# Compressed and penalized linear regression

Daniel J. McDonald
Indiana University, Bloomington
mypage.iu.edu/ dajmcdon

2 June 2017

# Obligatory "data is big" slide

Modern statistical applications — genomics, neural image analysis, text analysis, weather prediction — have large numbers of covariates $p$

Also frequently have lots of observations $n$.

Need algorithms which can handle these kinds of data sets, with good statistical properties

# Lesson of the talk

Many statistical methods use (perhaps implicitly) a singular value decomposition (SVD) to solve an optimization problem.

The SVD is computationally expensive.

We want to understand the statistical properties of some approximations which speed up computation and save storage.

Spoiler: sometimes approximations actually improve the statistical properties

# CORE TECHNIQUES

Suppose we have a matrix $\mathbb{X} \in \mathbb{R}^{n \times p}$ and vector $Y \in \mathbb{R}^n$

## LEAST SQUARES:

$$\widehat{\beta} = \operatorname*{argmin}_{\beta} ||\mathbb{X}\beta - Y||_2^2$$

## $\ell_2$ REGULARIZATION:

$$\widehat{\beta}(\lambda) = \operatorname*{argmin}_{\beta} ||\mathbb{X}\beta - Y||_2^2 + \lambda ||\beta||_2^2$$

# Core techniques

If $\mathbb{X}$ fits into RAM, there exist excellent algorithms in LAPACK that are

- Double precision
- Very stable
- cubic complexity, $O(np^2 + p^3)$, with small constants
- require extensive random access to matrix

There is a lot of interest in finding and analyzing techniques that extend these approaches to large(r) problems

# Out-of-core techniques

If $\mathbb{X}$ is too large to manipulate in RAM, there are other approaches

- (Stochastic) gradient descent
- Conjugate gradient
- Rank-one QR updates
- Krylov subspace methods

# Out-of-core techniques

Many techniques focus on randomized compression

This is sometimes known as sketching or preconditioning

- Rokhlin, Tygert, (2008) “A fast randomized algorithm for overdetermined linear least-squares regression.”
- Drineas, Mahoney, et al., (2011) “Faster least squares approximation.”
- Woodruff, (2014) “Sketching as a Tool for Numerical Linear Algebra.”
- Wang, Lee, Mahdavi, Kolar, Srebro, (2016) “Sketching meets random projection in the dual.”
- Ma, Mahoney, and Yu, (2015), “A statistical perspective on algorithmic leveraging.”
- Pilanci and Wainwright, (2015-2016). Multiple papers.
- Others.

# Compression

## Basic Idea:

- Choose some matrix $Q \in \mathbb{R}^{q \times n}$ .
- Use $Q\mathbb{X}$ (and) $QY$ instead in the optimization.

Finding $Q\mathbb{X}$ for arbitrary $Q$ and $\mathbb{X}$ takes $O(qnp)$ computations.

This can be expensive.

To get this approach to work, we need some structure on $Q$

# THE $Q$ MATRIX

- Gaussian:
  Well behaved distribution and eas(ier) theory. Dense matrix
- Fast Johnson-Lindenstrauss Methods
- Randomized Hadamard (or Fourier) transformation:
  Allows for $O(np\log(p))$ computations.
- $Q = \pi\tau$ for $\pi$ a permutation of $I$ and $\tau = [I_q\ 0]$:
  $Q\mathbb{X}$ means “read $q$ (random) rows”
- Sparse Bernoulli:

$$Q_{ij} \overset{i.i.d.}{\sim} \begin{cases} 1 & \text{with probability } 1/(2s) \\ 0 & \text{with probability } 1 - 1/s \\ -1 & \text{with probability } 1/(2s) \end{cases}$$

This means $Q\mathbb{X}$ takes $O\left(\frac{qnp}{s}\right)$ “computations” on average.

# Typical results

The general philosophy: Find an approximation algorithm that is as close as possible to the solution of the original problem.

For OLS, typical results would be to produce an $\tilde{\beta}$ such that

$$\begin{aligned} \left|\left|\mathbb{X}\tilde{\beta} - Y\right|\right|_2^2 &\leq (1+\epsilon)\left|\left|\mathbb{X}\beta_* - Y\right|\right|_2^2, \\ \left|\left|\mathbb{X}(\tilde{\beta} - \beta_*)\right|\right|_2^2 &\leq \epsilon\left|\left|\mathbb{X}\beta_*\right|\right|_2^2, \\ \left|\left|\tilde{\beta} - \beta_*\right|\right|_2^2 &\leq \epsilon\left|\left|\beta_*\right|\right|_2^2, \end{aligned}$$

Here, $\tilde{\beta}$ should be 'easier' to compute than

$$\beta_* = \operatorname*{argmin}_{\beta} \left|\left|\mathbb{X}\beta - Y\right|\right|_2^2$$

# Collaborator & Grant Support

Collaborator:
Darren Homrighausen
Department of Statistics
Southern Methodist University

Grant Support:
NSF, Institute for New Economic Thinking

# An alternate perspective

# Risk

Form a loss function $\ell : \Theta \times \Theta \to \mathbb{R}^+$

The quality of an estimator is given by its risk

$$R(\widehat{\theta}) = \mathbb{E}\left[\ell(\widehat{\theta}, \theta)\right]$$

We could use $\ell_2$ estimation risk:

$$R(\widehat{\theta}) = \mathbb{E}\left|\left|\theta - \widehat{\theta}\right|\right|_2^2$$

or excess $\ell_2$ prediction risk

$$\begin{aligned} R(\widehat{\theta}) = \mathbb{E}\left|\left|Y - \mathbb{X}\widehat{\theta}\right|\right|_2^2 &= \mathbb{E}\left|\left|Y - \mathbb{X}\theta + \mathbb{X}\theta - \mathbb{X}\widehat{\theta}\right|\right|_2^2 \\ &\propto \text{constant} + \mathbb{E}\left|\left|\mathbb{X}(\theta - \widehat{\theta})\right|\right|_2^2 \end{aligned}$$

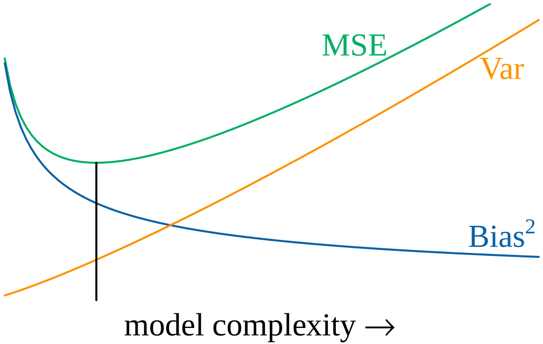
# RISK DECOMPOSITION

For an approximation $\tilde{\theta}$ of $\widehat{\theta}$,

$$\begin{aligned}\mathbb{E}\left|\left|\theta - \tilde{\theta}\right|\right|_2^2 &= \mathbb{E}\left|\left|\tilde{\theta} - \widehat{\theta} + \widehat{\theta} - \mathbb{E}\widehat{\theta} + \mathbb{E}\widehat{\theta} - \theta\right|\right|_2^2 \\ &\leq \mathbb{E}\text{Approx. error}^2 + \text{Variance} + \text{Bias}^2\end{aligned}$$

- Previous analyses focus only on the approximation error (with expectation over the algorithm, not the data).

# Bias-variance tradeoff



- Typical result compares to the zero-bias estimator which is assumed to have small variance.
- We examine $\mathbb{E}\left|\left|\theta - \tilde{\theta}\right|\right|_2^2$ where the expectation is over everything random.
- Closest similar analysis is Ma, Mahoney, and Yu (JMLR, 2015).

# Compressed regression

Let $Q \in \mathbb{R}^{q \times n}$

Call the fully compressed least squares estimator

$$\widehat{\beta}_{FC} = \operatorname*{argmin}_{\beta} ||Q(\mathbb{X}\beta - Y)||_2^2$$

→ A common way to solve least squares problems that are:

- Very large or
- Poorly conditioned

The numerical/theoretical properties generally depend on $Q$, $q$, $\mathbb{X}$

# Family of 4

1. Full compression:

$$\begin{aligned}\widehat{\beta}_{FC} &= \underset{\beta}{\operatorname{argmin}} ||Q(\mathbb{X}\beta - Y)||_2^2 \\ &= \underset{\beta}{\operatorname{argmin}} ||QY||_2^2 + ||Q\mathbb{X}\beta||_2^2 - 2Y^\top Q^\top Q\mathbb{X}\beta \\ &= (\mathbb{X}^\top Q^\top Q\mathbb{X})^{-1}\mathbb{X}^\top Q^\top QY\end{aligned}$$

2. Partial compression:[1]

$$\begin{aligned}\widehat{\beta}_{PC} &= \underset{\beta}{\operatorname{argmin}} ||Y||_2^2 + ||Q\mathbb{X}\beta||_2^2 - 2Y^\top \mathbb{X}\beta \\ &= (\mathbb{X}^\top Q^\top Q\mathbb{X})^{-1}\mathbb{X}^\top Y\end{aligned}$$

[1] Also called "Hessian Sketching".

# We also combine these

Write:

$$B = [\ \widehat{\beta}_{FC}\ \widehat{\beta}_{PC}\ ]$$

$$W = \mathbb{X}B$$

3. Linear combination compression:

$$\widehat{\alpha}_{lin} = \operatorname*{argmin}_{\alpha} ||W\alpha - Y||_2^2$$

$$\widehat{\beta}_{lin} = B\widehat{\alpha}_{lin}$$

4. Convex combination compression:

$$\widehat{\alpha}_{con} = \operatorname*{argmin}_{\substack{0 \leq \alpha \\ \sum \alpha = 1}} ||W\alpha - Y||_2^2$$

$$\widehat{\beta}_{con} = B\widehat{\alpha}_{con}$$

- These are simple to calculate given FC and PC.

# Why these?

- Turns out that FC is (approximately) unbiased, and therefore worse than OLS (has high variance)
- On the other hand, PC is biased and empirics demonstrate low variance
- Combination should give better statistical properties
- We do everything with an $\ell_2$ penalty

# Evidence from simulations

# Simulation setup

- Draw $\mathbb{X}_i \sim \mathbf{MVN}(0, (1-\rho)I_p + \rho\mathbf{1}\mathbf{1}^\top)$
  - We use $\rho = \{0.2,\ 0.8\}$.
- Draw $\beta \sim \mathrm{N}(0, \tau^2 I_p)$
- Draw $Y_i = \mathbb{X}_i^\top\beta + \epsilon_i$ with $\epsilon_i \sim \mathrm{N}(0, \sigma^2)$.

# BAYES ESTIMATOR

- For this model, the optimal estimator (in MSE) is

$$\widehat{\beta}_B = (\mathbb{X}^\top \mathbb{X} + \lambda_* I_p)^{-1} \mathbb{X}^\top Y$$

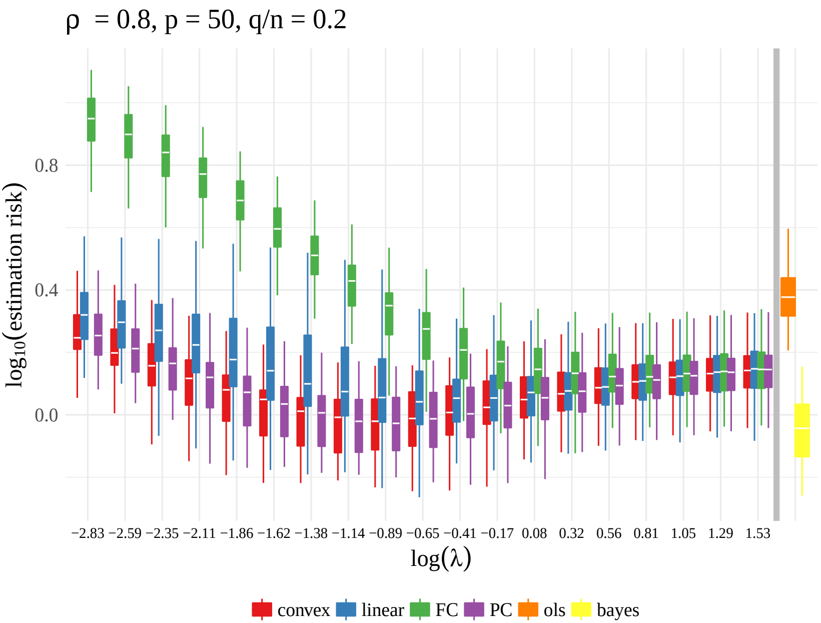
- In particular, with $\lambda_* = \frac{\sigma^2}{n\tau^2}$
- This is the posterior mode of the Bayes estimator under conjugate normal prior
- It is also the ridge regression estimator for a particular $\lambda$

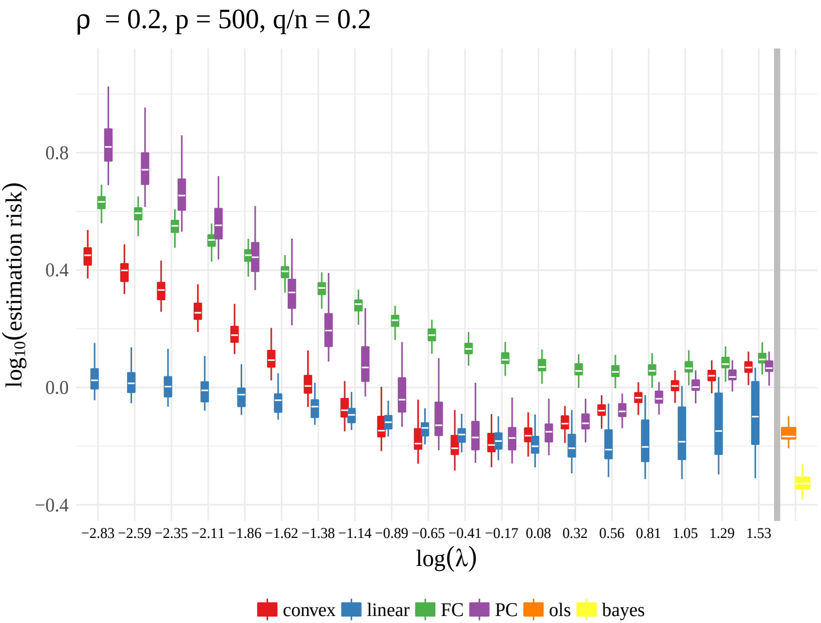
# GOLDILOCKS

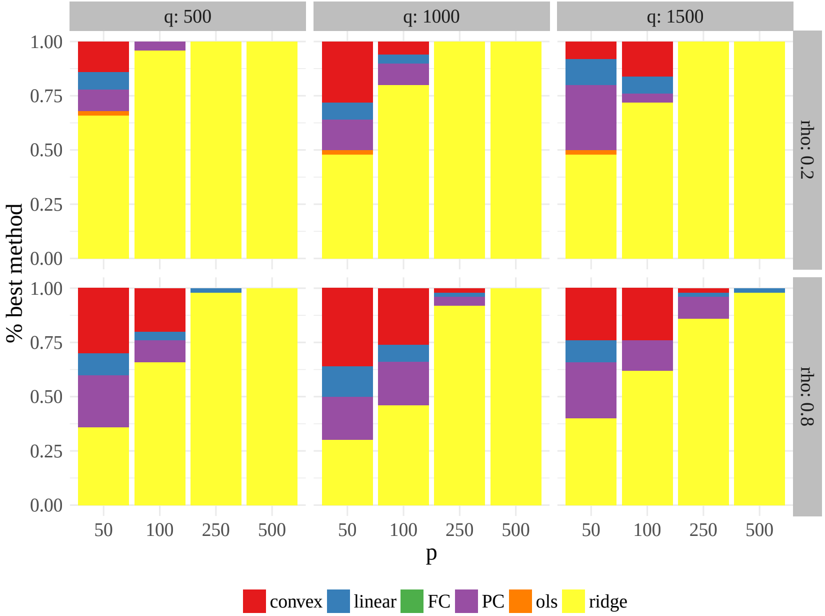
With $p < n$

1. If $\lambda_*$ is too big, we will tend to shrink all coefficients to 0.
   - This problem is too hard.
2. If $\lambda_*$ is too small, OLS will be very close to the optimal estimator.
   - This problem is too easy.
3. Need $\tau^2$, $\sigma^2$ “just right”.

- Take $\tau^2 = \pi/2$. This implies $\mathbb{E}[|\beta_i|] = 1$ (convenient)
- Take $n = 5000$. Big but computable.
- Take $\sigma = 50 \Rightarrow \log(\lambda_*) \approx -1.14$ (reasonable)
- Take $p \in \{50,\ 100,\ 250,\ 500\}$


ρ = 0.8, p = 50, q/n = 0.2
log10(estimation risk)
0.8
0.4
0.0
−2.83
−2.59
−2.35
−2.11
−1.86
−1.62
−1.38
−1.14
−0.89
−0.65
−0.41
−0.17
0.08
0.32
0.56
0.81
1.05
1.29
1.53
log(λ)
convex
linear
FC
PC
ols
bayes


ρ = 0.2, p = 500, q/n = 0.2
$\log_{10}$(estimation risk)
0.8
0.4
0.0
−0.4
−2.83
−2.59
−2.35
−2.11
−1.86
−1.62
−1.38
−1.14
−0.89
−0.65
−0.41
−0.17
0.08
0.32
0.56
0.81
1.05
1.29
1.53
log(λ)
convex
linear
FC
PC
ols
bayes

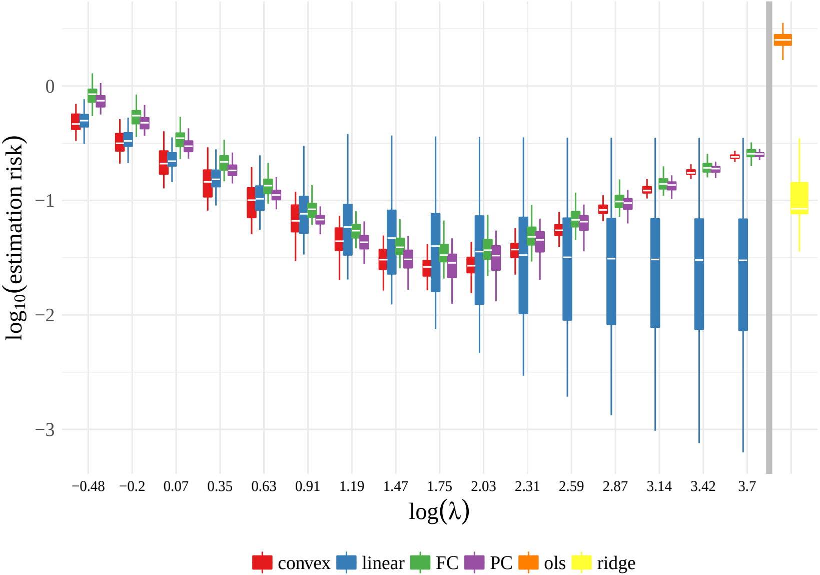

q: 500
q: 1000
q: 1500
rho: 0.2
rho: 0.8
% best method
1.00
0.75
0.50
0.25
0.00
50
100
250
500
p
convex
linear
FC
PC
ols
ridge

# Second verse...

In that case, ridge was optimal.

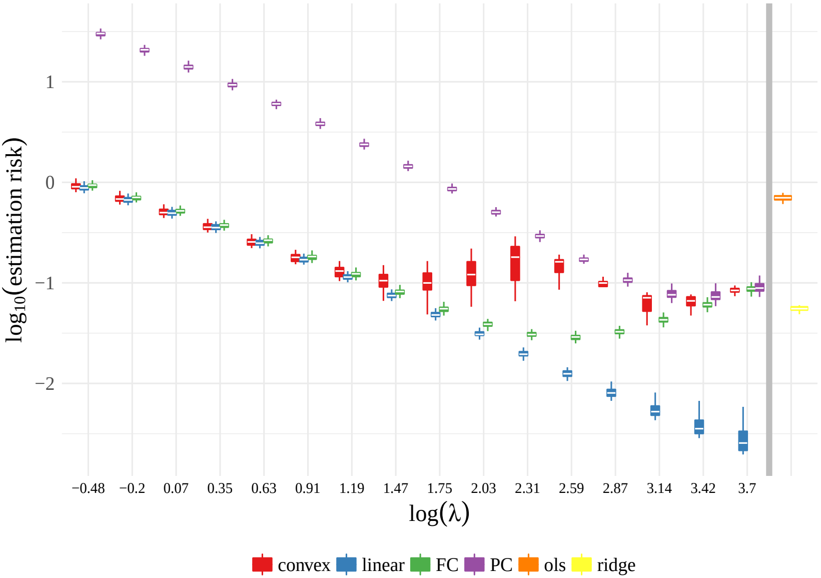
We did it again with $\beta_i \equiv 1$.

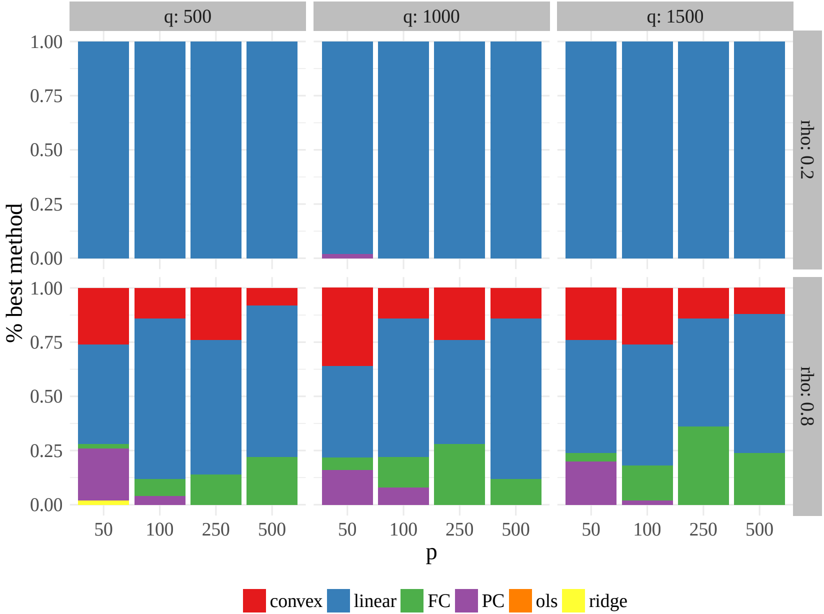
ρ = 0.8, p = 50, q/n = 0.2



log10(estimation risk)
0
−1
−2
−3
−0.48
−0.2
0.07
0.35
0.63
0.91
1.19
1.47
1.75
2.03
2.31
2.59
2.87
3.14
3.42
3.7
log(λ)
🟥 convex
🟦 linear
🟩 FC
🟪 PC
🟧 ols
🟨 ridge

ρ = 0.2, p = 500, q/n = 0.2



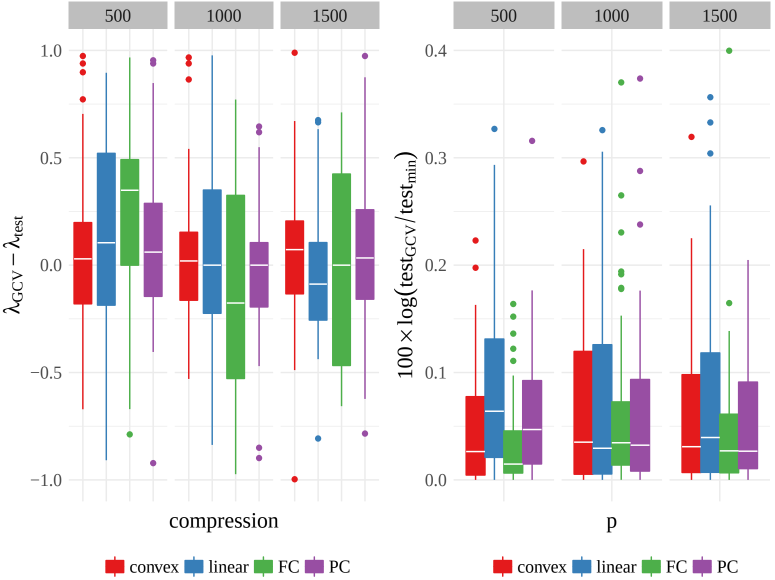

q: 500
q: 1000
q: 1500
rho: 0.2
rho: 0.8
% best method
1.00
0.75
0.50
0.25
0.00
50
100
250
500
p
convex
linear
FC
PC
ols
ridge

# SELECTING TUNING PARAMETERS

- We use GCV with the degrees of freedom:

$$\mathrm{GCV}(\lambda) = \frac{\left|\left|\mathbb{X}\widehat{\beta}(\lambda) - Y\right|\right|_2^2}{(1 - df/n)^2}$$

- $df$ is easy for full or partial compression.
- For the other cases, we do a dumb approximation: weighted average using $\widehat{\alpha}$.
- Easy to calculate for a range of $\lambda$ without extra computations.
- Also calculate the divergence to derive Stein's Unbiased Risk Estimate.
- After tedious algebra, it had odd behavior in practice (can be huge, or negative!?!)



$$n = 5000, \;\; p = 50, \;\; \rho = 0.8, \;\; \beta \sim \mathrm{N}(0, \pi/2)$$

# Theoretical results (sketch)

# Standard ridge results

Theorem

$$\text{bias}^2\left(\widehat{\beta}_{ridge}(\lambda) \mid \mathbb{X}\right) = \lambda^2 \beta_*^\top V(D^2 + \lambda I_p)^{-2} V^\top \beta_*.$$

$$\text{tr}\left(\mathbb{V}[\widehat{\beta}_{ridge}(\lambda) \mid \mathbb{X}]\right) = \sigma^2 \sum_{i=1}^{p} \frac{d_i^2}{(d_i^2 + \lambda)^2}.$$

# WHAT'S THE TRICK?

- Similar results are hard for compressed regression.
- All the estimators depend (at least) on

$$(\mathbb{X}^\top Q^\top Q \mathbb{X} + \lambda I_p)^{-1}$$

- We derived properties of $Q^\top Q$

$$\mathbb{E}\left[\frac{s}{q} Q^\top Q\right] = I_n$$

$$\mathbb{V}\left[\operatorname{vec}\left(\frac{s}{q} Q^\top Q\right)\right] = \frac{(s-3)_+}{q} \operatorname{diag}(\operatorname{vec}(I_n)) + \frac{1}{q} I_{n^2} + \frac{1}{q} K_{nn}$$

- So the technique is to do a Taylor expansion around $\frac{s}{q} Q^\top Q = I_n$.

# Main result

## Theorem

$$\begin{aligned}
\operatorname{bias}^2[\widehat{\beta}_{FC} \mid \mathbb{X}] &= \lambda^2 \beta_*^\top V(D^2 + \lambda I_p)^{-2} V^\top \beta_* + o_p(1) \\
\operatorname{tr}(\mathbb{V}[\widehat{\beta}_{FC} \mid \mathbb{X}]) &= \sigma^2 \sum_{i=1}^{p} \frac{d_i^2}{(d_i^2 + \lambda)^2} + o_p(1) \\
&\quad + \frac{(s-3)_+}{q} \operatorname{tr}\Big(\operatorname{diag}(\operatorname{vec}(I_n)) M^\top M \otimes (I - H) M \beta_* \beta_*^\top M^\top (I - H)\Big) \\
&\quad + \frac{\beta_*^\top M^\top (I - H)^2 M \beta_*}{q} \operatorname{tr}(M M^\top) \\
&\quad + \frac{1}{q} \operatorname{tr}\Big((I - H) M \beta_* \beta_*^\top M^\top (I - H) M^\top M\Big) .
\end{aligned}$$

Note: $M = (\mathbb{X}^\top \mathbb{X} + \lambda I_p)^{-1} \mathbb{X}^\top$ and $H = \mathbb{X} M$ (hat matrix for ridge regression)

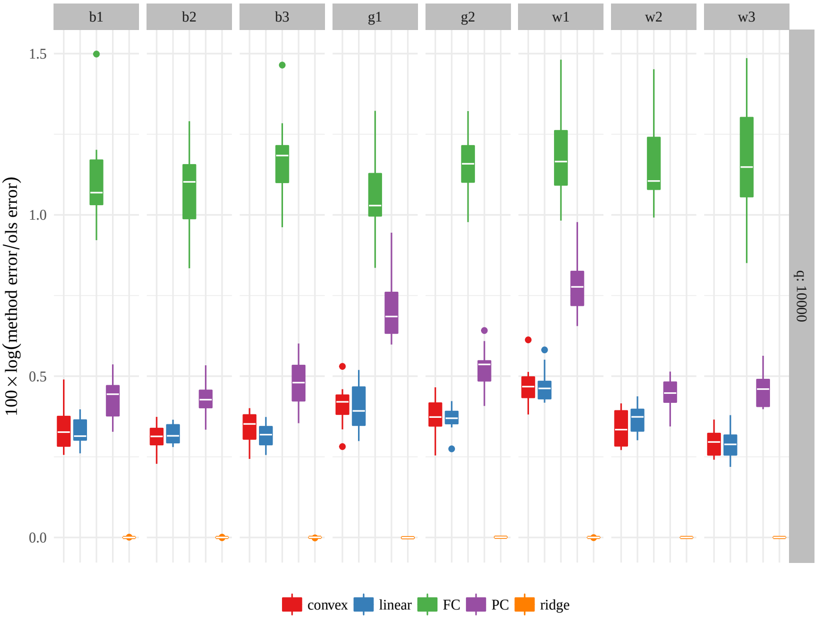
# Applications

# RNA-SEQ

- short-read RNA sequence data
- using a (Poisson) linear model to predict read counts based on the surrounding nucleotides
- 8 different tissues
- data is publicly available, preprocessing already done

# Results

- Test error averaged over 10 replications.
- On each replication, tuning parameters were chosen with GCV.
- At the best tuning parameter, the test error was computed and then these were averaged.
- For each dataset, we randomly chose 75% of the data as training on each replication.

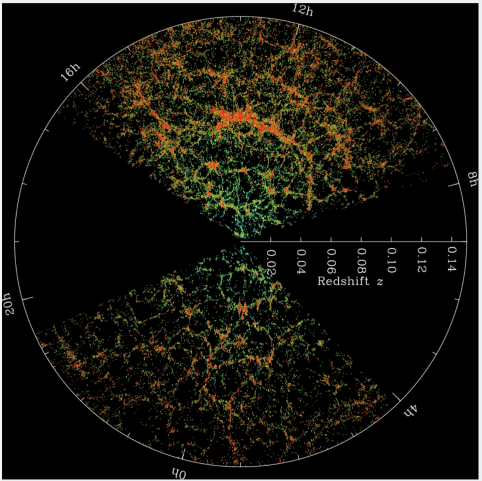

b1
b2
b3
g1
g2
w1
w2
w3
q: 10000
1.5
1.0
0.5
0.0
$100 \times \log(\text{method error/ols error})$
convex
linear
FC
PC
ridge

# Take-away message

- $q = 10000$ results in data reductions between $74\%$ and $93\%$
- $q = 20000$ gives reductions between $48\%$ and $84\%$
- ridge and ordinary least squares give equivalent test set performance (differing by less than $.001\%$)
- This is an “easy” problem.
- Full compression is the worst.

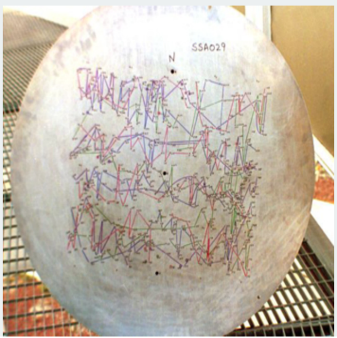
⏩ Skip galaxies

# Current application



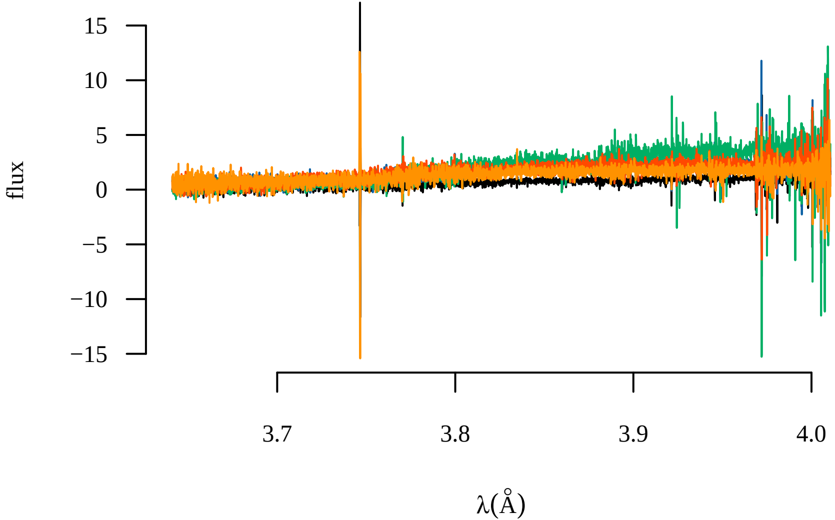
This image and the next are from the Sloan Digital Sky Survey

# SDSS

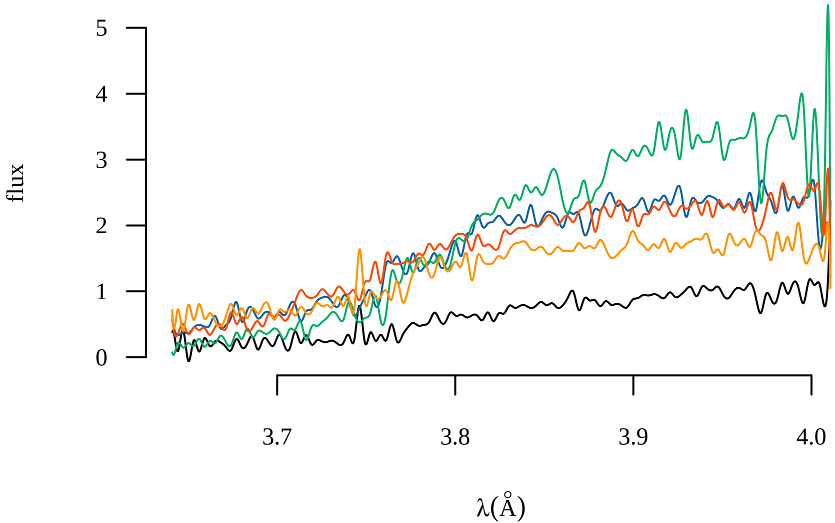


- Repeatedly “photograph” regions of the sky
- Collect information on piles of objects
- Around 200 million galaxy spectra at the moment
- Each spectra contains around 5000 wavelengths
- Want to predict the redshift from the spectra

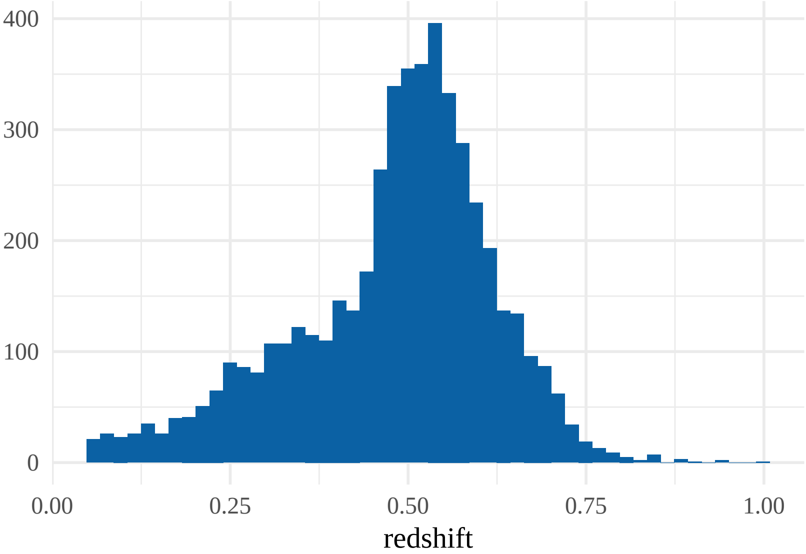
# A few galaxies

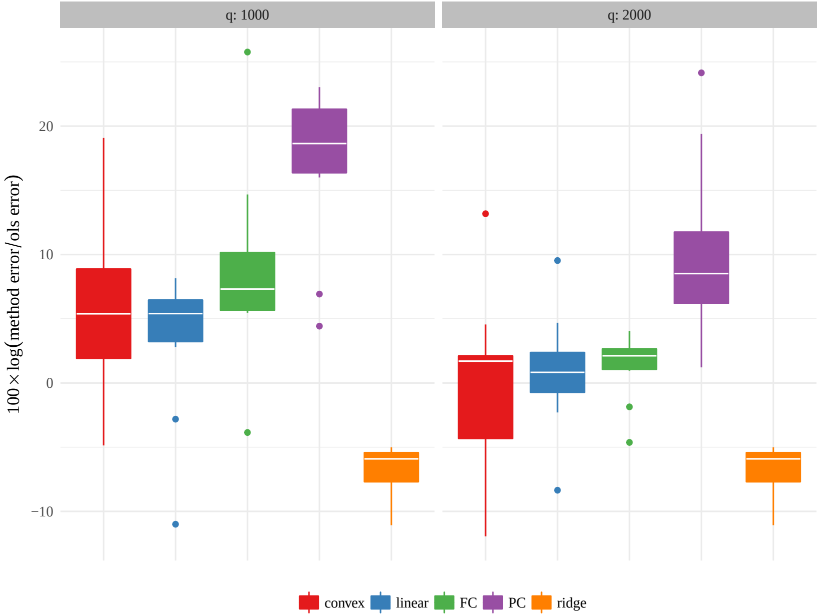


# SAME GALAXIES AFTER SMOOTHING



# 5000 REDSHIFTS




q: 1000
q: 2000
100 × log(method error/ols error)
20
10
0
−10
convex
linear
FC
PC
ridge

# Conclusions

# Conclusions

- Lots of people are looking at approximations to standard statistical methods (OLS, PCA, etc.)
- They characterize the approximation error
- We have been looking at whether we can actually benefit from the approximation
- Today looked at compressed regression

# Ongoing and future work

- We want to generalize our results here to other penalties
- Also generalized linear models
- We're also looking at how compression interacts with PCA
- Combine compression and random projection
- Connections to sufficient dimension reduction?
- Iterative versions?