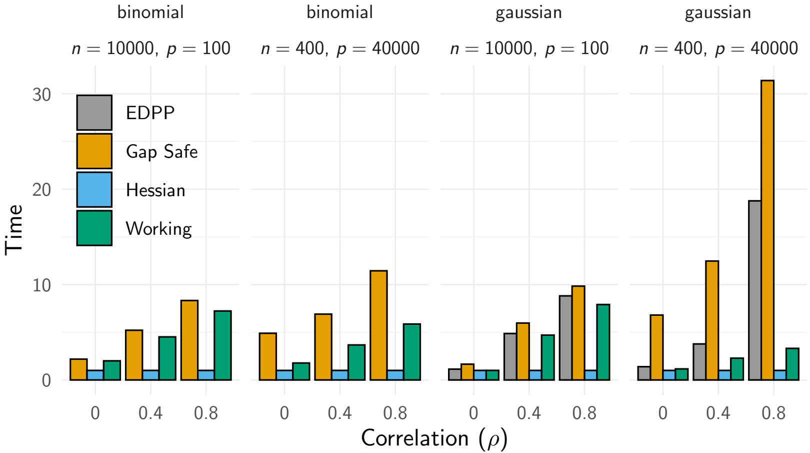

Figure 7: Time to fit a full regularization path for ℓ_1-regularized least-squares and logistic regression to a design with n observations, p predictors, and pairwise correlation between predictors of ρ. Time is relative to the minimal value for each group.

Real Data: Least-Squares Regression

Table 1: Time to fit a full regularization path of ℓ_1-regularized least-squares regression to real data sets.

				Time (s)			
Dataset	n	p	Density	Gap Safe	Hessian	Working	EDPP
cadata	20 640	8	1	1.65	0.196	1.42	1.49
e2006-log1p	16 087	4 272 227	0.0014	13 200	194	204	942
e2006-tfidf	16 087	150 360	0.0083	565	29.3	66.8	337
YearPredMSD	463 715	90	1	196	46.9	159	142

Real Data: Logistic Regression

Table 2: Time to fit a full regularization path of ℓ_1-regularized logistic regression to real data sets.

				Time (s)		
Dataset	n	p	Density	Gap Safe	Hessian	Working
arcene	100	10 000	0.54	11	8.0	7.1
colon-cancer	62	2000	1	0.45	0.19	0.36
duke-breast-cancer	44	7129	1	0.49	0.20	0.38
ijcnn1	35 000	22	1	29	9.9	29
madelon	2000	500	1	620	94	580
news20	19 996	1 355 191	0.000 34	33 000	1700	2300
rcv1	20 242	47 236	0.0016	940	530	380

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, lasso variations)
- a paper is submitted and under review, but a pre-print is available (Larsson and Wallin 2021).

Adaptive Lasso Path (Work in Progress)

Standard Grid Setup

The dominating choice for constructing the lasso path in the high-dimensional regime is to setup a log-spaced grid from λ_{min} to λ_{max} where

$$\lambda_{\text{min}} = \lambda_{\text{max}} \times \begin{cases} 10^{-2} & \text{if } p > n \\ 10^{-4} & \text{otherwise.} \end{cases}$$

Why this particular choice? We don't know.

Why Not Use LARS?

Homotopy methods scale poorly with p, in the worst case requiring $(3^p + 1)/2$ steps

The Problem with the Heuristic

If the λ values are not spaced densely enough around critical points, the interpolated values doesn't capture the path accurately.

On the other hand, if the values are spaced too densely where there are no critical points, then we are **wasting** resources.

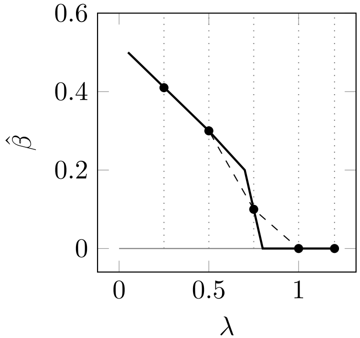

Figure 8: The exact path of a coefficient and the interpolated path based on the grid method as a dashed line.

Lasso Path Profiles

When and how predictors enter the models along the lasso path vary greatly.

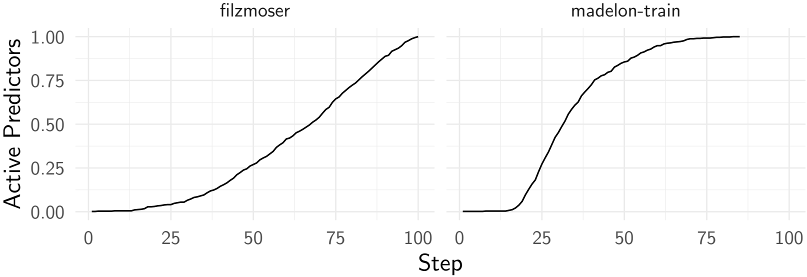

Figure 9: The number of included predictors at each step along the path for the standard grid path for the lasso. The number of included predictors varies considerably from data set to data set.

The Adaptive Lasso Path

The gradient estimate from the Hessian screening rule can be used to predict at which λ values the predictors will enter the model.

With the Adaptive Lasso Path, we use this information to find the next λ at which a desired number of predictors will enter the model

This lets us decide the resolution of the lasso path arbitrarily.

An Example

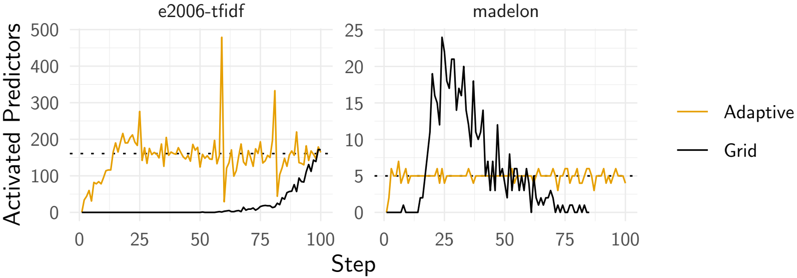

Figure 10: The number of predictors entering the model at each step for either the adaptive path or the grid path.

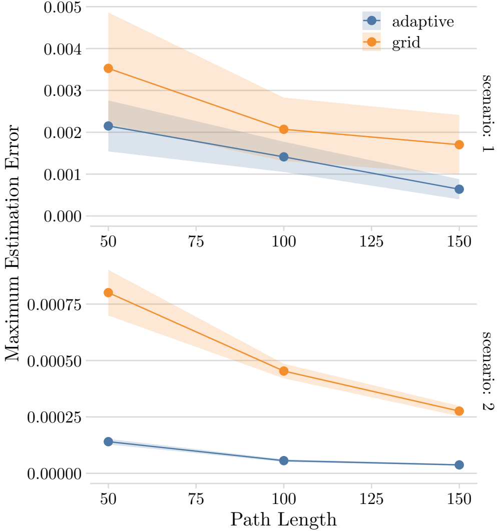

Figure 11: Maximum error along the regularization path for the adaptive and grid methods. Scenario 1 is a high-dimensional setting and 2 a low-dimensional setup.

Discussion

- The default grid heuristic is crude and sub-optimal in some cases.
- The adaptive lasso path adapts to the structure of the data and can be controlled by the user to tailor the resolution of the path.
- As n becomes smaller, the method converges towards a homotopy method.

Thank You!

Thank you for listening! Questions? Thoughts?

References I

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software* 33.1 (2010), pp. 1–22. DOI: 10.18637/jss.v033.i01.

Johan Larsson and Jonas Wallin. *The Hessian Screening Rule*. Apr. 27, 2021. arXiv: 2104.13026 [cs, stat]. URL: http://arxiv.org/abs/2104.13026 (visited on 04/29/2021).

Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288. ISSN: 0035-9246. JSTOR: 2346178.

Robert Tibshirani et al. "Strong Rules for Discarding Predictors in Lasso-Type Problems". In: *Journal of the Royal Statistical Society. Series B: Statistical Methodology* 74.2 (Mar. 2012), pp. 245–266. ISSN: 1369-7412. DOI: 10/c4bb85.