Randomized Proximal Algorithm with Automatic Dimension Reduction

Dmitry GRISHCHENKO
grishchenko.org

joint work with
F. IUTZELER, J. MALICK, M.-R. AMINI

Université Grenoble Alpes

ISMP 2018, Bordeaux
Distributed setup

- one **master** machine
- $M$ **worker** machines
- data stored locally
  on worker machines
- communication cost
  proportional to sending data size
Distributed Learning

Global objective:

$$\min_{x \in \mathbb{R}^d} \frac{1}{m} \sum_{j=1}^{m} \ell_j(x) + g(\lambda, x) + g(x)$$

$m$ examples individual losses ($\ell_j$)
empirical risk minimization regularizer $g$

Local data:

$$\min_{x \in \mathbb{R}^d} \sum_{i=1}^{M} \pi_i f_i(x) + g(x)$$

$M$ data blocks stored locally local functions ($f_i$)

$$f_i(x) = \frac{1}{|S_i|} \sum_{j \in S_i} \ell_j(x)$$

proportion $\pi_i = |S_i|/m$ at $i$
**Review on Proximal Gradient**

**Problem:**

\[
\min_{x \in \mathbb{R}^n} \ f(x) + g(x),
\]

- \( f(x) \) is differentiable, \( L \)-smooth and \( \mu \)-strongly convex
- \( g(x) \) is non-smooth but convex

**Algorithm:**

\[
x^{k+1} = \text{prox}_{\gamma g}(x^k - \gamma \nabla f(x^k)),
\]

where proximity operator of \( g \)

\[
\text{prox}_{\gamma g}(x) \triangleq \arg\min_{u} \left\{ g(u) + \frac{1}{2\gamma} \| u - x \|^2 \right\}
\]

**Convergence result:**

Let each \( f \) be \( L \)-smooth and \( \mu \)-strongly convex. Then, for \( \gamma \in (0, 2/(\mu + L)] \),

\[
\| x^k - x^* \|^2 \leq (1 - \alpha)^k \| x^0 - x^* \|^2,
\]

for \( \alpha = 2\gamma\mu L / (\mu + L) \)
Distributed Proximal Gradient

Problem:

$$\min_{x \in \mathbb{R}^d} \sum_{i=1}^{M} \pi_i f_i(x) + g(x)$$

Gradient property:

$$\nabla F(x) = \sum_{i=1}^{M} \pi_i \nabla f_i(x)$$

Algorithm: (on each iteration)

Master gathers local variables

$$x^{k+1} = \sum_{i=1}^{M} \pi_i x_i^{k+1/2} = x^k - \gamma \nabla F(x)$$

Master performs a proximity operation

$$x_1^{k+1} = \ldots = x_M^{k+1} = \text{prox}_{\gamma g}(x^{k+1})$$

Worker $i$ updates local variable

$$x_i^{k+1/2} = x_i^k - \gamma \nabla f_i(x_i^k)$$

for all $i = 1, \ldots, M$

It's exactly proximal gradient descent

$k =$ number of master updates

Convergence rate:

Let each $f_i$ be $L_i$-smooth and $\mu_i$-strongly convex. Then, for $\gamma \in (0, 2/(\mu + L)]$ and $L = \max\{L_i\}, \mu = \min\{\mu_i\}$,

$$\|x^k - x^*\|^2 \leq (1 - \alpha)^k \|x^0 - x^*\|^2$$
Communication Problem

**Question:**
What if dimension $d$ is extremely high?

**Answer:**
Sparsify data before sending!

---

Master ➔ Worker 1 ➔ $\nabla f_1$
Master ➔ Worker 2 ➔ $\nabla f_2$
...
Master ➔ Worker $M$ ➔ $\nabla f_M$

Map

$\sum_{i=1}^{M} \pi_i \cdot \text{prox}_{\gamma g}$

Reduce

$\text{prox}_{\gamma g}$
Let \((u^k)\) be a sequence converging to \(u^*\), verifying
\[
x^k := \text{prox}_{\gamma g}(u^k) \rightarrow x^*
\]
where \(x^*\) is the unique minimizer of the \(\min_x \sum_{i=1}^{M} \pi_i f_i(x) + g(x)\).

Then, there is \(K < \infty\) such that:

- \(g(x) = \lambda_1 \|x\|_1\).

\[
\text{supp}(x^*) \subseteq \text{supp}(x^k) \subseteq \text{supp}(y^*_\varepsilon) \quad \text{for all } k \geq K,
\]
where \(\text{supp}(x) = \{i \in [1, n] \mid x_i \neq 0\}\)

- \(g(x) = 1\)-dimensional TV \(x) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| \)

\[
\text{jumps}(x^*) \subseteq \text{jumps}(x^k) \subseteq \text{jumps}(y^*_\varepsilon) \quad \text{for all } k \geq K
\]
where \(\text{jumps}(x) = \{i \in [1, n-1] \mid x_i \neq x_{i+1}\}\)

where \(y^*_\varepsilon = \text{prox}_{\gamma(1-\varepsilon)g}(u^* - x^*)\) for any \(\varepsilon > 0\).
QUESTION:
What identification gives to us?

ANSWER:
For some regularizers proximal gradient points become sparse in some meaning:
- for $\ell_1$ regularizer - coordinate sparsity (small amount of nonzero coordinates)
- for TV regularizer - block sparsity (small amount of jumps)

CONCLUSION:
- master sends $\textbf{prox}_{\gamma g}$ which is “sparse”
- rightwards communications are “sparse”
Leftwards Sparsification

Ideas of sparsification:

- \( \text{prox}_{\gamma g} x_i^k \) is not an option to send – \( \sum_i \alpha_i \text{prox}_{\gamma g} x_i^k \) leads to nothing!
- master knows \( \bar{x}^k \) – we can send only gradient from slave!

QUESTION: How to sparsify gradient?

Option I:[Tong Zhang’ 17]
Use stochastic gradient against real one

Drawback:
- decreasing stepsize
- full gradient computation

Option II:[Peter Richtárik’ 16]
Use parallel coordinate descent

Drawback:
- block-separability
- shared memory

Our option: Use coordinate descent based algorithm taking into account sparsity structure of final solution
Some Notations

Projections:
Let \( \mathcal{P} \) be a set of orthogonal projections \( \{P_i\} \) such that:

- \( P_i \) is linear operator
- \((\forall i : P_i(z^*) = P_i(y^*)) \iff z^* = y^*\)

Expectation:
We select \( P \in \mathcal{P} \) random with the same probabilities
Let us denote by \( \bar{\mathcal{P}} = \mathbb{E}P \)
Also let \( \bar{Q} = \bar{\mathcal{P}}^{-\frac{1}{2}} \)

Examples:
Subspaces with sparsity equal to \( s \):
- \( \ell_1 \): \( s \)-dimensional subspace with fixed \( \text{supp} \) of size \( s \)
- \( \text{TV} \): \( s \)-dimensional subspace with fixed \( \text{jumps} \) of size \( s - 1 \)

Projections \( \mathcal{P} \):
- \( \ell_1 \): set of diagonal matrices with \( s \) ones and all other zeros
- \( \text{TV} \): set of projections, each projection is block-diagonal matrix with \( s \)–blocks; each block is fully filled with values equal to inverse of block’s size
Randomized Strata Descent

Master Initialization

Initialize $z^0$
Fix "measure of sparsity dimension", generate set $\mathcal{P}$ and calculate $\bar{\mathcal{P}}$, $\bar{\mathcal{Q}}$
Compute $x^0 = \text{prox}_{\gamma g}(\bar{\mathcal{Q}}^{-1}(z^0))$
Randomly select $P_0$ and send $P_0$, $x^0$, $\bar{\mathcal{Q}}$ to workers

Master

Initialize

\begin{align*}
\text{for } k=1,\ldots & \text{ do} \\
& \text{Receive } y_{i}^{k-1} \text{ from workers} \\
& z^k = z^{k-1} - P_{k-1}(z^{k-1}) \\
& \quad + P_{k-1}(\bar{\mathcal{Q}}(x^{k-1})) + \sum_{i=1}^{M} \pi_i y_{i}^{k-1} \\
& x^k = \text{prox}_{\gamma g}(\bar{\mathcal{Q}}^{-1}(z^k)) \\
& \text{Randomly select } P_k \\
& \text{Send } x^k, P_k \text{ to workers} \\
\text{end for}
\end{align*}

Worker $i$

\begin{align*}
\text{for } k=0,\ldots & \text{ do} \\
& \text{Receive } x^k, P_k \\
& y_{i}^{k} = P_{k} \bar{\mathcal{Q}}(\gamma \nabla f_i(x^k)) \\
& \text{Send } y_{i}^{k} \text{ to master} \\
\text{end for}
\end{align*}

Is it “coordinate descent”?

- **yes** because we use coordinate selection in gradient
- **no** because we don’t need regularizer to be separable
Experiments for LASSO
Randomized Strata Descent

- Synthetic LASSO problem
\[ \min \frac{1}{2} \|Ax - b\|^2_2 + \lambda_1 \|x\|_1 \]
dimension \( d = 30 \), \( \lambda_1 = 0.1 \)
- 10 machines (1CPU, 1GB) in a cluster
- Data divided uniformly

Analysis
positive Amount of iterations almost proportional to amount of coordinates selected
positive Identification works as expected
negative There is no relation between mask recognition and algorithm speedup
Experiments for Least Squares with 1-d TV Regularizer
Randomized Strata Descent

- Synthetic Least Squares problem with $1 - d$ TV regularizer

$$
\min \frac{1}{2} \|Ax - b\|_2^2 + \lambda_1 \sum_{i=1}^{d-1} |x_i - x_{i+1}|
$$

dimension $d = 30$, $\lambda_1 = 0.5$

- 10 machines (1CPU, 1GB) in a cluster

- Data divided uniformly

**Analysis**

**positive** Identification works as expected

**negative** Extremely big amount iterations for sparsified versions, does not correlate even with jumps’ amount

**negative** There is no relation between mask recognition and algorithm speedup
## Randomized Strata Descent with Automatic Dimension Reduction

### Master

Initialize

```latex
\textbf{for} k=1,p+1,.. \textbf{do}
\begin{align*}
\text{calculate sparsity structure of } x^k - S_k \\
\text{if } S_k \neq S_{k-p} \text{ then} \\
\text{Generate new } P, \bar{P}, \bar{Q} \text{ w.r.t to } S_k \text{ and } s\text{-extra} \\
\text{Send } S_k \text{ to slave}
\end{align*}
```

```latex
\textbf{end if}
```

```latex
\textbf{for} l=1,..,p \textbf{do}
\begin{align*}
\text{Receive } y_{i}^{k+l-1} \text{ from workers} \\
\text{Calculate:}
\begin{align*}
    z^{k+l} &= z^{k+l-1} - P_{k+l-1}(z^{k+l-1}) \\
    &+ P_{k+l-1} (\bar{Q} (x^{k+l-1})) + \sum_{i=1}^{M} \pi_i y_{i}^{k+l-1}
\end{align*}
\end{align*}
```

```latex
x^k = \text{prox}_{\gamma g} \left( \bar{Q}^{-1} (z^k) \right) \\
\text{Randomly select } P_k \\
\text{Send } x^k, P_k \text{ to workers}
```

```latex
\textbf{end for}
```

```latex
\textbf{end for}
```

### Worker i

```latex
\textbf{for} k=0,.. \textbf{do}
\begin{align*}
\text{if } S_k \text{ received then} \\
\text{Generate new } P, \bar{P}, \bar{Q} \text{ w.r.t to } S_k \text{ and } s\text{-extra}
\end{align*}
```

```latex
\textbf{end if}
```

```latex
\textbf{Receive} x^k, P_k \\
\begin{align*}
y_{i}^{k} &= P_k \bar{Q} \left( \gamma \nabla f_i(x^k) \right) \\
\text{Send } y_{i}^{k} \text{ to master}
\end{align*}
```

```latex
\textbf{end for}
```

### Is it “coordinate descent”?  

- **no** because we use adapted coordinate selection in gradient  
- **no** because we don’t need regularizer to be separable
Experiments for Least Squares with 1-d TV Regularizer
Randomized Strata Descent with Automatic Dimension Dimension Reduction

- Synthetic Least Squares problem with $1 - d \text{TV}$ regularizer

$$\min \frac{1}{2} \|Ax - b\|_2^2 + \lambda_1 \sum_{i=1}^{d-1} |x_i - x_{i+1}|$$

Dimension $d = 30$, $\lambda_1 = 0.5$

- 10 machines (1CPU, 1GB) in a cluster

- Data divided uniformly

**Analysis**

- **Positive** Identification works as expected
- **Positive** Small amount of iterations
- **Positive** Mask recognition leads to fast convergence
Theorem
Let each $f_i$ be $L_i$-smooth and $\mu_i$-strongly convex. Then, for $\gamma \in (0, 2/(\mu + L)]$, and $L = \max\{L_i\}, \mu = \min\{\mu_i\}$

$$
\mathbb{E} \left[ \|x^k - x^*\|_2^2 \right] \leq \left( 1 - \lambda_{\text{min}} \frac{2\gamma \mu L}{\mu + L} \right)^k \|x^0 - x^*\|_2^2,
$$

where $\lambda_{\text{min}}$ is minimal eigen value of $\bar{P}$

Fixed stepsize same as in standard Proximal Gradient

Example: $\ell_1$ regularizer
- $\lambda_{\text{min}} = p_{\text{min}}$, where $p_{\text{min}}$ is minimal probability for coordinate to be chosen
- $\text{prox}_{\gamma g}$ is separable
- $\bar{Q}$ - diagonal matrix
  $$\bar{Q}$$ could be skipped in the algorithm
Conclusion

Results

• Algorithm with automatic dimension reduction
• Importance of identification in sparsification

Future plans

• Asynchronous version
• Approximate computation of $\bar{Q}$
• Scarse communications
  make less exchanges
• Non-strongly-convex functions ($f_i$)

Thank you!