Distributed Optimization for Machine Learning

Martin Jaggi
EPFL Machine Learning and Optimization Laboratory
mlo.epfl.ch
Machine Learning Methods to Analyze Large-Scale Data
Machine Learning?

software that can

learn from data
Machine Learning Example

Training data
The Learning Algorithm

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

\( x_i \in \mathbb{R}^d \)

(Stochastic Gradient Descent)

iteration cost: \( O(d) \)

Perceptron
(Rosenblatt 1957)

Support-Vector-Machine
(Cortes & Vapnik 1995)
Machine Learning Systems
Machine Learning Systems

What if the data does not fit onto one computer anymore?
Machine Learning Systems
The Cost of Communication

- Reading \( \mathbf{v} \) from memory (RAM)
  
  \( 100 \text{ ns} \)

- Sending \( \mathbf{v} \) to another machine
  
  \( 500'000 \text{ ns} \)

- Typical Map-Reduce iteration
  
  \( 10'000'000'000 \text{ ns} \)
Usability
Parallel Programming is Hard

- no reusability of good single machine algorithms & code
Problem class

\[
\min_{\alpha \in \mathbb{R}^n} \quad f(A\alpha) + g(\alpha)
\]
repeat

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

2.1 Definitions

This formulation includes many popular methods in machine learning and signal processing, for convex functions.

2. Background and Setup

CoCoA: A General Framework for Communication-Efficient Distributed Optimization

Definition 2

In particular, we outline three separate cases: Case I, in which the function is smooth, and analogously if the same holds for all subgradients, in the case of a general closed convex function.

Definition 3

In particular, we outline three separate cases: Case I, in which the function is smooth, and analogously if the same holds for all subgradients, in the case of a general closed convex function.

Definition 4

In particular, we outline three separate cases: Case I, in which the function is smooth, and analogously if the same holds for all subgradients, in the case of a general closed convex function.

primal Lasso
dual L2-reg SVM/Log-Regr
primal L1-reg SVM/Log-Regr

Optimization: Primal-Dual Context

\[
\min_{\alpha \in \mathbb{R}^n} \left[ \mathcal{O}_A(\alpha) := f(A\alpha) + g(\alpha) \right]
\]

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

correspondence

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

\[
\min_{w \in \mathbb{R}^d} \left[ \mathcal{O}_B(w) := g^*(-A^T w) + f^*(w) \right]
\]
Distributed Experiments

Sparse Linear Regression

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Training</th>
<th>Features</th>
<th>Sparsity</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>2,396,130</td>
<td>3,231,961</td>
<td>3.5e-3%</td>
</tr>
<tr>
<td>epsilon</td>
<td>400,000</td>
<td>2,000</td>
<td>100%</td>
</tr>
<tr>
<td>kddb</td>
<td>19,264,097</td>
<td>29,890,095</td>
<td>9.8e-5%</td>
</tr>
<tr>
<td>webspam</td>
<td>350,000</td>
<td>16,609,143</td>
<td>0.02%</td>
</tr>
</tbody>
</table>

NIPS 2014, ICML 2015, arxiv.org/abs/1611.02189

Spark Code: github.com/gingsmith/proxcocoa

+ TensorFlow
+ Apache Flink
### Challenge

**Leveraging Memory Hierarchy**

Which data to put in which memory?
Leveraging Memory Hierarchy

duality gap as selection criterion

adaptive importance sampling
Experiments

RAM → GPU, 30GB dataset

Lasso

SVM
Conclusion

- try to improve **usability** of large-scale ML
- full **adaptivity** to the communication cost, memory hierarchy and bandwidth
- **re-usability** of good single machine solvers
- **accuracy** certificates
Open Research

- limited precision operations for efficiency of communication and computation
- asynchronous and fault tolerant algorithms
- multi-level approach on heterogenous systems
- more re-usable algorithmic building blocks
  - for more systems and problems
Project:
Distributed Machine Learning Benchmark

Goal:
Public and Reproducible Comparison of Distributed Solvers

github.com/mlbench/mlbench
Thanks!

mlo.epfl.ch

Celestine Dünner, Virginia Smith, Simone Forte, Chenxin Ma, Martin Takac, Dmytro Perekrestenko, Volkan Cevher, Michael I. Jordan, Thomas Hofmann