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:
  \[
  \min_{\beta} \sum_{i=1}^{N} (y_i - (\beta_0 + \beta^T x_i))^2 + \lambda \|\beta\|_1
  \]

  \[
  \min_\beta \text{RSS}(\beta) \quad \text{s.t.} \quad \|\beta\|_1 \leq B
  \]
Soft Threshholding

\[ f(\beta) = \text{RSS}(\beta) + \lambda \| \beta \|_1 \]

- Gradient of RSS term:
  \[
  \frac{\partial}{\partial \beta_j} \text{RSS}(\beta) = a_j \beta_j - c_j < 0 \rightarrow \frac{\partial}{\partial \beta_j} \text{RSS}(\beta) = 2 \sum_{i=1}^{N} \left( y_i - \beta_j x_{ij} \right)^2
  \]

- Subgradient of full objective:
  \[
  \frac{\partial}{\partial \beta_j} F(\beta) = (a_j \beta_j - c_j) + \lambda \frac{\partial}{\partial \beta_j} \| \beta \|_1
  \]

  - \[ a_j \beta_j - c_j - \lambda \beta_j < 0 \rightarrow a_j \beta_j - c_j + \lambda \beta_j > 0 \]
  - \[ [-c_j - \lambda, -c_j + \lambda] \beta_j = 0 \]
  - \[ a_j \beta_j - c_j + \lambda \beta_j > 0 \]

- Set subgradient = 0:
  \[
  \partial_{\beta_j} F(\beta) = \begin{cases} 
  a_j \beta_j - c_j - \lambda & \beta_j < 0 \\
  [-c_j - \lambda, -c_j + \lambda] & \beta_j = 0 \\
  a_j \beta_j - c_j + \lambda & \beta_j > 0 
  \end{cases}
  \]

  - If \( \beta_j < 0 \)
    \[ a_j \beta_j - c_j - \lambda = 0 \Rightarrow \beta_j = \frac{c_j - \lambda}{a_j} < 0 \Rightarrow c_j < -\lambda \text{ strong neg. corr.} \]
    then \( \beta_j < 0 \)
  
  - If \( \beta_j > 0 \)
    \[ a_j \beta_j - c_j + \lambda = 0 \Rightarrow \beta_j = \frac{c_j - \lambda}{a_j} > 0 \Rightarrow c_j > \lambda \text{ strong pos. corr.} \]
    then \( \beta_j > 0 \)
  
  - If \( \beta_j = 0 \)
    \[ -\lambda < c_j < \lambda \text{ otherwise, } \beta_j = 0 \]

- The value of \[ c_j = 2 \sum_{i=1}^{N} x_{ij}^2 (y_i - \beta'_{-j} x_{-ij}) \] constrains \( \beta_j \)

©Carlos Guestrin 2013
Soft Threshholding

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

From Kevin Murphy textbook

In LASSO, all coeff \( \hat{\beta}_j \) are shrunk relative to \( \hat{\beta}_j^{\text{OLS}} \)

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:
      \[ a_j = 2 \sum_{i=1}^{N} (x_{ij})^2 \quad c_j = 2 \sum_{i=1}^{N} x_{ij}(y_i - \hat{\beta}_{-j}'x_{-j}) \]
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 \( F(\beta) = \| X\beta - y \|_2^2 + \lambda \| \beta \|_1 \)

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

While not converged,
- Choose random coordinate \( j \),
- Update \( \beta_j \) (closed-form minimization)
Shotgun: Parallel SCD [Bradley et al ’11]

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

Shotgun (Parallel SCD)

While not converged,
- On each of \( P \) processors,
  - Choose random coordinate \( j \),
  - Update \( \beta_j \) (same as for Shooting)

---

Is SCD inherently sequential?

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

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: \[ \min_{\beta} F(\beta) \text{ where } F(\beta) = \| X\beta - y \|_2^2 + \lambda \| \beta \|_1 \]

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

\[
F(\beta + \Delta \beta) - F(\beta) \\
\leq - \sum_{i,j \in P} (\delta \beta_{ij})^2 + \sum_{i,j \in P, j \neq k} (X^TX)_{ij,ik} \delta \beta_{ij} \delta \beta_{ik}
\]

Is SCD inherently sequential?

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

\[
F(\beta + \Delta \beta) - F(\beta) \\
\leq - \sum_{i,j \in P} (\delta \beta_{ij})^2 + \sum_{i,j \in P, j \neq k} (X^TX)_{ij,ik} \delta \beta_{ij} \delta \beta_{ik}
\]

Nice case: Uncorrelated features

Bad case: Correlated features
Shotgun: Convergence Analysis

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

Assume \# parallel updates \( P < d / \rho + 1 \)

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

Convergence Analysis

**Theorem: Shotgun Convergence**

Assume \( P < d / \rho + 1 \) where \( \rho = \) spectral radius of \( X^TX \)

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

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

- Each processor in parallel:
  - Pick data point \( i \) at random
  - For \( j = 1 \ldots d \):

- 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!
Analysis of Distribute-then-Average

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