L1-Regularized Distributed Optimization
A Communication-Efficient Primal-Dual Framework

Martin Jaggi
ETH Zurich
Machine Learning Applications

Classification
- Support Vector Machine (SVM) \((L1,L2)\)
- Logistic Regression \((L1,L2)\)

Structured Prediction \((L1,L2)\)

Regression
- Ridge Regression
- Least Squares variants \((L1,L2):\)
  - Lasso, Elastic-Net \((Feature~Selection,~Compressed~Sensing)\)
Optimization Algorithms

Training data
Optimization Algorithms

\[ w := w + \lambda \cdot x \]

Support-Vektor-Machine

(Cortes & Vapnik 1995)
Distributed Machine Learning

What if the data does not fit onto one computer anymore?
Distributed Optimization

\[ w \leftarrow w + \sum_k \Delta w^{(k)} \]

Naive Distributed SGD/SDCA
Problem 1

The Cost of Communication

- Reading $\nu$ from Memory (RAM)
  
  $100 \text{ ns}$

- Sending $\nu$ to another Machine
  
  $500'000 \text{ ns}$

- One Typical Map-Reduce Iteration ($\text{Hadoop}$)
  
  $10'000'000'000 \text{ ns}$
Problem 2

* Parallel Algorithms are Hard
* Single Machine Solvers are Fast

* no reusability of good single machine algorithms
Distributed Optimization

Naive Distributed SGD

$\Delta w^{(1)} := \gamma x_i$

Repeat $T$ times

Reduce $w := w + \sum_k \Delta w^{(k)}$

$\Delta w^{(5)} := \gamma x_i$

# local datapoints read: $T$
# communications: $T$
convergence: ✓

“always communicate”
Naive Distributed SGD

Repeat T times:

\[ \Delta \mathbf{w}^{(1)} := \gamma \mathbf{x}_i \]

Reduce:

\[ \mathbf{w} := \mathbf{w} + \sum_k \Delta \mathbf{w}^{(k)} \]

Convergence:

✓

One-Shot Averaged Distributed Optimization

Do once:

\[ \mathbf{w}^{(1)} := \mathbf{w}^{(1)\ast} \]

\[ \mathbf{w}^{(5)} := \mathbf{w}^{(5)\ast} \]

Reduce:

\[ \mathbf{w} := \frac{1}{K} \sum_k \mathbf{w}^{(k)} \]

Convergence:

✗

Communication: Always / Never

Naive Distributed SGD:

- Local datapoints read: T
- Communications: T
- Convergence: ✓
- "always communicate"

One-Shot Averaged Distributed Optimization:

- Local datapoints read: T
- Communications: 1
- Convergence: ✗
- "never communicate"
One-Shot Averaging Does Not Work

\[ w^{(1)} := w^{(1)*} \]
\[ w^{(5)} := w^{(5)*} \]

Reduce

\[ w := \frac{1}{K} \sum_k w^{(k)} \]

Arjevani, Y., & Shamir, O. Communication Complexity of Distributed Convex Learning and Optimization. NIPS 2015
Communication Efficient
Distributed Block-Coordinate Ascent

\[ w(\alpha) := A\alpha \]

\[ \Delta w^{(1)} \]

Machine 1
\[ \alpha_1 \ldots \alpha_{1M} \]

Machine 2
\[ \alpha_{1M} \ldots \alpha_{2M} \]

Machine 3
\[ \alpha_{4M} \ldots \alpha_{5M} \]

Machine 4

Machine 5

Repeat T times

Reduce

\[ w := w + \frac{1}{K} \sum_k \Delta w^{(k)} \]

CoCoA
Optimization Problem

\[
\max_{\alpha \in \mathbb{R}^n} \left[ D(\alpha) := -\frac{1}{2} \| A\alpha \|^2_2 - \sum_{i=1}^{n} \ell_i^* (-\alpha_i) \right]
\]

\[
A_{1\text{oc}} \Delta \alpha[k] + w
\]

CoCoA+
Optimization Problem

\[
\max_{\alpha \in \mathbb{R}^n} \left[ D(\alpha) := -g^*(A\alpha) - \sum_{i=1}^{n} l_i^*(-\alpha_i) \right]
\]

where

\[
A_{\text{loc}} \Delta \alpha[k] + w
\]

prox CoCoA+
Convergence

Theorem
Have suboptimality $\epsilon$, and duality gap $\epsilon$, after $T$ rounds

$$T \geq \frac{1}{1-\Theta} \frac{\mu+n}{\mu} \log \frac{n}{\epsilon}$$

Theorem
Have suboptimality $\epsilon$, and duality gap $\epsilon$, after $T$ rounds

$$T \geq \tilde{O} \left( \frac{1}{1-\Theta} \frac{8L^2}{\lambda \epsilon} + \tilde{c} \right)$$
Primal-Dual Context

Primal

$$\min_{\mathbf{w} \in \mathbb{R}^d} \left[ \mathcal{P}(\mathbf{w}) := g(\mathbf{w}) + \sum_{i=1}^{n} \ell_i(\mathbf{w}^T \mathbf{x}_i) \right]$$

Dual

$$\max_{\mathbf{\alpha} \in \mathbb{R}^n} \left[ \mathcal{D}(\mathbf{\alpha}) := -g^*(A\mathbf{\alpha}) - \sum_{i=1}^{n} \ell_i^*(-\mathbf{\alpha}_i) \right]$$

$A_{i_{oc}} \Delta \mathbf{\alpha}[k] + \mathbf{w}$
Convergence

<table>
<thead>
<tr>
<th>Theorem</th>
<th>$g$ 1-s.c.</th>
<th>$g^*$ 1-smooth</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell_i(.)$ 1/$\mu$-smooth</td>
<td>$\ell_i^*(\cdot)$ $\mu$-s.c.</td>
<td></td>
</tr>
</tbody>
</table>

Have suboptimality $\epsilon$, and duality gap $\epsilon$, after $T$ rounds

$$T \geq \frac{1}{1-\Theta} \frac{\mu+n}{\mu} \log \frac{n}{\epsilon}$$

<table>
<thead>
<tr>
<th>Theorem</th>
<th>$g$ 1-s.c.</th>
<th>$g^*$ 1-smooth</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell_i(.)$ $L$-Lipschitz</td>
<td>$\ell_i^*(\cdot)$ $L$-bounded support</td>
<td></td>
</tr>
</tbody>
</table>

Have suboptimality $\epsilon$, and duality gap $\epsilon$, after $T$ rounds

$$T \geq \tilde{O} \left( \frac{1}{1-\Theta} \left( \frac{8L^2}{\lambda \epsilon} + \tilde{c} \right) \right)$$
Machine Learning Applications

\[ \max_{\alpha \in \mathbb{R}^n} \left[ \mathcal{D}(\alpha) := -g^*(A\alpha) - \sum_{i=1}^{n} \ell_i^*(-\alpha_i) \right] \]

Classification
Support Vector Machine (SVM)
(reg.: \text{L1, L2, elastic-net})
Logistic Regression
(reg.: \text{L1, L2, elastic-net})

Structured Prediction
(reg.: \text{L1, L2, elastic-net})

Regression
Least Squares
(reg.: \text{L1, L2, elastic-net})

CoCoA+
\( D = \text{dual} \)

prox CoCoA+
\( D = \text{dual} \)

primal prox CoCoA+
\( D = \text{primal} \)
L1: get bounded support!
Local Subproblems

$$\max_{\alpha \in \mathbb{R}^n} \left[ D(\alpha) := -g^*(A\alpha) - \sum_{i=1}^{n} \ell_i^*(-\alpha_i) \right]$$

$$- \sum_{k=1}^{K} G_k^{\sigma'} (\Delta \alpha_{[k]} ; v, \alpha_{[k]}) = L + g^*(v) + \nabla g^*(v)^T A \Delta \alpha$$

$$+ \frac{\sigma'}{2} \Delta \alpha^T \begin{bmatrix} A_{[1]}^T A_{[1]} & 0 \\ 0 & \ddots & \ddots \\ 0 & & A_{[K]}^T A_{[K]} \end{bmatrix} \Delta \alpha$$

where here $$L = \sum_{i \in [n]} \ell_i^* (- (\alpha + \Delta \alpha)_i)$$.

$$v = A\alpha$$
Local $\Theta$-Approximation

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

$$\mathbb{E}\left[g^\sigma_k(\Delta \alpha^*_k) - g^\sigma_k(\Delta \alpha_{[k]})\right] \leq \Theta \left(g^\sigma_k(\Delta \alpha^*_{[k]}) - g^\sigma_k(0)\right)$$
Conclusion

* full adaptivity to the communication cost
* re-usability of good single machine solvers
* primal-dual certificates
* theoretical and practical efficiency

Open Research

* incorporating second-order information
* multi-level approach on heterogenous systems
Dissolve\textsuperscript{struct}  
A Library for Distributed Structured Prediction

built on \texttt{Spark}

Structured SVM solver
\texttt{Block Coordinate Frank-Wolfe}

Distributed Optimization
\texttt{CoCoA}

Open Source
Approximate Inference \textit{allowed!}
drop-in replacement for \texttt{SVM}\textsuperscript{struct}

Applications:

Text
- Parsing
- POS tagging
- sentence alignment
- entity linking

Biology
Protein structure & function prediction

Vision
2d+3d Segmentation, OCR

more?
- Scene understanding
- object localization & recog.

Your Application?

dalab.github.io/dissolve-struct/
Project:
Distributed Machine Learning Benchmark

Goal:
Public and Reproducible Comparison of Distributed Solvers

github.com/dalab/distributed-ML-benchmark
Thanks

“Communication-Efficient Distributed Block-Coordinate Ascent”

CoCoA+ paper (ICML 2015)

CoCoA paper (NIPS 2014)

prox CoCoA / primal CoCoA: on arXiv soon

Spark code is available on github

joint work with Simone Forte, Virginia Smith, Martin Takáč, Chenxin Ma, Tribhuvanesh Orekondy, Aurelien Lucchi, Peter Richtarik, Thomas Hofmann, Michael I. Jordan