StingyCD: Safely Avoiding Wasteful Updates in Coordinate Descent

Tyler B. Johnson and Carlos Guestrin
University of Washington

Problem

Efficiently solve nonnegative Lasso:

$$\underset{\mathbf{x}\in\mathbb{R}^m}{\text{minimize}} \quad \tfrac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2 + \lambda\|\mathbf{x}\|_1 \qquad \text{s.t.} \quad \mathbf{x}\geq\mathbf{0}$$

Solution is ***sparse***—most weights have value 0

StingyCD can also solve:

- Unconstrained Lasso
- Linear SVM

Wasteful Updates in Coordinate Descent

Coordinate descent is a popular and efficient algorithm for solving this problem

Coordinate descent

for $t = 1, 2, \ldots$ **do**
- $i \leftarrow$ `get_next_coordinate()`
- $\delta \leftarrow$ `compute_update`$(i, \mathbf{x}^{(t-1)})$
- $\mathbf{x}^{(t)} \leftarrow \mathbf{x}^{(t-1)} + \delta\mathbf{e}_i$

Define residuals vector

$$\mathbf{r}^{(t)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(t)}$$

During iteration t, choose i and compute

$$g_i \leftarrow -\left\langle \mathbf{A}_i, \mathbf{r}^{(t-1)}\right\rangle + \lambda \qquad \delta \leftarrow \max\left\{-x_i^{(t-1)}, \frac{-g_i}{\|\mathbf{A}_i\|^2}\right\}$$

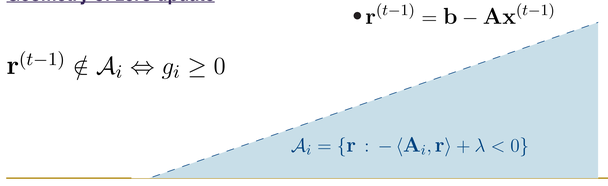
"***Zero updates***" During iteration t, $\delta = 0$ if

1. $g_i \geq 0$ and 2. $x_i^{(t-1)} = 0$

Due to sparsity, often ***majority*** of updates are zero

Zero updates are ***wasteful***—computing g_i requires time

Geometry of zero update



StingyCD

StingyCD skips over many updates guaranteed to be zero

Testing skip condition requires negligible time

Reference residuals (updated infrequently)

$$\mathbf{rr} = \mathbf{r}^{(t-k)}$$

Reference distance (updated after each nonzero update)

$$d^{(t)} = \left\|\mathbf{r}^{(t)} - \mathbf{rr}\right\|_2$$

Update to reference distance requires negligible time

$$\left(d^{(t)}\right)^2 = \underbrace{\left(d^{(t-1)}\right)^2 - 2\delta\langle \mathbf{A}_i, \mathbf{r}^{(t-1)}\rangle + 2\delta\langle \mathbf{A}_i, \mathbf{rr}\rangle + \delta^2\|\mathbf{A}_i\|^2}_{\text{Cached quantities}}$$

Active region distances (computed after each $\mathbf{rr}$ update)

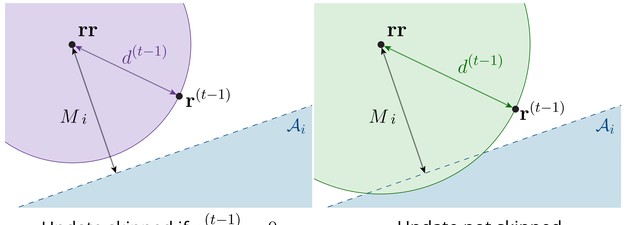
For all i, $\quad M_i = \frac{1}{\|\mathbf{A}_i\|}\left(\lambda - \langle \mathbf{A}_i, \mathbf{rr}\rangle\right)$

Distance between $\mathbf{rr}$ and $\mathcal{A}_i$ (but negative if $\mathbf{rr} \in \mathcal{A}_i$)

Full pass over data to compute M_i for all i

Skip condition StingyCD skips update t if

1. $d^{(t-1)} \leq M_i$ and 2. $x_i^{(t-1)} = 0$



Safeness guarantee $d^{(t-1)} \leq M_i \Rightarrow \mathbf{r}^{(t-1)} \notin \mathcal{A}_i \Rightarrow g_i \geq 0$

In StingyCD, every skipped update would, if computed, result in $\delta = 0$

StingyCD

for $t = 1, 2, \ldots$ **do**
- **if** `update_reference?()` **then**
  - $\mathbf{rr} \leftarrow \mathbf{r}^{(t-1)}$
  - $d^{(t-1)} \leftarrow 0$
  - $M_i \leftarrow$ `compute_M`$(\mathbf{rr})$ for all i
- $i \leftarrow$ `get_next_coordinate()`
- **if** $d^{(t-1)} \leq M_i$ **and** $x_i^{(t-1)} = 0$ **then**
  - $\mathbf{x}^{(t)} \leftarrow \mathbf{x}^{(t-1)}; \mathbf{r}^{(t)} \leftarrow \mathbf{r}^{(t-1)}; \ldots$
  - **continue**
- $\delta \leftarrow$ `compute_update`$(i, \mathbf{x}^{(t-1)})$
- $\mathbf{x}^{(t)} \leftarrow \mathbf{x}^{(t-1)} + \delta\mathbf{e}_i$
- $\mathbf{r}^{(t)} \leftarrow \mathbf{r}^{(t-1)} - \delta\mathbf{A}_i$
- $d^{(t)} \leftarrow$ `update_d`$(d^{(t-1)}, \delta, i)$

Compared to CD, a StingyCD iteration uses...

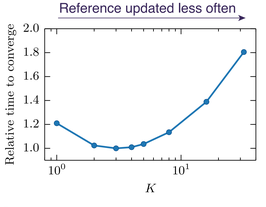
- **Less time** if update is skipped
- **More time** if reference is updated
- **The same amount of time** otherwise

Scheduling reference updates

Measure time T to first update $\mathbf{rr}$

Update $\mathbf{rr}$ after blocks of KT time

We set $K = 5$



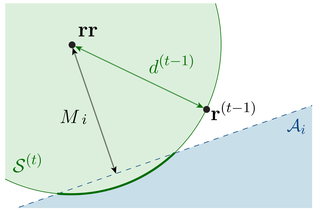
StingyCD+

StingyCD-inspired heuristic—skips updates more aggressively

Models probability of nonzero update, denoted $P(U^{(t)})$

If $x_i^{(t-1)} = 0$, then

$$P(U^{(t)}) = \frac{\text{Area}\left(\text{bd}\left(\mathcal{S}^{(t)}\right) \cap \mathcal{A}_i\right)}{\text{Area}\left(\text{bd}\left(\mathcal{S}^{(t)}\right)\right)}$$



Skip update if "expected relevant delay" is small:

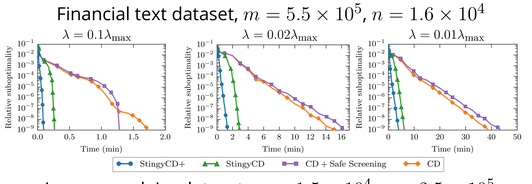
$$\underbrace{P(U^{(t)})D_i^{(t)}}_{\text{Number of updates computed since coordinate i last updated}} < \xi^{(t)}$$

We set $\xi^{(t)} = \text{NNZ}(\mathbf{x}^{(t-1)})$ (still converges to solution)

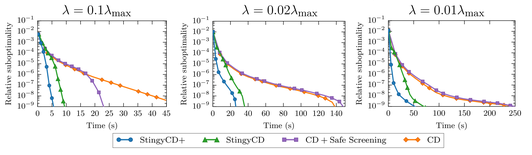
Empirical Results

Lasso problems

Financial text dataset, $m = 5.5 \times 10^5$, $n = 1.6 \times 10^4$



Insurance claim dataset, $m = 1.5 \times 10^4$, $n = 2.5 \times 10^5$



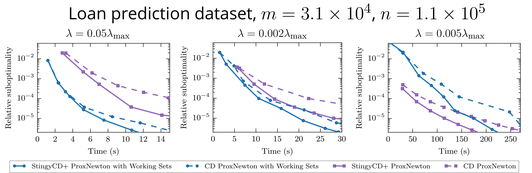
Sparse logistic regression problems

$$\underset{\mathbf{x}\in\mathbb{R}^m}{\text{minimize}} \ \sum_{i=1}^n \log(1 + \exp(-b_i\mathbf{a}_i^T\mathbf{x})) + \lambda\|\mathbf{x}\|_1$$

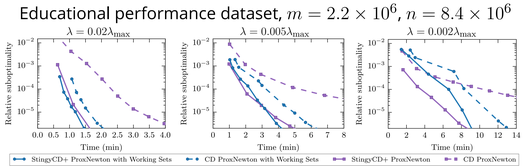
Proximal Newton method is efficient—solves sequence of Lasso problems

Compare CD, StingyCD+ as subproblem solvers

Loan prediction dataset, $m = 3.1 \times 10^4$, $n = 1.1 \times 10^5$



Educational performance dataset, $m = 2.2 \times 10^6$, $n = 8.4 \times 10^6$



Conclusions

StingyCD can avoid many wasteful CD updates

Relaxations (StingyCD+) can lead to further gains

Can also combine StingyCD with other methods