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

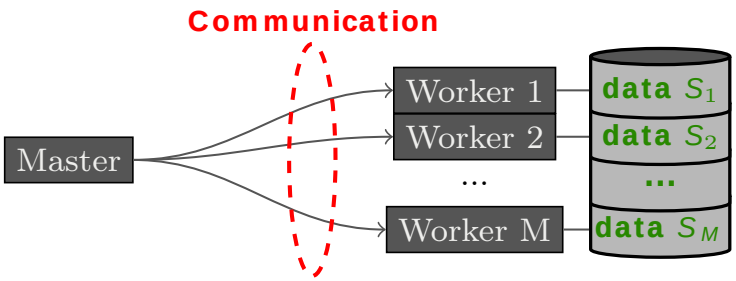
M **worker** machines

data stored locally

on worker machines

communication cost

proportional to sending data size



Distributed setup

one **master** machine

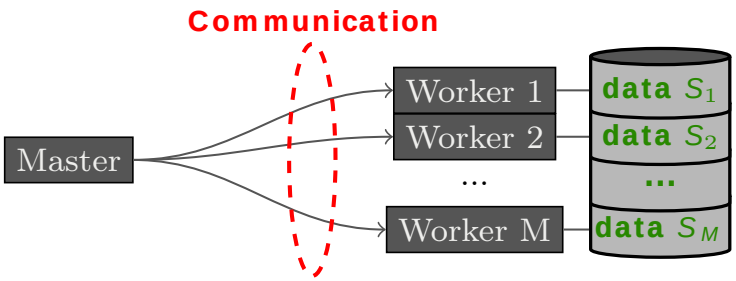
M **worker** machines

data stored locally

on worker machines

communication cost

proportional to sending data size



Distributed Learning

Global objective:

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

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

Distributed Learning

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:

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

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

Distributed Learning

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:

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

M data blocks stored locally local function (f_i)

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

proportion $\omega_i = |S_i|/m$ at i

Review on Proximal Gradient

Problem:

$$\min_{x \in \mathbb{R}^n} \; f(x) + g(x);$$

$f(x)$ is differentiable, L smooth and strongly convex

$g(x)$ is non-smooth but convex

Review on Proximal Gradient

Problem:

$$\min_{x \in \mathbb{R}^n} \quad f(x) + g(x);$$

$f(x)$ is differentiable, L smooth and strongly convex

$g(x)$ is non-smooth but convex

Algorithm:

$$x^{k+1} = \mathbf{prox}_{g}(x^k \quad r f(x^k));$$

where *proximity operator* of g

$$\mathbf{prox}_{g}(x) := \operatorname{argmin}_{u} \quad g(u) + \frac{1}{2} ku \quad xk^2$$

Review on Proximal Gradient

Problem:

$$\min_{x \in \mathbb{R}^n} \quad f(x) + g(x);$$

$f(x)$ is differentiable, L smooth and strongly convex

$g(x)$ is non-smooth but convex

Algorithm:

$$x^{k+1} = \mathbf{prox}_{g}(x^k \quad r\, f(x^k));$$

where *proximity operator* of g

$$\mathbf{prox}_{g}(x) := \operatorname{argmin}_{u} \quad g(u) + \frac{1}{2} k u \quad x k^2$$

Convergence result:

Let each f be L-smooth and -strongly convex. Then, for 2 $(0; 2=(\quad + L)]$,

$$k x^k \quad x^? k^2 \quad (1 \quad)^k k x^0 \quad x^? k^2;$$

for $= 2 \quad L=(\quad + L)$

Distributed Proximal Gradient

Problem:

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

Gradient property:

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

Distributed Proximal Gradient

Problem:

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

Gradient property:

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

Algorithm: on each iteration:

Master gathering of the local variables

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

Master performs a proximity operation

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

COMMUNICATION

Worker i update on local variable

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

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

Distributed Proximal Gradient

Problem:

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

Gradient property:

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

Algorithm: on each iteration:

Master gathering of the local variables

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

Master performs a proximity operation

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

COMMUNICATION

Worker i update on local variable

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

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

It's exactly proximal gradient descent

Distributed Proximal Gradient

Problem:

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

Gradient property:

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

Algorithm: on each iteration:

Master gathering of the local variables

$$x^{k+1} = \sum_{i=1}^{M} \alpha_i x_i^{k+1=2} = x^k \quad \nabla F(x)$$

Master performs a proximity operation

$$x_1^{k+1} = \quad = x_M^{k+1} = \textbf{prox}_{\quad g} \quad x^{k+1}$$

COMMUNICATION

Worker i update on local variable

$$x_i^{k+1=2} = x_i^k \quad \nabla f_i(x_i^k)$$

for all $i = 1; :::; M$

It's exactly proximal gradient descent

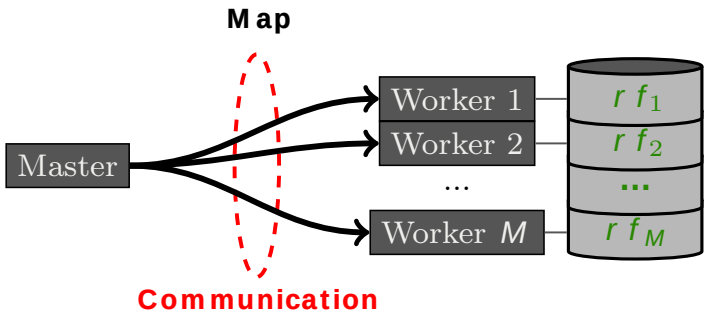
k = number of master updates

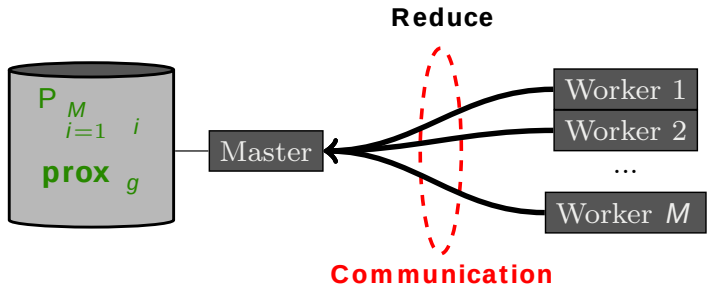
Convergence rate:

Let each f_i be L_i-smooth and $\quad_i$-strongly convex. Then, for $\quad 2 \ (0; 2=(\quad + L)]$ and $L = \max\{L_i\}; \quad = \min\{\quad_i\}$,

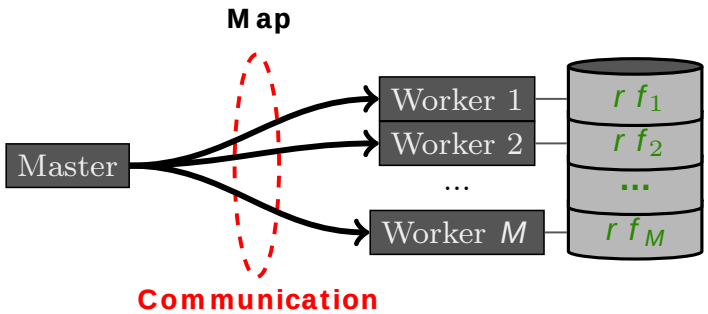
$$\|x^k \quad x^?\|^2 \quad (1 \quad)^k \|x^0 \quad x^?\|^2$$

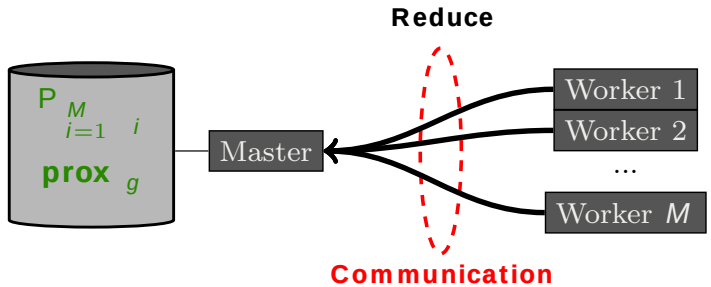
Communication Problem





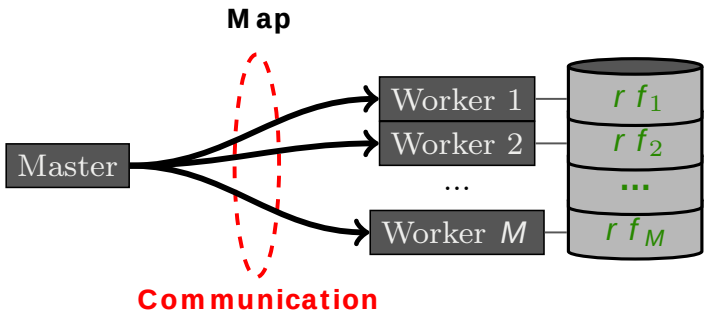
Communication Problem

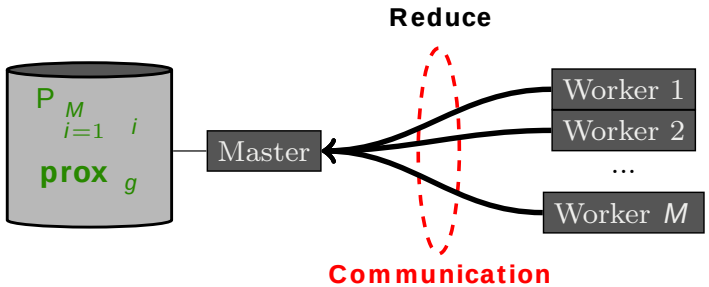




Question:
what if dimension d is extremely high?

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 $u^?$, verifying

$$x^k := \mathbf{prox}_g(u^k) \;!\; x^?$$

where $x^?$ is the unique minimizer of the $\min_x \sum_{i=1}^{M} {}_i f_i(x) + g(x)$.

Then, there is $K < 1$ such that:

- $g(x) = {}_1 kx k_1$.

  $$\mathrm{supp}(x^?) \quad \mathrm{supp}(x^k) \quad \mathrm{supp}(y^?_{"}) \qquad \text{for all } k \quad K;$$

  where $\mathrm{supp}(x) = f i \; 2 \; [1; n] \, j x_i \; 6= 0 g$

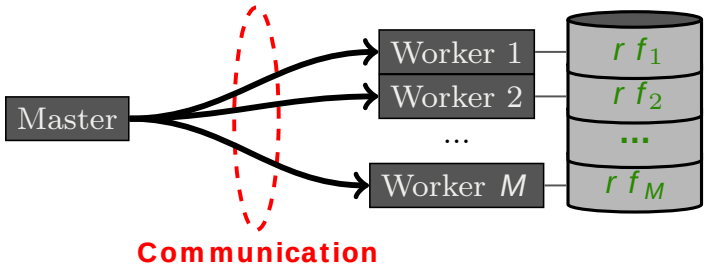
- $g(x) =$ 1-dimensional $\mathbf{TV}(x) = \sum_{i=1}^{n} {}^{1} j x_{i+1} \quad x_i j$

  $$\mathrm{jumps}(x^?) \quad \mathrm{jumps}(x^k) \quad \mathrm{jumps}(y^?_{"}) \qquad \text{for all } k \quad K$$

  where $\mathrm{jumps}(x) = \; i \; 2 \; [1; n \quad 1] j x_i \; 6= x_{i+1}$

where $y^?_{"} = \mathbf{prox}_{(1 \quad ")g}(u^? \quad x^?)$ for any $" > 0$.

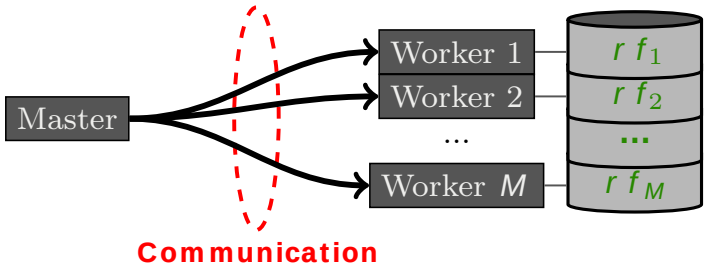
Rightwards Sparsification



QUESTION:

What identification gives to us?

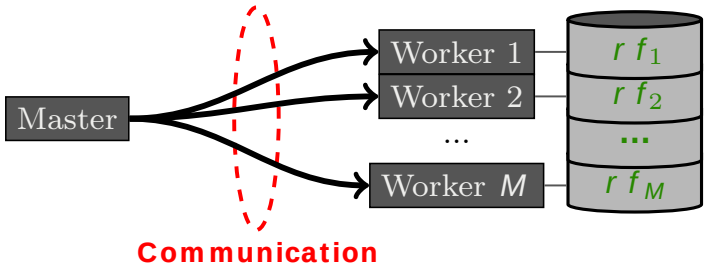
Rightwards Sparsification



QUESTION:
What identification gives to us?

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

Rightwards Sparsification

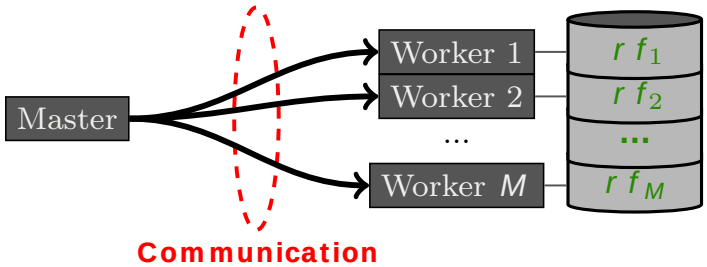


QUESTION:
What identification gives to us?

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)

Rightwards Sparsification



QUESTION:

What identification gives to us?

ANSWER:

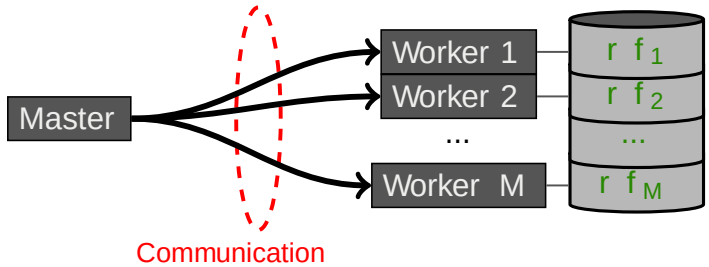
For some regularizers proximal gradient points become sparse in some meaning:

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

CONCLUSION:

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

Rightwards Sparsi cation



QUESTION:

What identi cation gives to us?

ANSWER:

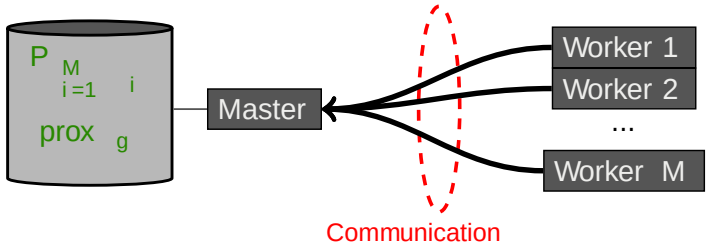
For some regularizers proximal gradient points become sparse in some meaning:

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

CONCLUSION:

- master sends prox_g which is \sparse"
- rightwards communications are \sparse"

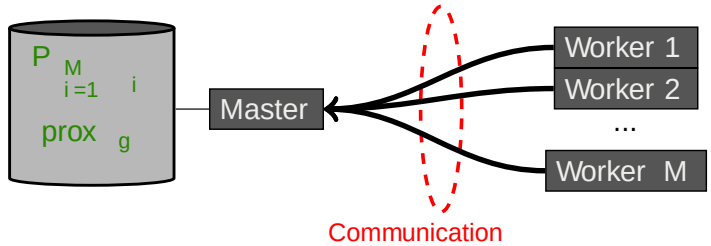
Leftwards Sparsi cation



Ideas of sparsi cation:

prox $_g$ x_i^k is not an option to send { $\sum_i$ $_i$ prox $_g$ x_i^k leads to nothing!
master knows x^k { we can send only gradient from slave!

Leftwards Sparsi cation

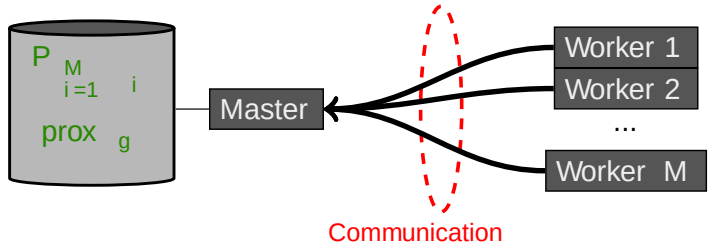


Ideas of sparsi cation:

prox $_g$ x_i^k is not an option to send { $\sum_i$ $_i$ prox $_g$ x_i^k leads to nothing!
master knows x^k { we can send only gradient from slave!

QUESTION: How to sparsify gradient?

Leftwards Sparsi cation



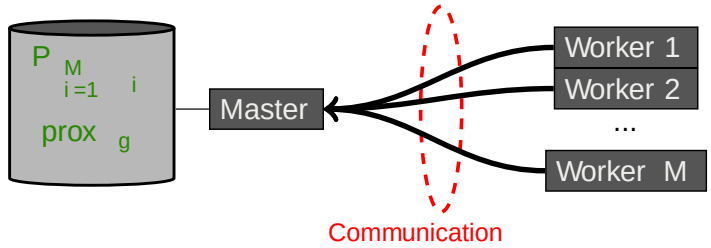
Ideas of sparsi cation:

prox $_g$ x_i^k is not an option to send { $\sum_i$ $_i$ prox $_g$ x_i^k leads to nothing!
master knows 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

Leftwards Sparsi cation



Ideas of sparsi cation:

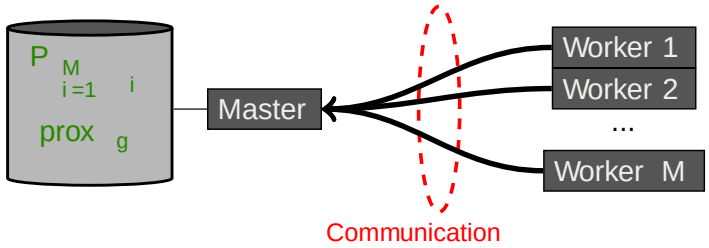
prox $_g$ x_i^k is not an option to send { $\sum_i$ $_i$ prox $_g$ x_i^k leads to nothing!
master knows x^k { we can send only gradient from slave!

QUESTION: How to sparsify gradient?

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

Drawback:
- decreasing stepsize
- full gradient computation

Leftwards Sparsi cation



Ideas of sparsi cation:

prox $_g$ x_i^k is not an option to send { $\sum_i$ $_i$ prox $_g$ x_i^k leads to nothing!

master knows 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

Option II:[Peter Richtarik' 16]

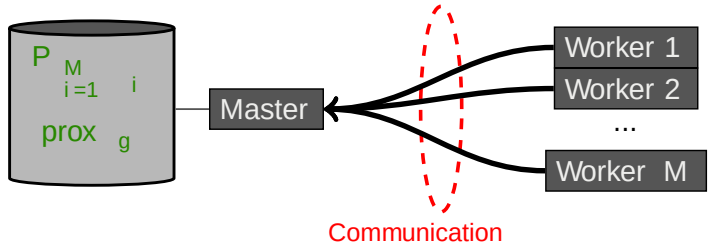
Use parallel coordinate descent

Drawback:

decreasing stepsize

full gradient computation

Leftwards Sparsi cation



Ideas of sparsi cation:

prox $_g$ x_i^k is not an option to send { $\sum_i$ $_i$ prox $_g$ x_i^k leads to nothing!
master knows x^k { we can send only gradient from slave!

QUESTION: How to sparsify gradient?

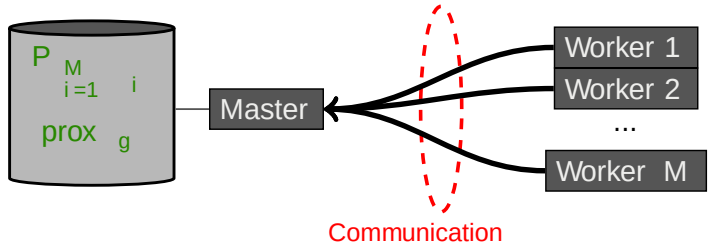
Option I:[Tong Zhang' 17]
Use stochastic gradient against real one

Drawback:
- decreasing stepsize
- full gradient computation

Option II:[Peter Richtarik' 16]
Use parallel coordinate descent

Drawback:
- block-separability
- shared memory

Leftwards Sparsi cation



Ideas of sparsi cation:

prox $_g$ x_i^k is not an option to send $\{ \sum_i$ $_i$ prox $_g$ x_i^k leads to nothing!

master knows 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 Richtarik' 16]

Use parallel coordinate descent

Drawback:

- block-separability
- shared memory

Our option: Use coordinate descent based algorithm taking into account sparsity structure of nal solution

Some Notations

Projections:

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

P_i is linear operator

$(\forall i : P_i(z^?) = P_i(y^?)) , \quad z^? = y^?$

Some Notations

Projections:

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

P_i is linear operator

$(\forall i : P_i(z^?) = P_i(y^?)) , \quad z^? = y^?$

Expectation:

We select $P \in \mathcal{P}$ random with the same probabilities

Let us denote by $P = E\mathcal{P}$

Also let $Q = P^{\frac{1}{2}}$

Some Notations

Projections:

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

P_i is linear operator

$(\forall i : P_i(z^?) = P_i(y^?)), \quad z^? = y^?$

Expectation:

We select $P \in \mathcal{P}$ random with the same probabilities

Let us denote by $P = E\mathcal{P}$

Also let $Q = P^{\frac{1}{2}}$

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

Some Notations

Projections:

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

P_i is linear operator

$(\forall i : P_i(z^?) = P_i(y^?)), \quad z^? = y^?$

Expectation:

We select $P \in \mathcal{P}$ random with the same probabilities

Let us denote by $P = E\mathcal{P}$

Also let $Q = P^{\frac{1}{2}}$

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

Some Notations

Projections:

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

P_i is linear operator

$(\forall i : P_i(z^?) = P_i(y^?)) , \quad z^? = y^?$

Expectation:

We select $P \in P$ random with the same probabilities

Let us denote by $P = EP$

Also let $Q = P^{\frac{1}{2}}$

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 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

Some Notations

Projections:

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

P_i is linear operator

$(\forall i : P_i(z^?) = P_i(y^?)) , \quad z^? = y^?$

Expectation:

We select $P \in P$ random with the same probabilities

Let us denote by $P = EP$

Also let $Q = P^{\frac{1}{2}}$

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 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 P and calculate P; Q

Compute $x^0 = \text{prox}_{g}(Q^{1} z^0)$

Randomly select P_0 and send P_0; x^0; Q to workers

Randomized Strata Descent

Master Initialization

Initialize z^0

Fix "measure of sparsity dimension", generate set P and calculate P ; Q

Compute $x^0 = \text{prox}_{g\,Q^{-1}}(z^0)$

Randomly select P_0 and send P_0 ; x^0 ; Q to workers

Master

Initialize

for k=1,.. do

Receive y_i^{k-1} from workers

$$z^k = z^{k-1} - P_{k-1}(z^{k-1}) + P_{k-1}\,Q^{-1}\,x^{k-1} + \sum_{i=1}^{M} {}_i\, y_i^{k-1}$$

$$x^k = \text{prox}_{g\,Q^{-1}}(z^k)$$

Randomly select P_k

Send x^k ; P_k to workers

end for

COMMUNICATION

Worker i

for k=0,.. do

Receive x^k ; P_k

$y_i^k =$

$P_k\,Q\,r\,f_i(x^k)$

Send y_i^k to master

end for

Randomized Strata Descent

Master Initialization

Initialize z^0

Fix "measure of sparsity dimension", generate set P and calculate P; Q

Compute $x^0 = \text{prox}_{g\,Q^{-1}}(z^0)$

Randomly select P_0 and send P_0; x^0; Q to workers

Master

Initialize

for k=1,.. do

Receive y_i^{k-1} from workers

$$z^k = z^{k-1} - P_{k-1}(z^{k-1}) + P_{k-1} Q^{-1} x^{k-1} + \sum_{i=1}^{M} {}_i y_i^{k-1}$$

$x^k = \text{prox}_{g\,Q^{-1}}(z^k)$

Randomly select P_k

Send x^k; P_k to workers

end for

COMMUNICATION

Worker i

for k=0,.. do

Receive x^k; P_k

$y_i^k = P_k Q \nabla f_i(x^k)$

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

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

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

Experiments for LASSO

Randomized Strata Descent

Synthetic LASSO problem

$$\min \frac{1}{2}kAx \quad bk_2^2 + \quad _1kxk_1$$

dimension $d = 30$, $_1 = 0:1$

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

Data divided uniformly

Analysis

positive Amount of iterations almost proportional to amount of coordinates selected

positive Identi cation works as expected

negative There is no relation between mask recognition and algorithm speedup

Experiments for Least Squares with 1-dTV 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

Experiments for Least Squares with 1-dTV 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

Experiments for Least Squares with 1-dTV 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 Identi cation works as expected

negative Extremely big amount iterations for sparsi ed 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 6= $S_{k\ p}$ then
    Generate new P ; P ; 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

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

$$+ P_{k+l\ 1}\ Q^{\ 1}\ x^{k+l\ 1} + \sum_{i=1}^{M} {}_i\, y_i^{k+l\ 1}$$

    $x^k = \text{prox}_{\ g}\ Q^{\ 1}\ z^k$
    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
    P ; P ; Q
      w.r.t to S_k
    and s-extra
  end if
  Receive x^k ; P_k

$$y_i^k = P_k\ Q\ \ r\ f_i(x^k)$$

  Send y_i^k to master
end for

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 6= $S_{k\ p}$ then
    Generate new P ; P ; 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

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

$$+ P_{k+l\ 1}\ Q^{1}\ x^{k+l\ 1} + \sum_{i=1}^{M} {}^{i}\ y_i^{k+l\ 1}$$

    $x^k = \text{prox}_{g}\ Q^{1}\ z^k$
    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
    P ; P ; Q
      w.r.t to S_k
    and s-extra
  end if
  Receive x^k ; P_k

$$y_i^k = P_k\ Q \quad r\ f_i(x^k)$$

  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

Experiments for Least Squares with 1-d TV Regularizer

Randomized Strata Descent with Automatic Dimension Reduction

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

$$\min \frac{1}{2}kAx \quad bk_2^2 + \quad _1 \sum_{i=1}^{d\chi^1} jx_i \quad x_{i+1}j$$

dimension $d = 30$, $\quad _1 = 0{:}5$

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

Data divided uniformly

Experiments for Least Squares with 1-d TV Regularizer

Randomized Strata Descent with Automatic Dimension Reduction

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

$$\min \frac{1}{2}kAx \quad bk_2^2 + \quad _1 \sum_{i=1}^{d\chi^1} jx_i \quad x_{i+1}j$$

dimension $d = 30$, $\quad _1 = 0{:}5$

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

Data divided uniformly

Experiments for Least Squares with 1-d TV Regularizer

Randomized Strata Descent with Automatic Dimension Reduction

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 $\quad_i$-strongly convex. Then, for $\quad 2\ (0; 2=(\quad + L)]$, and $L = \max\{L_i\}; \quad = \min\{\quad_i\}$

$$\mathsf{E}\left[\quad kx^k \quad x^?k_2^2\right] \qquad 1 \qquad \min \frac{2 \quad L}{+ L}^k \quad kx^0 \quad x^?k_2^2;$$

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

Convergence Rate

Theorem

Let each f_i be L_i-smooth and $\quad_i$-strongly convex. Then, for $\quad 2\ (0; 2=(\quad + L)]$, and $L = \max\{L_i\}; \quad = \min\{\quad_i\}$

$$\mathsf{E}\left[\quad kx^k \quad x^?k_2^2\right] \qquad 1 \qquad \min\frac{2 \quad L}{\quad + L}^k \quad kx^0 \quad x^?k_2^2;$$

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

Fixed stepsize same as in standard Proximal Gradient

Convergence Rate

Theorem

Let each f_i be L_i-smooth and $\quad_i$-strongly convex. Then, for $\quad 2\ (0; 2=(\quad + L)]$, and $L = \max\{L_i\}; \quad = \min\{\quad_i\}$

$$\mathsf{E}\left[\ kx^k \quad x^?k_2^2\right] \qquad 1 \qquad \min\frac{2 \quad L}{+L}^k \ kx^0 \quad x^?k_2^2;$$

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

Fixed stepsize same as in standard Proximal Gradient

Example: $\grave{}_1$ regularizer

$\quad_{\min} = p_{\min}$, where $p_{\min}$ is minimal probability for coordinate to be chosen

Convergence Rate

Theorem

Let each f_i be L_i-smooth and $_i$-strongly convex. Then, for $2 (0; 2=(+ L)]$, and $L = \max f L_i g; \quad = \min f \ _i g$

$$\mathsf{E}\left[kx^k \quad x^? k_2^2 \right] \qquad 1 \qquad \min \frac{2 \quad L}{+ L}^k kx^0 \quad x^? k_2^2;$$

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

Fixed stepsize same as in standard Proximal Gradient

Example: $`_1$ regularizer

$_{\min} = p_{\min}$, where $p_{\min}$ is minimal probability for coordinate to be chosen

prox $_g$ is separable

$\bar{Q}$ - diagonal matrix

Convergence Rate

Theorem

Let each f_i be L_i-smooth and $\quad_i$-strongly convex. Then, for $\quad 2\ (0; 2=(\quad + L)]$, and $L = \max\{L_i\}; \quad = \min\{\quad_i\}$

$$\mathsf{E}\left[\quad kx^k \quad x^?k_2^2\right] \qquad 1 \qquad \min\frac{2 \quad L}{\quad + L}^k \quad kx^0 \quad x^?k_2^2;$$

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

Fixed stepsize same as in standard Proximal Gradient

Example: ℓ_1 regularizer

$\quad_{\min} = p_{\min}$, where $p_{\min}$ is minimal probability for coordinate to be chosen

prox $_g$ 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

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

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!