Randomized Proximal Algorithm with Automatic Dimension Reduction

Dmitry GRISHCHENKO
grishchenko.org

joint work with

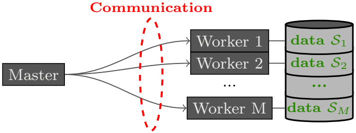
F. IUTZELER, J. MALICK, M.-R. AMINI

Université Grenoble Alpes

Optimization Days 2018, Grenoble

Distributed setup

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



Global objective:

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

m examples individual losses (ℓ_j) empirical risk minimization regularizer g

Local data:

$$\min_{x \in \mathbb{R}^d} \underbrace{\sum_{i=1}^{M} \pi_i f_i(x)}_{\text{convex, smooth}} + \underbrace{g(x)}_{\text{convex, nonsmooth}}$$

M data blocks stored locally local function (f_i)
$f_i(x) = \frac{1}{|\mathcal{S}_i|} \sum_{j \in \mathcal{S}_i} \ell_j(x)$
proportion $\pi_i = |\mathcal{S}_i|/m$ at i

Université Grenoble Alpes

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:

$$x^{k+1} = \mathbf{prox}_{\gamma g}(x^k - \gamma \nabla f(x^k)),$$

where *proximity operator* of g

$$\mathbf{prox}_{\gamma g}(x) := \operatorname*{argmin}_u \left\{ g(u) + \frac{1}{2\gamma} \|u - x\|^2 \right\}$$

Convergence result:

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

$$\|x^k - x^\star\|^2 \leq (1 - \alpha)^k \|x^0 - x^\star\|^2,$$

for $\alpha = 2\gamma\mu L/(\mu + L)$

Distributed Proximal Gradient

Problem:

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

Gradient property:

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

Algorithm: on each iteration:

Master		Worker
Master gathering of the 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 $x_1^{k+1} = \cdots = x_M^{k+1} = \mathbf{prox}_{\gamma g}\left(x^{k+1}\right)$	COMMUNICATION	Worker i update on local variable $x_i^{k+1/2} = x_i^k - \gamma\nabla f_i(x_i^k)$ for all $i = 1, .., M$

It's exactly proximal gradient descent

k = number of master updates

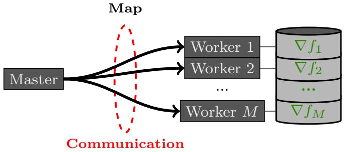
Convergence rate:

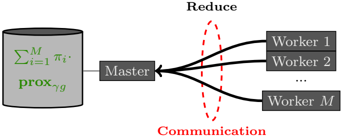
Let each f_i be L_i-smooth and μ_i-strongly convex. Then, for $\gamma \in (0, 2/(\mu + L)]$ and $L = \max\{L_i\}$, $\mu = \min\{\mu_i\}$,

$$\|x^k - x^\star\|^2 \leq (1-\alpha)^k \|x^0 - x^\star\|^2$$

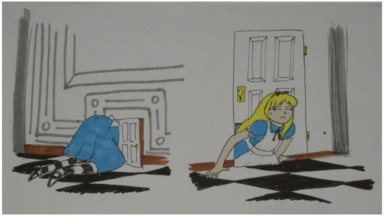
Université Grenoble Alpes

Communication Problem

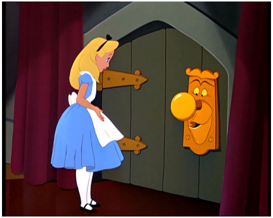




Question:
what if dimension d is extremely high?



Answer:
sparsify data before sending!



Identification

[Malick-Fadili-Peyré' 18]

Let (u^k) be a sequence converging to $u^\star$, verifying

$$x^k := \mathop{\mathbf{prox}}_{\gamma g}(u^k) \to x^\star$$

where $x^\star$ 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:

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

$$\mathrm{supp}(x^\star) \subseteq \mathrm{supp}(x^k) \subseteq \mathrm{supp}(y_\varepsilon^\star) \qquad \text{for all } k \geq K,$$

where $\mathrm{supp}(x) = \{i \in [1, n] \,|\, x_i \neq 0\}$

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

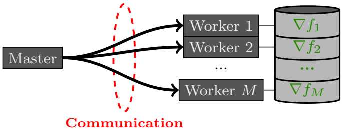
$$\mathrm{jumps}(x^\star) \subseteq \mathrm{jumps}(x^k) \subseteq \mathrm{jumps}(y_\varepsilon^\star) \qquad \text{for all } k \geq K$$

where $\mathrm{jumps}(x) = \left\{i \in [1, n-1] \,|\, x_i \neq x_{i+1}\right\}$

where $y_\varepsilon^\star = \mathbf{prox}_{\gamma(1-\varepsilon)g}(u^\star - x^\star)$ for any $\varepsilon > 0$.

Université Grenoble Alpes

Rightwards Sparsification



QUESTION:
What identification gives to us?

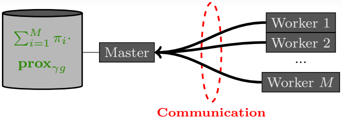
ANSWER:
For some regularizers proximal gradient points become sparse in some meaning:

- for ℓ_1 regularizer - coordinate sparsity (small amount of nonzero coordinates)
- for **TV** regularizer - block sparsity (small amount of jumps)

CONCLUSION:

- master sends $\mathbf{prox}_{\gamma g}$ which is "sparse"
- rightwards communications are "sparse"

Leftwards Sparsification



Ideas of sparsification:

- $\mathbf{prox}_{\gamma g}\, x_i^k$ is not an option to send – $\sum_i \alpha_i\, \mathbf{prox}_{\gamma g}\, x_i^k$ 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

UNIVERSITÉ Grenoble Alpes

Some Notations

Projections:

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

- P_i is linear operator
- $(\forall\, i: \; P_i(z^\star) = P_i(y^\star)) \Leftrightarrow z^\star = y^\star$

Expectation:

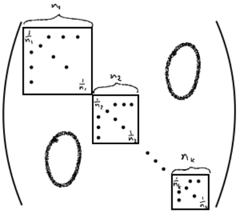
We select $P \in \mathcal{P}$ random with the same probabilities
Let us denote by $\bar{\mathcal{P}} = \mathbb{E}P$
Also let $\bar{\mathcal{Q}} = \bar{\mathcal{P}}^{-\frac{1}{2}}$

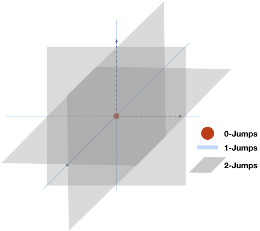
Examples:

Subspaces with sparsity equal to s:

ℓ_1 $s-$dimensional subspace with fixed **supp** of size s

TV $s-$dimensional subspace with fixed **jumps** of size $s-1$





Projections $\mathcal{P}$:

ℓ_1 set of diagonal matrices with s ones and all other zeros

TV set of projections, each projection is block-diagonal matrix with $s-$blocks; each blocks 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 $x^0 = \mathbf{prox}_{\gamma g}\left(\bar{\mathcal{Q}}^{-1}\left(z^0\right)\right)$
Randomly select P_0 and send P_0, x^0, $\bar{\mathcal{Q}}$ to workers

Master

Initialize
for k=1,.. **do**
Receive y_i^{k-1} from workers
$$z^k = z^{k-1} - P_{k-1}(z^{k-1}) + P_{k-1}\left(\bar{\mathcal{Q}}^{-1}\left(x^{k-1}\right)\right) + \sum_{i=1}^{M} \pi_i y_i^{k-1}$$
$x^k = \mathbf{prox}_{\gamma g}\left(\bar{\mathcal{Q}}^{-1}\left(z^k\right)\right)$
Randomly select P_k
Send x^k, P_k to workers
end for

C O M M U N I C A T I O N

Worker i

for k=0,.. **do**
Receive x^k, P_k
$y_i^k = P_k \bar{\mathcal{Q}}\left(\gamma \nabla f_i(x^k)\right)$
Send y_i^k to master
end for

Is it "coordinate descent"?

- **yes** because we use coordinate selection in gradient
- **no** because we don't need regularizer to be separable

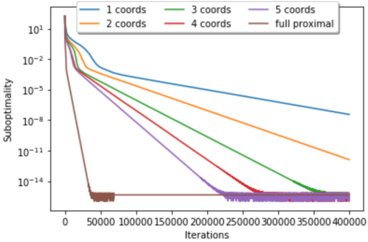
Experiments for LASSO

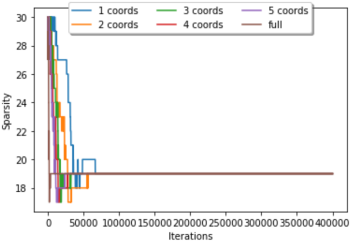
- Synthetic LASSO problem

$$\min \frac{1}{2}\|Ax - b\|_2^2 + \lambda_1\|x\|_1$$

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

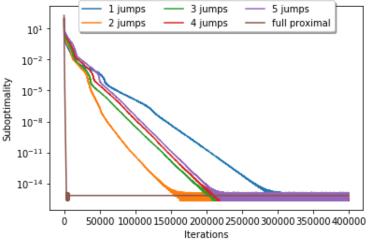
negative — There is no relation between mask recognition and algorithm speedup

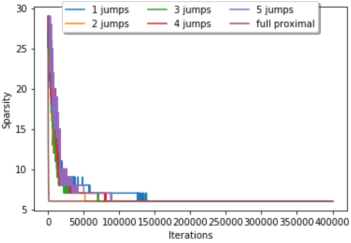
- Synthetic Least Squares problem with $1-d$ **TV** regularizer

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

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
for k=1,p+1,.. **do**
  calculate sparsity structure of x^k – S_k
  if $S_k \neq S_{k-p}$ **then**
    Generate new $\mathcal{P}, \bar{\mathcal{P}}, \bar{\mathcal{Q}}$ w.r.t to S_k and s-extra
    Send S_k to slave
  end if
  for l=1,..,p **do**
    Receive y_i^{k+l-1} from workers

$$z^{k+l} = z^{k+l-1} - P_{k+l-1}(z^{k+l-1})$$

$$+ P_{k+l-1}\left(\bar{\mathcal{Q}}^{-1}\left(x^{k+l-1}\right)\right) + \sum_{i=1}^{M} \pi_i y_i^{k+l-1}$$

$$x^k = \mathbf{prox}_{\gamma g}\left(\bar{\mathcal{Q}}^{-1}\left(z^k\right)\right)$$

    Randomly select P_k
    Send x^k, P_k to workers
  end for
end for

COMMUNICATION

Worker i

for k=0,.. **do**
  if S_k recieved **then**
    Generate new $\mathcal{P}, \bar{\mathcal{P}}, \bar{\mathcal{Q}}$ w.r.t to S_k and s-extra
  end if
  Receive x^k, P_k

$$y_i^k = P_k \bar{\mathcal{Q}}\left(\gamma \nabla f_i(x^k)\right)$$

  Send y_i^k to master
end for

Is it "coordinate descent"?

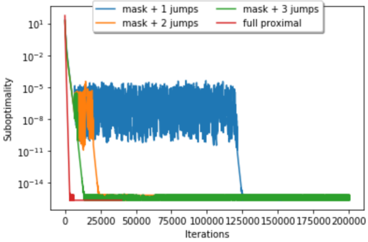
- **no** because we use adapted coordinate selection in gradient
- **no** because we don't need regularizer to be separable

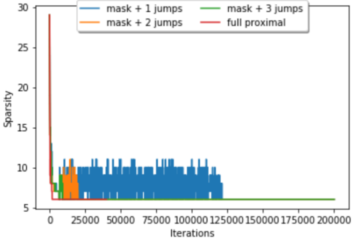
- Synthetic Least Squares problem with $1-d$ **TV** regularizer

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

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

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





Analysis

positive Identification works as expected

positive Small amount of iterations

positive Mask recognition leads to fast convergence

Theorem

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

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

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

Fixed stepsize same as in standard Proximal Gradient

Example: ℓ_1 regularizer

- $\lambda_{\min} = p_{\min}$, where $p_{\min}$ is minimal probability for coordinate to be chosen
- $\mathbf{prox}_{\gamma g}$ is separable
- $\bar{\mathcal{Q}}$ - diagonal matrix

} $\bar{\mathcal{Q}}$ 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

Thank you!