The Lasso: Past, Present and Future

Robert Tibshirani
Stanford University

ISI Founders of Statistics Prize Lecture, 2021

Support from the NSF and NIH gratefully acknowledged

Email:tibs@stanford.edu
https://statweb.stanford.edu/∼tibs

Some big thanks

I want to thank my many wonderful collaborators and students, past and present, including:

Trevor Hastie, Brad Efron, Jerome Friedman, Balasubramanian Narasimhan, Jacob Bien, Daniela Witten, Noah Simon and Ryan Tibshirani

and a large community of researchers worldwide who have contributed so much to this topic.

Outline

- Review of lasso and its history
- Computational approaches
- Some examples: Matrix completion, large scale lasso for GWAS, deep learning and LassoNet
- The future
- *Not covering:* theoretical results - eg asymptotic recovery of true model- interesting! Buhlmann, Candes, Donoho, Meinshausen, Wainwright, Yu,... But not enough time.

Regression shrinkage and selection via the Lasso

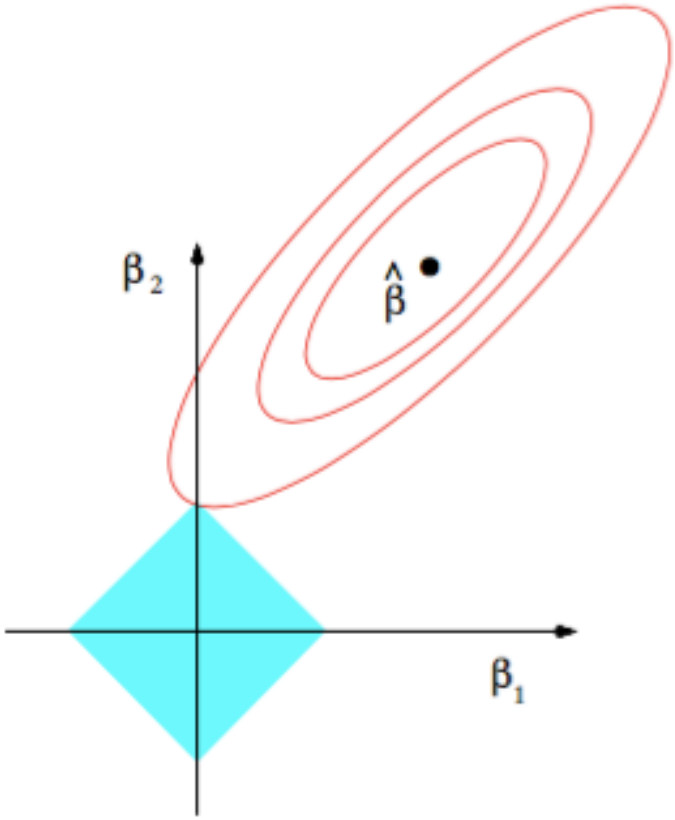
Tibshirani, JRSSB 1996

- Outcome variable y_i, for cases $i = 1, 2, \ldots n$, features x_{ij}, $j = 1, 2, \ldots p$
- Minimize

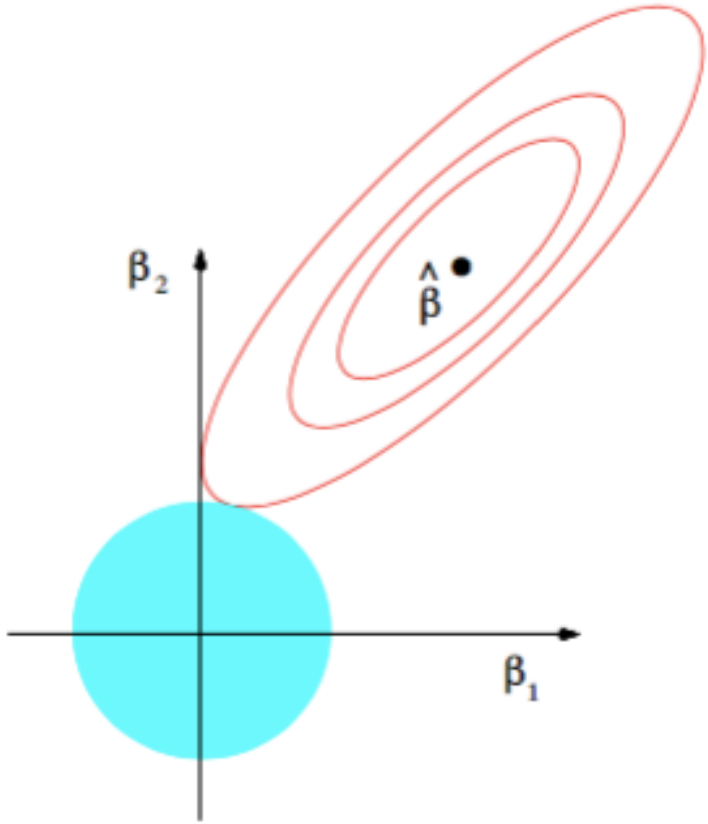
$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_j x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- Equivalent to minimizing sum of squares with constraint $\sum |\beta_j| \leq s$.
- Similar to *ridge regression*, which has constraint $\sum_j \beta_j^2 \leq t$
- Lasso does variable selection and shrinkage; ridge regression, in contrast only shrinks.

Picture of Lasso and Ridge regression

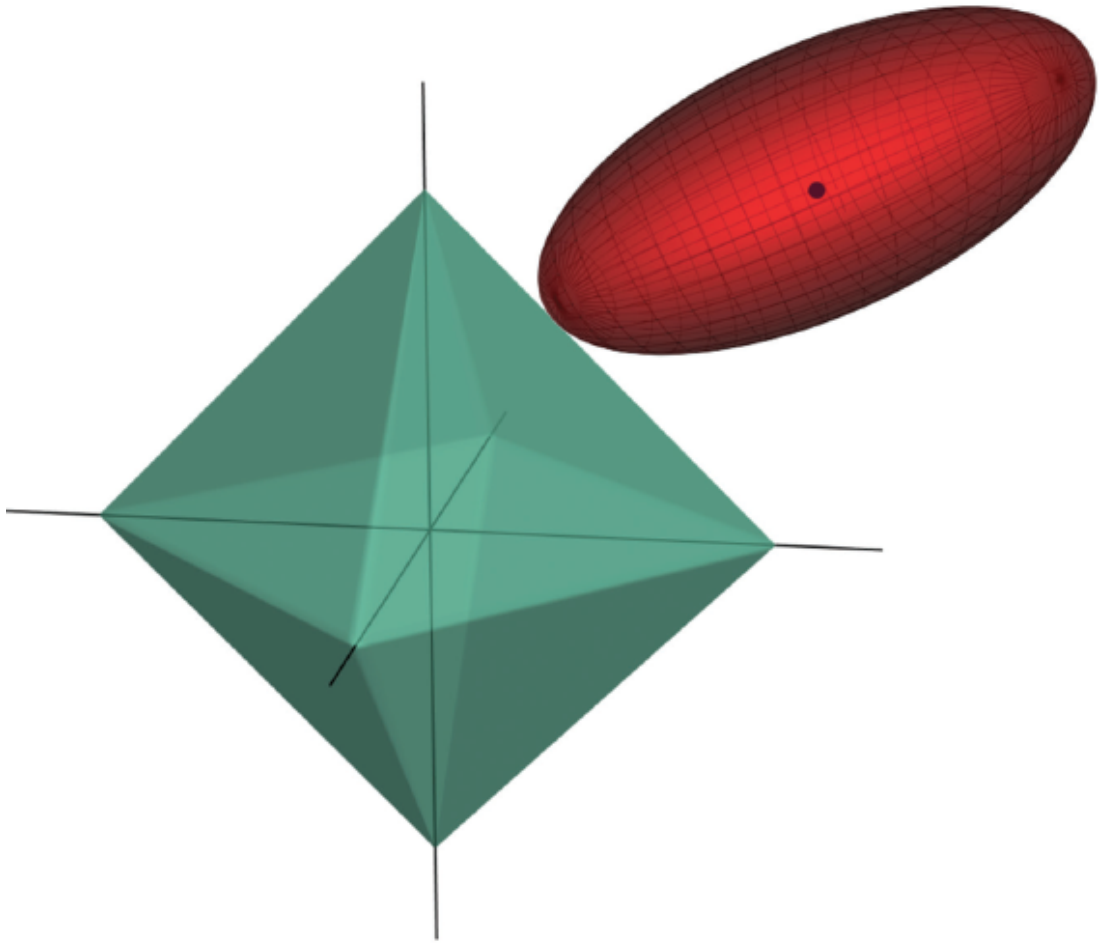


Lasso



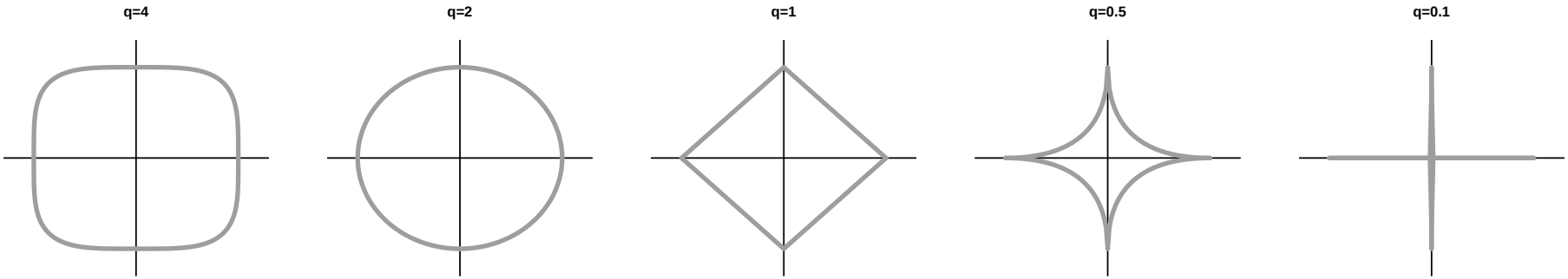
Ridge Regression

In 3D



From book "Statistical Learning with Sparsity" by Hastie, Tibshirani &Wainwright;

More ℓ_q norms

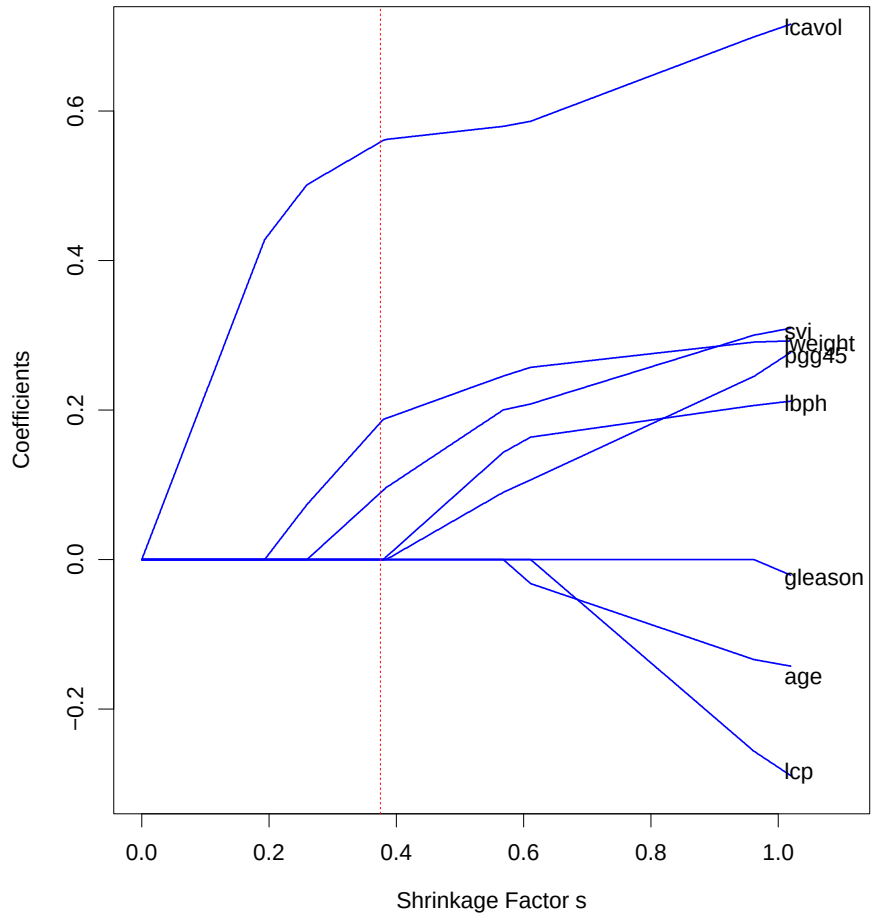


ridge *lasso* *subset selection*

Lasso uses $q = 1$, the value closest to subset selection ($q = 0$) that yields a convex problem.

Example: Prostate Cancer Data

$y_i = \log$ (PSA), x_{ij} measurements on a man and his prostate



History of the idea

- Lasso is regression with an ℓ_1 norm penalty. ℓ_1 norms have been around for long time!
- **Lots of concurrent work:** Frank and Friedman, "Bridge regression" 1993; using a penalty $\lambda \sum |\beta_j|^\gamma$, with both λ and γ estimated from the data.
- Chen, Donoho, Saunders "Atomic Decomposition by Basis Pursuit" IEEE 2001 (tech report 1994)
- For me, the most direct influence: Leo Breiman's **garotte**. Idea is to minimize

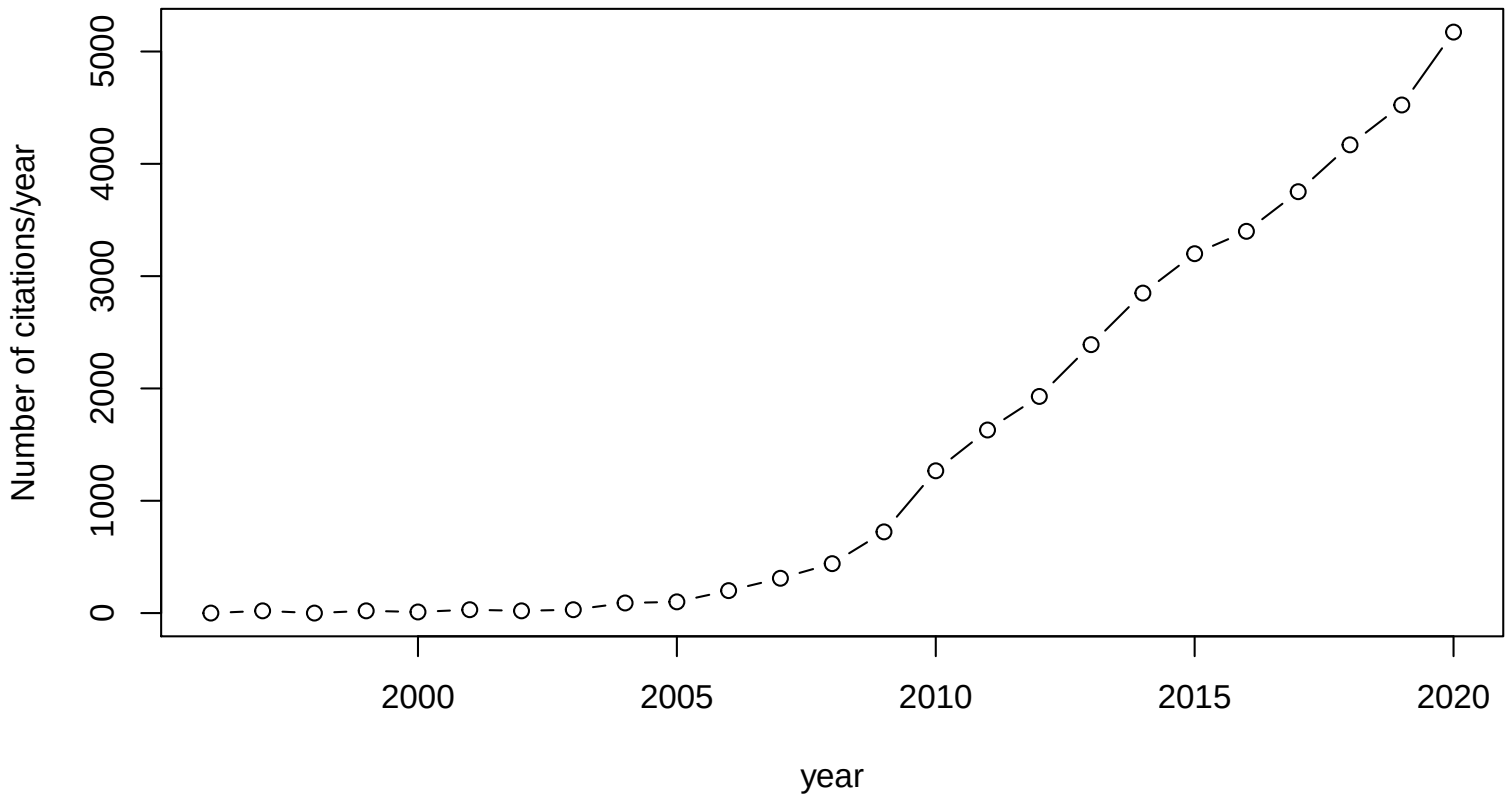
$$\sum_{i=1}^{n}(y_i - \sum_j c_j x_{ij}\hat{\beta}_j)^2 \text{ subject to } c_j \geq 0, \sum_{j=1}^{p} c_j \leq t$$

  where $\hat{\beta}_j$ are usual least squares estimates.
- This is undefined when $p > n$ (not a hot topic in 1995!) so I just combined the two stages into one (as a Canadian I also wanted a gentler name).

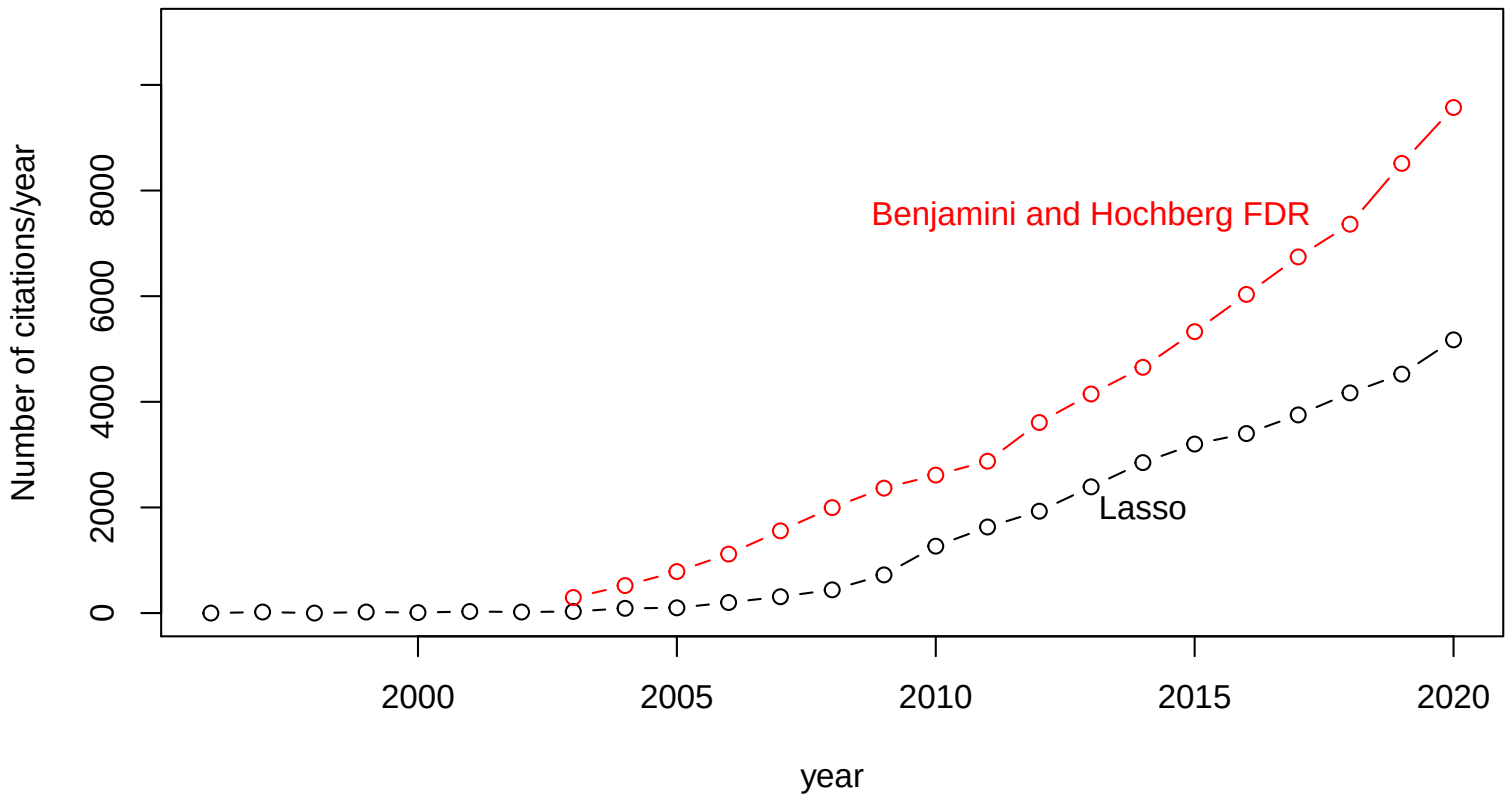
More on the paper

- submitted to JRSSB. Fairly positive reviews; one round of revisions.
- idea did not get much attention until years later
- why?
  - computation in 1996 was slow compared to today;
  - algorithms for lasso were black boxes, and not statistically motivated; (until *LARS*)
  - full *statistical* and *numerical* advantages of sparsity not appreciated
  - large data problems (in N, p or both) were rare;
  - community did not have the R language for fast, easy sharing of new software tools

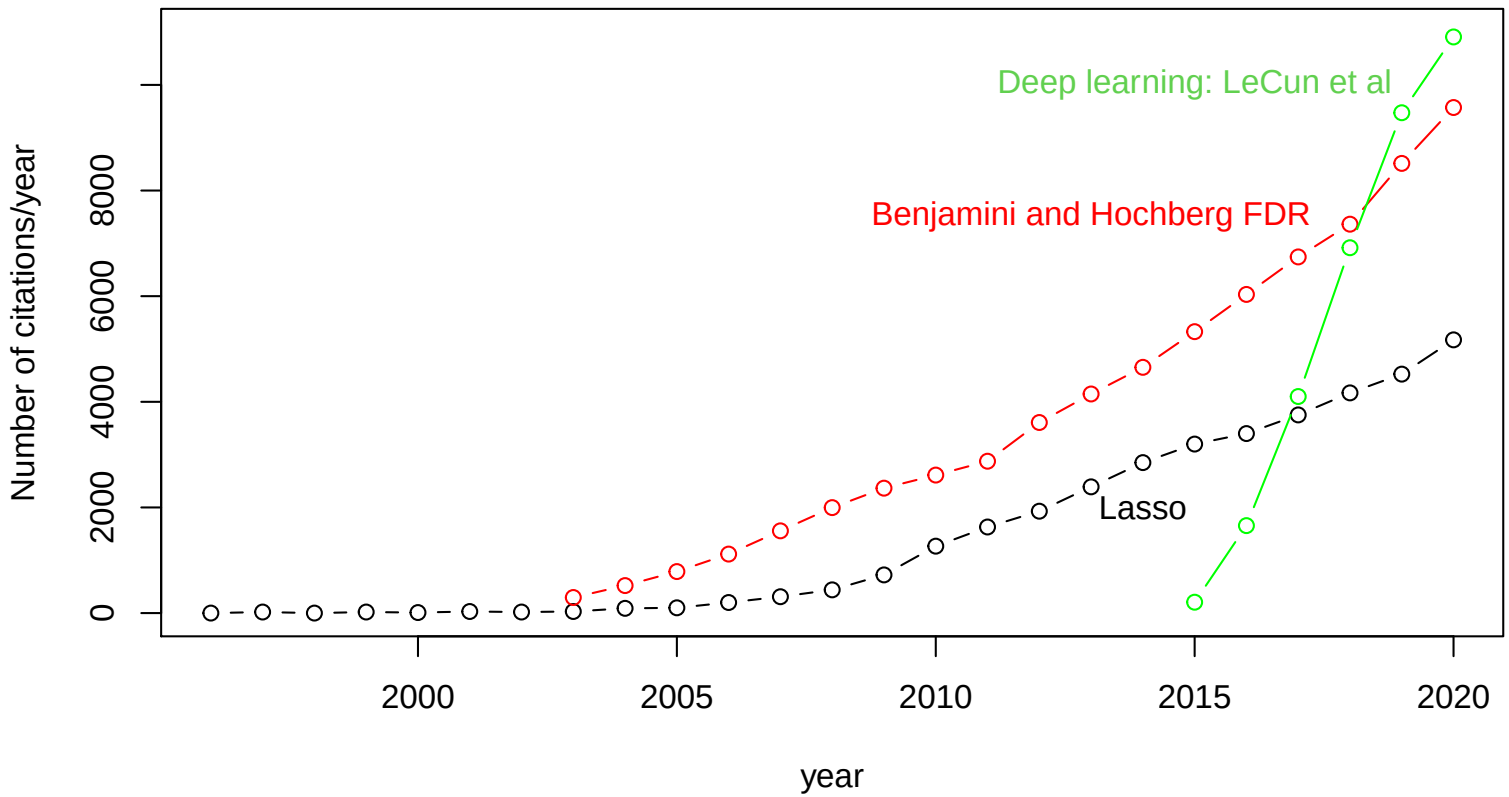
Yearly citation counts from google scholar



... including a control group



...and another


Deep learning: LeCun et al
Benjamini and Hochberg FDR
Lasso
Number of citations/year
0
2000
4000
6000
8000
2000
2005
2010
2015
2020
year

What we have learned since 1996

Advantages of sparsity and lasso are many-fold:

1. Simplicity of resulting model– ease of interpretation (we knew that)
2. Computational speed
3. Inherent shrinkage controls variance and reduces overfitting (it's not just a surrogate for the ℓ_0 penalty)
4. Convexity: enables the assessment of adaptivity
5. Theoretical performance— prediction error and support recovery.

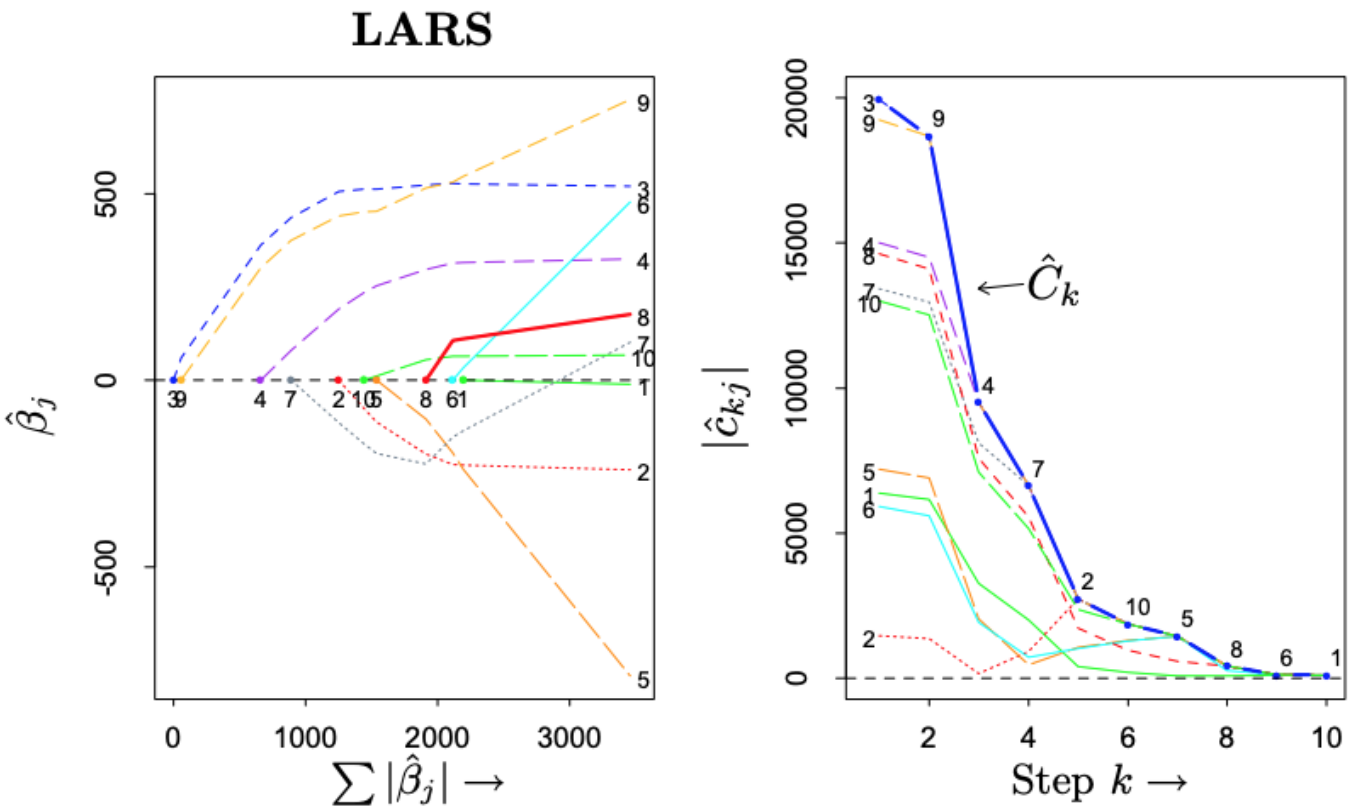
Computation: LAR and coordinate descent

- Original lasso paper used an off-the-shelf quadratic program solver. Doesn't scale well. Not transparent
- *LARS algorithm* (Efron, Hastie, Johnstone, Tibshirani 2002) gives an efficient way to solve the lasso, and connects the lasso to forward stagewise regression. Same algorithm is contained in the homotopy approach of Osborne, Presnell and Turlach (2000).
- *Coordinate descent* algorithms are extremely simple and fast, and exploit the assumed sparsity
- There's an explosion of activity in the optimization community in *first order methods* e.g. proximal gradient descent.

Least Angle Regression — LAR (Efron et al 2002)

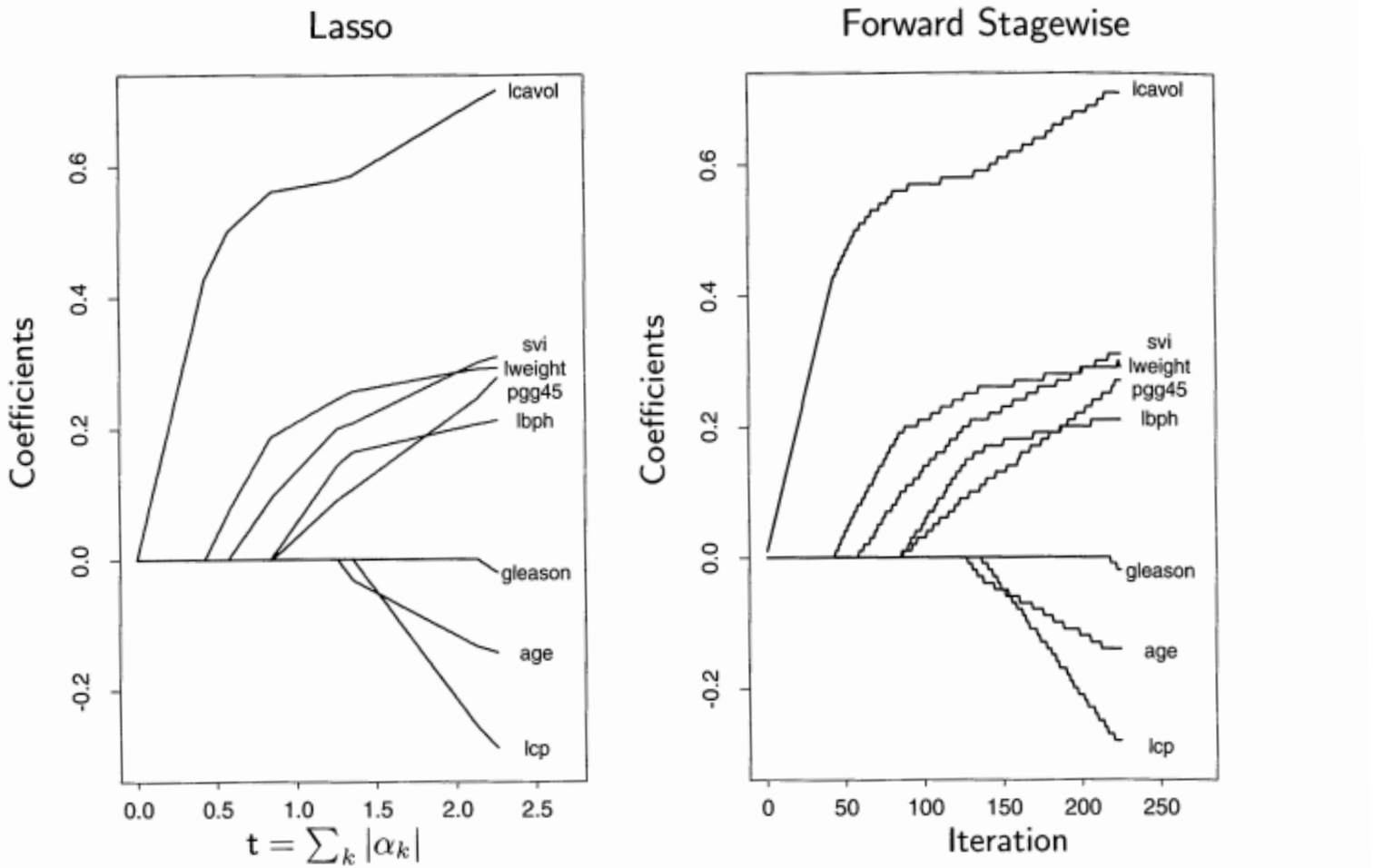
Like a "more democratic" version of forward stepwise regression.

1. Start with $r = y$, $\hat{\beta}_1, \hat{\beta}_2, \ldots \hat{\beta}_p = 0$. Assume x_j standardized.
2. Find predictor x_j most correlated with r.
3. Instead of simply entering the predictor, go slower: Increase β_j in the direction of sign(corr(r, x_j)) until some other competitor x_k has as much correlation with current residual as does x_j.
4. Move $(\hat{\beta}_j, \hat{\beta}_k)$ in the joint least squares direction for (x_j, x_k) until some other competitor x_ℓ has as much correlation with the current residual
5. Continue in this way until all predictors have been entered. Stop when corr$(r, x_j) = 0 \; \forall \; j$, i.e. OLS solution.


LARS
$\hat{\beta}_j$
500
0
-500
0
1000
2000
3000
$\sum |\hat{\beta}_j| \rightarrow$
$|\hat{c}_{kj}|$
20000
15000
10000
5000
0
$\leftarrow \hat{C}_k$
2
4
6
8
10
Step $k \rightarrow$

How LARS came to be

1. In trying to understand "boosting" for adaptive nonlinear function fitting, we looked it in the linear model setting where it is a form of forward stagewise fitting
2. In the process we discovered a possible relationship between forward stagewise and the lasso. We wrote about this in the first edition of *The Elements of Statistical Learning* (Figure next slide)
3. Brad Efron read it, and figured it all out; in the process he invented LARS
4. *Moral of the story*: be curious and have brilliant colleagues


Lasso
Coefficients
0.6
0.4
0.2
0.0
-0.2
lcavol
svi
lweight
pgg45
lbph
gleason
age
lcp
0.0
0.5
1.0
1.5
2.0
2.5
$t = \sum_k |\alpha_k|$
Forward Stagewise
Coefficients
0.6
0.4
0.2
0.0
-0.2
lcavol
svi
lweight
pgg45
lbph
gleason
age
lcp
0
50
100
150
200
250
Iteration

Pathwise coordinate descent for the lasso

- Coordinate descent: optimize one parameter (coordinate) at a time.
- How? suppose we had only one predictor. Problem is to minimize

$$\sum_i (y_i - x_i\beta)^2 + \lambda|\beta|$$

- Solution is the soft-thresholded estimate

$$\text{sign}(\hat{\beta})(|\beta| - \lambda)_+$$

  where $\hat{\beta}$ is usual least squares estimate.
- Idea: with multiple predictors, cycle through each predictor in turn. Compute residuals $r_i = y_i - \sum_{j \neq k} x_{ij}\hat{\beta}_k$ and apply soft-thresholding, pretending that our data is (x_{ij}, r_i).

Pathwise coordinate descent for the lasso–continued

- Turns out that this is coordinate descent for the lasso criterion

$$\sum_i (y_i - \sum_j x_{ij}\beta_j)^2 + \lambda \sum |\beta_j|$$

- like skiing to the bottom of a hill, going north-south, east-west, north-south, etc.
- algorithm starts with large value for λ (very sparse model) and slowly decreases it, using warms starts along λ path
- most coordinates that are zero never become non-zero

When does coordinate descent work?

Paul Tseng (1988), (2001)

If

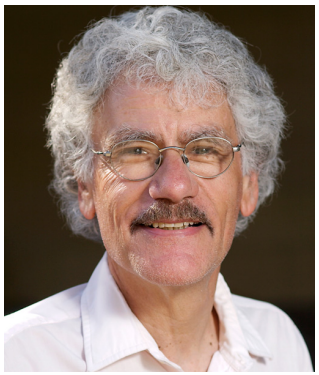
$$f(\beta_1 \dots \beta_p) = g(\beta_1 \dots \beta_p) + \sum h_j(\beta_j)$$

where $g(\cdot)$ is convex and differentiable, and $h_j(\cdot)$ is convex, then coordinate descent converges to a minimizer of f.

Non-differential part of loss function must be separable

The glmnet package in R

- Our lab has written a open-source R language package called `glmnet` for fitting lasso models. Numerics in FORTRAN(!)
- Many clever computational tricks were used to achieve it's impressive speed.
- 3.5 million downloads as of July 2021
- good software also available in Python (e.g. scikit.learn)



Jerry Friedman

Trevor Hastie

Balasubramanian Narasimhan

Features of the current version (glmnet 4.1)

- Gaussian, binomial, multinomial, poisson and user-defined "family" objects
- grouped lasso for multi-response Gaussian family
- support for sparse matrices
- feature filtering within cross-validation

Convexity enables the assessment of adaptivity

- The LAR work shows how the lasso is a more **theory-friendly** version of stepwise and best subset regression
- Remarkable **degrees for freedom** result for LAR and lasso: Define

$$\mathrm{df}(\hat{y}) \equiv \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{cov}(y_i, \hat{y}_i)$$

  For best subset of size k, $\mathrm{df}(\hat{y}) \geq k$, but no analytic expression exists

  For LAR/lasso with k non-zero coefficients, $\mathrm{df}(\hat{y}) = k$!!
- **Post-selection inference** — Lockhart et al; (covariance test); Ryan Tibshirani et, al, Fithian et al, Lee, Sun, Sun, Taylor; Markovic & Taylor– exact formulas for p-values and confidence intervals for LAR and lasso— fixed (or even CV-estimated) λ

Generalizations and related ideas

Method	*Authors*
Grouped lasso	Yuan and Lin
Elastic net	Zou and Hastie
Fused lasso	Tibs et. al
Adaptive lasso	Zou
Graphical lasso	Yuan & Lin, Fried., Hastie, Tibs
Dantzig selector	Candes and Tao
Near monotonic reg.	Tibs, Hoef. and Tibs
Matrix completion	Candes & Tao; Maz., Hastie, Tibs
Multivariate methods	Jolliffe, Witten and many others
Compressed sensing	Candes & Wakin; Donoho
Square-Root Lasso	Belloni et al

+ many more

The matrix completion problem

Example: *Movie recommendation systems*

	Lord of the rings	Pretty Woman	Harry Potter	Pulp Fiction	Kill Bill	Blue velvet
Daniela	5	5	4	1	1	1
Genevera	4	5	4	2	?	1
Larry	1	?	2	5	4	5
Jim	?	?	2	4	3	5
Andy	1	1	3	?	?	5

The matrix completion problem

- Data $X_{m\times n}$, for which only a relatively small number of entries are observed. The problem is to "complete" or impute the matrix based on the observed entries.
- For a matrix $X_{m\times n}$ let $\Omega \subset \{1,\ldots,m\}\times\{1,\ldots,n\}$ denote the indices of observed entries. Consider the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \operatorname{rank}(Z) \\ \text{subject to} \quad & Z_{ij} = X_{ij},\ \forall (i,j) \in \Omega \end{aligned} \tag{1}$$

Not convex!

- Make the problem convex by replacing "rank" with *nuclear norm* (sum of the singular values):

$$\begin{aligned} \text{minimize} \quad & \|Z\|_* \\ \text{subject to} \quad & Z_{ij} = X_{ij},\ \forall (i,j) \in \Omega \end{aligned} \tag{2}$$

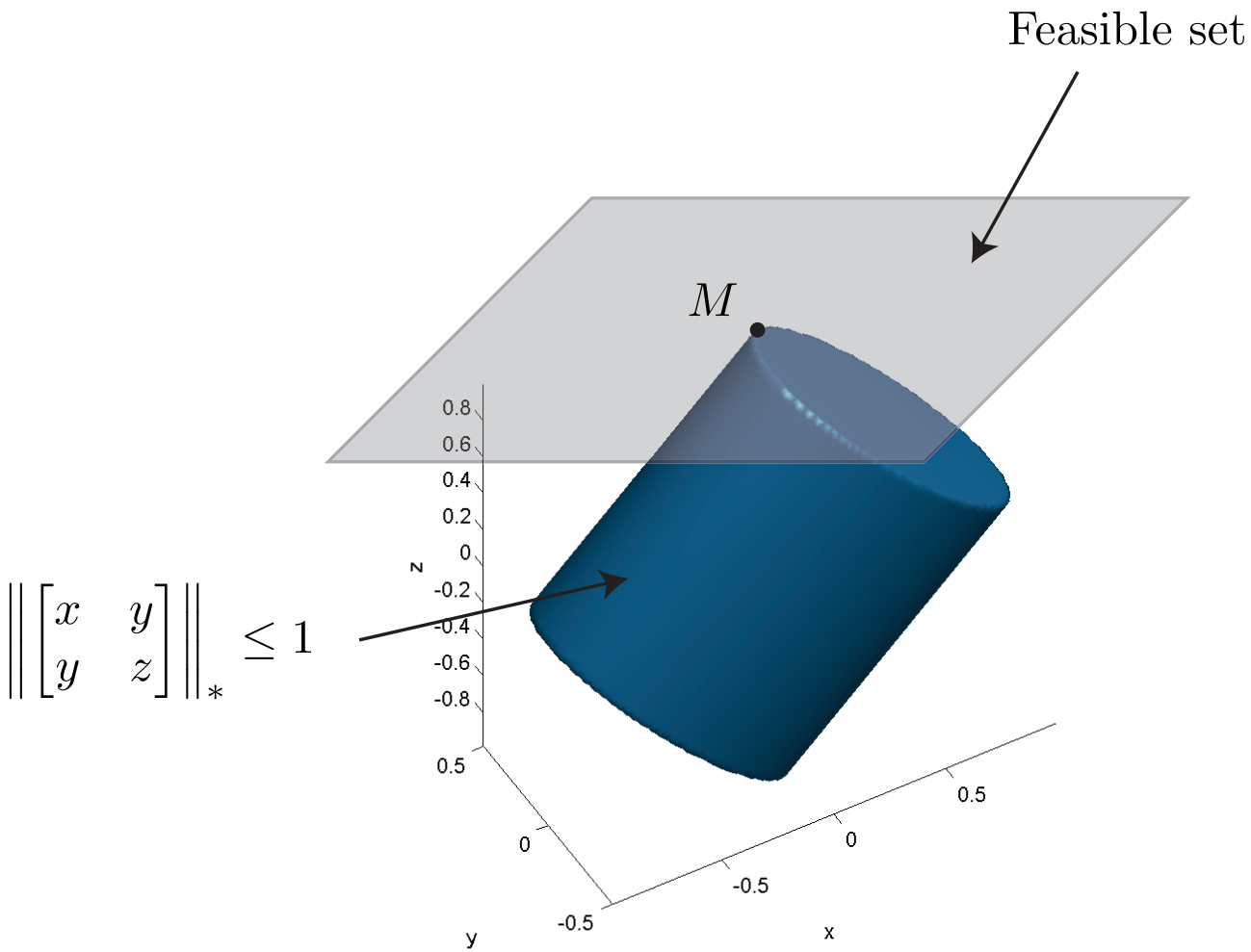
(Fazel 2002).

- This criterion is used by Candes et al 2009. Fascinating work on conditions for *exact* reconstruction.
- But this criterion requires the training error to be zero. This is too harsh and can overfit.
- Instead we use penalized reconstruction error:

$$\text{minimize} \sum_{(i,j)\in\Omega} (Z_{ij} - X_{ij})^2 + \lambda \cdot \|Z\|_* \tag{3}$$

Nuclear norm is like L_1 norm for matrices

Geometry



Thanks to Emmanuel Candes

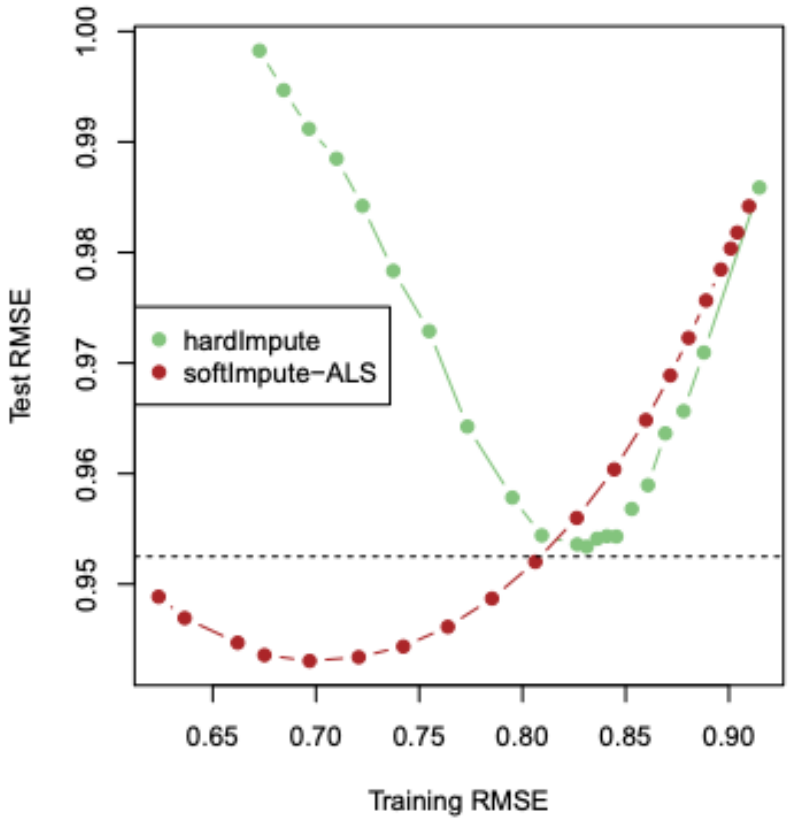
Idea of Algorithm

(Mazunder, Hastie , Tibshirani 2010)

1. impute the missing data with some initial values
2. compute the singular value decomposition (SVD) of the current imputed matrix, and soft-threshold the singular values:
3. reconstruct the SVD and hence obtain new imputations for missing values
4. repeat steps 2,3 until convergence

$$\hat{Z} \quad = \quad UD_{d_i}V^T \rightarrow UD_{S(d_i,\lambda)}V^T \tag{4}$$

Results on Netflix data



Lasso penalty not only selects features, but its shrinkage controls variance. *It is not just a surrogate for the ℓ_0 penalty.*

This behavior is also observed in comparisons with subset selection in regression ("Best Subset, Forward Stepwise or Lasso?" Hastie, Tibs + Tibs, Statistical Science 2020)

Lasso and elastic net for GWAS

Estimation of Polygenic risk scores

- with current software (eg **glmnet**), lasso and elastic net cannot be applied to data of size say 500K (patients) by 800K (SNPs)
- we have developed a new approach using the idea of strong screening rules (Tibs et al JRSSB 2012), that successfully carries out this computation in hours (exact, within machine precision)
- Joint work with PhD students **Junyang Qian** , Yosuke Tanigawa, Trevor Hastie and the Manny Rivas group at Stanford DBDS
- "A Fast and Flexible Algorithm for Solving the Lasso in Large-scale and Ultrahigh-dimensional Problems" , Qian et al bioRxiv 2019

Strong Rules

For lasso fit, with *active set* $\mathcal{A}$:

$$\left|\langle \mathsf{x}_j, \mathsf{y} - \mathsf{X}\hat{\beta}(\lambda)\rangle\right| \quad \begin{array}{ll} = \lambda & \forall j \in \mathcal{A} \\ \leq \lambda & \forall j \notin \mathcal{A} \end{array}$$

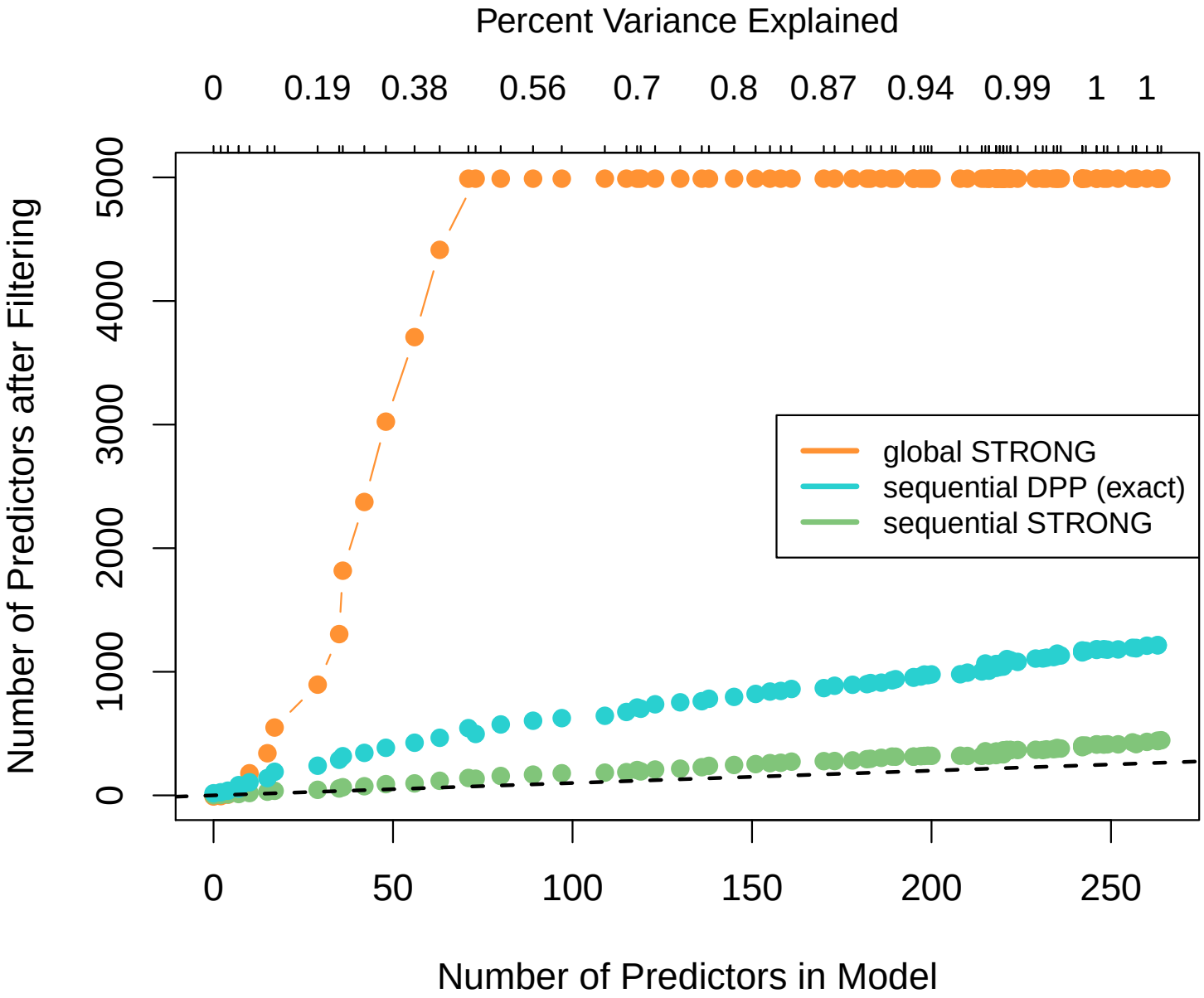
So variables *nearly in* $\mathcal{A}$ will have inner-products with the residuals *nearly* equal to λ.

Suppose fit at λ_ℓ is $\mathsf{X}\hat{\beta}(\lambda_\ell)$, and we want to compute the fit at $\lambda_{\ell+1} < \lambda_\ell$. Strong rules gamble on set

$$\mathcal{S}_{\ell+1} = \left\{j : |\langle \mathsf{x}_j, \mathsf{y} - \mathsf{X}\hat{\beta}(\lambda_\ell)\rangle| > \lambda_{\ell+1} - (\lambda_\ell - \lambda_{\ell+1})\right\}$$

GLMNET screens at every λ step, and after convergence, checks if any violations. Mostly $\mathcal{A}_{\ell+1} \subseteq \mathcal{S}_{\ell+1}$.

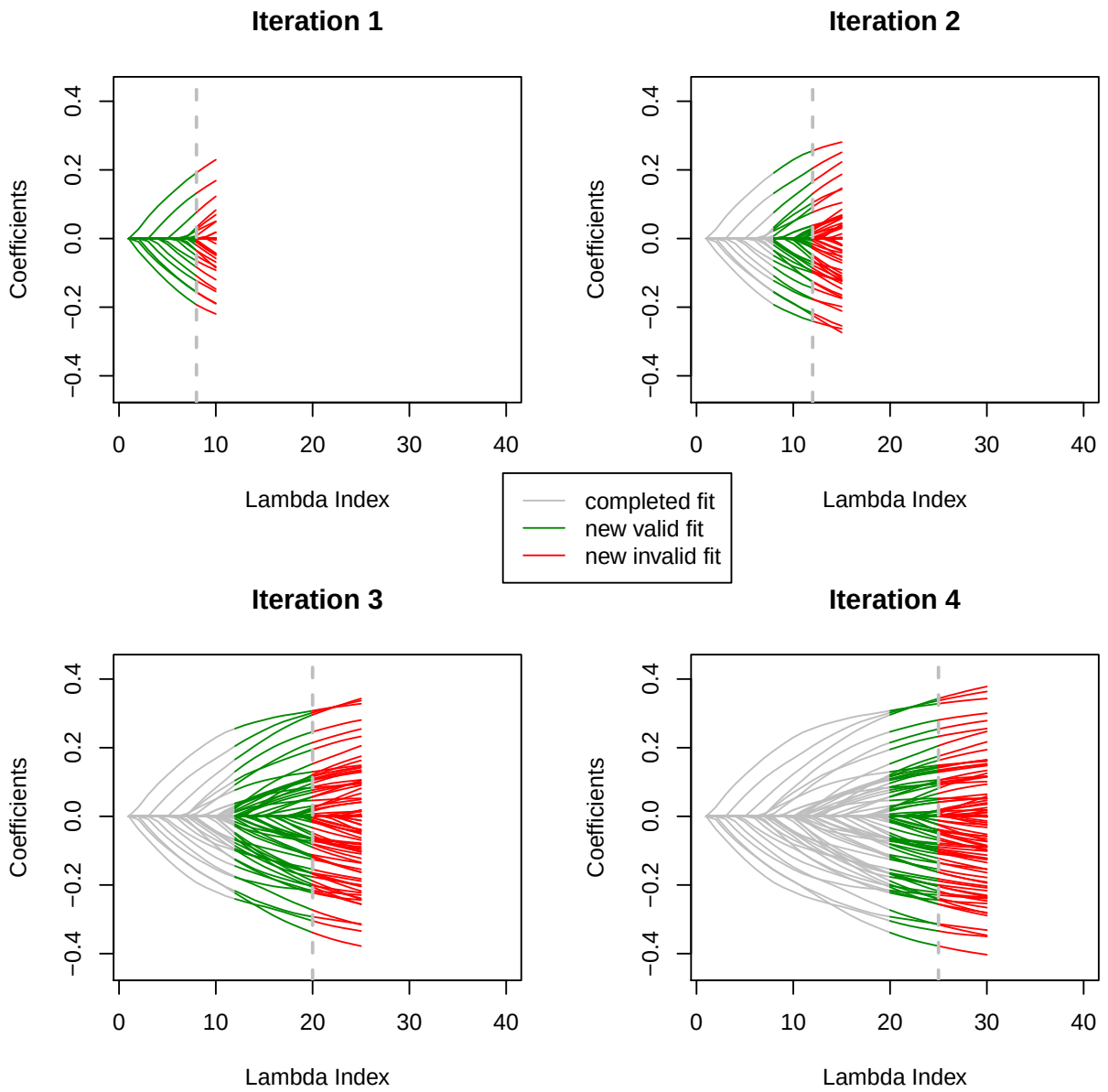
* Tibshirani, Bien, Friedman, Hastie, Simon, Taylor, Tibshirani (JRSSB 2012)



Strong rules inspired by El Ghaoui, Viallon and Rabbani (2010)

Sequential DPP due to Wang, Lin, Gong, Wonka and Ye (2013)

"SNPnet" in pictures

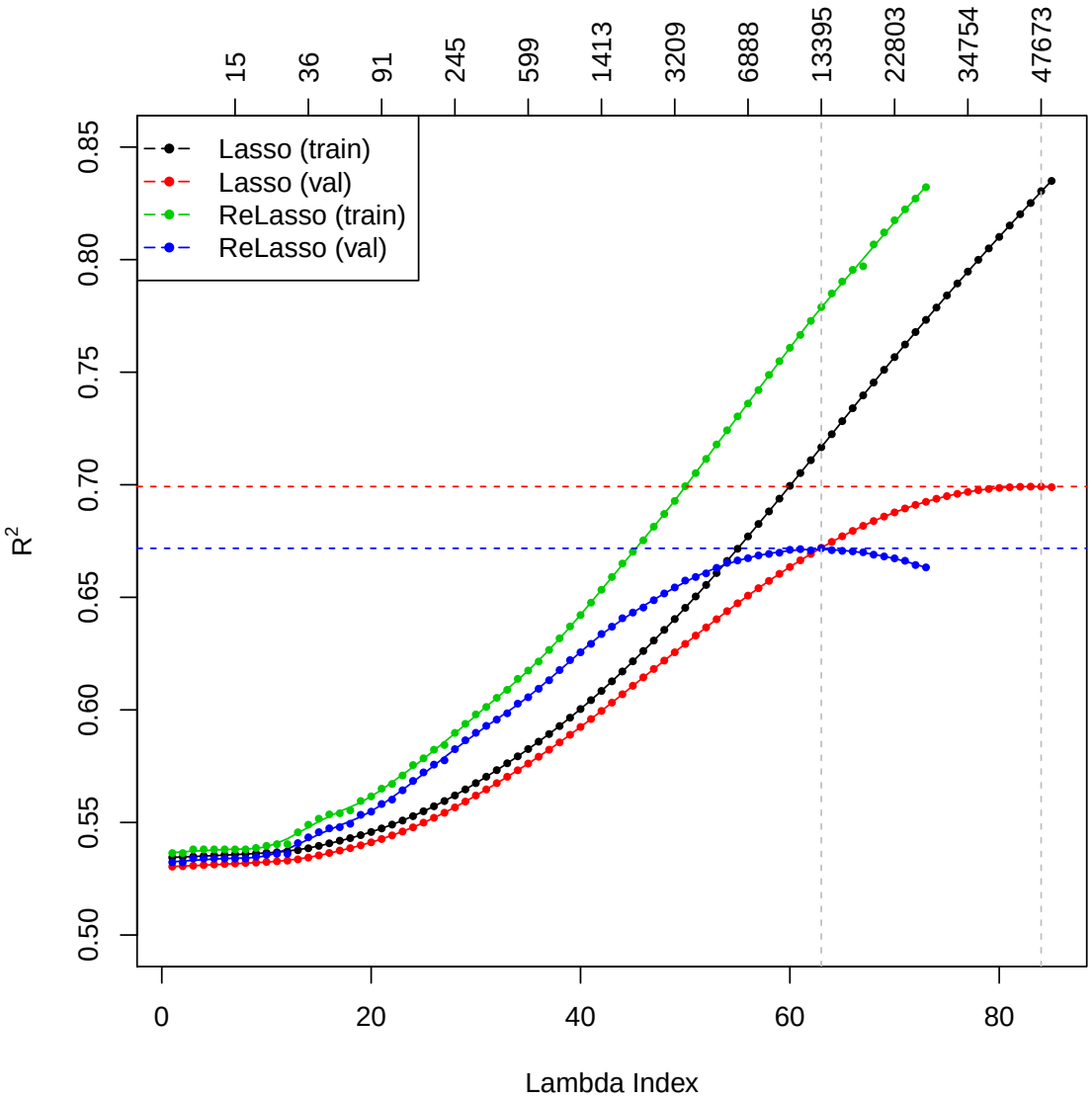


GWAS example

- Large cohort of 500K British adults (Bycroft et al 2018)
- Each individual genotyped at 805K locations (AA, Aa, aa or NA)
- 100s of phenotypes measured on each subject
- We looked at white British subset of 337K, and illustrate with `height` phenotype
- Divided the data 60% training, 20% validation, 20% test.
- computation took a few hours (128GB memory and 16 cores)

Package SNPNET available on Github (link on Hastie's website, and to 2019 report)

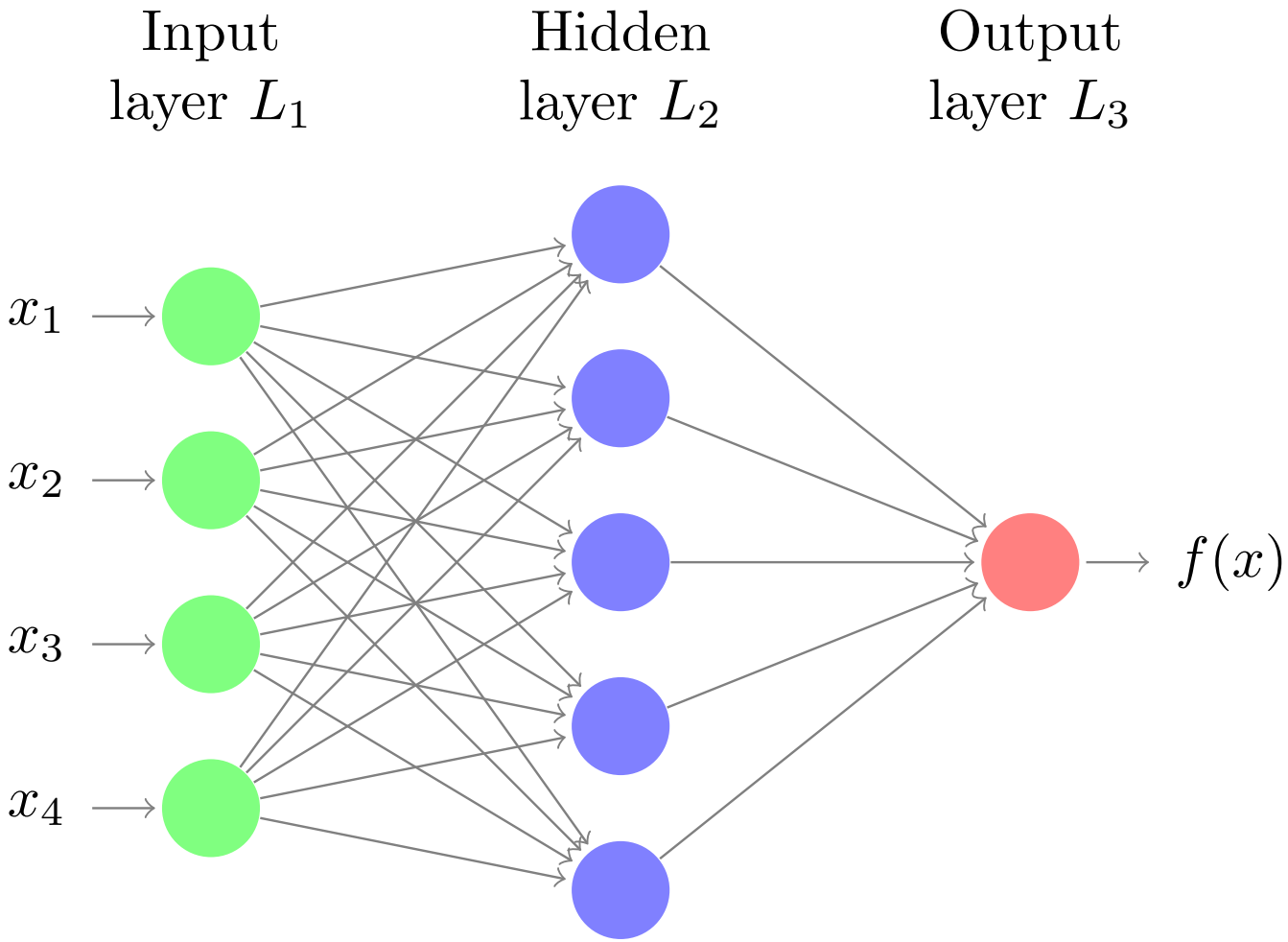
Lasso fit to height



The Elephant in the Room: DEEP LEARNING

Will it eat the lasso and other statistical models?

Deep Nets/Deep Learning



Neural network diagram with a single hidden layer. The hidden layer derives transformations of the inputs — nonlinear transformations of linear combinations — which are then used to model the output

What makes Deep Nets so powerful

(and challenging to analyze!)

It's not one "mathematical model" but a **customizable framework**– a set of **engineering tools** that can exploit the special aspects of the problem (weight-sharing, convolution, feedback, recurrence ...)

Confession: I was Geoff Hinton's colleague at Univ. of Toronto (1985-1998) and didn't appreciate the potential of Neural Networks!

Will Deep Nets eat the lasso and other statistical models?

Deep Nets are especially powerful when the features have some spatial or temporal organization (signals, images), and SNR is high

But they are not a good approach when

- we have moderate #obs or wide data (#obs $<$ #features),
- SNR is low, or
- interpretability is important
- It's difficult to find examples where Deep Nets beat lasso or GBM in low SNR settings, with "generic " features

LassoNet: "If you can't beat'em ....

Feature sparse neural networks

Lemhadri, Ruan, Abraham Tibshirani, JMLR 2021

LassoNet in two minutes
Click for video

LassoNet in detail

- We assume the model

$$y_i = \beta_0 + \sum_j x_{ij}\beta_j + \sum_{k=1}^{K}[\alpha_k + \gamma_k \cdot f(\theta_k^T x_i)] + \epsilon_i \tag{5}$$

with $\epsilon \sim (0, \sigma^2)$. Here f is a monotone, nonlinear function such as a sigmoid or rectified linear unit, and each $\theta_k = (\theta_{1k}, \ldots \theta_{pk})$ is a p-vector.

- Our objective is to minimize

$$\begin{aligned} J(\beta, \Theta, \alpha, \gamma) &= \frac{1}{2}\sum_i (y_i - \hat{y}_i)^2 + \lambda \sum_j |\beta_j| + \bar{\lambda} \sum_{jk} |\theta_{jk}| \\ &\quad \text{subject to } |\theta_{jk}| \le |\beta_j| \ \ \forall j, k. \quad \leftarrow \textit{secret sauce} \end{aligned}$$

In a nutshell

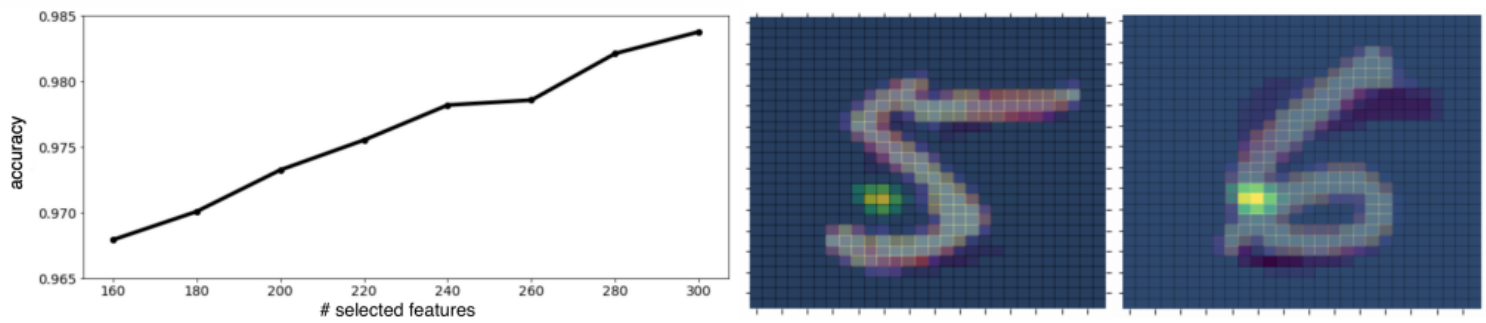
"*Features can participate in the hidden layer only if they have non-zero main effects*"

- A neural network is a complex model, but is made simpler if the model is a function of only a subset of the features
- Eg a protein-based blood test for a disease: if only 10 proteins out of 1000 need to be measured, then the complicated neural net mapping of these 10 proteins is still acceptable to scientists

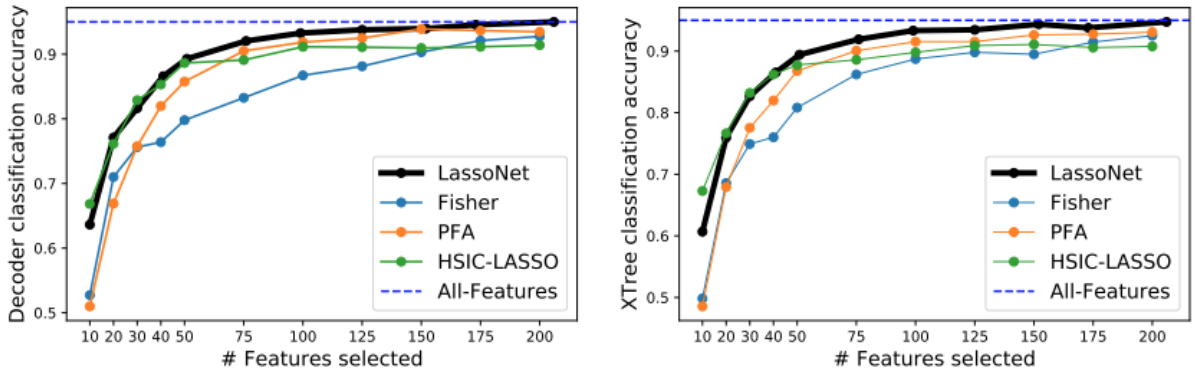
Computational strategy

- We seek the solution over a path of λ values, using current solutions as warm starts. We use a projected proximal gradient procedure at each λ
- Unlike Lasso, objective is **non-convex**. Makes optimization much harder. Eg the starting values for (β, θ) matter!
- In lasso (our glmnet package), we compute a path of solutions from sparse (λ large) to dense (λ small)
- This worked badly for LassoNet; we tried many tricks; finally we tried going from Dense to Sparse. **Worked!**

Examples



Differentiating 5s from 6s



ISOLET (26 classes); LassoNet vs other feature selection methods; two different classifiers

Discussion

- lasso (ℓ_1) penalties are a useful method for encouraging and inducing sparsity
- **Sparsity** is clearly an important idea, for model interpretation, statistical and computational efficiency
- There is emerging evidence that general learners like Deep Nets seek a basis of features that yield a sparse representation in the weight space— without the explicit use of an ℓ_1 penalty (Pilanci et al 2018)
- understanding the role of sparsity— and how to find sparse models in practice– is an important direction for the future