Distributed SDCA: CoCoA$^+$ vs. Mini-Batch

Martin Takáč

LEHIGH UNIVERSITY

Joint work with Peter Richtárik, Nathan Srebro, Chenxin Ma
Virginia Smith, Martin Jaggi, Michael I. Jordan

ISMP, 16th of July, 2015

The Problem - Regularized Empirical Loss Minimization

Let $\{(x_i, y_i)\}_{i=1}^n$ be our training data data, $x_i \in \mathbf{R}^d$ and $y_i \in \mathbf{R}$.

$$\min_{w \in \mathbf{R}^d} \left[P(w) := \frac{1}{n} \sum_{i=1}^{n} \ell_i(w^T x_i) + \frac{\lambda}{2} \|w\|^2 \right] \qquad \text{(P)}$$

where

- $\lambda > 0$ is a regularization parameter
- $\ell_i(\cdot)$ is convex loss function which can depend on the label y_i
  Examples:
  - Logistic loss: $\ell_i(\zeta) = \log(1 + \exp(-y_i \zeta))$
  - Hinge loss: $\ell_i(\zeta) = \max\{0, 1 - y_i \zeta\}$

The dual problem

$$\max_{\alpha \in \mathbf{R}^n} \left[D(\alpha) := -\frac{\lambda}{2} \|A\alpha\|^2 - \frac{1}{n} \sum_{i=1}^{n} \ell_i^*(-\alpha_i) \right] \qquad \text{(D)}$$

where $A = \frac{1}{\lambda n} X^T$ and $X^T = [x_1, x_2, \ldots, x_n] \in \mathbf{R}^{d \times n}$

- ℓ_i^* is convex conjugate of ℓ_i
- wlog $\|x_i\| \leq 1$

Duality

Primal-Dual mapping

For any $\alpha \in \mathrm{dom}(D)$ we can define

$$w_\alpha = w(\alpha) := A\alpha \tag{1}$$

From strong duality we have that $w^* = w(\alpha^*)$ is optimal to (P) if α^* is optimal solution to (D).

Gap function

$$G(\alpha) = P(w(\alpha)) - D(\alpha)$$

The Setting & Challenges

- The size of matrix A is huge (e.g. TBs of data)
- We want to use many nodes of computer cluster (or cloud) to speed-up the computation

Challenges

- **distributed data:** no single machine can load the whole instance
- **expensive communication:**

	latency
RAM	100 nanoseconds
standard network connection	250,000 nanoseconds

- **unreliable nodes:** we assume that the node can die at any point during the computation (we want to have fault tolerant solution)

The Serial/Parallel/Distributed SDCA Algorithm

Serial **S**tochastic **D**ual **C**oordinate **A**scent

choose $\alpha^{(0)} \in \mathbf{R}^n$
repeat
  $\alpha^{(t+1)} = \alpha^{(t)}$
  pick a random coordinate $i \in \{1, \ldots, n\}$
  compute the update: $h_t^i(\alpha^{(t)}) := \arg\max_h D(\alpha^{(t)} + he_i)$
  apply the update: $\alpha_i^{(t+1)} = \alpha_i^{(t+1)} + h_t^i(\alpha^{(t)})e_i$

The Serial/Parallel/Distributed SDCA Algorithm

Parallel Stochastic Dual Coordinate Ascent

choose $\alpha^{(0)} \in \mathbf{R}^n$

repeat

$\alpha^{(t+1)} = \alpha^{(t)}$

pick a random coordinate $i \in \{1, \dots, n\}$

pick a random subset $S \subset \{1, \dots, n\}$ with $|S| = H$

for each $i \in S$ in **parallel** do

compute the update: $h_t^i(\alpha^{(t)}) := \arg\max_h D(\alpha^{(t)} + he_i)$

apply the update: $\alpha_i^{(t+1)} = \alpha_i^{(t+1)} + h_t^i(\alpha^{(t)})e_i$

apply the update: $\alpha_i^{(t+1)} = \alpha_i^{(t+1)} + \frac{\mathbf{1}}{\mathbf{H}} \sum_{i \in S} h_t^i(\alpha^{(t)})e_i$

The Serial/Parallel/Distributed SDCA Algorithm

- assume we have K nodes (computers) each with parallel processing power
- we **partition** the coordinates $\{1, 2, \dots, n\}$ into K **balanced** sets $\mathcal{P}_1, \dots, \mathcal{P}_K$ $\forall k \in \{1, \dots, K\}$ we have $|\mathcal{P}_k| = \frac{n}{K}$

Distributed Stochastic Dual Coordinate Ascent

choose $\alpha^{(0)} \in \mathbf{R}^n$
repeat
  $\alpha^{(t+1)} = \alpha^{(t)}$
  for each **computer** $k \in \{1, \dots, K\}$ in **parallel** do
    pick a random subset $S \subset \{1, \dots, n\}$ with $|S| = H$
    pick a random subset $S_k \subset \mathcal{P}_k$ with $|S| = H \le \frac{n}{K}$
    for each $i \in S_k$ in **parallel** do
      compute the update: $h_t^i(\alpha^{(t)}) := \arg\max_h D(\alpha^{(t)} + he_i)$
  apply the update: $\alpha_i^{(t+1)} = \alpha_i^{(t+1)} + \frac{1}{H} \sum_{i \in S} h_t^i(\alpha^{(t)}) e_i$
  apply the update: $\alpha_i^{(t+1)} = \alpha_i^{(t+1)} + \frac{1}{\mathbf{KH}} \sum_{k \in \{1,\dots,K\}} \sum_{i \in S_k} h_t^i(\alpha^{(t)}) e_i$

The distributed algorithm can need (in the worst case) the **same number of iterations as a serial one!**

Can We do Better Then Averaging?

Definition (Expected Separable Overapproximation)

Assume that $\forall k$ S_k has uniform marginals: $\forall i, j : \mathbf{P}(i \in S_k) = \mathbf{P}(j \in S_k)$.
Then we say that **convex and smooth** function f admits v-ESO with respect to the sampling S if $\forall x, t \in \mathbf{R}^N$ we have

$$\mathbf{E}[f(\alpha + t_{[S]})] \leq f(\alpha) + \frac{\mathbf{E}[|S|]}{n}(\langle \nabla f(\alpha), t \rangle + \frac{1}{2}\|t\|_{\mathbf{v}}{}^2)$$

Can We do Better Then Averaging?

Definition (Expected Separable Overapproximation)

Assume that $\forall k\ S_k$ has uniform marginals: $\forall i,j : \mathbf{P}(i \in S_k) = \mathbf{P}(j \in S_k)$.
Then we say that **convex and smooth** function f admits v-ESO with respect to the sampling S if $\forall x, t \in \mathbf{R}^N$ we have

$$\mathbf{E}[f(\alpha + t_{[S]})] \leq f(\alpha) + \frac{\mathbf{E}[|S|]}{n}(\langle \nabla f(\alpha), t\rangle + \frac{1}{2}\|t\|_{\mathbf{v}}{}^2)$$

Expected Separable Lowerbound

$$\mathbf{E}[D(\alpha + t_{[\hat{S}]})] \geq \frac{b}{n}\mathcal{H}(t,\alpha) + \left(1 - \frac{b}{n}\right) D(\alpha),$$

where

$$\mathcal{H}(t,\alpha) := -\frac{1}{n}\sum_{i=1}^{n}\ell_i^*(-(\alpha_i + t_i)) - \frac{\lambda}{2}\|w_\alpha\|^2 - \frac{\lambda}{2}\|\frac{1}{\lambda n}t\|_{\mathbf{v}}^2 - \frac{1}{n}t^T X w_\alpha,$$

Can We do Better Then Averaging?

Definition (Expected Separable Overapproximation)

Assume that $\forall k$ S_k has uniform marginals: $\forall i, j : \mathbf{P}(i \in S_k) = \mathbf{P}(j \in S_k)$.
Then we say that **convex and smooth** function f admits v-ESO with respect to the sampling S if $\forall x, t \in \mathbf{R}^N$ we have

$$\mathbf{E}[f(\alpha + t_{[S]})] \leq f(\alpha) + \frac{\mathbf{E}[|S|]}{n}(\langle \nabla f(\alpha), t \rangle + \frac{1}{2}\|t\|_{\mathbf{v}}{}^2)$$

Expected Separable Lowerbound

$$\mathbf{E}[D(\alpha + t_{[\hat{S}]})] \geq \frac{b}{n}\mathcal{H}(t, \alpha) + \left(1 - \frac{b}{n}\right) D(\alpha),$$

where

$$\mathcal{H}(t, \alpha) := -\frac{1}{n}\sum_{i=1}^{n} \ell_i^*(-(\alpha_i + t_i)) - \frac{\lambda}{2}\|w_\alpha\|^2 - \frac{\lambda}{2}\|\frac{1}{\lambda n} t\|_{\mathbf{v}}^2 - \frac{1}{n} t^T X w_\alpha,$$

compute the update: $h_t^i(\alpha^{(t)}) := \arg\max_h D(\alpha^{(t)} + h e_i)$
compute the update: $h_t^i(\alpha^{(t)}) := \arg\max_h \mathcal{H}(h e_i, \alpha^{(t)})$

Can We do Better Then Averaging?

Definition (Expected Separable Overapproximation)

Assume that $\forall k$ S_k has uniform marginals: $\forall i, j : \mathbf{P}(i \in S_k) = \mathbf{P}(j \in S_k)$.
Then we say that **convex and smooth** function f admits v-ESO with respect to the sampling S if $\forall x, t \in \mathbf{R}^N$ we have

$$\mathbf{E}[f(\alpha + t_{[S]})] \le f(\alpha) + \frac{\mathbf{E}[|S|]}{n}(\langle \nabla f(\alpha), t\rangle + \frac{1}{2}\|t\|_{\mathbf{v}}{}^2)$$

Expected Separable Lowerbound

$$\mathbf{E}[D(\alpha + t_{[\hat{S}]})] \ge \frac{b}{n}\mathcal{H}(t, \alpha) + \left(1 - \frac{b}{n}\right) D(\alpha),$$

where

$$\mathcal{H}(t, \alpha) := -\frac{1}{n}\sum_{i=1}^n \ell_i^*(-(\alpha_i + t_i)) - \frac{\lambda}{2}\|w_\alpha\|^2 - \frac{\lambda}{2}\|\frac{1}{\lambda n} t\|_{\mathbf{v}}^2 - \frac{1}{n} t^T X w_\alpha,$$

compute the update: $h_t^i(\alpha^{(t)}) := \arg\max_h D(\alpha^{(t)} + h e_i)$

compute the update: $h_t^i(\alpha^{(t)}) := \arg\max_h \mathcal{H}(h e_i, \alpha^{(t)})$

apply the update: $\alpha_i^{(t+1)} = \alpha_i^{(t+1)} + \frac{1}{\mathbf{KH}} \sum_{k \in \{1,\dots,K\}} \sum_{i \in S_k} h_t^i(\alpha^{(t)}) e_i$

apply the update: $\alpha_i^{(t+1)} = \alpha_i^{(t+1)} + \sum_{k \in \{1,\dots,K\}} \sum_{i \in S_k} h_t^i(\alpha^{(t)}) e_i$

How to Compute ESO Parameter v?

Let $D = \operatorname{diag}(XX^T)$ and let us define

$$\sigma^2 \stackrel{\text{def}}{=} \max_{\alpha \in \mathbf{R}^n : \|\alpha\|=1} \frac{1}{n} \|X^T D^{-\frac{1}{2}} \alpha\|^2 \in \left[\frac{1}{n}, 1\right]$$

Serial SDCA

$v_i = \|x_i\|^2$

Parallel SDCA

$v_i = \left(1 + \frac{(H-1)(n\sigma^2-1)}{\max\{1,n-1\}}\right)\|x_i\|^2$ Compare $\left(1 + \frac{(H-1)(n\sigma^2-1)}{\max\{1,n-1\}}\right)$ with H

Distributed SDCA, $H \geq 2$

$v_i = \frac{H}{H-1}\left(1 + \frac{K(H-1)(n\sigma^2-1)}{\max\{K,n-K\}}\right)\|x_i\|^2$ Compare $\frac{H}{H-1}\left(1 + \frac{K(H-1)(n\sigma^2-1)}{\max\{K,n-K\}}\right)$ with HK

Parallel/Distributed SDCA: Convergence Guarantee

Theorem ($(1/\gamma)$-Smooth Loss)

If

- *losses are* $(1/\gamma)$*-smooth*
- $f(\alpha) = \|\frac{1}{\lambda n} X^T \alpha\|^2$ *admits* v*-ESO*
- *choose desired duality gap* $\epsilon > 0$

After

$$T \geq \frac{\|v\|_\infty}{|S|}\left(\frac{1}{\lambda\gamma} + \frac{n}{\|v\|_\infty}\right)\log\left(\frac{\|v\|_\infty}{|S|}\left(\frac{1}{\lambda\gamma} + \frac{n}{\|v\|_\infty}\right)\frac{1}{\epsilon}\right)$$

iterations we have that $\mathbf{E}[P(w_T) - D(\alpha_T)] \leq \epsilon$.

Guarantees and Speed-ups for Specific Sampling

Parallel SDCA

$$\beta = 1 + \frac{(|S|-1)(n\sigma^2-1)}{\max\{1,n-1\}}$$

Distributed SDCA

$$\beta = \frac{|S|}{|S|-K}\left(1 + \frac{(|S|-K)(n\sigma^2-1)}{\max\{K,n-K\}}\right)$$

Smooth Loss

$$T \geq \frac{\beta}{|S|}\left(\frac{1}{\lambda\gamma} + \frac{n}{\beta}\right) \log\left(\frac{\beta}{|S|}\left(\frac{1}{\lambda\gamma} + \frac{n}{\beta}\right)\frac{1}{\epsilon}\right).$$

ignoring logarithmic term:

$$\tilde{\mathcal{O}}\left(\frac{n}{|S|} + \frac{\beta}{|S|}\frac{1}{\lambda\gamma}\right)$$

- **linear improvement** in the second term, i.e. potential for linear speedup, as long as $\beta = O(1)$.
- $\beta \approx 1 + |S|\sigma^2 \Rightarrow$ we obtain linear speedups as long as $|S| = O(1/\sigma^2)$

Speed-up in SGD vs. SDCA: Smooth Loss

- SGD: linear reduction in the iteration complexity with mini-batch size of up to $\mathcal{O}(\sqrt{n})$ without any data-dependent assumption
- **Is this possible also with SDCA?**
- SDCA: $\beta \leq |S|$

  $$\mathcal{O}\left(\left(\frac{1}{\lambda\gamma} + \frac{n}{|S|}\right)\log(1/\epsilon)\right)$$

  a larger mini-batch scales the second term (unconditional on any data dependence), and as long as it is the dominant term, we get linear speedups
- $\lambda = \Theta(1/\sqrt{n})$: linear speedups up to a mini-batch of size $\mathcal{O}(\gamma\sqrt{n})$ (this is the same as the mini-batch SGD guarantee)

CoCoA vs. CoCoA+

Communication-Efficient Distributed Dual Coordinate Ascent

Input: $T \geq 1$, $\gamma \in [\frac{1}{K}, 1]$, $\sigma' \in [1, \infty)$
Data: $\{(x_i, y_i)\}_{i=1}^n$ distributed over K machines
Initialize: $\alpha_{[k]}^{(0)} \leftarrow 0$ for all machines k, and $w^{(0)} \leftarrow 0$
for $t = 1, 2, \dots, T$
  for all machines $k = 1, 2, \dots, K$ **in parallel**
    approximately $\max \mathcal{G}^{\sigma'}(\Delta\alpha_{[k]}; w^{(t)})$ to obtain $\Delta\alpha_{[k]}$ computation
    $\alpha_{[k]}^{(t)} \leftarrow \alpha_{[k]}^{(t-1)} + \gamma\Delta\alpha_{[k]}$
    $\Delta w_k \leftarrow \gamma A \Delta\alpha_{[k]}$
  reduce $w^{(t)} \leftarrow w^{(t-1)} + \sum_{k=1}^K \Delta w_k$ communication

- If $\gamma = \frac{1}{K}$, $\sigma' = 1$ we obtain CoCoA
- CoCoA+ $\gamma = 1$, $\sigma' = K$

$$\mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}; w^{(t)}) = -\frac{1}{n}\sum_{i \in \mathcal{P}_k} \ell_i^*(-(\alpha_{[k]}^{(t)} + \Delta\alpha_{[k]})_i) - \lambda (w^{(t)})^T A \Delta\alpha_{[k]} - \frac{\lambda}{2}\left\|A\Delta\alpha_{[k]}\right\|^2 - \frac{\lambda}{2}(\sigma' - 1)\left\|A\Delta\alpha_{[k]}\right\|^2.$$

$(\alpha_{[k]})_i = \alpha_i$ if $i \in \mathcal{P}_k$ and 0 otherwise

How Accurately?

Assumption: Θ-approximate solution

We assume that there exists $\Theta \in [0, 1)$ such that $\forall k \in [K]$, the local solver at any iteration t produces a **(possibly) randomized** approximate solution $\Delta\alpha_{[k]}$, which satisfies

$$\mathbf{E}\left[\mathcal{G}_k^{\sigma'}(\Delta\alpha^*_{[k]}, w) - \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}, w)\right] \leq \Theta\left(\mathcal{G}_k^{\sigma'}(\Delta\alpha^*_{[k]}, w) - \mathcal{G}_k^{\sigma'}(\mathbf{0}, w)\right), \tag{2}$$

where

$$\Delta\alpha^* \in \arg\min_{\Delta\alpha \in \mathbf{R}^n} \sum_{k=1}^{K} \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}, w). \tag{3}$$

- because the subproblem is **not really** what one wants to solve, therefore in practise $\Theta \approx 0.9$ (depending on the cluster and problem)
- what about convergence guarantees?
- how to get Θ approximate solution?

Iteration Complexity - Smooth Loss

Theorem

Assume the loss functions functions ℓ_i are $(1/\mu)$-smooth, for $i \in \{1, 2, \dots, n\}$. We define

$$\sigma_k \stackrel{def}{=} \max_{\alpha_{[k]} \in \mathbf{R}^n} \frac{\|A\alpha_{[k]}\|^2}{\|\alpha_{[k]}\|^2} \leq |\mathcal{P}_k| \tag{4}$$

and $\sigma_{\max} = \max_{k \in [K]} \sigma_k$.
Then after T iterations of CoCoA$^+$, with

$$T \geq \frac{1}{\gamma(1-\Theta)} \frac{\lambda\mu n + \sigma_{\max}\sigma'}{\lambda\mu n} \log \frac{1}{\epsilon},$$

it holds that $\mathbf{E}[D(\alpha^*) - D(\alpha^T)] \leq \epsilon$.
Furthermore, after T iterations with

$$T \geq \frac{1}{\gamma(1-\Theta)} \frac{\lambda\mu n + \sigma_{\max}\sigma'}{\lambda\mu n} \log \left(\frac{1}{\gamma(1-\Theta)} \frac{\lambda\mu n + \sigma_{\max}\sigma'}{\lambda\mu n} \frac{1}{\epsilon} \right),$$

we have the expected duality gap

$$\mathbf{E}[P(w(\alpha^{(T)})) - D(\alpha^{(T)})] \leq \epsilon.$$

Averaging vs. Adding

The leading term is $\frac{1}{\gamma(1-\Theta)} \frac{\lambda\mu n + \sigma_{\max}\sigma'}{\lambda\mu n}$. Let us assume that $\forall k : |\mathcal{P}_k| = \frac{n}{K}$

Averaging

$\gamma = \frac{1}{K}$
$\sigma' = 1$

$$\frac{K}{1-\Theta} \frac{\lambda\mu n + \frac{n}{K}}{\lambda\mu n}$$

$$\frac{1}{1-\Theta} \frac{\lambda\mu K + 1}{\lambda\mu}$$

Adding

$\gamma = 1$
$\sigma' = K$

$$\frac{1}{1-\Theta} \frac{\lambda\mu n + \frac{n}{K} K}{\lambda\mu n}$$

$$\frac{1}{1-\Theta} \frac{\lambda\mu + 1}{\lambda\mu}$$

Note: this is in the worst case (for the worst case example)

SDCA as a Local Solver

SDCA

Input: $\alpha_{[k]}, w = w(\alpha)$
Data: Local $\{(x_i, y_i)\}_{i \in \mathcal{P}_k}$
Initialize: $\Delta\alpha_{[k]}^{0} = 0 \in \mathbb{R}^n$
for $h = 0, 1, \ldots, H-1$ **do**
  choose $i \in \mathcal{P}_k$ uniformly at random
  $\delta_i^* = \arg\max_{\delta_i \in \mathbf{R}} \mathcal{G}_k^{\sigma'}(\Delta\alpha_{[k]}^{h} + \delta_i e_i, w)$
  $\Delta\alpha_{[k]}^{(h+1)} = \Delta\alpha_{[k]}^{(h)} + \delta_i^* e_i$
end for
Output: $\Delta\alpha_{[k]}^{(H)}$

Theorem

Assume the functions ℓ_i are $(1/\mu)-$smooth for $i \in \{1, 2, \ldots, n\}$. If

$$H \geq n_k \frac{\sigma' + \lambda n \mu}{\lambda n \mu} \log \frac{1}{\Theta} \tag{5}$$

then SDCA will produce a Θ-approximate solution.

Total Runtime

- To get ϵ accuracy we need

$$\mathcal{O}\left(\frac{1}{1-\Theta}\log\frac{1}{\epsilon}\right)$$

- Recall $\Theta = \left(1 - \frac{\lambda n\gamma}{1+\lambda n\gamma}\frac{K}{n}\right)^{H}$

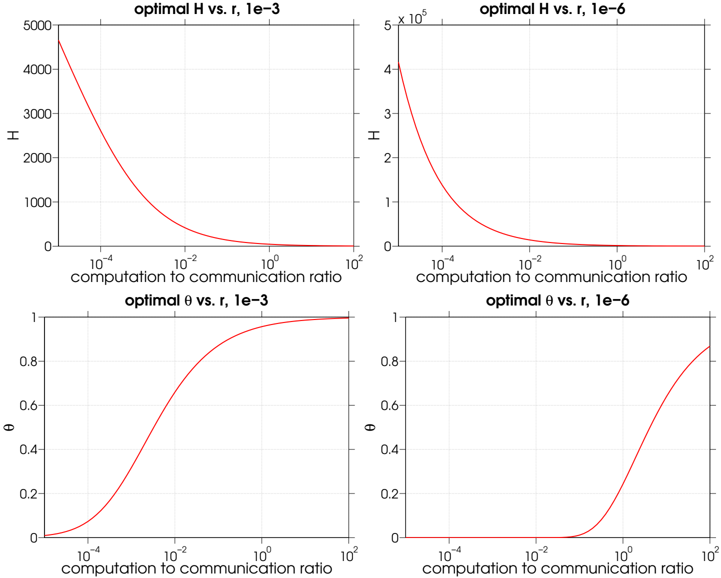
Let

- τ_o be the duration of communication per iteration
- τ_c be the duration of **ONE** coordinate update during the inner iteration

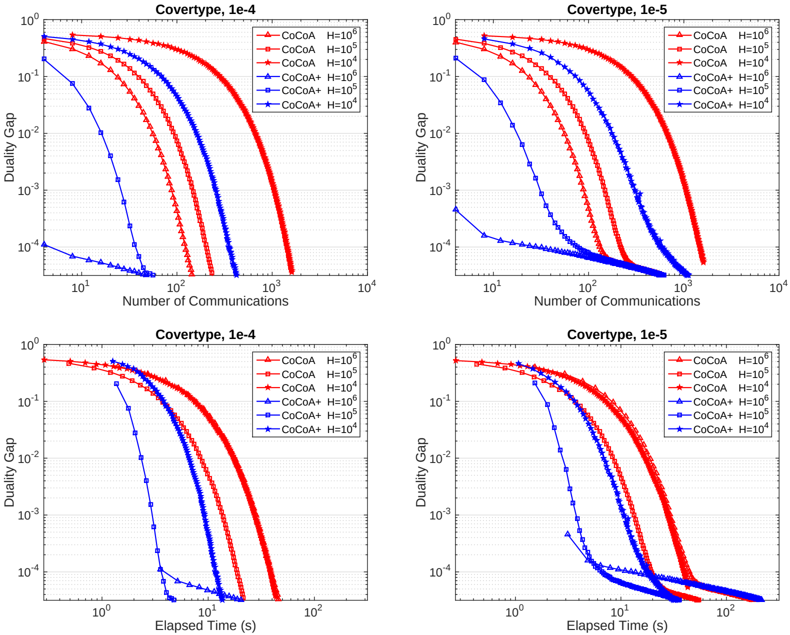
Total runtime

$$\mathcal{O}\left(\frac{1}{1-\Theta}\left(\tau_O + H\tau_c\right)\right) = \mathcal{O}\left(\frac{1}{1-\Theta}\left(1 + H\underbrace{\frac{\tau_c}{\tau_o}}_{r_{c/o}}\right)\right)$$

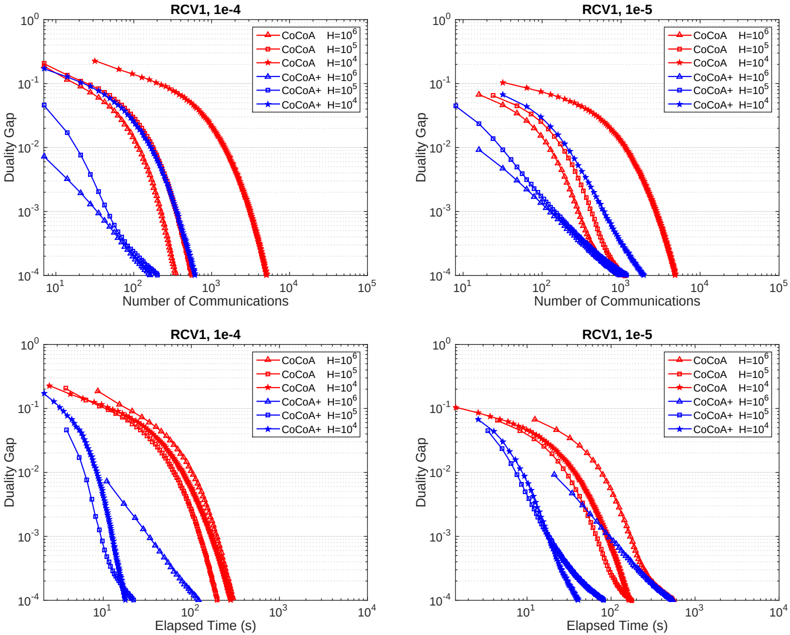
$H(\tau_c/\tau_o)$, $\Theta(\tau_c/\tau_o)$



CoCoA vs. CoCoA$^+$



CoCoA vs. CoCoA$^+$



mSDCA vs. CoCoA+: Theory

Setting $|S| = K$ and $H = 1$

- mini-batch SDCA with a minibatch of size $|S|$
- we would expect the CoCoA+ analysis to yield the same guarantee

mSDCA

$$\tilde{O}\left(\frac{n}{b} + \frac{1}{b\lambda} + \frac{\sigma^2}{\lambda}\right)$$

CoCoA+

$$\tilde{O}\left(\frac{n}{b} + \frac{n\tilde{\sigma}^2}{b\lambda} + \frac{1}{\lambda} + \frac{\tilde{\sigma}^2}{\lambda^2}\right)$$

where $\tilde{\sigma}^2 = \max_k \max_{\alpha:\sum_{i\in\mathcal{P}_k}\|\alpha_i x_i\|^2=1} \left(\frac{K}{n}\|\sum_{i\in\mathcal{P}_k}\alpha_i x_i\|\right) \geq \sigma^2 \geq 1/n$

Setting $|S| = n$ and $H \to \infty$ ($\Theta = 0$)

- CoCoA+ do much more work, we expect the guarantees to be better

mSDCA

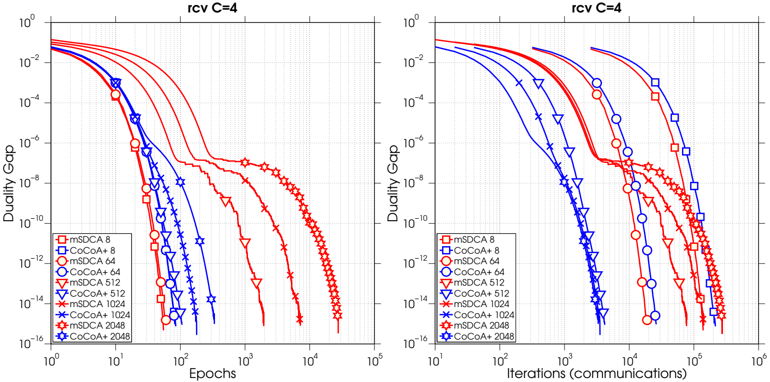
$$\tilde{O}\left(1 + \frac{\sigma^2}{\lambda}\right)$$

CoCoA+

$$\tilde{O}\left(1 + \frac{\sigma'\tilde{\sigma}^2}{\lambda}\right)$$

where $\sigma' = \max_\alpha \frac{1}{K}\frac{\|X^T\alpha\|}{\sum_k\|\sum_{i\in\mathcal{P}_k}x_i\alpha_i\|}$ and so $\sigma'\tilde{\sigma}^2 \geq \sigma^2$

mSDCA vs. CoCoA+: Experiments



References

1. Chenxin Ma, Virginia Smith, Martin Jaggi, Michael I. Jordan, Peter Richtárik and Martin Takáč: *Adding vs. Averaging in Distributed Primal-Dual Optimization*, arXiv: 1502.03508, 2015.
2. Martin Jaggi, Virginia Smith, Martin Takáč, Jonathan Terhorst, Thomas Hofmann and Michael I. Jordan: *Communication-Efficient Distributed Dual Coordinate Ascent*, NIPS 2014.
3. Richtárik, P. and Takáč, M.: *On optimal probabilities in stochastic coordinate descent methods*, arXiv:1310.3438, 2013.
4. Richtárik, P. and Takáč, M.: *Parallel coordinate descent methods for big data optimization*, arXiv:1212.0873, 2012.
5. Richtárik, P. and Takáč, M.: *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, Mathematical Programming, 2012.
6. Takáč, M., Bijral, A., Richtárik, P. and Srebro, N.: *Mini-batch primal and dual methods for SVMs*, In ICML, 2013.
7. Takáč, M., Richtárik, P. and Srebro, N.: *Distributed Mini-Batch SDCA*, arXiv, 2015.
8. Qu, Z., Richtárik, P. and Zhang, T.: *Randomized dual coordinate ascent with arbitrary sampling*, arXiv:1411.5873, 2014.
9. Qu, Z., Richtárik, P., Takáč, M. and Fercoq, O.: *SDNA: Stochastic Dual Newton Ascent for Empirical Risk Minimization*, arXiv:1502.02268, 2015.
10. Tappenden, R., Takáč, M. and Richtárik, P., *On the Complexity of Parallel Coordinate Descent*, arXiv: 1503.03033, 2015.