**overview**

**Goal:** Fixing the communication bottleneck for distributed optimization in supervised ML.

**Contributions**
- COCOA+ is a primal-dual framework for distributed optimization
  - efficient additive aggregation of local updates
  - strong convergence guarantees
  - framework: guarantees for arbitrary local solvers
  - significant practical speedup

**COCOA+**

**Main Idea**
Propose a local subproblem to allow additive, data dependent aggregation

![Diagram showing COCOA+](image)

- $\gamma = \frac{1}{2}$ => averaging, $\gamma = 1$ => adding

**Local Subproblem**

\[
\max_{\Delta \alpha_i \in [0, \Delta] \mid \text{ aggregation parameter } \gamma} g''_i(\Delta \alpha_i) \mid w, \alpha_i
\]

- $\sigma^r$ = measure of difficulty of data partitioning
- $\sigma^s := \gamma K \in [1, K]$ safe in practice

**CoCoA**

**Framework**
- Input: Aggregation parameter $\gamma \in (0, 1]$
- Subproblem parameter $\sigma$

**CoCoA+**
- Rate independent of K
- Applies also to duality gap

**convergence results**

**Local $\Theta$-Approximation**
For $\Theta \in [0, 1]$, we assume the local solver finds a (possibly) randomized approximate solution satisfying:

\[
\begin{align*}
\mathbb{E}[g''_i(\Delta \alpha_i) - g''_i(0)] \\
\leq \Theta(g''_i(\Delta \alpha_i) - g''_i(0))
\end{align*}
\]

**Theorem.** Let $\ell_i(.)$ be $L$-Lipschitz
Obtain suboptimality $\epsilon$, after $T$ iterations, with:

- CoCoA, averaging $\gamma = 1/K$
  
  $$T \geq \frac{K}{1 - \Theta} \left( \frac{SL^2}{\lambda x} + \epsilon \right)$$

- CoCoA+, adding $\gamma = 1$
  
  $$T \geq \frac{1}{1 - \Theta} \left( \frac{SL^2}{\lambda x} + \epsilon \right)$$

**Local $\Theta$-Approximation**
For $\Theta \in [0, 1]$, we assume the local solver finds a (possibly) randomized approximate solution satisfying:

**Theorem.** Let $\ell_i(.)$ be $\mu$-smooth
Obtain suboptimality $\epsilon$, after $T$ iterations, with:

- CoCoA, averaging $\gamma = 1/K$
  
  $$T \geq \frac{1}{1 - \Theta} \left( \frac{\mu K}{\lambda x} + \log T \right)$$

- CoCoA+, adding $\gamma = 1$
  
  $$T \geq \frac{1}{1 - \Theta} \left( \frac{\mu K}{\lambda x} + \log T \right)$$

**setup**

**Primal problem formulation**

\[
\min_{w \in \mathbb{R}^d} \frac{1}{2} \|w\|^2 + \frac{1}{n} \sum_{i=1}^n \ell_i(w^T x_i)
\]

**data partitioned by examples** $A_i := x_i$

**Dual problem**

\[
\max_{\alpha \in \mathbb{R}^n} -\frac{1}{2} \|\alpha\|^2 - \frac{1}{n} \sum_{i=1}^n \ell_i^*(\alpha_i) - \frac{1}{\lambda n} A_i \Delta \alpha_i
\]

**flexible: can use arbitrary local solver**

**experiments in Spark**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$d$</th>
<th>$K$</th>
<th>$\epsilon$</th>
<th>$\mu$</th>
<th>$\lambda$</th>
<th>$\Theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>438</td>
<td>2</td>
<td>0.01</td>
<td>50</td>
<td>1000</td>
<td>0.1</td>
</tr>
<tr>
<td>Spambase</td>
<td>58</td>
<td>16</td>
<td>0.005</td>
<td>2</td>
<td>100</td>
<td>0.001</td>
</tr>
<tr>
<td>Webspam</td>
<td>154</td>
<td>64</td>
<td>0.001</td>
<td>0.1</td>
<td>5</td>
<td>0.99</td>
</tr>
</tbody>
</table>

*Experiments with BDCA as a local solver -> reduces to BCDCA [1]*

$H =$ number of local updates per round

code at: [github.com/gingsmith/cocoa](github.com/gingsmith/cocoa)

**scaling**

**references**