Case Study 3: fMRI Prediction

Stochastic Coordinate Descent (SCD) for LASSO (Shooting)
Parallel SCD (Shotgun)
Parallel SGD
Averaging Solutions

Machine Learning/Statistics for Big Data
CSE599C1/STAT592, University of Washington
Carlos Guestrin

February 21st, 2013

Today

- One 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?

LASSO Regression

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

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

$$\Updownarrow$$

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

Soft Threshholding

$F(\beta) = RSS(\beta) + \lambda \|\beta\|_1$

- Gradient of RSS term:

$$\frac{\partial}{\partial \beta_j} RSS(\beta) = a_j \beta_j - c_j$$

$a_j = 2\sum_{i=1}^{N} (x_j^i)^2$

$c_j = 2\sum_{i=1}^{N} x_j^i (y^i - \beta_{-j}^T x_{-j}^i)$

β_{-j}: all but the j^{th} coeff.

$c_j \propto corr(x_j, r_{-j})$

msr how relevant x_j is for pred y beyond what the others can

r_{-j}: residual from model w/o j^{th} cov.

- Subgradient of full objective:

$$\partial_{\beta_j} F(\beta) = (a_j \beta_j - c_j) + \lambda \partial_{\beta_j} \|\beta\|_1$$

$$= \begin{cases} a_j \beta_j - c_j - \lambda & \beta_j < 0 \\ [-c_j - \lambda, -c_j + \lambda] & \beta_j = 0 \\ a_j \beta_j - c_j + \lambda & \beta_j > 0 \end{cases}$$

Soft Threshholding

- Set subgradient = 0:

$$\partial_{\beta_j} F(\beta) = \left\{ \begin{array}{ll} a_j\beta_j - c_j - \lambda & \beta_j < 0 \\ [-c_j - \lambda, -c_j + \lambda] & \beta_j = 0 \\ a_j\beta_j - c_j + \lambda & \beta_j > 0 \end{array} \right. = 0$$

If $\beta_j < 0$

$a_j \beta_j - c_j - \lambda = 0$

$\Rightarrow \beta_j = \frac{c_j + \lambda}{a_j} < 0 \Rightarrow c_j < -\lambda$

strong neg. corr., then $\beta_j < 0$

If $\beta_j > 0$

$a_j \beta_j - c_j + \lambda = 0 \Rightarrow \beta_j = \frac{c_j - \lambda}{a_j} > 0 \Rightarrow c_j > \lambda$

strong pos. corr., then $\beta_j > 0$

If $\beta_j = 0$ $\quad -\lambda < c_j < \lambda$

otherwise, $\beta_j = 0$

- The value of $c_j = 2\sum_{i=1}^{N} x_j^i (y^i - \beta'_{-j} x_{-j}^i)$ constrains β_j

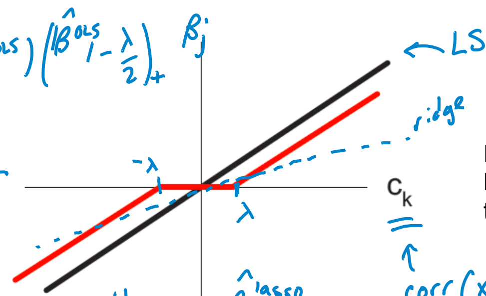
Soft Threshholding

$$\hat{\beta}_j = \begin{cases} (c_j + \lambda)/a_j & c_j < -\lambda \\ 0 & c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & c_j > \lambda \end{cases} = \text{sign}\left(\frac{c_j}{a_j}\right)\left(\frac{|c_j|}{a_j} - \frac{\lambda}{a_j}\right)_+$$

If $X^T X = I$

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

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



$= \uparrow$ corr(x_j, r_{-j})

From
Kevin Murphy
textbook

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

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?

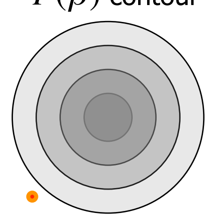
Shooting: Sequential SCD

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

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

While not converged,
- Choose random coordinate j,
- Update β_j (closed-form minimization)

$F(\beta)$ contour



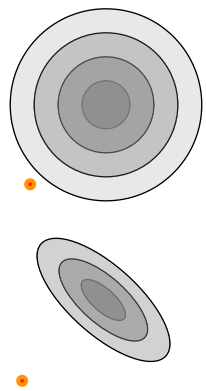
Shotgun: Parallel SCD [Bradley et al '11]

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)



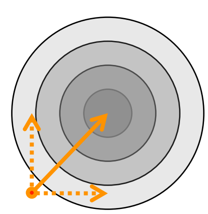
Is SCD inherently sequential?

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

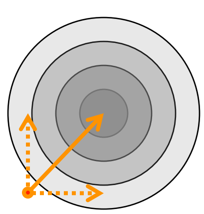
Theorem: If X is normalized s.t. diag($X^T X$)=1,

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

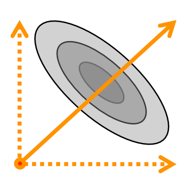
Is SCD inherently sequential?

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

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



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 $P < d/\rho + 1$

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

Convergence Analysis

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

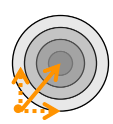
Theorem: Shotgun Convergence

Assume $P < d/\rho + 1$

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

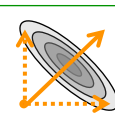
$$E\left[F(\beta^{(T)})\right] - F(\beta^*) \le \frac{d\left(\frac{1}{2}\| \beta^* \|_2^2 + F(\beta^{(0)})\right)}{TP}$$

Nice case:
Uncorrelated
features



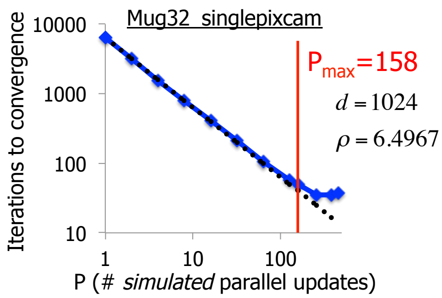
$\rho =$ __ $\Rightarrow P_{\max} =$ __

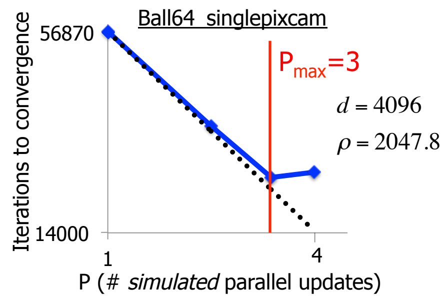
Bad case:
Correlated
features



$\rho =$ __ $\Rightarrow P_{\max} =$ __(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…d:
- 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

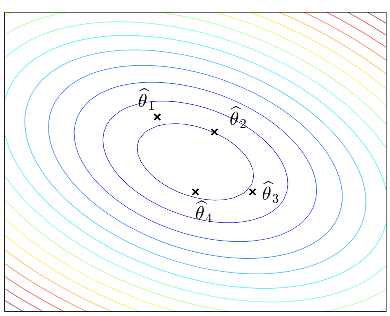
Simplest Distributed Optimization Algorithm Ever Made

- Given *N* data points & *m* 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 *m* 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

- One 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