Fast Stagewise Algorithms for Approximate Regularization Paths

Ryan Tibshirani

Dept. of Statistics
Dept. of Machine Learning
Carnegie Mellon University

Lasso regression

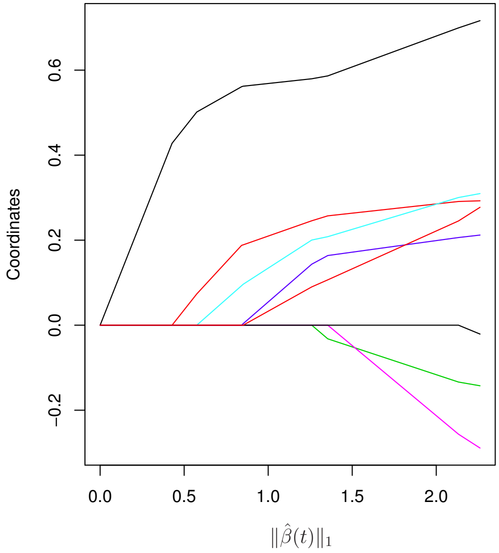
Given outcome $y \in \mathbb{R}^n$, predictors $X \in \mathbb{R}^{n\times p}$, the lasso estimate (Tibshirani, 1996 and others) is:

$$\hat{\beta}(t) \in \underset{\beta\in\mathbb{R}^p}{\arg\min}\ \frac{1}{2}\|y - X\beta\|_2^2 \ \text{ subject to } \ \|\beta\|_1 \le t$$

Regularized least squares estimation, using an ℓ_1 norm constraint (note $\|\beta\|_1 = \sum_{j=1}^p |\beta_j|$). Here t is a tuning parameter, controlling level of regularization

Important property of ℓ_1 norm: induces sparsity in estimate $\hat{\beta}(t)$. Lower t, greater degree of sparsity

Small lasso example with $p = 8$ predictors (diabetes data):



Success of lasso: plentiful, both theoretical and computational

Computational: many fast algorithms, e.g.,

- LARS (Efron et al., 2004)
- Coordinate descent (Friedman et al., 2007)
- ISTA and FISTA (Beck and Teboulle, 2009)
- NESTA (Candes et al., 2010)
- ADMM (Boyd et al., 2010)
- and at least 5 more ...

Said once a famous optimization guru:

"The world doesn't need another lasso algorithm"

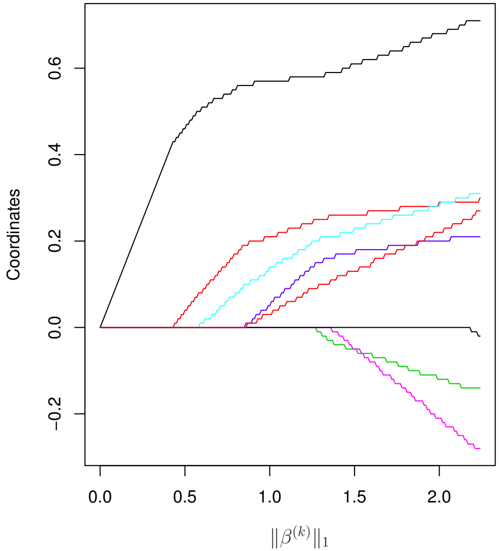
Forward stagewise regression

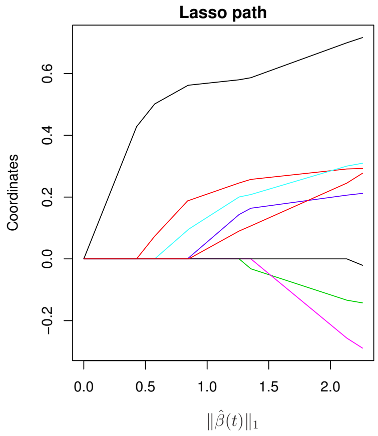
Compare now forward stagewise regression, a very simple iterative approach to regularized estimation:

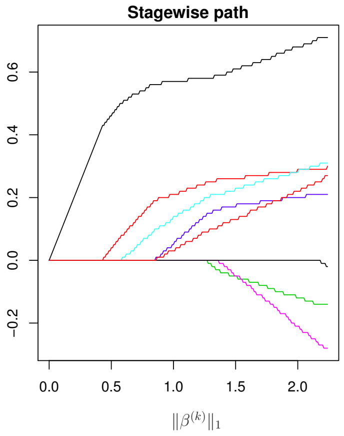
- Start with $\beta^{(0)} = 0$
- Repeat, for $k = 1, 2, 3, \ldots$
  1. Find i such that
  $$\left|X_i^T(y - X\beta^{(k-1)})\right| = \max_{j=1,\ldots p} \left|X_j^T(y - X\beta^{(k-1)})\right|$$
  2. Update
  $$\beta^{(k)} = \beta^{(k-1)} + \epsilon \cdot \mathrm{sign}\big(X_i^T(y - X\beta^{(k-1)})\big) \cdot e_i$$

Here $\epsilon > 0$ is a fixed (small) constant. This is like forward stepwise regression, just less greedy

Same example, now fit with forward stagewise (and $\epsilon = 0.1$):







$$\hat\beta(t) = \underset{\beta\in\mathbb{R}^p}{\arg\min}\ \frac{1}{2}\|y - X\beta\|_2^2$$
$$\text{subject to } \|\beta\|_1 \le t$$

$$i = \underset{j=1,\ldots p}{\arg\max}\ \left|X_j^T(y - X\beta^{(k-1)})\right|$$
$$\beta^{(k)} = \beta^{(k-1)} + \epsilon\cdot\mathrm{sign}\big(X_i^T(y - X\beta^{(k-1)})\big)\cdot e_i$$

Why are these so similar?

A short history:

- First noticed in Hastie et al. (2001)
- Efron et al. (2004) proved that the stagewise paths converge (locally) to the lasso paths, as $\epsilon \to 0$, assuming monotonicity
- Rosset et al. (2004) extended this result to ℓ_1 constrained minimization with any convex, differentiable loss function
- Zhao and Yu (2007) showed that with forward and backward steps, stagewise paths always converge to lasso paths

These results mostly focus on mathematical details and offer little intuition ... so intuitively, why are these so similar?

Forward stagewise, revisited

Let $f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$ denote the least squares loss. Note that $\nabla_j f(\beta) = -X_j^T(y - X\beta)$, hence stagewise updates are:

$$\beta^{(k)} = \beta^{(k-1)} - \epsilon \cdot \mathrm{sign}\big(\nabla_i f(\beta^{(k-1)})\big) \cdot e_i$$

$$\text{where } \big|\nabla_i f(\beta^{(k-1)})\big| = \max_{j=1,\dots p} \big|\nabla_j f(\beta^{(k-1)})\big|$$

Equivalently, forward stagewise updates are:

$$\boxed{\begin{gathered}\beta^{(k)} = \beta^{(k-1)} + \Delta \\ \text{where } \Delta \in \mathop{\mathrm{arg\,min}}_{z \in \mathbb{R}^p} \langle \nabla f(\beta^{(k-1)}), z\rangle \ \text{ subject to } \ \|z\|_1 \le \epsilon\end{gathered}}$$

This is just like steepest descent with respect to the ℓ_1 norm!

Regularization beyond the ℓ_1 norm

Lots of interesting regularization schemes, beyond ℓ_1 norm and pure sparsity:

- *Generalized lasso regularization*—e.g., fused lasso, trend filtering, image/graph denoising
- *Group structured regularization*—e.g., group lasso, multitask learning
- *Spectral regularization*—e.g., matrix completion, reduced rank regression or classification
- *Hierarchical regularization*—e.g., hierNet, CAP, others

Generally speaking, computation with such regularizers is more difficult than it is with the ℓ_1 norm.

Stagewise for general convex problems

Consider generic regularization problem:

$$\hat{\beta}(t) \in \underset{\beta \in \mathbb{R}^p}{\arg\min}\ f(\beta) \ \text{ subject to } \ g(\beta) \leq t$$

Loss f is convex, differentiable; regularizer g is convex

Start with $\beta^{(0)} = \hat{\beta}(t_0)$, solution at some value t_0. Inspired by last stagewise formulation, consider repeating, for $k = 1, 2, 3, \ldots$,

$$\boxed{\begin{gathered} \beta^{(k)} = \beta^{(k-1)} + \Delta \\ \text{where } \Delta \in \underset{z \in \mathbb{R}^p}{\arg\min}\ \langle \nabla f(\beta^{(k-1)}), z \rangle \ \text{ subject to } \ g(z) \leq \epsilon \end{gathered}}$$

This is just like steepest descent with respect to g!

"Luckily" the stagewise update

$$\Delta \in \operatorname*{arg\,min}_{z \in \mathbb{R}^p} \langle \nabla f(\beta^{(k-1)}), z\rangle \;\text{ subject to }\; g(z) \le \epsilon$$

is easy to compute for many statistical learning problems, including most of those mentioned previously

Note that the difficulty in computing Δ depends entirely on regularizer g and not on loss f (assuming ∇f can be evaluated)

If g is a norm (or seminorm), then

$$\begin{aligned}\Delta &\in -\epsilon \cdot \Big(\operatorname*{arg\,max}_{z \in \mathbb{R}^p} \langle \nabla f(\beta^{(k-1)}), z\rangle \;\text{ subject to }\; g(z) \le 1 \Big) \\ &= -\epsilon \cdot \partial g^*\big(\nabla f(\beta^{(k-1)})\big)\end{aligned}$$

where g^* is the dual norm (or dual seminorm), and $\partial g^*(x)$ denotes its subdifferential at a point x

Related work

Statistics/ML focused:

- All of the work on stagewise and lasso mentioned previously: Efron et al. (2004), Rosset et al. (2004), Zhao and Yu (2007)
- Also Hastie et al. (2007), Friedman (2008), Obozinski et al. (2010)

Optimization focused:

- Frank-Wolfe algorithm: Frank and Wolfe (1956), Jaggi (2013)
- Cutting plane and bundle methods: Teo et al. (2007)

The biggest difference with the above optimization work: the proposed stagewise algorithm iterates along the path, rather than iterating for a fixed value of the regularization parameter

Outline

The rest of the talk:

- Group structured regularization
- Trace norm regularization
- Generalized lasso regularization
- Some theory
- Strengths, shortcomings, future work

Group structured regularization

Consider the group structured regularization problem:

$$\hat{\beta}(t) \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \; f(\beta) \;\; \text{subject to} \;\; \sum_{j=1}^{G} w_j \|\beta_{\mathcal{I}_j}\|_2 \leq t$$

Here $\mathcal{I}_1, \ldots \mathcal{I}_G$ is a partition of $\{1, \ldots p\}$, and $w_1, \ldots w_G > 0$ are fixed weights

Think of, e.g.,

- Gaussian group lasso: $f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$
- logistic group lasso:
  $f(\beta) = \sum_{i=1}^n \Big[- y_i x_i^T \beta + \log\big(1 + \exp(x_i^T \beta)\big)\Big]$

Stagewise algorithm starts at $t = 0$ and $\beta^{(0)} = 0$. Update rule: $\beta^{(k)} = \beta^{(k-1)} + \Delta$, where

$$\Delta \in \operatorname*{arg\,min}_{z \in \mathbb{R}^p} \ \langle \nabla f(\beta^{(k-1)}), z \rangle \ \text{ subject to } \ \sum_{j=1}^{G} w_j \|z_{\mathcal{I}_j}\|_2 \le \epsilon$$

Abbreviating $\nabla f = \nabla f(\beta^{(k-1)})$, we can show that Δ has the form:

$$\Delta_{\mathcal{I}_i} = \frac{-\epsilon \cdot (\nabla f)_{\mathcal{I}_i}}{w_i \|(\nabla f)_{\mathcal{I}_i}\|_2}$$

$$\Delta_{\mathcal{I}_j} = 0 \quad \text{for all } j \neq i$$

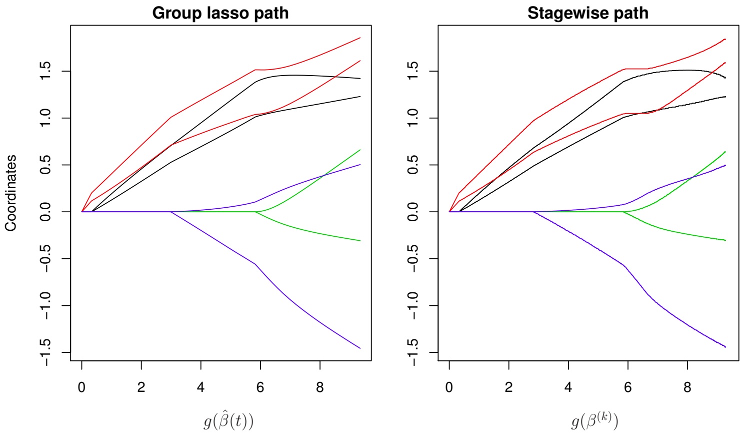
where

$$\frac{\|(\nabla f)_{\mathcal{I}_i}\|_2}{w_i} = \max_{j=1,\ldots G} \frac{\|(\nabla f)_{\mathcal{I}_j}\|_2}{w_j}$$

I.e., update one group at a time, corresponding to the largest block of the gradient (scaled by the weights)

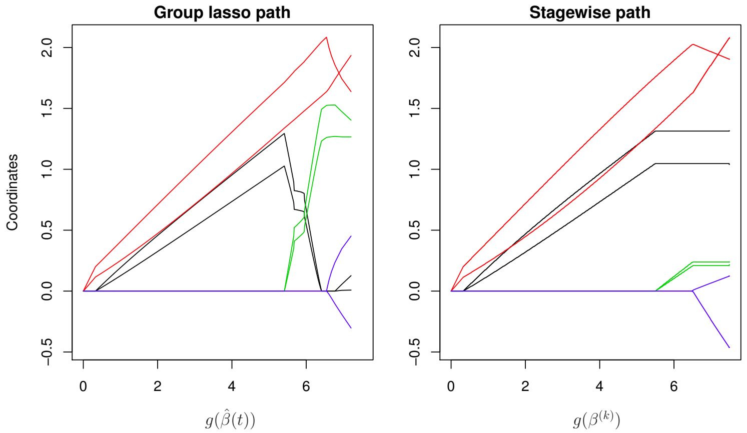
Small group lasso example

Small example with $n = 20$, $p = 8$, and $G = 4$ groups:



Red and black paths correspond to truly active groups, predictor variables are uncorrelated in population

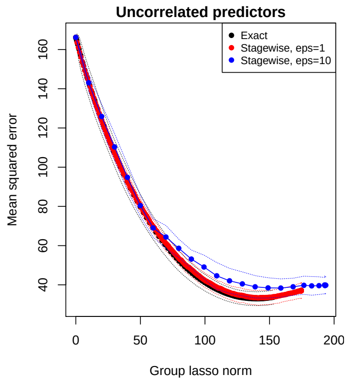
Same setup, but now predictor variables are highly correlated:

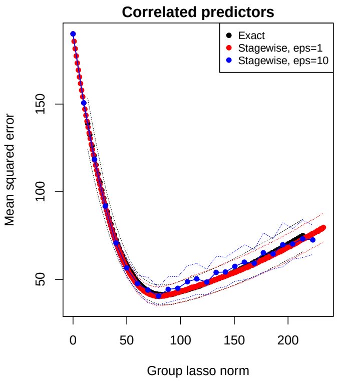


Stagewise paths appear more stable that exact paths ... so how do they compare statistically?

Big group lasso example

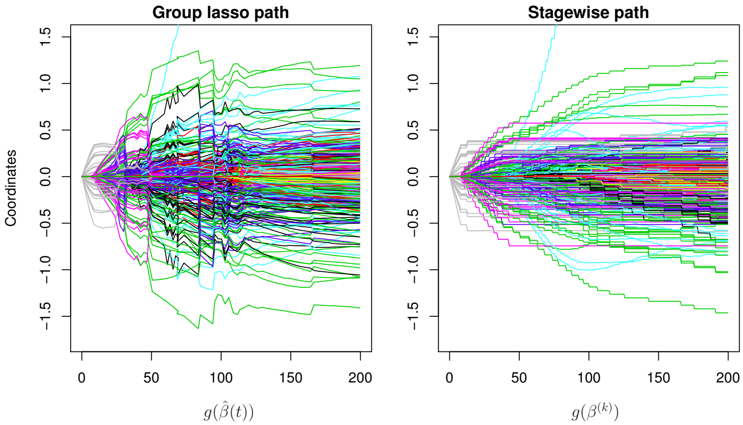
Big example with $n = 200$, $p = 4000$, and $G = 100$ groups:





Algorithm timings (in seconds)		
Method	Uncorrelated case	Correlated case
Exact: coordinate descent, 100 solutions	9.08 (1.06)	78.64 (17.92)
Stagewise: $\epsilon = 1$, 250 estimates	0.93 (0.00)	0.94 (0.01)
Stagewise: $\epsilon = 10$, 25 estimates	0.09 (0.00)	0.10 (0.01)

Coefficient paths for with correlated predictors:



Again, broadly speaking, stability is the key difference here

Trace norm regularization

Consider the trace norm regularization problem:

$$\hat{B}(t) \in \operatorname*{arg\,min}_{B \in \mathbb{R}^{m \times n}} \; f(B) \;\; \text{subject to} \;\; \|B\|_* \leq t$$

Here B is an $m \times n$ matrix and $\|B\|_*$ is the trace norm (or nuclear norm) of B, i.e., the sum of its singular values

Think of, e.g., matrix completion: observe entries of $Y \in \mathbb{R}^{m \times n}$ only for $(i,j) \in \Omega$, and

$$f(B) = \frac{1}{2} \sum_{(i,j) \in \Omega} (Y_{ij} - B_{ij})^2$$

Stagewise algorithm starts at $t = 0$ and $B^{(0)} = 0$. Update rule: $B^{(k)} = B^{(k-1)} + \Delta$, where

$$\Delta \in \underset{Z \in \mathbb{R}^{m \times n}}{\arg\min} \; \langle \nabla f(B^{(k-1)}), Z \rangle \;\; \text{subject to} \;\; \|Z\|_* \leq \epsilon$$

Again we can show that Δ has the explicit form:

$$\Delta = -\epsilon \cdot uv^T$$

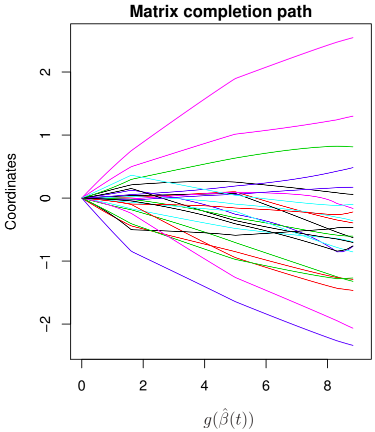
where u and v are the leading left and right singular vectors of $\nabla f(B^{(k-1)})$

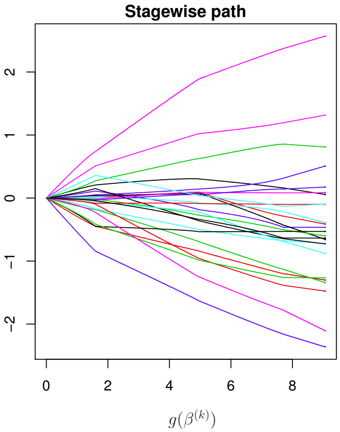
I.e., each update requires only one singular vector computation

Compare proximal gradient descent, which for a single value of the regularization parameter, iterates over (partial) SVD computations

Small matrix completion example

Small example with $Y \in \mathbb{R}^{5\times 5}$, and 50% observed:

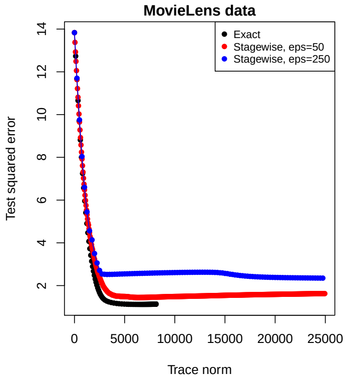




Stagewise paths appear a little different here ... how do they compare statistically?

Big matrix completion example

MovieLens data with $Y \in \mathbb{R}^{943 \times 1682}$, about 6% observed:



Algorithm timings (in seconds)	
Method	MovieLens data
Exact: proximal gradient, 100 solutions	334.67
Stagewise: $\epsilon = 50$, 500 estimates	107.66
Stagewise: $\epsilon = 250$, 100 estimates	21.22

Generalized lasso regularization

Consider the generalized lasso regularization problem:

$$\hat{\beta}(t) \in \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} \; f(\beta) \;\; \text{subject to} \;\; \|D\beta\|_1 \leq t$$

Here $D \in \mathbb{R}^{m \times p}$ is a penalty matrix

Think of, e.g., fused lasso: $f(\beta) = \frac{1}{2}\|y - \beta\|_2^2$ and each row of D is of the form

$$(0, 0, \ldots -1, \ldots 1, \ldots 0)$$

corresponding to an edge in some underlying graph. I.e.,

$$\|D\beta\|_1 = \sum_{(i,j) \in E} |\beta_i - \beta_j|$$

Trouble: the stagewise update rule

$$\Delta \in \operatorname*{arg\,min}_{z \in \mathbb{R}^p} \langle \nabla f(\beta^{(k-1)}), z \rangle \text{ subject to } \|Dz\|_1 \leq \epsilon$$

is not easy to compute

Fix: go to the dual! The dual problem is

$$\hat{u}(t) \in \operatorname*{arg\,min}_{u \in \mathbb{R}^m} f^*(-D^T u) + t \cdot \|u\|_\infty$$

where f^* is the convex conjugate of f

Strategy: compute (an approximate) dual regularization path using stagewise algorithm, then convert to primal regularization path

Note: regularization direction is reversed between primal and dual

Dual stagewise update:

$$\Delta \in \underset{z \in \mathbb{R}^m}{\arg\min} \ \langle -D\nabla f^*(-D^T u^{(k-1)}), z\rangle \ \text{ subject to } \ \|z\|_\infty \le \epsilon$$

It is not hard to show that Δ takes the form:

$$\boxed{\Delta_i = -\epsilon \cdot \begin{cases} 1 & \left[D\nabla f^*(-D^T u^{(k-1)})\right]_i < 0 \\ -1 & \left[D\nabla f^*(-D^T u^{(k-1)})\right]_i > 0 \\ 0 & \left[D\nabla f^*(-D^T u^{(k-1)})\right]_i = 0 \end{cases} \quad \text{for } i = 1, \ldots m}$$

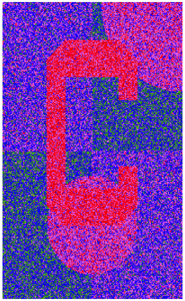
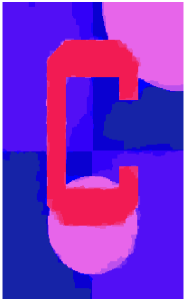
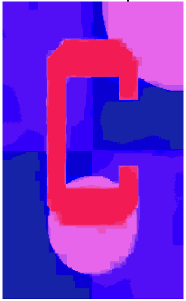
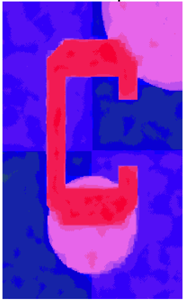
Note: if $f(\beta) = \frac{1}{2}\|y - \beta\|_2^2$, then $f^*(v) = \frac{1}{2}\|y + v\|_2^2$, and

$$-D\nabla f^*(-D^T u) = -D(y - D^T u)$$

Hence calculation of Δ above requires one multiplication by D^T and one multiplication by D

Big image denoising example

Synthetic 300×200 image (graph with 60k nodes, 120k edges):

Noisy image	Exact, $t = 2055.9$	Stagewise, $\epsilon = 0.0005$, 2323 steps	Stagewise, $\epsilon = 0.005$, 211 steps
			

Algorithm timings	
Method	Runtime
Exact: maximum flow, 100 solutions	109.04 (6.21)
Stagewise: $\epsilon = 0.0025$, 6000 estimates	15.11 (0.18)
Stagewise: $\epsilon = 0.25$, 500 estimates	1.26 (0.02)

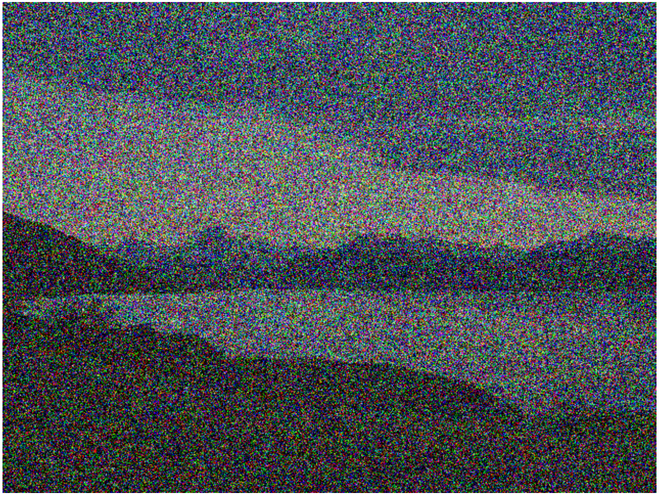
Bigger image denoising example

Real 640×480 image (graph with 307k nodes, 612k edges):

True image of Mount Cook, NZ

Bigger image denoising example

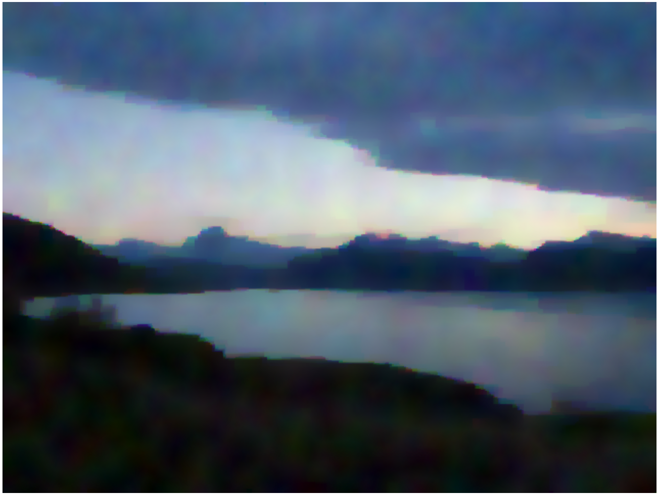
Real 640×480 image (graph with 307k nodes, 612k edges):



Corrupted image

Bigger image denoising example

Real 640×480 image (graph with 307k nodes, 612k edges):



Stagewise estimate, with $\epsilon = 0.001$, 650 steps
(computed in 21.34 seconds)

Some theory

Back to general problem: $\min_{\beta \in \mathbb{R}^p} \ f(\beta) \ \text{subject to} \ g(\beta) \leq t.$

Theorem (Stagewise suboptimality bound)

Assume that g is a seminorm, and ∇f is Lipschitz with respect to the pair g^*, g with constant L. Fix a parameter value t of interest, and run stagewise from $\beta^{(0)} = \hat{\beta}(t_0)$, where $t_0 \leq t$. After k steps, with step size ϵ, such that $t_k = t_0 + k\epsilon = t$, the stagewise estimate $\beta^{(k)}$ satisfies

$$f(\beta^{(k)}) - f(\hat{\beta}(t)) \leq L(t^2 - t_0^2) + L(t - t_0)\epsilon.$$

Therefore, if we consider the limiting stagewise estimate at the parameter value t, by $\tilde{\beta}(t)$, as $\epsilon \to 0$, then

$$f(\tilde{\beta}(t)) - f(\hat{\beta}(t)) \leq L(t^2 - t_0^2).$$

At a high level, what keeps stagewise from optimality? It struggles to undo incremental decisions made in previous steps ... e.g., think of the lasso regularization case

Modification: shrunken stagewise, which repeats for $k = 1, 2, 3, \ldots$,

$$\boxed{\begin{gathered} \beta^{(k)} = \alpha \cdot \beta^{(k-1)} + \Delta \\ \text{where } \Delta \in \operatorname*{arg\,min}_{z \in \mathbb{R}^p} \langle \nabla f(\beta^{(k-1)}), z \rangle \ \text{ subject to } \ g(z) \leq \epsilon \end{gathered}}$$

where $\alpha < 1$ is a shrinkage factor

Theorem (Shrunken stagewise suboptimality bound)

Under same conditions as the previous result, suppose that $\epsilon \to 0$ and $\alpha \to 1$ so that $\frac{1-\alpha}{\epsilon} \to 0$. Then the limiting stagewise estimate $\tilde{x}(t)$ satisfies

$$f(\tilde{x}(t)) - f(\hat{x}(t)) \leq L(t - t_0)^2.$$

Theorem (Shrunken stagewise exact convergence)

Under the same conditions as before, and additionally: $\frac{1-\alpha}{\epsilon^2} \to \infty$. *Let* $k = k(\epsilon, \alpha)$ *denote the number of steps taken by the shrunken stagewise algorithm to reach the parameter value* $t_k = t$. *Define the effective Lagrange parameters* $\lambda_i = g^*(\nabla f(x^{(i)}))$, $i = 1, \dots k$, *and assume that these parameters exhibit a weak type of decay:*

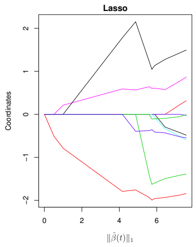
$$\lambda_i / t_i \geq CL, \quad i = 1, \dots r-1,$$
$$\lambda_r / t_r \leq \frac{(C+1)\theta^2 - 2}{2} L,$$

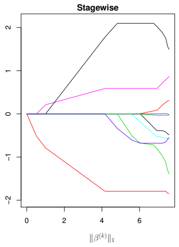
for some $r < k$, *with* $r/k \to \theta \in (0,1)$, *and a constant* C. *Then the limiting shrunken stagewise estimate* $\tilde{x}(t)$ *satisfies*

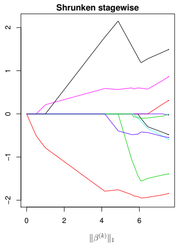
$$f(\tilde{x}(t)) = f(\hat{x}(t)),$$

i.e., $\tilde{x}(t)$ *is an exact solution at the parameter value* t.

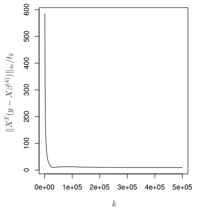
Small lasso example with $n = 20$, $p = 10$:







Observed decay:



Important note: shrunken stagewise is not practically efficient; it is useful conceptual and theoretical bridge between the stagewise and exact solution paths

Strengths, shortcomings, future work

Strengths: simple, efficient algorithm for many problems

Shortcomings:

- Cannot easily handle constraints
- Cannot easily handle a mix of penalties
- For generalized lasso, cannot start at regularized end
- For generalized lasso, cannot handle arbitrary loss functions

Future work:

- Computational improvements: smarter iterations? Efficient data structures?
- Theoretical development: when do stagewise estimates exhibit favorable statistical properties?

Acknowledgements

Thanks to Geoff Gordon and the Convex Optimization (ML 725) class at CMU

Thanks to Rob Tibshirani for helpful conversations

Bonus time

Frank-Wolfe comparison

Let $t^{(k-1)} = g(\beta^{(k-1)})$, and at next parameter value $t = t^{(k-1)} + \epsilon$, consider computing estimate $\beta^{(k)} = \beta^{(k-1)} + z$

Local linear approximation of f around $\beta^{(k-1)}$ (first-order Taylor approximation):

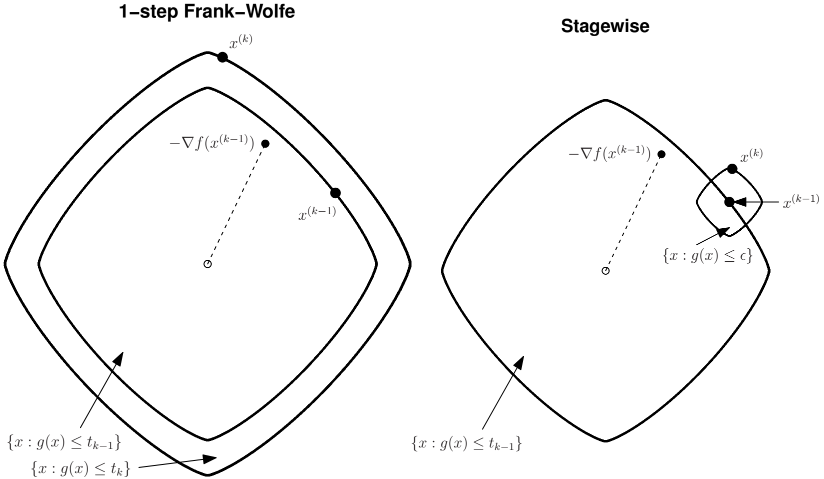
$$f(\beta^{(k-1)} + z) \approx f(\beta^{(k-1)}) + \langle \nabla f(\beta^{(k-1)}), z \rangle$$

Frank-Wolfe

$$\begin{aligned} \underset{z \in \mathbb{R}^p}{\arg\min} \; & \langle \nabla f(\beta^{(k-1)}), z \rangle \\ \text{subject to } \; & g(\beta^{(k-1)} + z) \\ & \qquad \leq t^{(k-1)} + \epsilon \end{aligned}$$

Stagewise

$$\begin{aligned} \underset{z \in \mathbb{R}^p}{\arg\min} \; & \langle \nabla f(\beta^{(k-1)}), z \rangle \\ \text{subject to } \; & g(z) \leq \epsilon \end{aligned}$$


1−step Frank−Wolfe
$x^{(k)}$
$-\nabla f(x^{(k-1)})$
$x^{(k-1)}$
$\{x : g(x) \le t_{k-1}\}$
$\{x : g(x) \le t_k\}$
Stagewise
$-\nabla f(x^{(k-1)})$
$x^{(k)}$
$x^{(k-1)}$
$\{x : g(x) \le \epsilon\}$
$\{x : g(x) \le t_{k-1}\}$

Group regularization under arbitrary norms

Consider general group structured regularization problem:

$$\hat{\beta}(t) \in \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} f(\beta) \;\; \text{subject to} \;\; \sum_{j=1}^{G} w_j h_j(\beta_{\mathcal{I}_j}) \le t$$

Here each h_j is a (semi)norm. Let h_j^* denote its dual (semi)norm; e.g., if $h_j(x) = \|x\|_{q_j}$, then is $h_j^*(x) = \|x\|_{r_j}$, for $1/q_j + 1/r_j = 1$

Stagewise updates are $\beta^{(k)} = \beta^{(k-1)} + \Delta$, and Δ has form:

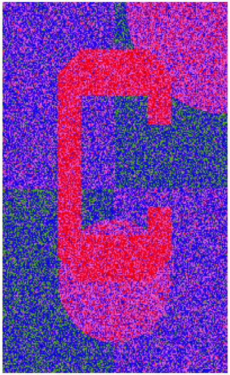
$$\Delta_{\mathcal{I}_i} \in -\frac{\epsilon}{w_i} \cdot \partial h_i^*\big((\nabla f)_{\mathcal{I}_i}\big) \quad \text{and} \quad \Delta_{\mathcal{I}_j} = 0 \quad \text{for all } j \neq i$$

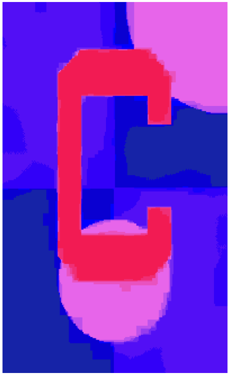
where

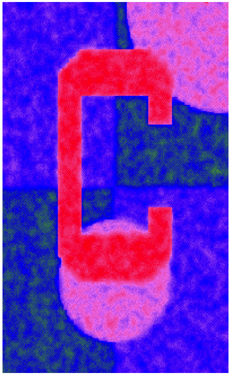
$$\frac{h_i^*\big((\nabla f)_{\mathcal{I}_i}\big)}{w_i} = \max_{j=1,\ldots G} \frac{h_j^*\big((\nabla f)_{\mathcal{I}_j}\big)}{w_j}$$

Self-tuning regularization?

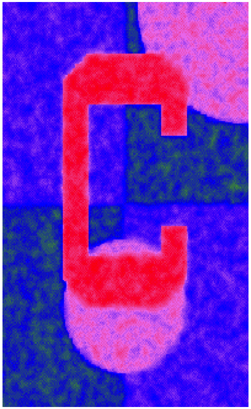
Back to image denoising example with 2d fused lasso; recall we ran 2000 stagewise steps with $\epsilon = 0.0005$. Question: what happens if we take 20 steps with $\epsilon = 0.05$?

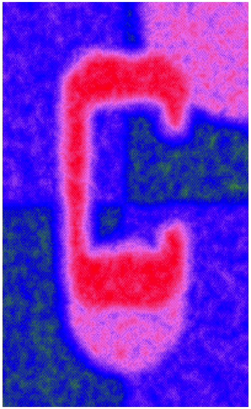

Noisy image


2000 steps, $\epsilon = 0.0005$

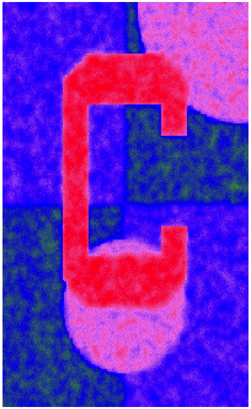

20 steps, $\epsilon = 0.05$

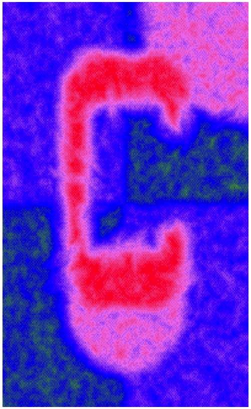
This estimate is really not that bad, considering how cheap it was! Here is where things get interesting:


20 steps, $\epsilon = 0.05$


200 steps, $\epsilon = 0.05$

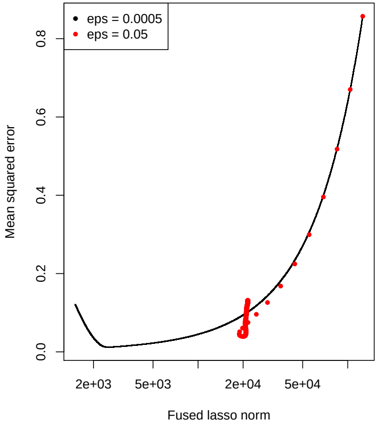
This estimate is really not that bad, considering how cheap it was! Here is where things get interesting:


20 steps, $\epsilon = 0.05$


2000 steps, $\epsilon = 0.05$

Why does the amount of regularization seem to halt, even though we're taking more steps?

Mean squared error to the true underlying image, across steps of the stagewise algorithm:



Something like an implicit termination rule for regularization!