Case Study 3: fMRI Prediction

"Scalable" LASSO Solvers:
Parallel SCD (Shotgun)
Parallel SGD
Averaging Solutions
ADMM

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

February 6th, 2014

Scaling Up LASSO Solvers

- 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
- ADMM

Stochastic Coordinate Descent for LASSO (aka Shooting Algorithm)

- Repeat until convergence
  - Pick a coordinate *j* at random
    - Set:

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

    - Where:

cache

$$a_j = 2\sum_{i=1}^{N}(x_j^i)^2 \qquad c_j = 2\sum_{i=1}^{N} x_j^i (y^i - \beta'_{-j} x^i_{-j})$$

cost per iteration $O(N)$

Can be done more efficiently. Proof: HW!

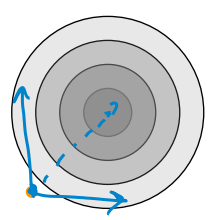
Shotgun: Parallel SCD [Bradley et al '11]

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

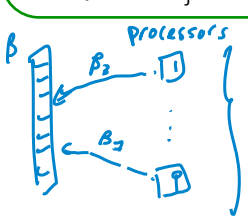
Shotgun (Parallel SCD)

While not converged,

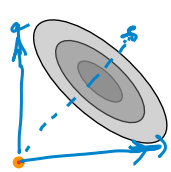
- On each of *P* processors,
  - Choose random coordinate j,
  - Update β_j (same as for Shooting)



yes! Features are uncorrelated



act as if they were the only children



no ☹ Features are highly corr.

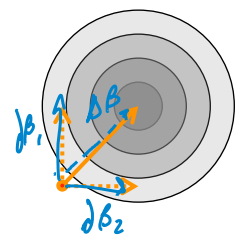
Is SCD inherently sequential?

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:

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

Convergence Analysis

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

Theorem: Shotgun Convergence

Assume $P < p/\rho + 1$

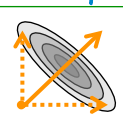
where ρ = spectral radius of $\mathbf{X^T X}$

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

Nice case:
Uncorrelated features

$\rho = 1 \Rightarrow P_{\max} = p$

Bad case:
Correlated features



$\rho = p \Rightarrow P_{\max} = 1$ (at worst)

Stepping Back…

- Stochastic coordinate ascent SCD
  - Optimization: pick a coord. j, find $\min_{\beta_j}$
  - Parallel SCD: pick P coord.
  - Issue: coordinates may interfere on P coord. spectral radius
  - Solution: bound possible interference based ρ
- Natural counterpart: SGD
  - Optimization: pick a datapoint i $\beta \leftarrow \beta - \eta \nabla F(x^i; \beta)$
  - Parallel pick P datapoints + ind. update β
  - Issue: can interfere on all coord.
  - Solution: bound interference

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*:

$$\beta_j \leftarrow \beta_j - \eta \left(\nabla F(x^i; \beta)\right)_j$$

- Assume atomicity of: $\beta_j \leftarrow \beta_j + a$

other interferences

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

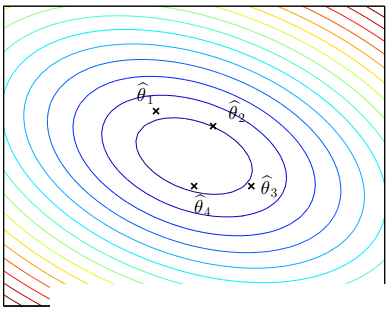
Simplest Distributed Optimization Algorithm Ever Made

- 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

Alternating Directions Method of Multipliers

- A tool for solving convex problems with separable objectives:

- LASSO example:

- Know how to minimize $f(\beta)$ or $g(\beta)$ separately

ADMM Insight

- Try this instead:

- Solve using method of multipliers
- Define the augmented Lagrangian:

  - Issue: L2 penalty destroys separability of Lagrangian
  - Solution: Replace minimization over (x, z) by alternating minimization

ADMM Algorithm

- Augmented Lagrangian:

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(x - z) + \frac{\rho}{2}||x - z||_2^2$$

- Alternate between:

  1. $x \leftarrow$
  2. $z \leftarrow$
  3. $y \leftarrow$

ADMM for LASSO

- Objective: $L_\rho(x,z,y) = f(x) + g(z) + y^T(x-z) + \frac{\rho}{2}||x-z||_2^2$

- Augmented Lagrangian:

$L_\rho(\beta, z, a) =$

- Alternate between:

  1. $\beta \leftarrow$
  2. $z \leftarrow$
  3. $a \leftarrow$

ADMM Wrap-Up

- When does ADMM converge?
  - Under very mild conditions
  - Basically, *f* and *g* must be convex

- ADMM is useful in cases where
  - *f(x)* + *g(x)* is challenging to solve due to coupling
  - We can minimize
    - $f(x) + (x-a)^2$
    - $g(x) + (x-a)^2$

- Reference
  - Boyd, Parikh, Chu, Peleato, Eckstein (2011) "Distributed optimization and statistical learning via the alternating direction method of multipliers." *Foundations and Trends in Machine Learning*, 3(1):1-122.

What you need to know

- 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
- ADMM
  - General idea
  - Application to LASSO