Compressed and Penalized Linear Regression

-Combining Statistical & Computational Efficiency-

Darren Homrighausen
Visiting Assistant Professor
Department of Statistical Science
Southern Methodist University

Collaborators & Grant Support

Collaborator:

Daniel J. McDonald, Indiana University, Bloomington

(~~Assistant~~ Associate Professor in the Department of Statistics)

Grant Support:

- NSF Grant DMS14-07543
- INET Grant INO-14-00020

Outline

- Overview of problem
  - Approximation schemes
  - Background and motivation
  - Approximation-regularization
- Methods
  - Compressed Regression
  - Sparsified mean PCA

Overview of problem

A big problem

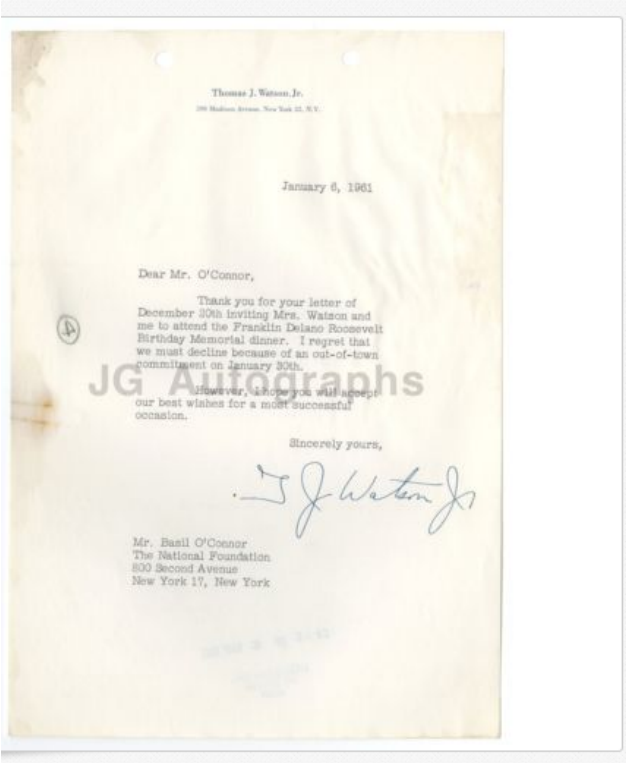
Large data sets have become very common

Astrophysics: Modern cosmology relies on creating a very large database of a particular type of supernova.

(Can we classify and record the type of the ∼30 billion/year supernovae observable from Earth?)

Text processing: Comments left by buyer/seller in eBay auctions along with sales price of the item

eBay auctions



Thomas Watson, Jr. - IBM Chairman - Authentic Autographed Letter (TLS)

Item condition: --

Ended: May 27, 2014 16:59:11 PDT

Winning bid: **US $11.61** [6 bids]

Shipping: **$3.99** Standard Shipping | See details
Item location: **United States**
Ships to: **Worldwide**

Delivery: Estimated within 3-6 business days

Payments: PayPal | See details

Returns: 14 days money back, buyer pays return shipping | See details

Guarantee: ebay MONEY BACK GUARANTEE | See details
Get the item you ordered or get your money back.
Covers your purchase price and original shipping.

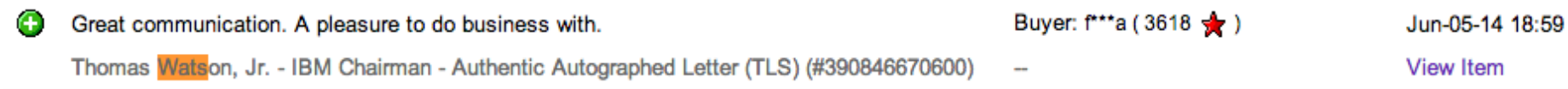
Seller information
jgautographs (64927 ★)
100% Positive feedback

Follow this seller
See other items
Visit store: JG Autographs

Example big data problem

Buyer:

Seller:



The data (∼750 Gb, millions of rows, thousands of columns):

	always	pleas	smooth	transact	great	commun	busin		Sales Price ($)
$X_1^\top =$	[1	2	1	1	0	0	0],	$Y_1 =$	[17.53]
$X_2^\top =$	[0	1	0	0	1	1	1],	$Y_2 =$	[17.53]

Core techniques

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

(eBay review: each column of $\mathbb{X}$ is a count for each word, Y is the sales price)

Least squares:

Finding

$$\hat{\beta}_{LS} \text{ such that } \min_{\beta} ||\mathbb{X}\beta - Y||_2^2 = \left|\left|\mathbb{X}\hat{\beta} - Y\right|\right|_2^2$$

Principal Components Analysis (PCA):

(Or graph Laplacian or diffusion map or..)

Finding U, V orthogonal and D diagonal such that

$$\mathbb{X} - \overline{\mathbb{X}} = UDV^\top$$

where

$$\overline{\mathbb{X}} = \mathbf{1}\mathbf{1}^\top \mathbb{X}$$

Core techniques

If $\mathbb{X}$ fits into random access memory (RAM), there exist excellent algorithms in LAPACK that...

- ... have double precision
- ... are very stable
- ... have cubic complexity with small constants

  (General least squares problem: $O(np^2)$)
- ... 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, we can use:

- (Stochastic) gradient descent
- Conjugate gradient
- iterative QR updates
- Krylov subspace methods (e.g. `SLEPc` or `IRLBA`)

(These can use less storage/computations but more read/write latency and are approximate)

Out of core techniques

Many techniques focus on randomized compression

(This is sometimes known as sketching)

Least squares:

1. Rokhlin, Tygert "A fast randomized algorithm for overdetermined linear least-squares regression" (2008)
2. Drineas, Mahoney, et al. "Faster least squares approximation' (2011)
3. Woodruff "Sketching as a tool for numerical linear algebra" (2013)
4. Pilanci and Wainwright "Iterative Hessian Sketch.." (2016)
5. Homrighausen, McDonald "Compressed and Penalized Linear Regression" (under review)

Spectral Decomposition:

1. Halko, et al. "Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions" (2011)
2. Gittens, Mahoney "Revisiting the Nyström method for improved large-scale machine learning" (2013)
3. Pourkamali "Memory and computation efficient PCA via very sparse random projections" (2014)
4. Homrighausen, McDonald "On the Nyström and column-sampling methods for the approximate PCA of large data sets" (2016)

(Of course, there are many other papers not included for brevity's sake)

General problem specifics

Reminder: The matrix $\mathbb{X} \in \mathbb{R}^{n \times p}$ and the vector $Y \in \mathbb{R}^n$

We will be concerned with the scenario in which $n \gg p$

(This makes sense in the Ebay example as the number of auctions grows much faster than the vocabulary)

The idea of sketching is define a compression parameter $n \gg q \gg p$

The procedure is then applied to the sketched/compressed data

Typical results

This q parameter needs to be chosen sensibly so that the induced procedure...

- has "good" statistical properties
- reduces the computational/storage burden

Some examples of "good":

Least Squares:

A typical result would be to find an $\tilde{\beta}$ such that

$$\frac{1}{2n}\left|\left|\mathbb{X}\tilde{\beta} - Y\right|\right|_2^2 \leq (1+\epsilon)^2 \left(\min_{\beta} \frac{1}{2n} ||\mathbb{X}\beta - Y||_2^2\right)$$

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

Typical results

This q parameter needs to be chosen sensibly so that the induced procedure...

- has "good" statistical properties
- reduces the computational/storage burden

Some examples of "good":

PCA:

A typical result would be to find an approximate $\tilde{V}$ such that

$$\text{angle}(V, \tilde{V}) \leq \sqrt{\frac{p}{n}} \left(\frac{1}{\text{spectral gap}} \right)$$

(This is the same order of convergence as PCA [Homrighausen, McDonald (2016)])

Compressed regression

Fully compressed regression

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

(The exact form of this matrix will be discussed later. Though the choice of Q is important, the choice of q is the relevant topic for now)

Let's look at the fully compressed least squares problem

$$\hat{\beta}_{FC} = \underset{\beta}{\text{argmin}} \frac{1}{2n} ||Q(\mathbb{X}\beta - Y)||_2^2$$

(This is also known as preconditioning)

This is a commonly suggested way of compressing least squares problems that are either...

- ... very large
- ... or poorly conditioned (typically in these applications, $q = n$ and $Q\mathbb{X}$ is a more stable matrix)

(e.g. Boutsidis & Drineas (2009), Mahoney (2011), Drineas et al. (2011), ...)

Fully compressed regression

There are various ways to analyze $\hat{\beta}_{FC}$

We've already discussed one:

$$\frac{1}{2n}\left|\left|\mathbb{X}\hat{\beta}_{FC} - Y\right|\right|_2^2 \leq (1+\epsilon)^2\left(\min_{\beta}\frac{1}{2n}||\mathbb{X}\beta - Y||_2^2\right)$$

Typically need $q \succeq \frac{p}{\epsilon^2}$

(Mahoney (2011))

What about other criteria?

Fully compressed regression

Instead of comparing the value through the residual sums of squares, we can compare to the least squares solution itself

$$\hat{\beta}_{LS} := \underset{\beta}{\text{argmin}} \, ||\mathbb{X}\beta - Y||_2^2$$

For instance, we can compare the predictions made by $\hat{\beta}_{FC}$ to those made by $\hat{\beta}_{LS}$

$$\frac{1}{n} \left|\left| \mathbb{X}\hat{\beta}_{FC} - \mathbb{X}\hat{\beta}_{LS} \right|\right|_2^2$$

Let's suppose that there is a true β_* such that

$$Y = \mathbb{X}\beta_* + \epsilon,$$

where ϵ is a "nice" stochastic term

(For example, ϵ_i are i.i.d Gaussian)

Fully compressed regression

Under typical assumptions,

$$\frac{1}{n}\left|\left|\mathbb{X}\hat{\beta}_{LS} - \mathbb{X}\beta_*\right|\right|_2^2 = \frac{1}{n}\left|\left|\text{Projection}_{\mathbb{X}}\epsilon\right|\right|_2^2 \asymp \frac{\sigma^2 p}{n}$$

Combining these two results together, we find that

$$q \succeq \frac{p}{\epsilon^2} \asymp \frac{n}{\sigma^2}$$

Conclusion: We must pick q as the same order as n, which (in an asymptotic order sense) defeats the whole purpose of compression

So, $\hat{\beta}_{FC}$ seems like a flawed approach. We can push this even further..

Fully compressed regression

Note:

$$||Q(\mathbb{X}\beta - Y)||_2^2 \propto \beta^\top \mathbb{X}^\top Q^\top Q\mathbb{X}\beta - 2\beta^\top \mathbb{X}^\top Q^\top QY$$

$$\rightarrow (\mathbb{X}^\top Q^\top Q\mathbb{X})\hat{\beta}_{FC} = \mathbb{X}^\top Q^\top QY$$

We can decompose $\mathbb{R}^n = \mathrm{col}(\mathbb{X}) \oplus \mathrm{null}(\mathbb{X}^\top)$

Two facts immediately follow:

Fact 1: $Y = \mu + R$ and $\mathbb{P}(R \in \mathrm{col}(\mathbb{X})) = 0$

Fact 2: If $\mu = \mathbb{X}\beta_*$ for some β_*, then $\mathbb{E}R = 0$

(e.g. if linear model is true)

Fully compressed regression

Assume: $\mathbb{X}^\top Q^\top Q\mathbb{X}$ is invertible

Then

$$\hat{\beta}_{FC} = \beta_* + (\mathbb{X}^\top Q^\top Q\mathbb{X})^{-1}\mathbb{X}^\top Q^\top QR$$

$\rightarrow \hat{\beta}_{FC}$ is unbiased!

Hence, $\hat{\beta}_{FC}$ is provably worse in a risk sense than $\hat{\beta}_{LS}$

(As $\hat{\beta}_{LS}$ is UMVUE and hence has the same bias but lower variance)

Yet, this is by far the most commonly taken approach in the approximation literature!

Partially compressed regression

Instead of:

$$\min_\beta \left(\beta^\top \mathbb{X}^\top Q^\top Q \mathbb{X}\beta - 2\beta^\top \mathbb{X}^\top Q^\top Q Y\right)$$

Consider:

$$\min_\beta \left(\beta^\top \mathbb{X}^\top Q^\top Q \mathbb{X}\beta - 2\beta^\top \mathbb{X}^\top Y\right)$$

$$\begin{aligned} \to \hat{\beta}_{PC} &= \operatorname*{argmin}_\beta \left(\beta^\top \mathbb{X}^\top Q^\top Q \mathbb{X}\beta - 2\beta^\top \mathbb{X}^\top Y\right) \\ &= (\mathbb{X}^\top Q^\top Q \mathbb{X})^{-1} \mathbb{X}^\top \mathbb{X}\hat{\beta} \end{aligned}$$

(Again, under the invertibility assumption)

$\hat{\beta}_{PC}$ has the "opposite" behavior: high bias, low variance

Compressed regression

Recap: To do good predictions/estimation, we need to calibrate bias and variance

We have two estimators

- Low bias/ high variance
- High bias/ low variance

$\rightarrow$ Combine them!

Compressed regression

1. Form the matrix

$$B = \left[\; \hat{\beta}_{FC}, \hat{\beta}_{PC} \; \right] \in \mathbb{R}^{p \times 2}$$

2. and compute

$$\hat{\alpha} = \underset{\alpha}{\text{argmin}} \, ||B\alpha - Y||_2^2$$

(Can add a convex constraint $\alpha_1 + \alpha_2 = 1$)

The estimator: $\hat{\beta}_C = B\hat{\alpha}$

Back to the Q matrix

The Q matrix

Example: Let the entries in Q, Q_{ij}, to be i.i.d standard normal

This is attractive as $Q\mathbb{X}$ is now a Gaussian matrix

(In general, we could assume that Q_{ij} are i.i.d sub-Gaussian and control $Q\mathbb{X}$ via non-commutative concentration inequalities)

Problem: Finding $Q\mathbb{X}$ for an arbitrary dense Q and $\mathbb{X}$ takes $O(qnp)$ computations using matrix multiplication

This immediately destroys the advantage of compression as q must be larger than p

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

The Q matrix

- Random orthogonal subsampling (e.g. Hadamard, or Fourier)

  (Allows for $O(np\log(p))$ computations)
- Row sampling

  (Very easy/fast computationally. However, we should sample proportionate to the leverage scores, which are expensive to compute)

The Q matrix

- Random orthogonal subsampling (e.g. Hadamard, or Fourier)

  (Allows for $O(np\log(p))$ computations)
- Row sampling

  (Very easy/fast computationally. However, we should sample proportionate to the leverage scores, which are expensive to compute)
- 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"

Compressed regression

With this Q, $\hat{\beta}_C$ "works" in practice:

- Computational savings: $O\left(\frac{qnp}{s} + qp^2\right)$
- Approximately the same risk: $R(\hat{\beta}_C) \approx R(\hat{\beta}_{LS})$

(Details omitted)

This is good, but we had a realization:

Constrained methods outperform OLS in terms of risk

(e.g. Hoerl and Kennard (1970))

So, we should seek to compress a constrained least squares procedure

Compressed ridge regression

This means introducing a tuning parameter λ and defining:

$$\hat{\beta}_{PC}(\lambda) = (\mathbb{X}^\top Q^\top Q\mathbb{X} + \lambda Q^\top Q)^{-1}\mathbb{X}^\top Y$$
$$\hat{\beta}_{FC}(\lambda) = (\mathbb{X}^\top Q^\top Q\mathbb{X} + \lambda Q^\top Q)^{-1}\mathbb{X}^\top Q^\top QY$$

(Everything else about the procedures is the same)

This has the same computational complexity

Let's look at a typical result..

Simulation setup

- Draw $X_i \sim \text{MVN}(0, (1-\rho)I_p + \rho 11^\top)$
  - We use $\rho = \{0.2,\ 0.8\}$.
- Draw $\beta_* \sim \text{N}(0, \tau^2 I_p)$
- Draw $Y_i = X_i^\top \beta_* + \epsilon_i$ with $\epsilon_i \sim \text{N}(0, \sigma^2)$.

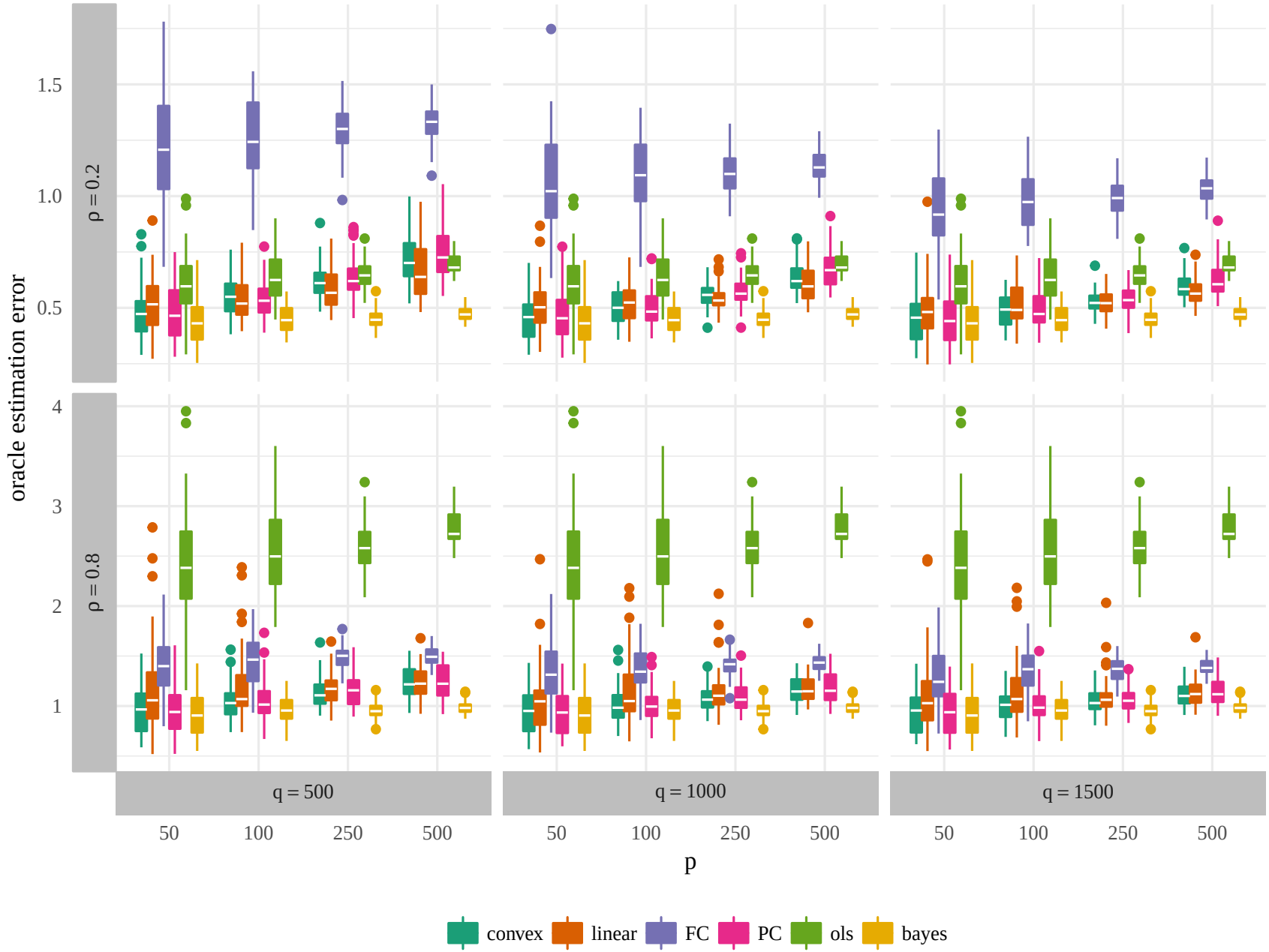
Bayes' estimator:

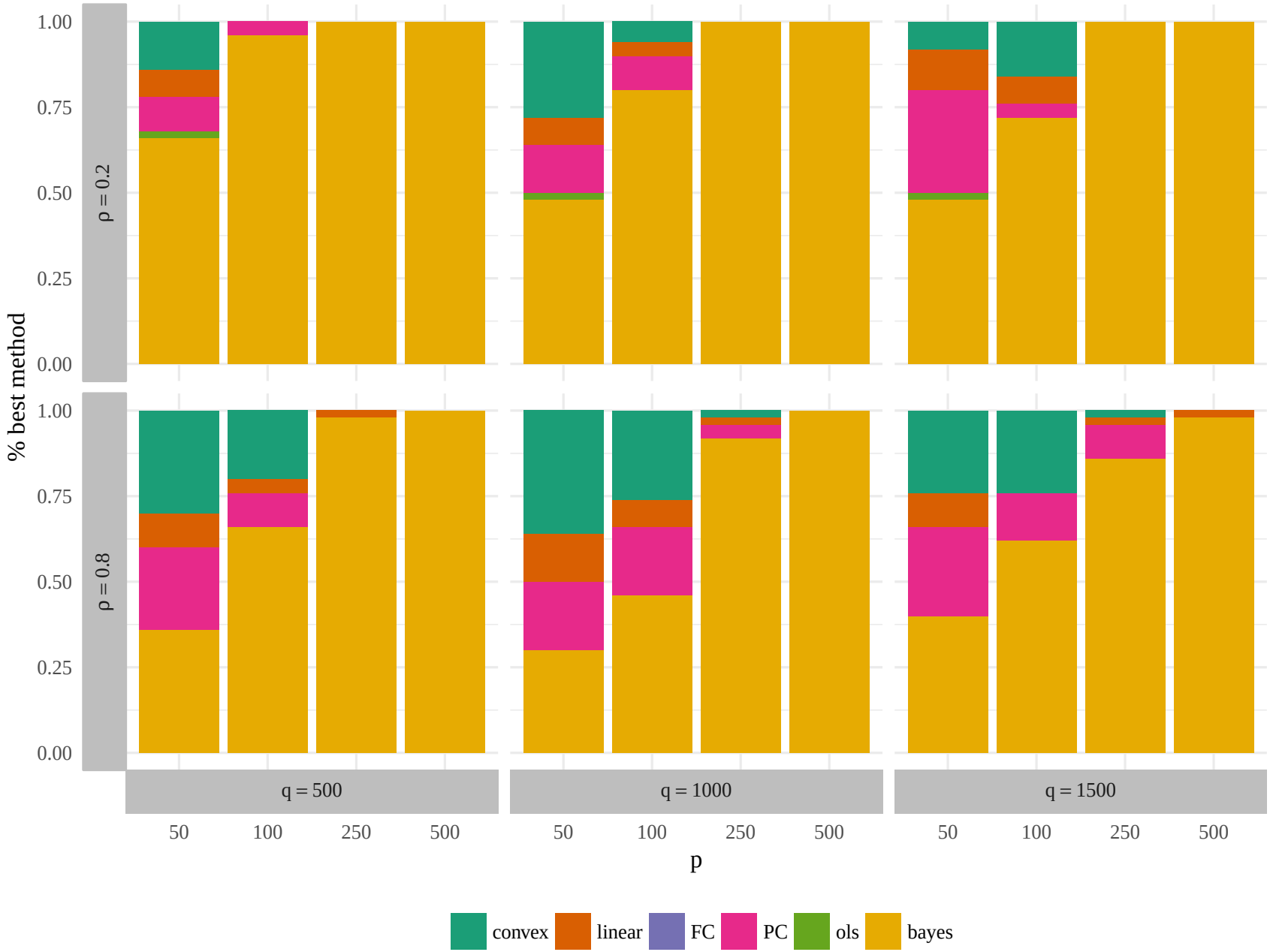
- For this model, the optimal estimator (in MSE) is

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

- In particular, with $\lambda_* = \frac{\sigma^2}{n\tau^2}$

  (This is the mean/mode of posterior under conjugate normal prior)

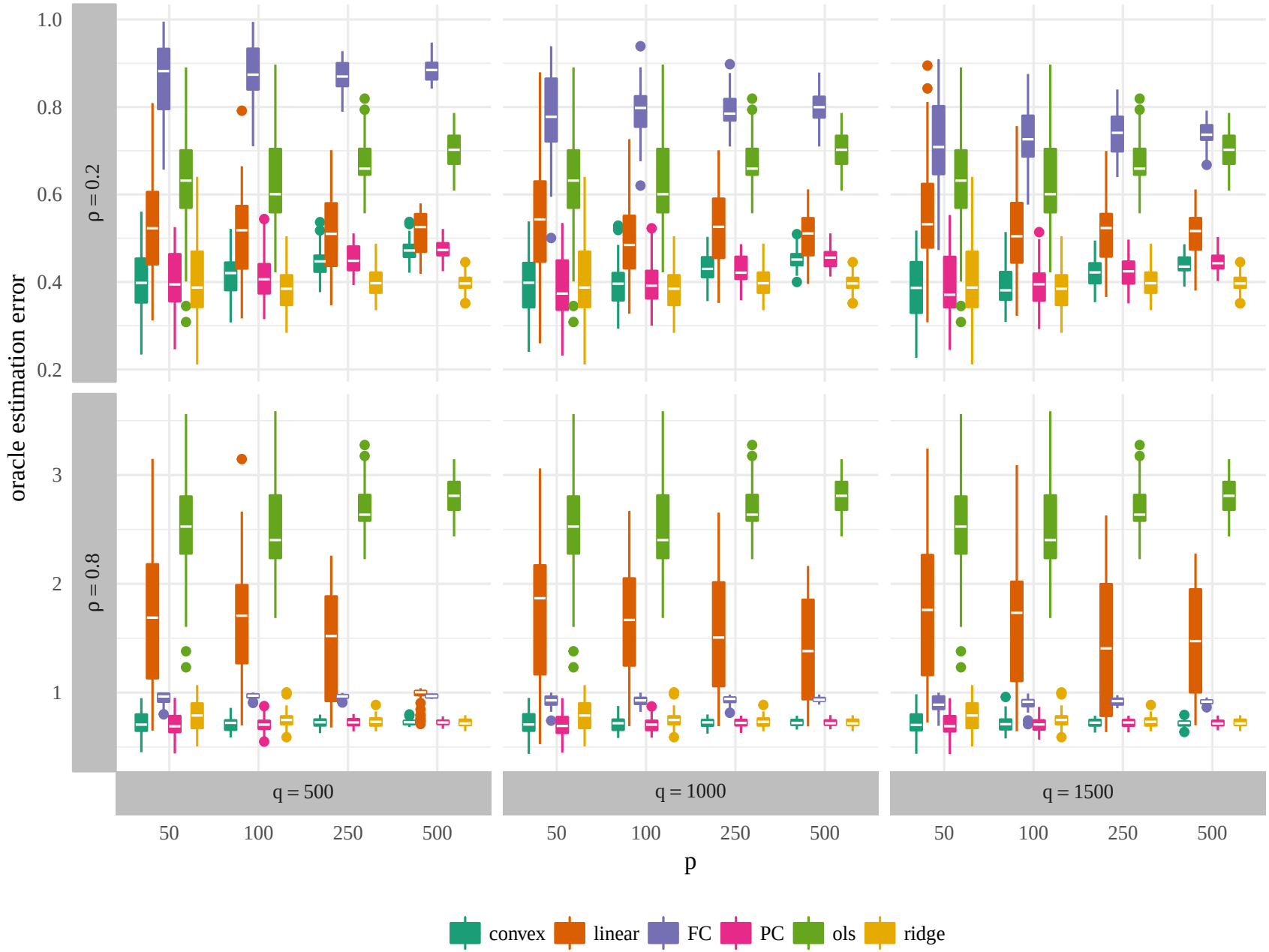

oracle estimation error
ρ = 0.2
ρ = 0.8
q = 500
q = 1000
q = 1500
p
50
100
250
500
convex
linear
FC
PC
ols
bayes

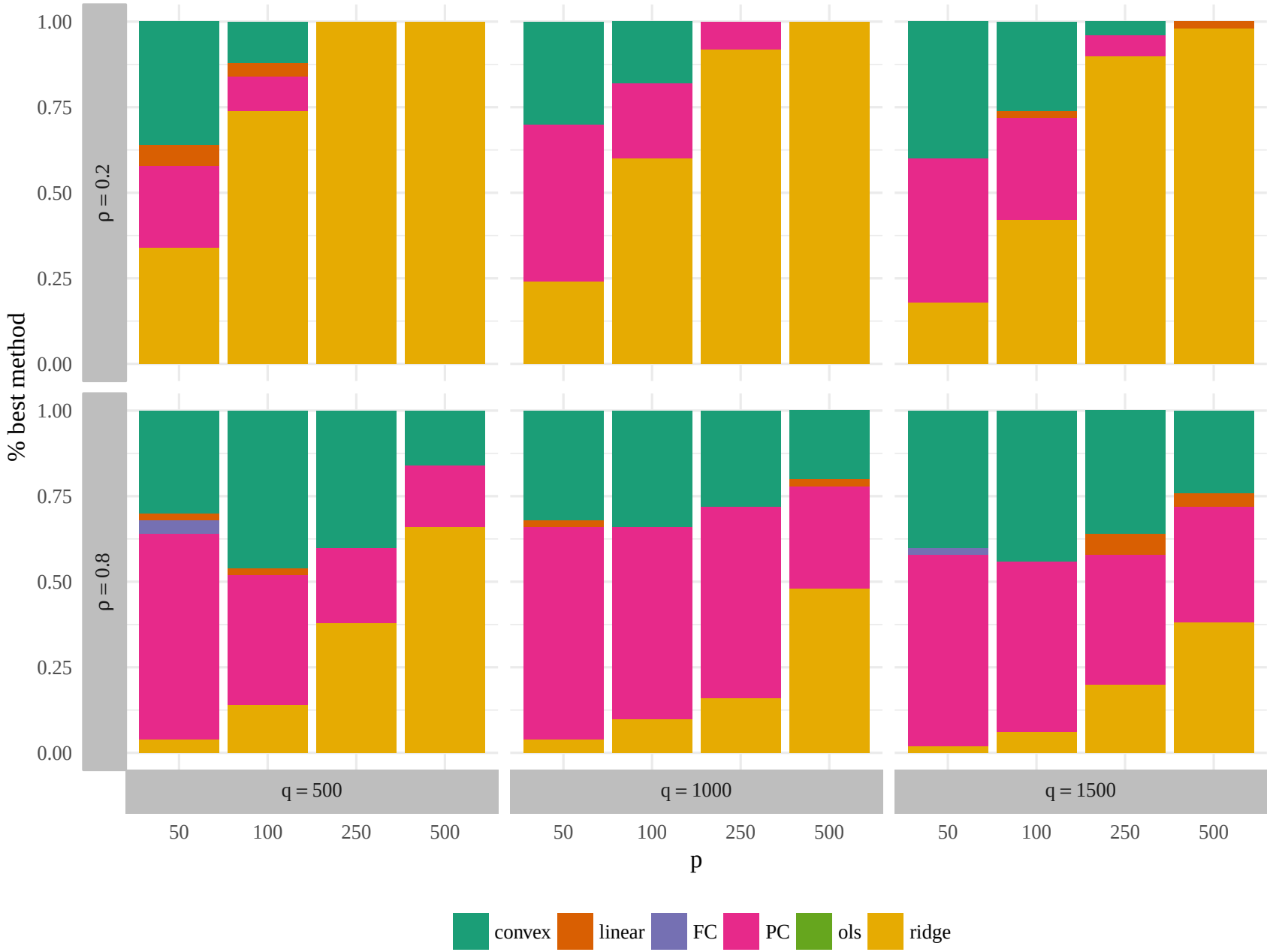

% best method
1.00
0.75
0.50
0.25
0.00
ρ = 0.2
ρ = 0.8
q = 500
q = 1000
q = 1500
50
100
250
500
p
convex
linear
FC
PC
ols
bayes

Second simulation. . .

In the previous case, ridge was optimal.

Let's look at another scenario where $\beta_* \in \{-1, 1\}^p$.


oracle estimation error
ρ = 0.2
ρ = 0.8
q = 500
q = 1000
q = 1500
p
50
100
250
500
convex
linear
FC
PC
ols
ridge


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

Tuning parameter selection

Setting λ

Now that we introduced a tuning parameter (λ), we need a way to set it

A resampling-based risk estimate (e.g. some flavor of cross-validation or bootstrap) wouldn't work

$\rightarrow$ too computationally intensive

So, we use a risk estimate based on degrees of freedom instead

Setting λ

The degrees of freedom for a generic predictor f is

$$\mathrm{df}(f) := \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(Y_i, f_i(Y)).$$

Example: For a linear procedure (e.g. $f(Y) = HY$), we have

$$\mathrm{df} = \mathrm{trace}(H) \underbrace{=}_{\mathrm{OLS}} \mathrm{rank}(\mathbb{X})$$

Setting λ

We use GCV with the degrees of freedom:

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

This requires an estimate of df

(But not of the variance)

This is easy for full or partial compression

(they are linear, after all)

The linear/convex combination is more difficult as they are nonlinear

Setting λ

Reminder:

$$B = \left[\; \hat{\beta}_{FC}, \hat{\beta}_{PC} \; \right] \quad \text{and} \quad \hat{\alpha} = \underset{\alpha}{\text{argmin}} \, ||B\alpha - Y||_2^2$$

For the linear/convex combination, we can approximate df with

$$\hat{\alpha}^\top \begin{bmatrix} df_{FC} \\ df_{PC} \end{bmatrix}$$

(This is sometimes done with neural networks, e.g. Ingrassia, S. and Morlini, I. (2007))

This approach has worked well in practice but will underestimate df

So, we derive an estimate via Stein's lemma instead..

Setting λ

An expression for the degrees of freedom can be found via Stein's Lemma:

$$\mathrm{df} = \mathbb{E} \sum_{i=1}^{n} \frac{\partial \hat{Y}_i}{\partial Y_i}$$

(This is the divergence. This result requires normality and almost sure differentiability)

This gives us an unbiased estimator of the degrees of freedom:

$$\widehat{\mathrm{df}} = \sum_{i=1}^{n} \frac{\partial \hat{Y}_i}{\partial Y_i}$$

Theoretical results

Standard ridge results

Theorem

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

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

(Here, we are writing $\mathbb{X} = UDV^\top$ as the SVD)

Preliminary details

Results conditional on Q are more relevant for analyzing the $\hat{\beta}$'s presented thus far

However, if $n \gg q$, then we can still save on computations if we average a few $\hat{\beta}$'s with different draws of Q

(This is similar to classical multiple imputation schemes)

Results unconditional on Q are more relevant in this case

We have theoretical results both conditional on Q & not

(only the unconditional results are stated for brevity)

Preliminary details

All the estimators depend (theoretically) on $Q^\top Q$

(Note: we wouldn't want to form $Q^\top Q$ explicitly in practice)

Some properties of $Q^\top Q$

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

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

So the technique is to do a Taylor expansion around

$$\frac{s}{q}Q^\top Q = I_n$$

MSE of Full Compression

Theorem:

$$\text{bias}^2[\hat{\beta}_{FC}|\mathbb{X}] = \lambda^2 \beta_*^\top V(D^2 + \lambda I_p)^{-2} V^\top \beta_* + o_p(1)$$

$$\begin{aligned}\text{trace}(\mathbb{V}[\hat{\beta}_{FC}|\mathbb{X}]) = \sigma^2 &\sum_{i=1}^{p} \frac{d_i^2}{(d_i^2+\lambda)^2} + o_p(1) \\ &+ \frac{(s-3)_+}{q}\text{trace}\left(\text{diag}(\text{vec}I_n)M^\top M \otimes (I-H)M\beta_*\beta_*^\top M^\top (I-H)\right) \\ &+ \frac{\beta_*^\top M^\top (I-H)^2 M\beta_*}{q}\text{trace}(MM^\top) \\ &+ \frac{1}{q}\text{trace}\left((I-H)M\beta_*\beta_*^\top M^\top (I-H)M^\top M\right).\end{aligned}$$

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

(hat matrix for ridge regression)

Special case

Corollary:
If $\mathbb{X}^\top \mathbb{X} = nI_p$,

$$\mathrm{MSE}(\hat{\beta}_{ridge}) = b^2 \left(\frac{\theta}{1+\theta}\right)^2 + \frac{p\sigma^2}{n(1+\theta)^2}$$

$$\mathrm{MSE}(\hat{\beta}_{FC}) = b^2 \left(\frac{\theta}{1+\theta}\right)^2 + \frac{p\sigma^2}{n(1+\theta)^2} + \frac{b^2 p\theta^2 (s-2)_+}{q(1+\theta)^4} + \frac{p^2\theta^2 b^2}{q(1+\theta)^4}$$

$$\mathrm{MSE}(\hat{\beta}_{PC}) = b^2 \left(\frac{\theta}{1+\theta}\right)^2 + \frac{p\sigma^2}{n(1+\theta)^2} + \frac{p(s-2)_+ b^2}{q(1+\theta)^2} + \frac{pb^2}{q(1+\theta)^4}$$

Where $b^2 := ||\beta_*||_2^2$, and $\theta := \lambda/n$

PCA

Least squares applied to PCA

A similar approach can be applied to the Affine embedding problem

Find

$$\min_{W:\mathrm{rank}(W)=k} ||\mathbb{X} - W||_F^2$$

For PCA:

$$\min_{\mu,(d_i),V\in\mathcal{S}_k} \sum_{i=1}^{n} ||X_i - \mu - Vd_i||_2^2 = \min_{(d_i),V\in\mathcal{S}_k} \sum_{i=1}^{n} \left|\left|X_i - \overline{X} - Vd_i\right|\right|_2^2$$

($\mathcal{S}_k$ is the Stiefel manifold of rank-k orthogonal matrices)

Conclusion

Thank you listening!

Grant Support:

- NSF Grant DMS14-07543
- INET Grant INO-14-00020