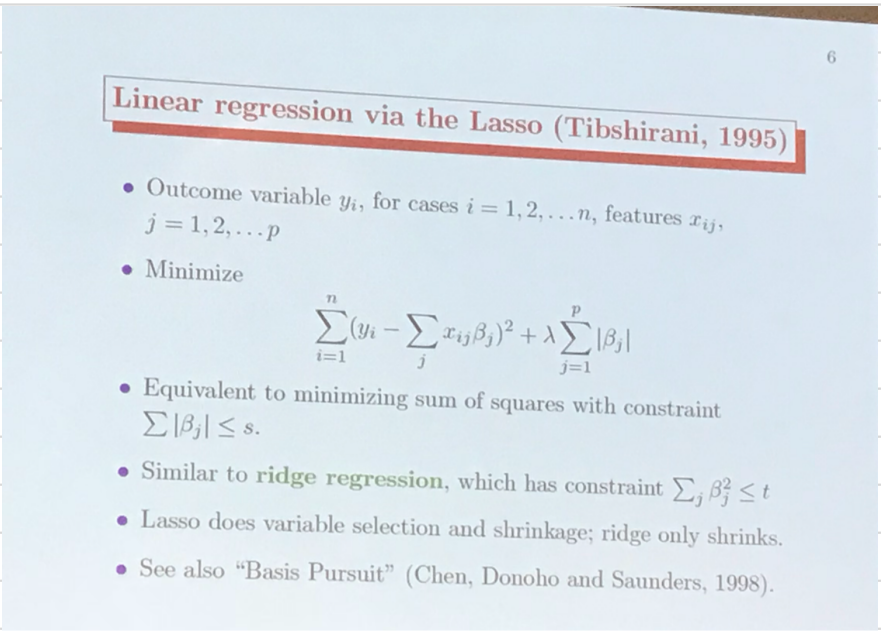
5/29/2018



6

Linear regression via the Lasso (Tibshirani, 1995)

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

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

- Equivalent to minimizing sum of squares with constraint $\sum|\beta_j| \le s$.
- Similar to **ridge regression**, which has constraint $\sum_j \beta_j^2 \le t$
- Lasso does variable selection and shrinkage; ridge only shrinks.
- See also "Basis Pursuit" (Chen, Donoho and Saunders, 1998).

LS: $p << \sqrt{n}$

$p \approx \sqrt{n}$ or $p >> n$

$\approx n$ $\log(p) = n$

Ridge Regression Lasso Regression

variance of LS explodes	infinite many least squares
reduce the var	limit to one var

Ridge : Minimizes variance $\hat{\beta}$ (least square estimator)

$\hat{\beta}_{ls} = \underbrace{(X^TX)^{-1}}X^TY$

Expectation of this is the variance

when this is small ⇒ shouldn't use least squares

$\hat{\beta}_{Ridge} = (X^TX + \lambda I_p)^{-1}X^TY$

↳ Adding identity to enlarge minimum eigenvalue

smaller λ ⇒ closer to least squares

too large λ ⇒ losing variety of Y

} ⇒ λ → tuning parameter, $\lambda \ge 0$

Idea:

Increase $\lambda_{min}(X^TX)$ to reduce variance in estimation ($p \approx n$)

Background

Ridge → 60's

Thykanov (regularization)

↳ Regularize your estimator by assuming properties / minimizes variance

Non-parametric Regression

$Y_i = f(X_i) + \varepsilon_i$ f ⇒ unknown ← want to estimate → $\hat{f} = ?$

unknown $f(x) = ?$

(1) Approaches ⇒ Basis Expansion (Kernalization)

- Polynomials X ⇒ fitty overlap
- Kernel functions
- Spline basis (B-splines)

$f(x) = \sum_{k=1}^{\infty} \cdot B_k(x)$

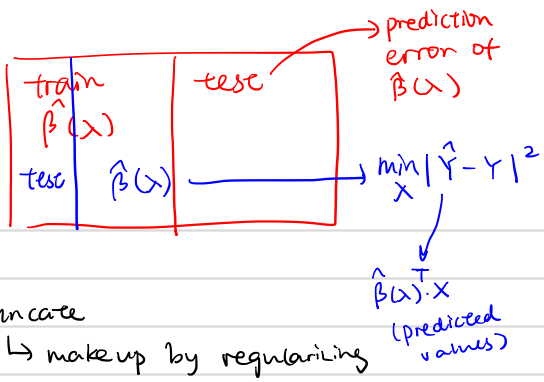
B_k's should spread (spanning the whole space)

Ridge:

choose λ that gives you best prediction

↳ cross-validation

↳ split into (train + test)



② how many Bases we should choose

$$f(x) = \sum_{k=1}^{\infty} B_k(x) \cdot a_k$$

$$\hat{f}(x) = \sum_{k=1}^{?} B_k(x) \cdot \hat{a}_k \quad \leftarrow \text{estimating}$$

Truncate

↳ make up by regularizing

Thykanov Regularization:

⇒ we can choose p as long as what we have left is small enough

$$\min_{a_1, a_2, \cdots, a_p} \sum_{i=1}^{n} (Y_i - \sum_{k=1}^{p} B_k(X_i) \cdot a_k)^2 + \lambda \cdot \sum_{k=1}^{n} \sum_{i=1}^{n} a_k^2 \cdot B_k^2(X_i)$$

(a_k: unknown; the penalty term → penalty function)

minimizes the variance, ensure that $\hat{f}$ is a smooth function

↳ smooth instead of overfitting

controls the size of the bias

(introduce the bias, but λ minimizes the bias)

Linear Regression:

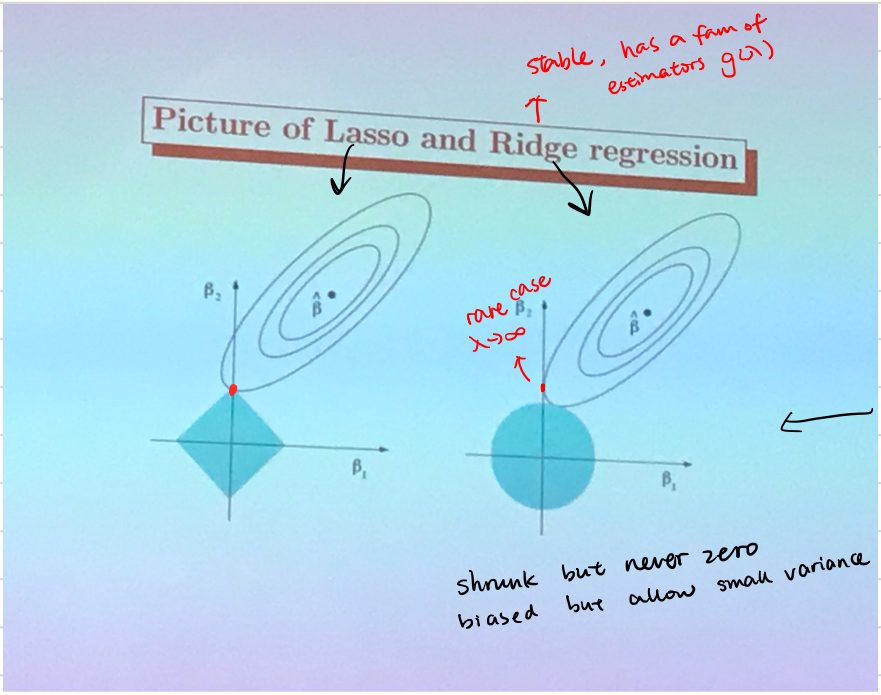
$$Y_i = \beta^T \cdot X_i + \varepsilon_i$$

Thykanov ⇒ $\min \sum_{i=1}^{n} (Y_i - \beta^T X_i)^2 + \lambda \cdot \|\beta\|_2^2$

$\lambda \cdot \sum_{j=1}^{p} \beta_j^2$

← seem like **Lagarange**

← related to { Budget constraint and indifference curve }



$$\min_{\beta} \sum_{i=1}^{n} (Y_i - \beta^T X_i)^2$$

such that $\sum_{j=1}^{n} \beta_j^2 \le t$

↳ $\sum_{j=1}^{n} \beta_j = t$

← solution is at the edge

Ridge vs L.S.

$\hat{\beta}_j(\lambda)$ is smaller than $\hat{\beta}_{jLS}$ $\Rightarrow$ shrinkage effect

$\hat{\beta}_j(\lambda) \neq 0$, $\hat{\beta}_{jLS} \neq 0$

No hypothesis testing for because $\hat{\beta}_j(\lambda)$ is biased, meaningless to do hypo-test.

↳ not able to reduce the bias by adding corrector

↳ λ is too huge $\Rightarrow$ larger bias

↳ λ is too small $\Rightarrow$ overfitting

minimizes $\sum$ variance^2 + bias ?

Lasso Estimator ($\log(p) \approx n$)

↳ (least absolute Shrinkage and Selection Operator)

$$\min_{\beta} \sum_{i=1}^{n} (Y_i - \beta^T X_i)^2 + \lambda \underbrace{\|\beta\|_1}_{\sum_{j=1}^{p} |\beta_j|}$$

← as long as is not differentiable get the result of 0

$\min \sum_{i=1}^{n} (Y_i - \beta^T X_i)^2$ such that $\|\beta\|_1 \leq t$ Constraint problem

not convex

SCAD

smallest not convex.

Lasso vs Ridge

$\hat{\beta}_{Lasso}(\lambda)_j = 0$ $\hat{\beta}_{Ridge}(\lambda)_j \neq 0$ $\lambda > 0$

↑

not used for prediction

minimizing prediction error → that's it ⇒ no way of having p-val / other indicators

↳ λ-step forward stepwise ⇒ p-value, y's variability

$\bar{J}$: prediction

$\bar{J}$: features explaining variability in Y

LARS Algorithm → find variable highly related with Y

margin line of residuals correlates with other variables

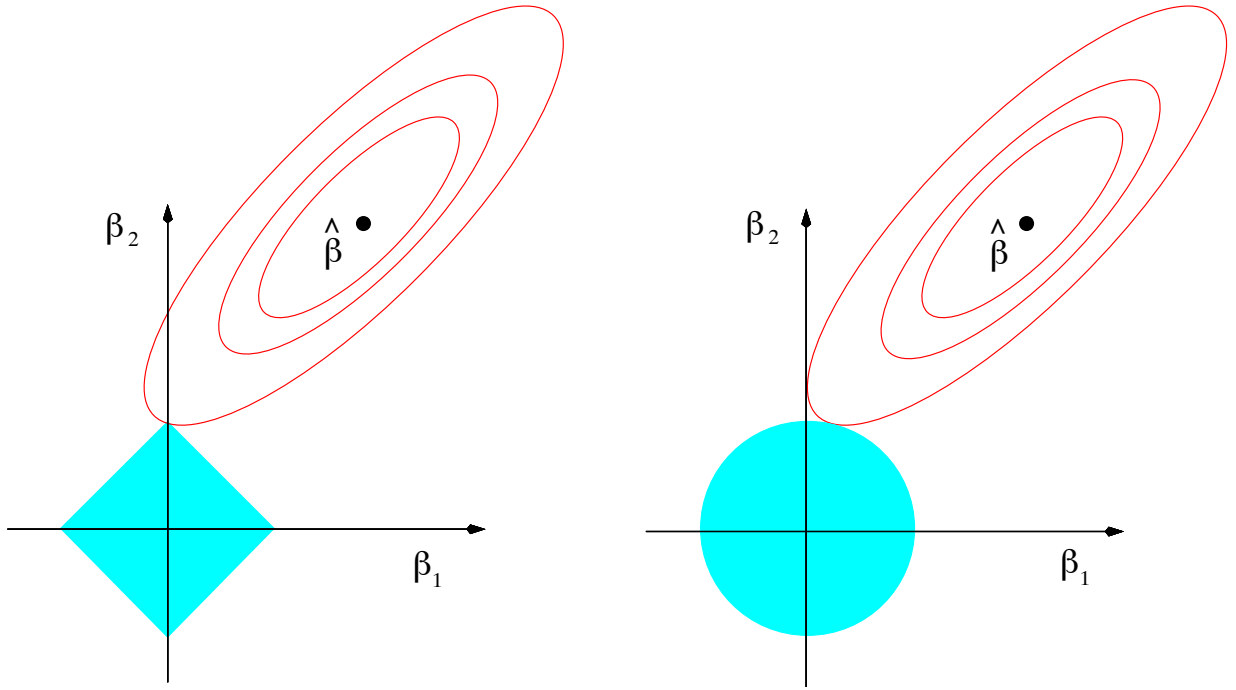
Linear regression via the Lasso (Tibshirani, 1995)

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

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

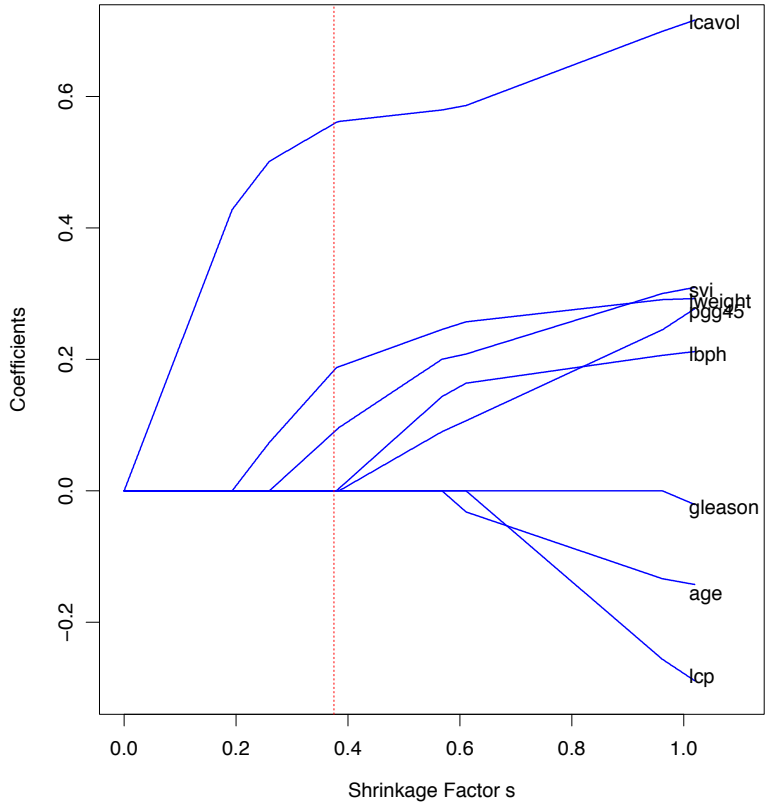
- Equivalent to minimizing sum of squares with constraint $\sum |\beta_j| \leq s$.
- Similar to **ridge regression**, which has constraint $\sum_j \beta_j^2 \leq t$
- Lasso does variable selection and shrinkage; ridge only shrinks.
- See also "Basis Pursuit" (Chen, Donoho and Saunders, 1998).

Picture of Lasso and Ridge regression



Example: Prostate Cancer Data

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



Emerging themes

- Lasso (ℓ_1) penalties have powerful **statistical** and **computational** advantages
- ℓ_1 penalties provide a natural to encourage/enforce sparsity and simplicity in the solution.
- **"Bet on sparsity principle"** (In the *Elements of Statistical learning*). Assume that the underlying truth is sparse and use an ℓ_1 penalty to try to recover it. If you're right, you will do well. If you're wrong— the underlying truth is not sparse—, then no method can do well. [Bickel, Buhlmann, Candes, Donoho, Johnstone,Yu ...]
- ℓ_1 penalties are convex and the assumed sparsity can lead to significant computational advantages

Outline

- New fast algorithm for lasso- Pathwise coordinate descent
- Three examples of applications/generalizations of the lasso:
  - Logistic/multinomial for classification. Example later of classification from microarray data
  - **Near-isotonic regression** - a modern take on an old idea
  - The matrix completion problem
- Not covering: **sparse multivariate methods- Principal components, canonical correlation, clustering** (Daniela Witten's thesis). Google 'Daniela Witten' $->$ "Penalized matrix decomposition"

Algorithms for the lasso

- Standard convex optimizer
- Least angle regression (LAR) - Efron et al 2004- computes entire path of solutions. State-of-the-Art until 2008
- Pathwise coordinate descent- new

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

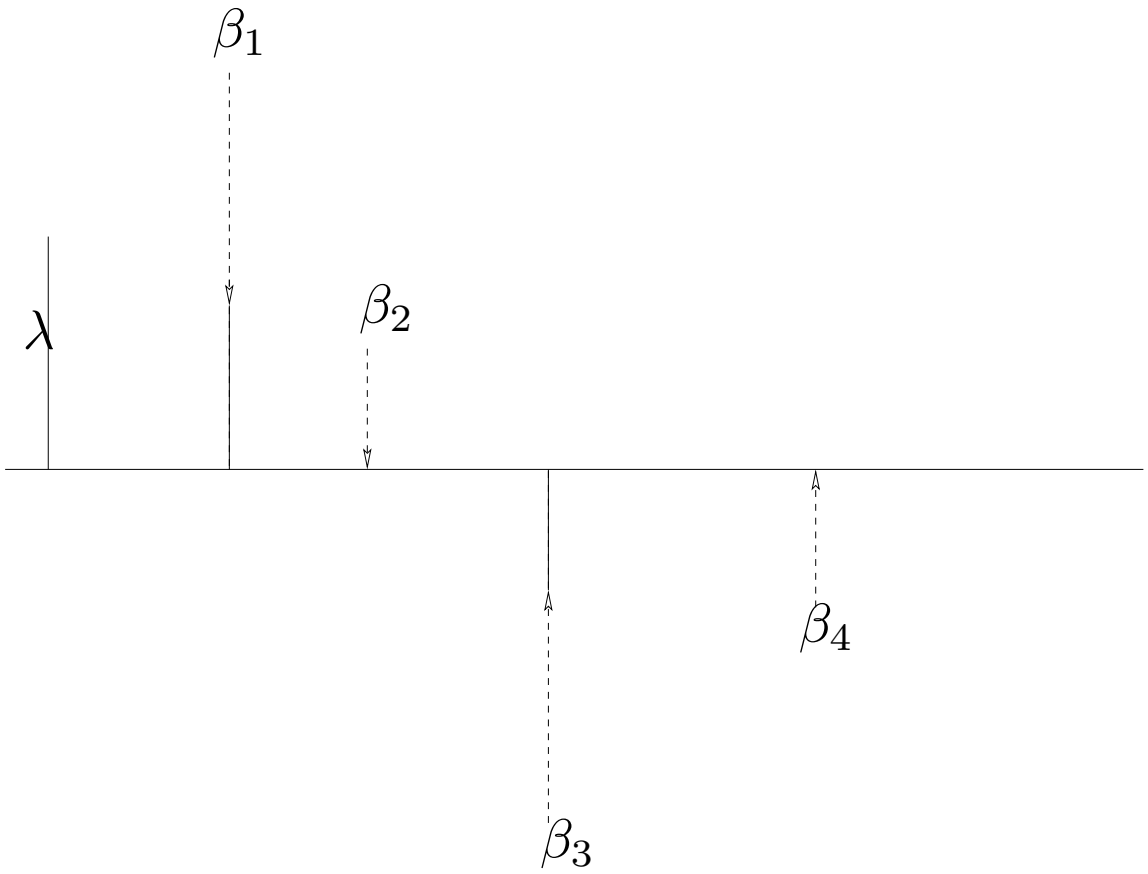
$$\sum_i (y_i - x_i\beta)^2 + \lambda|\beta|$$

- Solution is the soft-thresholded estimate

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

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

Soft-thresholding



- 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. [Show movie]
- **Too simple?!**

A brief history of coordinate descent for the lasso

- 1997: Tibshirani's student Wenjiang Fu at University of Toronto develops the "shooting algorithm" for the lasso. Tibshirani doesn't fully appreciate it
- 2002 Ingrid Daubechies gives a talk at Stanford, describes a one-at-a-time algorithm for the lasso. Hastie implements it, makes an error, and Hastie +Tibshirani conclude that the method doesn't work
- 2006: Friedman is the external examiner at the PhD oral of Anita van der Kooij (Leiden) who uses the coordinate descent idea for the Elastic net. Friedman wonders whether it works for the lasso. Friedman, Hastie + Tibshirani start working on this problem. See also Wu and Lange (2008)!

Pathwise coordinate descent for the lasso

- Start with large value for λ (very sparse model) and slowly decrease it
- most coordinates that are zero never become non-zero
- **coordinate descent code for Lasso is just 73 lines of Fortran!**

Extensions

- Pathwise coordinate descent can be generalized to many other models: logistic/multinomial for classification, graphical lasso for undirected graphs, fused lasso for signals.
- Its speed and simplicity are quite remarkable.
- `glmnet` R package now available on CRAN

Logistic regression

- Outcome $Y = 0$ or 1; Logistic regression model

$$\log(\frac{Pr(Y=1)}{1-Pr(Y=1)}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots$$

- Criterion is binomial log-likelihood +absolute value penalty
- Example: sparse data. $N = 50,000$, $p = 700,000$.
- State-of-the-art interior point algorithm (Stephen Boyd, Stanford), exploiting sparsity of features : **3.5 hours** for 100 values along path

Logistic regression

- Outcome $Y = 0$ or 1; Logistic regression model

$$\log(\frac{Pr(Y=1)}{1-Pr(Y=1)}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots$$

- Criterion is binomial log-likelihood +absolute value penalty
- Example: sparse data. $N = 50,000$, $p = 700,000$.
- State-of-the-art interior point algorithm (Stephen Boyd, Stanford), exploiting sparsity of features : **3.5 hours** for 100 values along path
- Pathwise coordinate descent: **1 minute**

Multiclass classification

Microarray classification: 16,000 genes, 144 training samples 54 test samples, 14 cancer classes. Multinomial regression model.

Methods	CV errors out of 144	Test errors out of 54	# of genes used
1. Nearest shrunken centroids	35 (5)	17	6520
2. L_2-penalized discriminant analysis	25 (4.1)	12	16063
3. Support vector classifier	26 (4.2)	14	16063
4. Lasso regression (one vs all)	30.7 (1.8)	12.5	1429
5. K-nearest neighbors	41 (4.6)	26	16063
6. L_2-penalized multinomial	26 (4.2)	15	16063
7. **Lasso-penalized multinomial**	17 (2.8)	13	269
8. Elastic-net penalized multinomial	22 (3.7)	11.8	384

Near Isotonic regression

Ryan Tibshirani, Holger Hoefling, Rob Tibshirani (2010)

- generalization of isotonic regression: data sequence $y_1, y_2, \ldots y_n$.

$$\text{minimize} \sum (y_i - \hat{y}_i)^2 \text{ subject to } \hat{y}_1 \leq \hat{y}_2 \ldots$$

Solved by Pool Adjacent Violators algorithm.

- Near-isotonic regression:

$$\beta_\lambda = \text{argmin}_{\beta \in \mathcal{R}^n} \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_i)^2 + \lambda \sum_{i=1}^{n-1} (\beta_i - \beta_{i+1})_+,$$

with x_+ indicating the positive part, $x_+ = x \cdot 1(x > 0)$.

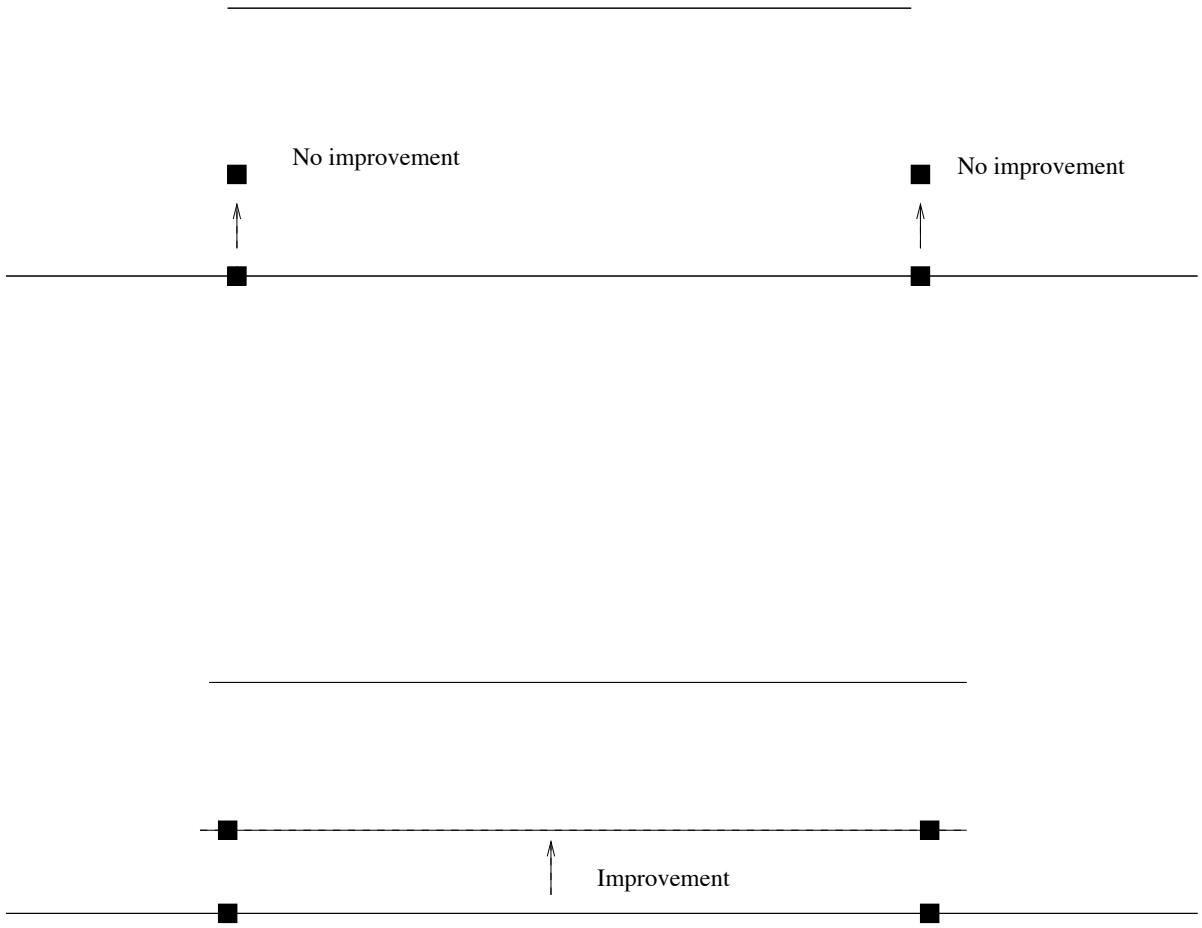
Near-isotonic regression- continued

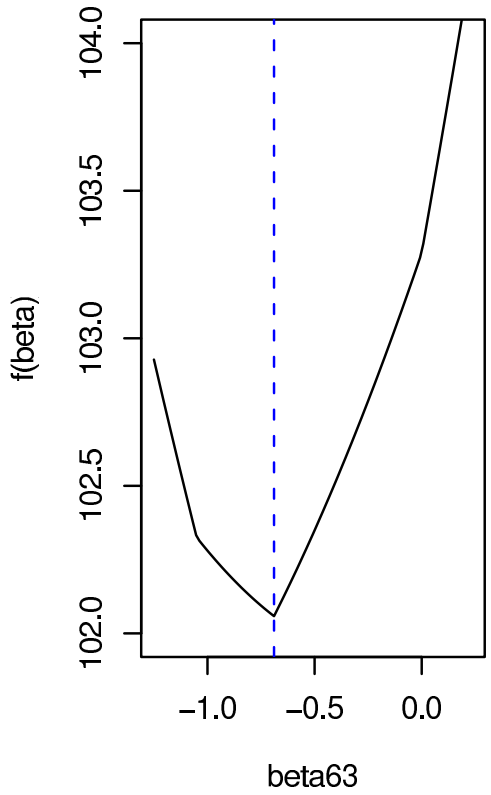
- Convex problem. Solution path $\hat{\beta}_i = y_i$ at $\lambda = 0$ and culminates in usual isotonic regression as $\lambda \to \infty$. Along the way gives **near monotone** approximations.

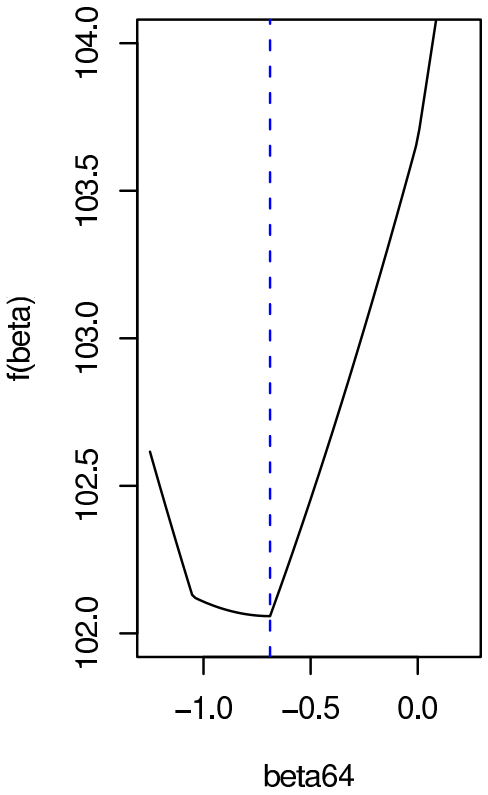
Numerical approach

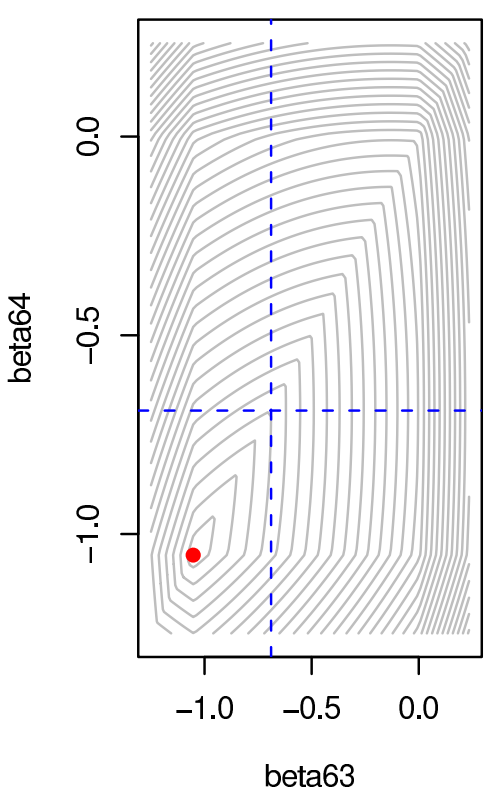
How about using coordinate descent?

- **Surprise!** Although criterion is convex, it is not differentiable, and coordinate descent can get stuck in the "cusps"


No improvement
No improvement
Improvement


f(beta)
102.0
102.5
103.0
103.5
104.0
−1.0
−0.5
0.0
beta63


f(beta)
102.0
102.5
103.0
103.5
104.0
−1.0
−0.5
0.0
beta64


beta64
0.0
−0.5
−1.0
−1.0
−0.5
0.0
beta63

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.

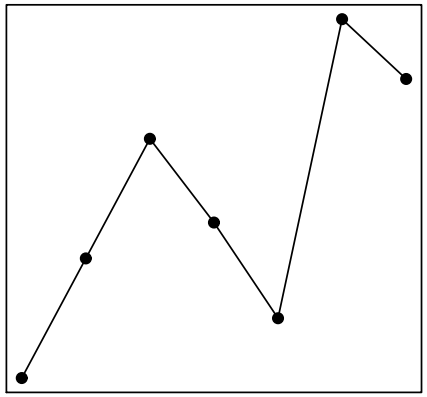
Non-differential part of loss function must be separable

Solution: devise a path algorithm

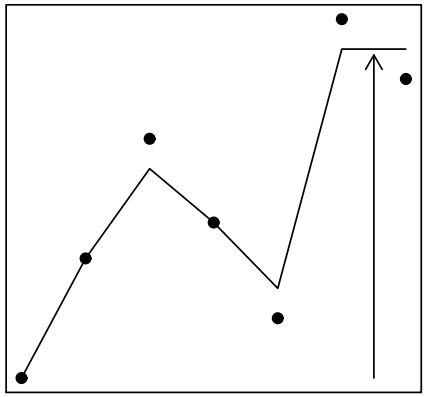
- Simple algorithm that computes the entire path of solutions, a modified version of the well-known **pool adjacent violators**
- Analogous to LARS algorithm for lasso in regression
- Bonus: we show that the degrees of freedom is the number of "plateaus" in the solution. Using results from **Ryan Tibshirani's** PhD work with **Jonathan Taylor**

Toy example

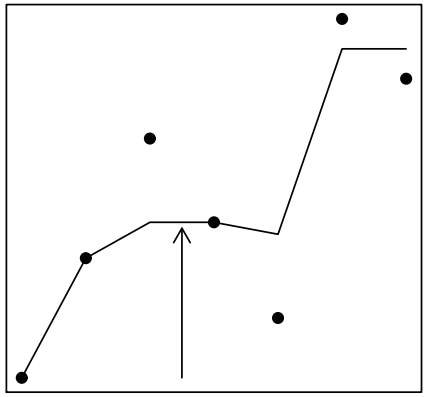
$\lambda = 0$



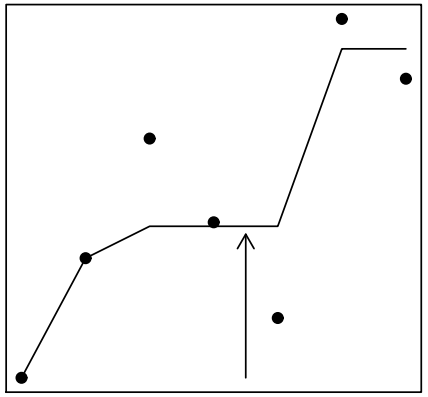
$\lambda = 0.25$



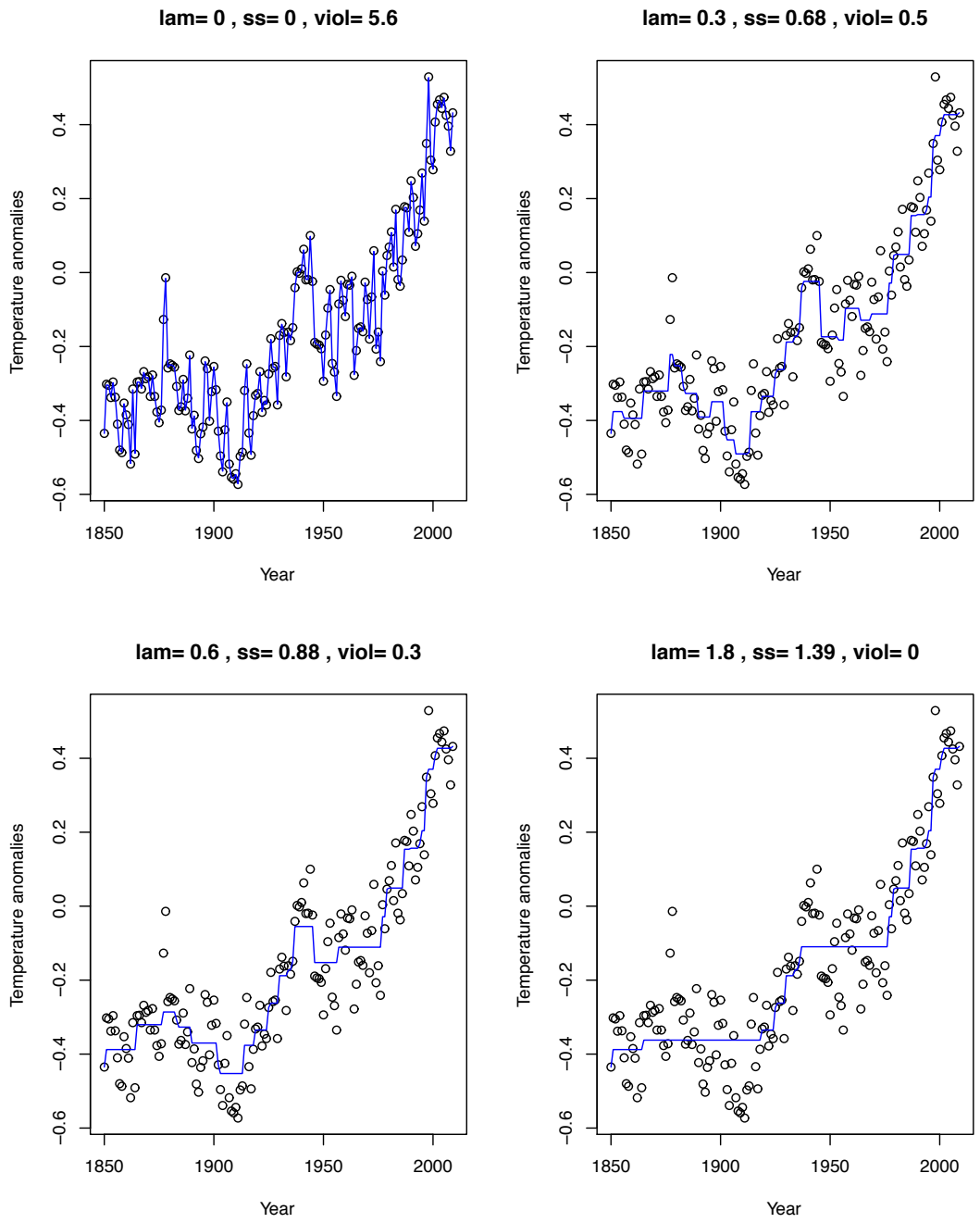
$\lambda = 0.7$



$\lambda = 0.77$



Global warming data


lam= 0 , ss= 0 , viol= 5.6
lam= 0.3 , ss= 0.68 , viol= 0.5
lam= 0.6 , ss= 0.88 , viol= 0.3
lam= 1.8 , ss= 1.39 , viol= 0
Temperature anomalies
Year
−0.6
−0.4
−0.2
0.0
0.2
0.4
1850
1900
1950
2000

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. Eg the Netflix database (see next slide).
- 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:

$$\begin{array}{ll} \text{minimize} & \text{rank}(Z) \\ \text{subject to} & Z_{ij} = X_{ij},\ \forall (i,j) \in \Omega \end{array} \qquad (1)$$

Not convex!

	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 following seemingly small modification to (1)

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

  makes the problem convex [Faz02]. Here $\|Z\|_*$ is the nuclear norm, or the sum of the singular values of Z.

- This criterion is used by [CT09, CCS08, CR08]. Fascinating work! See figure.

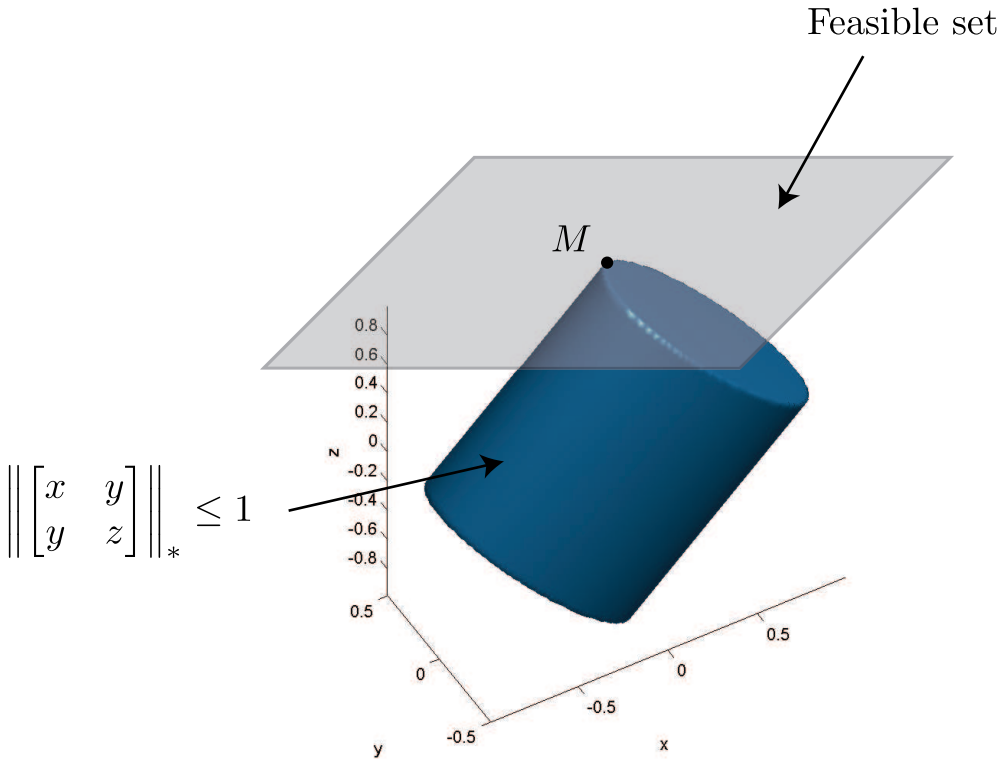
- But this criterion requires the training error to be zero. This is too harsh and can overfit!

- Instead we use the criterion:

$$\begin{aligned} \text{minimize} \quad & \|Z\|_* \\ \text{subject to} \quad & \sum_{(i,j)\in\Omega} (Z_{ij} - X_{ij})^2 \leq \delta \end{aligned} \tag{3}$$

Nuclear norm is like L_1 norm for matrices

Geometry



Idea of Algorithm

1. impute the missing data with some initial values
2. compute the SVD of the current 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

Notation

- Define a matrix $P_\Omega(X)$ (with dimension $n \times m$)

$$P_\Omega(X)\ (i,j) = \begin{cases} X_{ij} & \text{if } (i,j) \in \Omega \\ 0 & \text{if } (i,j) \notin \Omega, \end{cases} \tag{4}$$

  which is a projection of the matrix X onto the observed entries.

- Let

$$\mathbf{S}_\lambda(W) \equiv UD_\lambda V' \quad \text{with} \quad D_\lambda = \text{diag}\left[(d_1 - \lambda)_+, \ldots, (d_r - \lambda)_+\right], \tag{5}$$

  where UDV' is the singular value decomposition of W,

Algorithm

1. Initialize $Z^{\text{old}} = 0$ and create a decreasing grid Λ of values $\lambda_1 > \ldots > \lambda_K$.
2. For every fixed $\lambda = \lambda_1,\ \lambda_2, \ldots \in \Lambda$ iterate till convergence:

   Compute $Z^{\text{new}} \leftarrow \mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z^{\text{old}}))$
3. Output the sequence of solutions $\hat{Z}_{\lambda_1}, \ldots, \hat{Z}_{\lambda_K}$.

It X is sparse, then at each step the non-sparse matrix has the structure:

$$X = X_{SP}\ \text{(Sparse)} \quad + \quad X_{LR}\ \text{(Low Rank)} \tag{6}$$

Can apply Lanczos methods to compute the SVD efficiently.

Properties of Algorithm

We show this iterative algorithm converges to the solution to

$$\underset{Z}{\text{minimize}} \quad \frac{1}{2}\|P_\Omega(X) - P_\Omega(Z)\|_F^2 + \lambda\|Z\|_*. \tag{7}$$

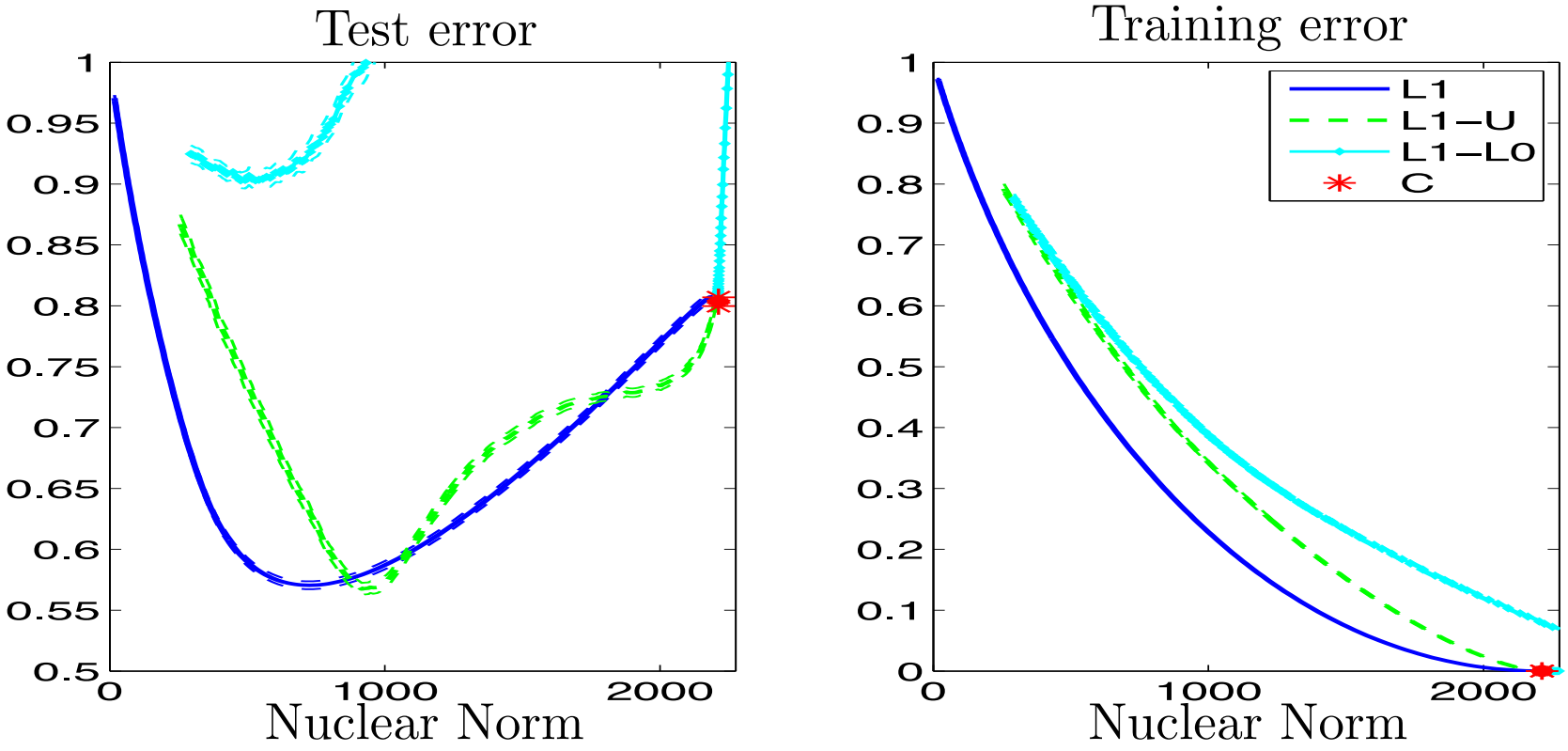
which is equivalent to the bound version (3),

Timings

(m, n)	$\lvert\Omega\rvert$	true rank	SNR	effective rank	time(s)
$(3 \times 10^4, 10^4)$	10^4	15	1	$(13, 47, 80)$	$(41.9, 124.7, 305.8)$
$(10^5, 10^5)$	10^4	15	10	$(5, 14, 32, 62)$	$(37, 74.5, 199.8, 653)$
$(10^5, 10^5)$	10^5	15	10	$(18, 80)$	$(202, 1840)$
$(5 \times 10^5, 5 \times 10^5)$	10^4	15	10	11	628.14
$(5 \times 10^5, 5 \times 10^5)$	10^5	15	1	$(3, 11, 52)$	$(341.9, 823.4, 4810.75)$
$(10^6, 10^6)$	10^5	15	1	80	8906

Accuracy

50% missing entries with SNR=1, true rank =10



Discussion

- lasso penalties are useful for fitting a wide variety of models to large datasets; pathwise coordinate descent enables to fit these models to large datasets for the first time
- In CRAN: coordinate descent in R: **glmnet**- linear regression, logistic, multinomial, Cox model, Poisson
- Also: LARS, nearIso, cghFLasso, glasso
- Matlab software for glm.net and matrix completion
  http://www-stat.stanford.edu/∼ tibs/glmnet-matlab/
  http://www-stat.stanford.edu/∼rahulm/SoftShrink

Ongoing work in lasso/sparsity

- grouped lasso (Yuan and Lin) and many variations (Peng, Zhu...Wang "RemMap")
- multivariate- principal components, canonical correlation, clustering (Witten and others)
- matrix-variate normal (Genevera Allen)
- graphical models, graphical lasso (Yuan+Lin, Friedman, Hastie+Tibs, Peng, Wang et al- "SPACE")
- Compressed sensing (Candes and co-authors)
- "Strong rules" (Tibs et al 2010) provide a 5-80 fold speedup in computation, with no loss in accuracy

Some challenges

- develop tools and theory that allow these methods to be used in statistical practice: standard errors, p-values and confidence intervals that account for the adaptive nature of the estimation.
- while it's fun to develop these methods, as statisticians, our ultimate goal is to provide better answers to scientific questions

References

[CCS08] Jian-Feng Cai, Emmanuel J. Candes, and Zuowei Shen. A singular value thresholding algorithm for matrix completion, 2008.

[CR08] Emmanuel Candès and Benjamin Recht. Exact matrix completion via convex optimization. **Foundations of Computational Mathematics**, 2008.

[CT09] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion, 2009.

[Faz02] M. Fazel. **Matrix Rank Minimization with Applications**. PhD thesis, Stanford University, 2002.