Case Study 3: fMRI Prediction

Fused LASSO
LARS
Parallel LASSO Solvers

Machine Learning for Big Data
CSE547/STAT548, University of Washington

Emily Fox

February 4th, 2014

LASSO Regression

- **LASSO:** least absolute shrinkage and selection operator
- New objective:

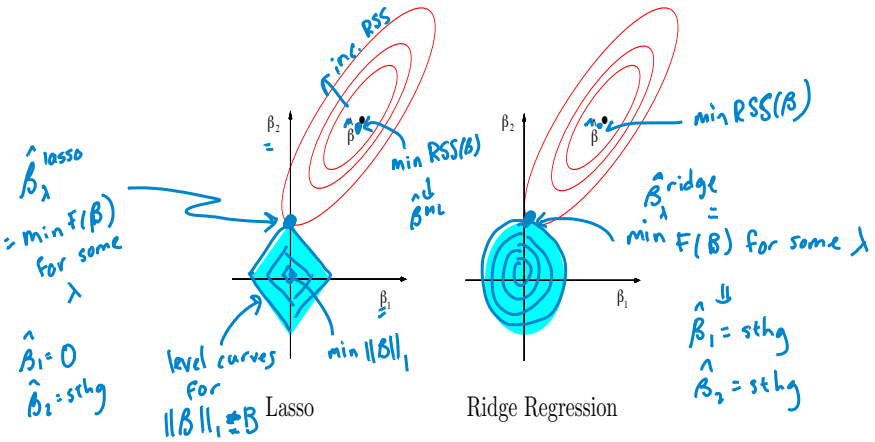
$$\min_{\beta} \sum_{i=1}^{N} \underbrace{\left(y^i - (\beta_0 + \beta^T x^i)\right)^2}_{RSS(\beta)} + \lambda \|\beta\|_1$$

$$\Updownarrow$$

$$\min_{\beta} RSS(\beta) \quad \text{s.t.} \quad \|\beta\|_1 \leq B$$

Geometric Intuition for Sparsity

overall obj: $F(\beta) = RSS(\beta) + \lambda \|\beta\|$ ← 1 or 2 norm; "lasso" "ridge"

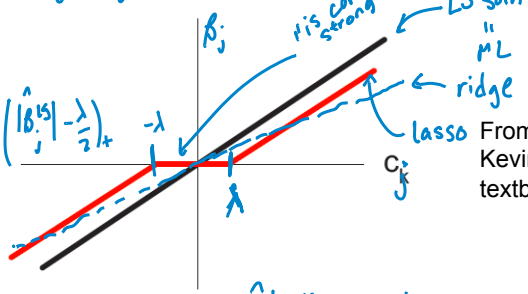


Soft Threshholding

$c_j \propto corr(x_{\cdot j}, r_{-j})$ — all examples of feature j; residual from model w/o using j^{th} covariate

$$\hat{\beta}_j = \begin{cases} (c_j + \lambda)/a_j & c_j < -\lambda \\ 0 & c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & c_j > \lambda \end{cases}$$

$= sign\left(\frac{c_j}{a_j}\right)\left(\frac{|c_j|}{a_j} - \frac{\lambda}{a_j}\right)_+$



From Kevin Murphy textbook

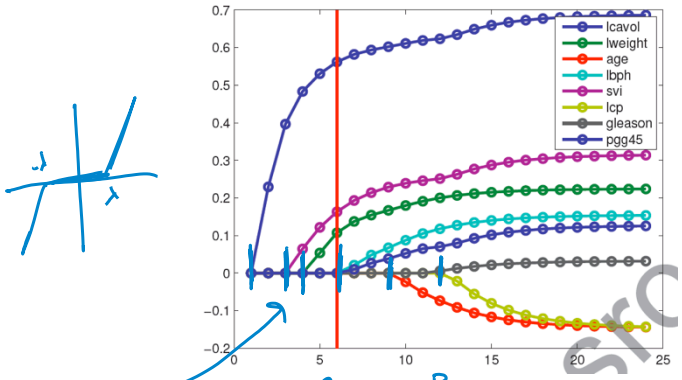
If $X^TX = I$

$\hat{\beta}_j^{lasso} = sign\left(\hat{\beta}_j^{LS}\right)\left(|\hat{\beta}_j^{LS}| - \frac{\lambda}{2}\right)_+$

$\hat{\beta}_j^{ridge} = \frac{\hat{\beta}_j^{LS}}{1+\lambda}$

In LASSO, all coeff. $\hat{\beta}_j^{lasso}$ are shrunk relative to $\hat{\beta}^{LS}$

LASSO Coefficient Path



From Kevin Murphy textbook

Sparsistency

- Typical Statistical Consistency Analysis:
  - Holding model size (p) fixed, as number of samples (N) goes to infinity, estimated parameter goes to true parameter
- Here we want to examine $p >> N$ domains
- Let both model size p and sample size N go to infinity!
  - Hard case: $N = k \log p$

Sparsistency

- Rescale LASSO objective by *N*:

- Theorem (Wainwright 2008, Zhao and Yu 2006, …):
  - Under some constraints on the design matrix *X*, if we solve the LASSO regression using

    Then for some $c_1>0$, the following holds with at least probability

    - The LASSO problem has a unique solution with support contained within the true support
    - If $\min_{j \in S(\beta^*)} |\beta_j^*| > c_2 \lambda_n$ for some $c_2>0$, then $S(\hat{\beta}) = S(\beta^*)$

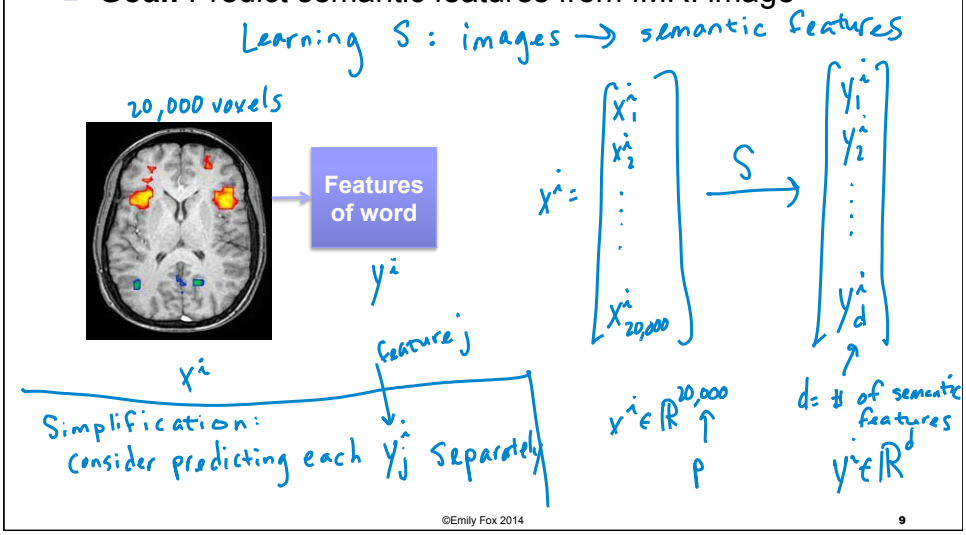
Case Study 3: fMRI Prediction

Fused LASSO

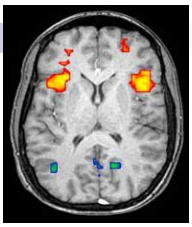
Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
February 4th, 2014

fMRI Prediction Subtask

- **Goal:** Predict semantic features from fMRI image



Fused LASSO



- Might want coefficients of neighboring voxels to be similar
- How to modify LASSO penalty to account for this?
- Graph-guided fused LASSO
  - Assume a 2d lattice graph connecting neighboring pixels in the fMRI image
  - Penalty:

Generalized LASSO

- Assume a structured linear regression model:

- If *D* is invertible, then get a new LASSO problem if we substitute

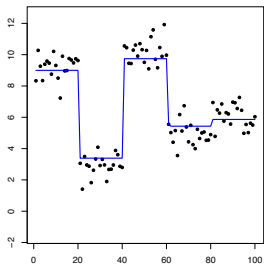
- Otherwise, not equivalent

- For solution path, see
  Ryan Tibshirani and Jonathan Taylor, "The Solution Path of the Generalized Lasso." Annals of Statistics, 2011.

Generalized LASSO

$$\hat{\beta}_\lambda = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{2}\|y - \beta\|_2^2 + \lambda \|D\beta\|_1$$

Let $D = \begin{bmatrix} -1 & 1 & 0 & 0 & \dots \\ 0 & -1 & 1 & 0 & \dots \\ 0 & 0 & -1 & 1 & \dots \\ \vdots & & & & \end{bmatrix}$. This is the 1d fused lasso.



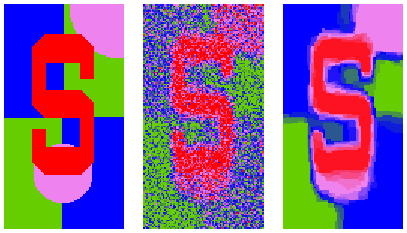
Generalized LASSO

$$\hat{\beta}_\lambda = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{2}\|y - \beta\|_2^2 + \lambda \|D\beta\|_1$$

Suppose D gives "adjacent" differences in β:

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

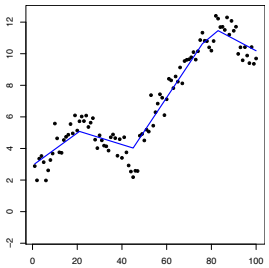
where adjacency is defined according to a graph $\mathcal{G}$. For a 2d grid, this is the 2d fused lasso.



Generalized LASSO

$$\hat{\beta}_\lambda = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{2}\|y - \beta\|_2^2 + \lambda \|D\beta\|_1$$

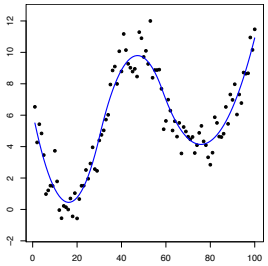
Let $D = \begin{bmatrix} -1 & 2 & -1 & 0 & \ldots \\ 0 & -1 & 2 & -1 & \ldots \\ 0 & 0 & -1 & 2 & \ldots \\ \vdots & & & & \end{bmatrix}$. This is linear trend filtering.



Generalized LASSO

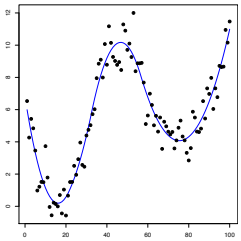
$$\hat{\beta}_\lambda = \underset{\beta \in \mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{2}\|y - \beta\|_2^2 + \lambda \|D\beta\|_1$$

Let $D = \begin{bmatrix} -1 & 3 & -3 & 1 & \dots \\ 0 & -1 & 3 & -3 & \dots \\ 0 & 0 & -1 & 3 & \dots \\ \vdots & & & & \end{bmatrix}$. Get quadratic trend filtering.

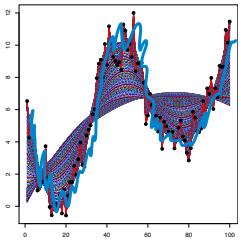


Generalized LASSO

- Tracing out the fits as a function of the regularization parameter



$\hat{\beta}_\lambda$ for $\lambda = 25$



$\hat{\beta}_\lambda$ for $\lambda \in [0, \infty]$

Acknowledgements

- Some material relating to the fused/generalized LASSO slides was provided by Ryan Tibshirani

Case Study 3: fMRI Prediction

LASSO Solvers – Part 1: LARS

Machine Learning for Big Data
CSE547/STAT548, University of Washington

Emily Fox

February 4th, 2014

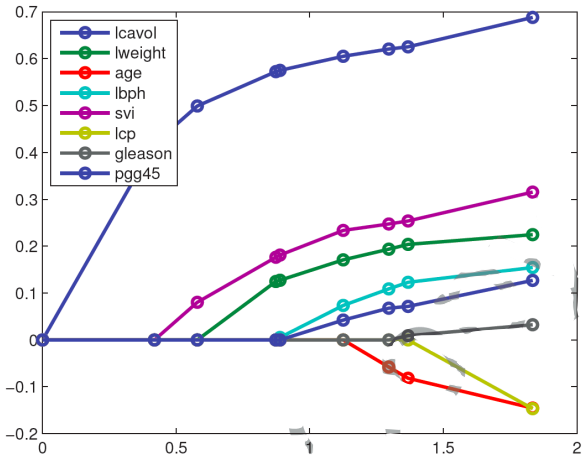
LASSO Algorithms

- Standard convex optimizer
- Now: Least angle regression (LAR)
  - Efron et al. 2004
  - Computes entire path of solutions
  - State-of-the-art until 2008
- Next up:
  - Pathwise coordinate descent ("shooting") – new
  - Parallel (approx.) methods

LARS – Efron et al. 2004

- LAR is an efficient stepwise variable selection algorithm
  - "useful and less greedy version of traditional forward selection methods"
- Can be modified to compute regularization path of LASSO
  - → LARS (Least angle regression and *shrinkage*)
- Increasing upper bound B, coefficients gradually "turn on"
  - Few critical values of B where support changes
  - Non-zero coefficients increase or decrease linearly between critical points
  - Can solve for critical values analytically
- Complexity:

LASSO Coefficient Path



From
Kevin Murphy
textbook

LARS – Algorithm Overview

- Start with all coefficient estimates
- Let $\mathcal{A}$ be the "active set" of covariates most correlated with the "current" residual
- Initially, $\mathcal{A} = \{x_{j_1}\}$ for some covariate x_{j_1}
- Take the largest possible step in the direction of x_{j_1} until another covariate x_{j_2} enters $\mathcal{A}$
- Continue in the direction equiangular between x_{j_1} and x_{j_2} until a third covariate x_{j_3} enters $\mathcal{A}$
- Continue in the direction equiangular between $x_{j_1}, x_{j_2}, x_{j_3}$ until a fourth covariate x_{j_4} enters $\mathcal{A}$
- This procedure continues until all covariates are added at which point

Comments

- LARS increases $\mathcal{A}$, but LASSO allows it to decrease
- Only involves a single index at a time
- If *p* > *N,* LASSO returns at most *N* variables
- If group of variables are highly correlated, LASSO tends to choose one to include rather arbitrarily
  - Straightforward to observe from LARS algorithm….Sensitive to noise.

More Comments

- In general, can't solve analytically for GLM (e.g., logistic reg.)
  - Gradually decrease λ and use efficiency of computing $\hat{\beta}(\lambda_k)$ from $\hat{\beta}(\lambda_{k-1})$ = warm-start strategy
  - See Friedman et al. 2010 for coordinate ascent + warm-starting strategy
- If *N* > *p,* but variables are correlated, ridge regression tends to have better predictive performance than LASSO (Zou & Hastie 2005)
  - Elastic net is hybrid between LASSO and ridge regression

Case Study 3: fMRI Prediction

LASSO Solvers – Part 2:
SCD for LASSO (Shooting)
Parallel SCD (Shotgun)
Parallel SGD
Averaging Solutions

Machine Learning for Big Data
CSE547/STAT548, University of Washington

Emily Fox

February 4th, 2014

Scaling Up LASSO Solvers

- Another way to solve LASSO problem:
  - Stochastic Coordinate Descent (SCD)
  - Minimizing a coordinate in LASSO
- A simple SCD for LASSO (Shooting)
  - Your HW, a more efficient implementation! ☺
  - Analysis of SCD
- Parallel SCD (Shotgun)
- Other parallel learning approaches for linear models
  - Parallel stochastic gradient descent (SGD)
  - Parallel independent solutions then averaging

Coordinate Descent

- Given a function F
  - Want to find minimum
- Often, hard to find minimum for all coordinates, but easy for one coordinate
- Coordinate descent:
- How do we pick a coordinate?
- When does this converge to optimum?

Soft Threshholding

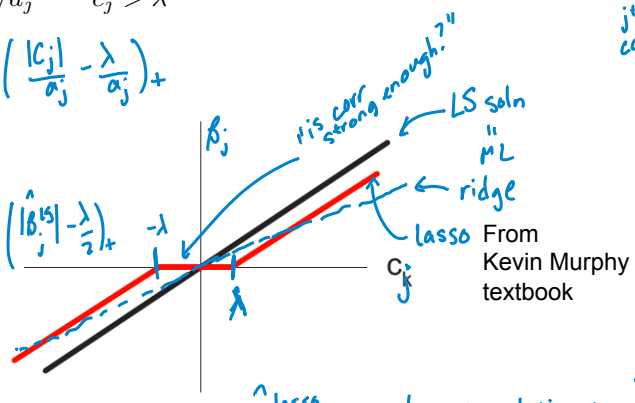
$$\hat{\beta}_j = \begin{cases} (c_j + \lambda)/a_j & c_j < -\lambda \\ 0 & c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & c_j > \lambda \end{cases}$$

$= \text{sign}\left(\frac{c_j}{a_j}\right)\left(\frac{|c_j|}{a_j} - \frac{\lambda}{a_j}\right)_+$

$c_j \propto \text{corr}(x_{:j}, r_{-j})$

all examples of feature j

residual from model w/o using j^{th} covariate



From Kevin Murphy textbook

If $X^T X = I$

$\hat{\beta}_j^{lasso} = \text{sign}\left(\hat{\beta}_j^{LS}\right)\left(|\hat{\beta}_j^{LS}| - \frac{\lambda}{2}\right)_+$

$\hat{\beta}_j^{ridge} = \frac{\hat{\beta}_j^{LS}}{1+\lambda}$

In LASSO, all coeff. $\hat{\beta}_j^{lasso}$ are shrunk relative to $\hat{\beta}^{LS}$

Stochastic Coordinate Descent for LASSO (aka Shooting Algorithm)

- Repeat until convergence
  - Pick a coordinate *j* at random
    - Set:
$$\hat{\beta}_j = \begin{cases} (c_j + \lambda)/a_j & c_j < -\lambda \\ 0 & c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & c_j > \lambda \end{cases}$$
    - Where:
$$a_j = 2\sum_{i=1}^{N}(x_j^i)^2 \qquad c_j = 2\sum_{i=1}^{N} x_j^i(y^i - \beta'_{-j}x^i_{-j})$$

Analysis of SCD [Shalev-Shwartz, Tewari '09/'11]

- Analysis works for LASSO, L1 regularized logistic regression, and other objectives!
- For (coordinate-wise) strongly convex functions:
- Theorem:
  - Starting from
  - After T iterations
  - Where E[ ] is wrt random coordinate choices of SCD
- Natural question: How does SCD & SGD convergence rates differ?

Shooting: Sequential SCD

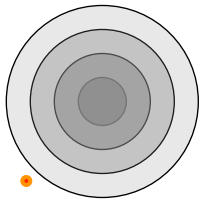
Lasso: $\min_{\beta} F(\beta)$ where $F(\beta) = \| X\beta - \mathbf{y} \|_2^2 + \lambda \| \beta \|_1$

Stochastic Coordinate Descent (SCD)
(e.g., Shalev-Shwartz & Tewari, 2009)

While not converged,

- Choose random coordinate j,
- Update β_j (closed-form minimization)

$F(\beta)$ contour



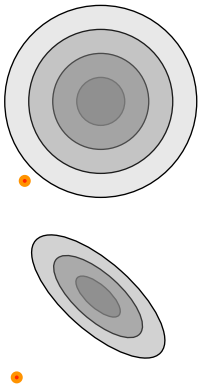
Shotgun: Parallel SCD [Bradley et al '11]

Lasso: $\min_{\beta} F(\beta)$ where $F(\beta) = \| X\beta - \mathbf{y} \|_2^2 + \lambda \| \beta \|_1$

Shotgun (Parallel SCD)

While not converged,

- On each of *P* processors,
  - Choose random coordinate j,
  - Update β_j (same as for Shooting)



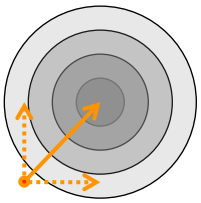
Is SCD inherently sequential?

Lasso: $\min_{\beta} F(\beta)$ where $F(\beta) = \| X\beta - \mathbf{y} \|_2^2 + \lambda \| \beta \|_1$

Coordinate update:

$\beta_j \leftarrow \beta_j + \delta\beta_j$

(closed-form minimization)



Collective update:

$$\Delta\beta = \begin{pmatrix} \delta\beta_i \\ 0 \\ 0 \\ \delta\beta_j \\ 0 \end{pmatrix}$$

Is SCD inherently sequential?

Lasso: $\min_{\beta} F(\beta)$ where $F(\beta) = \| X\beta - \mathbf{y} \|_2^2 + \lambda \| \beta \|_1$

Theorem: If X is normalized s.t. diag($X^T X$)=1,

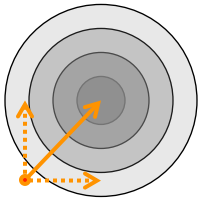
$$F(\beta + \Delta\beta) - F(\beta)$$

$$\leq -\sum_{i_j \in \mathcal{P}} \left(\delta\beta_{i_j}\right)^2 + \sum_{\substack{i_j, i_k \in \mathcal{P}, \\ j \neq k}} \left(X^T X\right)_{i_j, i_k} \delta\beta_{i_j} \delta\beta_{i_k}$$

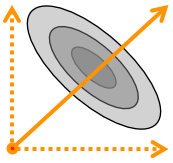
Is SCD inherently sequential?

Theorem: If X is normalized s.t. diag(X^TX)=1,

$$F(\beta+\Delta\beta)-F(\beta) \leq -\sum_{i_j\in\mathcal{P}}\left(\delta\beta_{i_j}\right)^2 + \sum_{\substack{i_j,i_k\in\mathcal{P},\\ j\neq k}}\left(X^{\mathrm{T}}X\right)_{i_j,i_k}\delta\beta_{i_j}\delta\beta_{i_k}$$



Nice case: Uncorrelated features



Bad case: Correlated features

Shotgun: Convergence Analysis

Lasso: $\min_\beta F(\beta)$ where $F(\beta) = \|X\beta-\mathbf{y}\|_2^2 + \lambda\|\beta\|_1$

Assume # parallel updates $P < p/\rho + 1$

Generalizes bounds for Shooting (Shalev-Shwartz & Tewari, 2009)

Convergence Analysis

Lasso: $\min_{\beta} F(\beta)$ where $F(\beta) = \| X\beta - \mathbf{y} \|_2^2 + \lambda \| \beta \|_1$

Theorem: Shotgun Convergence

Assume $P < p/\rho + 1$

where ρ = spectral radius of $\mathbf{X}^T\mathbf{X}$

$$E\left[F(\beta^{(T)})\right] - F(\beta^*) \leq \frac{p\left(\frac{1}{2}\| \beta^* \|_2^2 + F(\beta^{(0)})\right)}{TP}$$

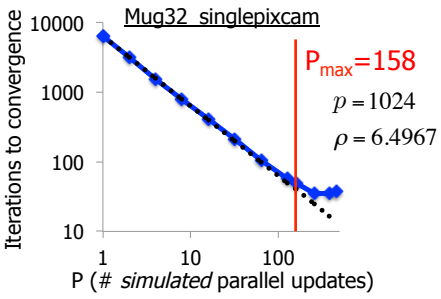
Nice case: Uncorrelated features

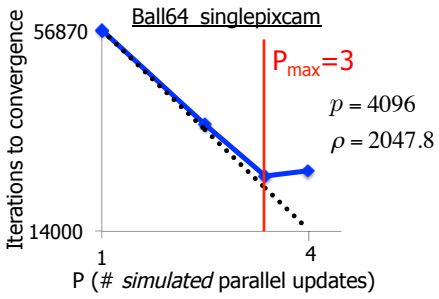
$\rho = \_\_ \Rightarrow P_{\max} = \_\_$

Bad case: Correlated features

$\rho = \_\_ \Rightarrow P_{\max} = \_\_$(at worst)

Empirical Evaluation





Stepping Back…

- Stochastic coordinate ascent
  - Optimization:
  - Parallel SCD:
  - Issue:
  - Solution:
- Natural counterpart:
  - Optimization:
  - Parallel
  - Issue:
  - Solution:

Parallel SGD with No Locks

[e.g., Hogwild!, Niu et al. '11]

- Each processor in parallel:
  - Pick data point i at random
  - For j = 1…*p*:
- Assume atomicity of:

Addressing Interference in Parallel SGD

- Key issues:
  - Old gradients
  - Processors overwrite each other's work
- Nonetheless:
  - Can achieve convergence and some parallel speedups
  - Proof uses weak interactions, but through sparsity of data points

Problem with Parallel SCD and SGD

- Both Parallel SCD & SGD assume access to current estimate of weight vector
- Works well on shared memory machines
- Very difficult to implement efficiently in distributed memory
- Open problem: Good parallel SGD and SCD for distributed setting…
  - Let's look at a trivial approach

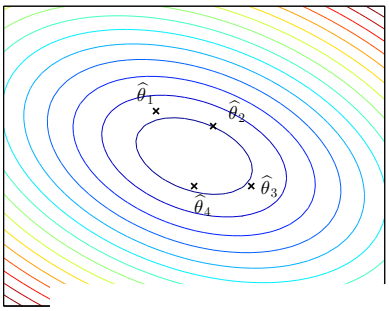
Simplest Distributed Optimization Algorithm Ever Made

- Given *N* data points & *P* machines
- Stochastic optimization problem:
- Distribute data:
- Solve problems independently
- Merge solutions
- Why should this work at all????

For Convex Functions…

- Convexity:
- Thus:

Hopefully…



- Convexity only guarantees:
- But, estimates from independent data!

Figure from John Duchi

Analysis of Distribute-then-Average

[Zhang et al. '12]

- Under some conditions, including strong convexity, lots of smoothness, and more…
- If all data were in one machine, converge at rate:
- With *P* machines, converge at a rate:

Tradeoffs, tradeoffs, tradeoffs,…

- Distribute-then-Average:
  - "Minimum possible" communication
  - Bias term can be a killer with finite data
    - Issue definitely observed in practice
  - Significant issues for L1 problems:

- Parallel SCD or SGD
  - Can have much better convergence in practice for multicore setting
  - Preserves sparsity (especially SCD)
  - But, hard to implement in distributed setting

What you need to know

- Sparsistency
- Fused LASSO
- LASSO Solvers
  - LARS
  - A simple SCD for LASSO (Shooting)
    - Your HW, a more efficient implementation! ☺
    - Analysis of SCD
  - Parallel SCD (Shotgun)
- Other parallel learning approaches for linear models
  - Parallel stochastic gradient descent (SGD)
  - Parallel independent solutions then averaging