2021 ZJU-CSE Summer School

Lecture VIII: Distributed Composite Optimizaiton

Jinming Xu

Zhejiang University

August 06, 2021

Outline

Proximal gradient descent

Dual proximal gradient methods

Primal-dual gradient methods

Distributed primal-dual gradient methods

Outline

Proximal gradient descent

Dual proximal gradient methods

Primal-dual gradient methods

Distributed primal-dual gradient methods

Composite optimization

- Composite optimization problem

$$F^\star = \min_{x \in \mathbb{R}^d} F(x) := f(x) + h(x)$$

  - f: convex and smooth
  - h: convex (potentially non-smooth)
- Examples
  - l_1-regularization (e.g., compressive sensing) to promote sparsity

$$\min_{x \in \mathbb{R}^d} f(x) + \underbrace{\|x\|_1}_{h(x):l_1 \text{ norm}}$$

  - TV-regualization (e.g., image recovery) to promote?

$$\min_{x \in \mathbb{R}^d} f(x) + \underbrace{\|x\|_{TV}}_{h(x):\text{Total Variation}}$$

Proximal operator

- Proximal operator

$$\mathbf{prox}_h(x) := \arg\min_z \left\{ h(z) + \frac{1}{2}\|z - x\|^2 \right\}$$

  for any convex function h.
- Why consider proximal operators?
  - well-defined under very general conditions (including nonsmooth convex functions)
  - can be evaluated efficiently for many widely used functions (regularizers)
  - provide a conceptually and mathematically simple way to cover many optimization algorithms, including PGD, PPA, ADMM and so on.

Examples of Proximal Operators

- If $h(x) = \|x\|_1$, then

$$\mathbf{prox}_{\lambda h}(x) = \begin{cases} x - \lambda, & \text{if } x > \lambda \\ x + \lambda, & \text{if } x < -\lambda \\ 0, & \text{else} \end{cases} \qquad \text{(Soft-thresholding)}$$

- If $h(x) = \iota_{\mathcal{X}}(x) = \begin{cases} 0, & \text{if } x \in \mathcal{X} \\ \infty, & \text{else} \end{cases}$, then

$$\mathbf{prox}_{\lambda h}(x) = \mathcal{P}_{\mathcal{X}}(x) \qquad \text{(Projection)}$$

- many other examples...

Properties of Proximal operator

- **Firmly nonexpansive**

$$\langle \mathbf{prox}_h(x) - \mathbf{prox}_h(y), x - y \rangle \geq \|\mathbf{prox}_h(x) - \mathbf{prox}_h(y)\|^2$$

- **Nonexpansive**

$$\|\mathbf{prox}_h(x) - \mathbf{prox}_h(y)\| \leq \|x - y\|$$

Proof of sketch: $z_1 = \mathbf{prox}_h(x_1), z_2 = \mathbf{prox}_h(x_2)$

- $x_1 - z_1 \in \partial h(z_1)$ and $x_2 - z_2 \in \partial h(z_2)$
- due to convexity of h, we have

$$\begin{cases} h(z_2) \geq h(z_1) + \langle z_2 - z_1, x_1 - z_1 \rangle \\ h(z_1) \geq h(z_2) + \langle z_1 - z_2, x_2 - z_2 \rangle \end{cases}$$

- $\Rightarrow \langle x_1 - x_1 - (z_1 - z_2), z_1 - z_2 \rangle \geq 0$
- $\Leftrightarrow \langle x_1 - x_1, z_1 - z_2 \rangle \geq \|z_1 - z_2\|^2 \Rightarrow$ firmly nonexpansive
- together with Cauchy-Schwarz, we obtain the nonexpansiveness.

Proximal gradient methods

- Proximal gradient descent

$$x^{k+1} = \mathbf{prox}_{\gamma h}\left(x^k - \gamma \nabla f(x^k)\right)$$

  - alternates between gradient updates on f and proximal minimizaiton on h
  - useful when $\mathbf{prox}_{\gamma h}(\cdot)$ is simple to evaluate
- Which is equivalent to

$$\begin{aligned} x^{k+1} &= \arg\min_x \left\{ \frac{1}{2\gamma} \left\| x - (x^k - \gamma \nabla f(x^k)) \right\|^2 + h(x) \right\} \\ &= \arg\min_x \left\{ \underbrace{\frac{1}{2\gamma} \left\| x - x^k \right\|^2}_{\text{proximal term}} + \gamma \underbrace{\left\langle x - x^k, \nabla f(x^k) \right\rangle}_{\text{first-order approximation}} + \underbrace{h(x)}_{\text{regularization}} \right\} \end{aligned}$$

Linear Convergence of Proximal Gradient Methods

Theorem (Linear Convergence Rate)

Let f be μ-strongly convex and L-smooth. If $\eta_k \equiv \gamma = \frac{1}{L}$, then

$$\left\|x^k - x^\star\right\|^2 \le \left(1 - \frac{1}{\kappa}\right)^k \left\|x^0 - x^\star\right\|^2$$

where $\kappa := L/\mu$ is condition number; $x^\star$ is minimizer.

- dimension-free in iteration complexity: need $\mathcal{O}(\kappa \log \frac{1}{\epsilon})$ number of iterations to reach an accuracy of ϵ.
- slightly weaker than that of unconstrained cases.

Sublinear Convergence of Proximal Gradient Methods

Theorem (Sublinear Convergence Rate)

Let f be convex and L-smooth. If $\eta_k \equiv \gamma = \frac{1}{L}$, then

$$F(x^k) - F^\star \leq \frac{L \left\| x^0 - x^\star \right\|^2}{k}$$

where $x^\star$ is any minimizer attaining the optimal value of $f(x^\star)$

- dimension-free in iteration complexity: need $\mathcal{O}(\frac{1}{\epsilon})$ number of iterations to reach an accuracy of ϵ
- better than subgradient methods which gives $\mathcal{O}(1/\epsilon^2)$
- fast if $\mathbf{prox}_h(\cdot)$ can be efficiently implemented

Comparing to gradient methods

- Gradient descent

	stepsize rule	convergence rate	iteration complexity
convex & smooth problems	$\gamma_k = \frac{1}{L}$	$\mathcal{O}(1/k)$	$\mathcal{O}(\frac{1}{\epsilon})$
strongly convex & smooth problems	$\gamma_k = \frac{2}{L+\mu}$	$\mathcal{O}((\frac{\kappa-1}{\kappa+1})^k)$	$\mathcal{O}(\kappa \log \frac{1}{\epsilon})$

- Proximal gradient descent

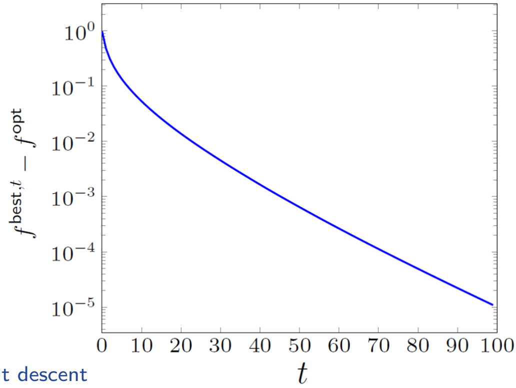
	stepsize rule	convergence rate	iteration complexity
convex & smooth problems	$\gamma_k = \frac{1}{L}$	$\mathcal{O}(1/k)$	$\mathcal{O}(\frac{1}{\epsilon})$
strongly convex & smooth problems	$\gamma_k = \frac{1}{L}$	$\mathcal{O}((1-\frac{1}{\kappa})^k)$	$\mathcal{O}(\kappa \log \frac{1}{\epsilon})$

Numerical example: LASSO

- A LASSO problem (Compressive Sensing)

$$\min_{x\in\mathbb{R}^d} F(x) = \frac{1}{2}\left\|Ax-b\right\|^2 + \|x\|_1$$

with i.i.d Gaussian $A \in \mathbb{R}^{2000\times 1000}, \gamma = 1/L, L = \lambda_{\max}(A^T A)$



Outline

Proximal gradient descent

Dual proximal gradient methods

Primal-dual gradient methods

Distributed primal-dual gradient methods

Conjugate convex functions

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ be an extend-valued convex function.

- Convex conjugate function

$$f^*(y) := \sup_{x \in \mathbb{R}^n} \{\langle x, y\rangle - f(x)\}$$

  where $f^* : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ is the convex conjugate of f

- Similar to Fourier Transformation
- Useful in primal-dual convex analysis

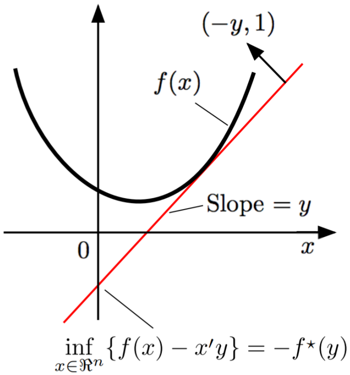


Figure: Geometric intepretion (courtesy to Bertsekas)

Conjugate convex functions

Examples: $f^*(y) := \sup_{x\in\mathbb{R}^n} \{\langle x, y\rangle - f(x)\}$

- linear function

$$f(x) := a \cdot x - b \quad \to \quad f^*(y) = \begin{cases} 0, & y = a \\ +\infty, & y \neq a \end{cases}$$

- stricly convex quadratic funciton $f(x) = \frac{1}{2}x^T A x$ with $A \succ 0$

$$f^*(y) = \sup_x \left\{ \langle x, y\rangle - \frac{1}{2}x^T A x \right\} = \frac{1}{2}x^T A^{-1} x$$

- power function (DIY)

$$f(x) := \frac{|x|^p}{p} (\text{where } p > 1) \quad \to \quad f^*(y) := \frac{|y|^q}{q} (\text{where } \frac{1}{p} + \frac{1}{q} = 1)$$

- when $f = f^*$? ($f = \frac{1}{2}\|\cdot\|^2$)

Properties of conjugate functions

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ be an extend-valued convex function and f^* be its convex conjugate function.

Theorem (Fenchel's inequality)

For any x, y, we have

$$\langle x, y \rangle \leq f(x) + f^*(y)$$

When $f = \frac{|x|^p}{p}$, the above reduces to Young inequality. Also,

- f^* is always convex no matter f is convex or not
- Let f be proper and convex. Then, $y \in \partial f(x) \Leftrightarrow x \in \partial f^*(y)$
- if f is μ-strongly convex, then f^* is $1/\mu$-smooth and vice versa.
- **Question**: when $f = f^{**}$? (HW)

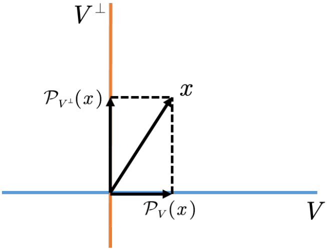
Moreau decomposition

Lemma (Moreau decomposition)

Suppose f is closed, proper and convex. Then, we have

$$x = \mathbf{prox}_f(x) + \mathbf{prox}_{f^*}(x)$$

- key relationship between proximal mapping and duality
- generalization of orthogonal decomposition



A special case for a subspace V, we have $x = \mathcal{P}_V(x) + \mathcal{P}_{V^\perp}(x)$

Convex optimization with affine constraints

- Consider the problem

$$\min_{x\in\mathbb{R}^n} f(x), \quad s.t. \quad \underbrace{Ax = b}_{\text{affine constraint}}$$

where f is convex and smooth.

- Can be rewritten as

$$\min_{x\in\mathbb{R}^n} f(x) + h(Ax)$$

where $h(u)$ is an indicator function defined as

$$h(\cdot) = \begin{cases} 0, & \text{if } Ax = b \\ \infty, & \text{otherwise} \end{cases}$$

- proximal operator w.r.t. $\tilde{h}(x) := h(Ax)$ could be very difficult (even when $\mathbf{prox}_h(\cdot)$ is simle due to the complication of A)

Fenchel Duality

- Consider the problem

$$P^\star := \min_{x\in\mathbb{R}^n} f(x) + h(Ax)$$

whose dual problem is

$$D^\star := \min_{y} -f^*(-A^T y) - h^*(y)$$

where * denotes the (Fenchel) conjugate.

- **dual formulation**

$$P^\star = \min_{x\in\mathbb{R}^n} \{f(x) + \underbrace{\max_{y\in\mathbb{R}^n} \langle Ax, y\rangle - h^*(y)\}}_{:=h(Ax)}$$

$$= \min_{x\in\mathbb{R}^n} \max_{y\in\mathbb{R}^n} \{f(x) + \langle Ax, y\rangle - h^*(y)\} \quad \text{(saddle point formualtion)}$$

$$= \max_{y\in\mathbb{R}^n} \underbrace{\min_{x\in\mathbb{R}^n} \{f(x) + \langle Ax, y\rangle\}}_{:=-f^*(-A^T y)} - h^*(y) = D^\star \quad \text{(minmax theorem)}$$

Connection to Lagarange Duality

- Consider the problem

$$P^\star := \min_{x \in \mathbb{R}^n} f(x) + h(Ax)$$

- Let $z = Ax$. Then, we have

$$\min_{x \in \mathbb{R}^n} f(x) + h(z), \textit{s.t. } z = Ax.$$

- The Lagarange dual function

$$\begin{aligned} g(y) = \min_{x,z} L(x, z, y) &= \min_{x,z} f(x) + h(z) + y^T(Ax - z) \\ &= \min_x \{f(x) + y^T Ax\} + \min_z \{h(z) - y^T z\} \\ &= \min_x \{f(x) - (-A^T y)^T x\} + \min_z \{h(z) - y^T z\} \\ &= -f^*(-A^T y) - h^*(y) \end{aligned}$$

which is exactly the above dual problem

Dual proximal gradient methods

Dual proximal gradient methods

$$y^{k+1} = \mathbf{prox}_{\gamma h^*}\left(y^k + \gamma A \nabla f^*(A^T y^k)\right)$$

- $\mathbf{prox}_{\gamma h^*}(x)$ can be calculated from the primal $I - \mathbf{prox}_{\gamma h}(x/\gamma)$

Theorem (Sublinear Convergence Rate)

Let f be μ-strongly convex. If $\gamma_k \equiv \gamma = \frac{\mu}{\lambda_{\max}(A)^2}$, then

$$D(y^k) - D^\star \leq \frac{\mu \left\|x^0 - x^\star\right\|^2}{\lambda_{\max}(A)^2 k}$$

What if A is not full rank? (HW)

Dual proximal gradient methods

Dual proximal gradient methods

$$y^{k+1} = \mathbf{prox}_{\gamma h^*}\left(y^k + \gamma A \nabla f^*(A^T y^k)\right)$$

- $\mathbf{prox}_{\gamma h^*}(x)$ can be calculated from the primal $I - \mathbf{prox}_{\gamma h}(x/\gamma)$

Theorem (Linear Convergence Rate)

Let f be μ-strongly convex and L-smooth and A be a full-rank matrix with $\kappa_A = \lambda_{\max}(A)/\lambda_{\min}(A)$. If $\gamma_k \equiv \gamma = \frac{2L\mu}{L\lambda_{\max}(A)^2 + \mu\lambda_{\min}(A)^2}$, then

$$\left\|y^k - y^\star\right\|^2 \leq \left(1 - \frac{1}{\kappa\kappa_A^2}\right)^k \left\|y^0 - y^\star\right\|^2$$

where $y^\star$ is the optimum for the dual problem.

What if A is not full rank? (HW)

Primal representation of dual proximal gradient methods

- Let $x^k = \nabla f^*(A^T y^k)$. This means that $A^T y^k = \nabla f(x^k)$
- By first-order optimality, the above is equivalent to

$$x^k = \arg\min_x \{f(x) + \langle A^T y^k, x\rangle\}$$

Dual proximal gradient methods

$$x^k = \arg\min_x \{f(x) + \langle A^T y^k, x\rangle\}$$
$$y^{k+1} = \mathbf{prox}_{\gamma h^*}\left(y^k + \gamma A x^k\right)$$

- $\{x^k\}$ is primal sequence, which is not always feasible!
- Can we approximately solve the sub-problem involving x^k?

Outline

Proximal gradient descent

Dual proximal gradient methods

Primal-dual gradient methods

Distributed primal-dual gradient methods

A saddle-point formulation

A saddle-point formulation

$$\min_x \max_y f(x) + \langle y, Ax \rangle - h^*(y)$$

remember how to derive it? (HW)

- KKT conditions

$$\begin{cases} 0 \in \nabla f(x) + A^T y \\ 0 \in Ax - \partial h^*(y) \end{cases}$$

- Can be rewriten as

$$0 \in \begin{bmatrix} \nabla f & A^T \\ -A & \partial h^* \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} := F(x, y)$$

- **Key idea**: iteratively update (x, y) to solve the above inclusion

Monotone operator

- a relation T on a set $\mathbb{R}^n$ is a subset of $\mathbb{R}^n \times \mathbb{R}^n$ (e.g., set-valued mapping $\partial f := \{(x, \partial f(x)) | x \in \mathbb{R}^n\}$)
- relation T on $\mathbb{R}^n$ is monotone if

$$(u - v)^T (x - y) \geq 0 \quad \forall (x, u), (y, v) \in T$$

- **Examples**
  - $T(x) = \partial f(x)$ is monotone
  - Skew-symmetric matrix is also monotone

$$\begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix}$$

  - Why? (Using the definition)

Resolvent operator and cocoercive property

- for $\lambda \in \mathbb{R}$, resolvent of relation T is

$$R = (I + \lambda T)^{-1}$$

  when $F = \partial f$, the above reduces to $\mathbf{prox}_{\lambda f}(\cdot)$
- We say T is β-cocoercive in G-space if

$$\beta \|Tx - Ty\|_G^2 \leq \langle Tx - Ty, x - y \rangle_G$$

- if T is monotone, then R is 1-cocoercive
  - suppose $(x, u) \in R$ and $(y, v) \in R$, i.e.,

$$x \in u + \lambda T(u), \quad y \in v + \lambda T(v)$$

  - substract to get $x - y \in u - v + \lambda(T(u) - T(v))$
  - multiply by $(u - v)^T$ and use the monotonicity of T

(Generalized) Forward-backward splitting

- Motivated by solving composite problem, e.g.,

$$\text{find } x \quad \mathbf{s.t.}\ 0 \in (M+F)x$$

  where M: monotone and F: cocoercive.
- Usually difficult to be solved together
- Examples: $\min_x \ \frac{1}{2}\|Mx-b\|_2^2 + \|x\|_1$
- Equivalent to finding fixed point of $\underbrace{(I-\gamma F)}_{T_F} x \in \underbrace{(I+\gamma M)}_{T_M} x$
- which can be solved by:

$$\begin{cases} x_{k+\frac{1}{2}} = (I-\gamma F)x_k, & (T_F : \text{gradient operator}) \\ x_{k+1} = \mathbf{prox}_{\gamma M}(x_{k+\frac{1}{2}}), & (T_M : \text{resolvent operator}) \end{cases}, \ \textit{separated!}$$

- Since M is monotone and F is cooercive, with proper stepsize γ $\Rightarrow (x_k)_{k\in\mathbb{N}}$ converges to x^*

(Generalized) Forward-backward splitting

- Motivated by solving composite problem, e.g.,

$$\text{find } x \quad \textbf{s.t. } 0 \in (M+F)x$$

where M: monotone and F: cocoercive.

- Usually difficult to be solved together
- Examples: $\min_x \ \frac{1}{2}\|Mx-b\|_2^2 + \|x\|_1$
- Equivalent to finding fixed point of $\underbrace{(I-\gamma G^{-1}F)}_{T_F} x \in \underbrace{(I+\gamma G^{-1}M)}_{T_M} x$
- which can be solved by:

$$\begin{cases} x_{k+\frac{1}{2}} = (I - G^{-1}F)x_k, & \text{(gradient operator)} \\ x_{k+1} = \mathbf{prox}_{G^{-1}M}(x_{k+\frac{1}{2}}), & \text{(proximal operator)} \end{cases}, \ \textit{separated!}$$

- $G^{-1}F$, $G^{-1}M$ is cooercive and monotone in G-space, respectively (why?), with proper stepsize $G \Rightarrow (x_k)_{k\in\mathbb{N}}$ converges to x^*

(Inexact) Primal-dual gradient methods

- Recall the primal-dual problem

$$0 \in \begin{bmatrix} \nabla f & A^T \\ -A & \partial h^* \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- which can be rewritten as

$$0 \in \underbrace{\begin{bmatrix} \nabla f & 0 \\ 0 & 0 \end{bmatrix}}_{:=F} \begin{bmatrix} x \\ y \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & A^T \\ -A & \partial h^* \end{bmatrix}}_{:=M} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Using the forward-backward splitting, we have

$$\left(\begin{bmatrix} \frac{1}{\gamma} I & 0 \\ 0 & \frac{1}{\tau} I \end{bmatrix} + \begin{bmatrix} 0 & A^T \\ -A & \partial h^* \end{bmatrix} \right) \begin{bmatrix} x^{k+1} \\ y^{k+1} \end{bmatrix} = \left(\begin{bmatrix} \frac{1}{\gamma} I & 0 \\ 0 & \frac{1}{\tau} I \end{bmatrix} - \begin{bmatrix} \nabla f & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} x^k \\ y^k \end{bmatrix}$$

(Inexact) Primal-dual gradient methods-cont'

- Which is equivalent to

$$\begin{bmatrix} x^{k+1} \\ y^{k+1} \end{bmatrix} = \underbrace{\left(\begin{bmatrix} I & \gamma A^T \\ -\tau A & I + \tau \partial h^* \end{bmatrix} \right)^{-1}}_{(G+M)^{-1}} \underbrace{\begin{bmatrix} I - \gamma \nabla f & 0 \\ 0 & I \end{bmatrix}}_{G-F} \begin{bmatrix} x^k \\ y^k \end{bmatrix}$$

- and can be rewritten as

$$\begin{aligned} x^{k+1} &= x^k - \gamma \nabla f(x^k) - \gamma A^T y^{k+1} \\ y^{k+1} &= \mathbf{prox}_{\tau h^*} \left(y^k - \tau A x^{k+1} \right) \end{aligned}$$

- still coupled in x^{k+1} and y^{k+1} due to the complication of A
- how can we further avoid the calculation of the inverse of A? note that it is not always possible to do this in dsitributed settings.

Efficient Primal-dual gradient methods

- Recall the primal-dual problem

$$0 \in \begin{bmatrix} \nabla f & A^T \\ -A & \partial h^* \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- which can be rewritten as

$$0 \in \underbrace{\begin{bmatrix} \nabla f & 0 \\ 0 & 0 \end{bmatrix}}_{:=F} \begin{bmatrix} x \\ y \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & A^T \\ -A & \partial h^* \end{bmatrix}}_{:=M} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Using the (generalized) forward-backward splitting, we have

$$\left(\begin{bmatrix} \frac{1}{\gamma} I & -A^T \\ -A & \frac{1}{\tau} I \end{bmatrix} + \begin{bmatrix} 0 & A^T \\ -A & \partial h^* \end{bmatrix} \right) \begin{bmatrix} x^{k+1} \\ y^{k+1} \end{bmatrix} = \left(\begin{bmatrix} \frac{1}{\gamma} I & -A^T \\ -A & \frac{1}{\tau} I \end{bmatrix} - \begin{bmatrix} \nabla f & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} x^k \\ y^k \end{bmatrix}$$

Efficient Primal-dual gradient methods

- Using the forward-backward splitting, we have

$$\begin{bmatrix} x^{k+1} \\ y^{k+1} \end{bmatrix} = \left(\begin{bmatrix} I & 0 \\ -2\tau A & I + \tau \partial h^* \end{bmatrix} \right)^{-1} \begin{bmatrix} I - \gamma \nabla f & -\gamma A^T \\ -\tau A & I \end{bmatrix} \begin{bmatrix} x^k \\ y^k \end{bmatrix}$$

- which can be rewritten as

$$\begin{aligned} x^{k+1} &= x^k - \gamma \nabla f(x^k) - \gamma A^T y^k \\ y^{k+1} &= \mathbf{prox}_{\tau h^*}\left(y^k - \tau A(2x^{k+1} - x^k) \right) \end{aligned}$$

- now x and y is no longer coupled!
- this way allows us to avoid the calculation of the inverse of A

Outline

Proximal gradient descent

Dual proximal gradient methods

Primal-dual gradient methods

Distributed primal-dual gradient methods

Distributed Optimization with Regularization

- Want to solve the following original problem

$$\min_{x \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^{m} f_i(x) + h_i(x), \qquad \text{(P)}$$

  - $x \in \mathbb{R}^d$: the global decision variable
  - $f_i : \mathbb{R}^d \to \mathbb{R}$ the cost funciton **known only** by the associated agent i.
  - $h_i : \mathbb{R}^d \to \mathbb{R} \cup \{\pm\infty\}$ is a (potentially nonsmooth) function of agent i.

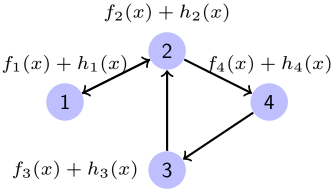


Figure: A network model

- Equivalent to solve the problem as follows

$$\min_{\mathbf{x} \in \mathcal{R}^m} f(\mathbf{x}) = \sum_{i=1}^{m} f_i(x_i) + h_i(x_i) \qquad \text{s.t. } x_i = x_i,\ \forall i, j \in \mathcal{V},$$

  - $\mathbf{x} = [x_1, x_2, ...x_m]^T$: local estimates of agents for global optimum $x^\star$.

Distributed proximal gradient method

- Distributed proximal gradient method (DPGM)

$$x_{i,k+1} = \mathbf{prox}_{\gamma h_i}\left(\sum_{j=1}^{m} w_{ij} x_{j,k} - \gamma \nabla f_i(x_{i,k})\right)$$

  - γ: the constant stepsize chosen by agents,
  - $\mathbf{prox}_{\gamma h_i}$: the proximal operator[1] of h_i with the parameter γ.

- Convergence result ($\bar{x}_k = \frac{\mathbf{1}\mathbf{1}^T}{m} x_k, \gamma \leq 1/L$):

$$\max\{\underbrace{\left\|\mathbf{x}^k - \bar{\mathbf{x}}^k\right\|}_{\text{Disagreement}}, \underbrace{\left|f(\mathbf{x}^k) - f(\mathbf{x}^\star)\right|}_{\text{Optimality gap}}\} \leq \mathcal{O}(1/k) + \mathcal{O}(\gamma)$$

  - steady state error $O(\gamma)$,
  - need bounded (sub)gradient assumption: $\|\nabla f_i\| < C$

- Only update primal variables; can we do it from dual or even primal-dual simulaneously?

[1] $\mathbf{prox}_{\gamma\phi} = arg\min_u \left(\phi(u) + \frac{1}{2\gamma}\|u - x\|^2\right)$

Distributed Optimization with Regularization

- Recalling the following original problem

$$\min_{x \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^{m} f_i(x) + g_i(x), \qquad \text{(P)}$$

  - $x \in \mathbb{R}^d$: the global decision variable
  - $f_i : \mathbb{R}^d \to \mathbb{R}$ the cost funciton **known only** by the associated agent i.
  - $g_i : \mathbb{R}^d \to \mathbb{R} \cup \{\pm\infty\}$ is a (potentially nonsmooth) function of agent i.

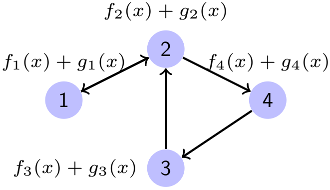


Figure: A network model

- Equivalent to solve the problem as follows

$$\min_{\mathbf{x} \in \mathcal{R}^m} f(\mathbf{x}) = \sum_{i=1}^{m} f_i(x_i) + g_i(x_i) \qquad \underbrace{\text{s.t. } (\mathbf{I}-\mathbf{W})^{1/2}\mathbf{x} = 0}_{\text{consensus when } \mathbf{null}\{\mathbf{I}-\mathbf{W}\}=\mathbf{span}\{\mathbf{1}\}},$$

  - $\mathbf{x} = [x_1, x_2, ...x_m]^T$: local estimates of agents for global optimum $x^\star$.

Derivation of Distributed Primal-dual gradient methods

- KKT conditions ($\mathbf{L} = (\mathbf{I} - \mathbf{W})^{\mathbf{1/2}}$)

$$0 \in \begin{bmatrix} \nabla f + \partial g & \mathbf{L} \\ -\mathbf{L} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- which can be rewritten as

$$0 \in \underbrace{\begin{bmatrix} \nabla f & 0 \\ 0 & 0 \end{bmatrix}}_{:=F} \begin{bmatrix} x \\ y \end{bmatrix} + \underbrace{\begin{bmatrix} \partial g & \mathbf{L} \\ -\mathbf{L} & 0 \end{bmatrix}}_{:=M} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Using the (generalized) forward-backward splitting, we have

$$\left(\begin{bmatrix} \frac{1}{\gamma} I & \mathbf{L} \\ \mathbf{L} & \frac{1}{\tau} I \end{bmatrix} + \begin{bmatrix} \partial g & \mathbf{L} \\ -\mathbf{L} & 0 \end{bmatrix} \right) \begin{bmatrix} x^{k+1} \\ y^{k+1} \end{bmatrix} = \left(\begin{bmatrix} \frac{1}{\gamma} I & \mathbf{L} \\ \mathbf{L} & \frac{1}{\tau} I \end{bmatrix} - \begin{bmatrix} \nabla f & 0 \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} x^k \\ y^k \end{bmatrix}$$

Derivation of Distributed Primal-dual gradient methods

- KKT conditions ($\mathbf{L} = (\mathbf{I} - \mathbf{W})^{\mathbf{1/2}}$)

$$0 \in \begin{bmatrix} \nabla f + \partial g & \mathbf{L} \\ -\mathbf{L} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- which can be rewritten as

$$0 \in \underbrace{\begin{bmatrix} \nabla f & 0 \\ 0 & 0 \end{bmatrix}}_{:=F} \begin{bmatrix} x \\ y \end{bmatrix} + \underbrace{\begin{bmatrix} \partial g & \mathbf{L} \\ -\mathbf{L} & 0 \end{bmatrix}}_{:=M} \begin{bmatrix} x \\ y \end{bmatrix}$$

- can be rewritten as

$$\begin{aligned} x^{k+1} &= \mathbf{prox}_{\gamma g}\left(x^k - \gamma \nabla f(x^k) - \gamma \mathbf{L}(2y^{k+1} - y^k)\right) \\ y^{k+1} &= y^k - \tau \mathbf{L} x^k \end{aligned}$$

Derivation of Distributed Primal-dual gradient methods

- KKT conditions ($\mathbf{L} = (\mathbf{I} - \mathbf{W})^{\mathbf{1/2}}$)

$$0 \in \begin{bmatrix} \nabla f + \partial g & \mathbf{L} \\ -\mathbf{L} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- which can be rewritten as

$$0 \in \underbrace{\begin{bmatrix} \nabla f & 0 \\ 0 & 0 \end{bmatrix}}_{:=F} \begin{bmatrix} x \\ y \end{bmatrix} + \underbrace{\begin{bmatrix} \partial g & \mathbf{L} \\ -\mathbf{L} & 0 \end{bmatrix}}_{:=M} \begin{bmatrix} x \\ y \end{bmatrix}$$

- can be rewritten as

$$\begin{aligned} x^{k+1} &= \mathbf{prox}_{\gamma g}\left(x^k - \gamma \nabla f(x^k) - \gamma \mathbf{L}(2y^k - y^{k-1})\right) \\ y^{k+1} &= y^k - \tau \mathbf{L} x^{k+1} \end{aligned}$$

Derivation of Distributed Primal-dual gradient methods

- KKT conditions ($\mathbf{L} = (\mathbf{I} - \mathbf{W})^{\mathbf{1/2}}$)

$$0 \in \begin{bmatrix} \nabla f + \partial g & \mathbf{L} \\ -\mathbf{L} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- which can be rewritten as

$$0 \in \underbrace{\begin{bmatrix} \nabla f & 0 \\ 0 & 0 \end{bmatrix}}_{:=F} \begin{bmatrix} x \\ y \end{bmatrix} + \underbrace{\begin{bmatrix} \partial g & \mathbf{L} \\ -\mathbf{L} & 0 \end{bmatrix}}_{:=M} \begin{bmatrix} x \\ y \end{bmatrix}$$

- can be rewritten as ($\tau = 1/\gamma$)

$$\begin{aligned} x^{k+1} &= \mathbf{prox}_{\gamma g}\left(\mathbf{W}x^k - \gamma \nabla f(x^k) - \gamma \mathbf{L} y^k\right) \\ y^{k+1} &= y^k - 1/\gamma \mathbf{L} x^{k+1} \end{aligned}$$

Derivation of Distributed Primal-dual gradient methods

- KKT conditions ($\mathbf{L} = (\mathbf{I} - \mathbf{W})^{\mathbf{1/2}}$)

$$0 \in \begin{bmatrix} \nabla f + \partial g & \mathbf{L} \\ -\mathbf{L} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- which can be rewritten as

$$0 \in \underbrace{\begin{bmatrix} \nabla f & 0 \\ 0 & 0 \end{bmatrix}}_{:=F} \begin{bmatrix} x \\ y \end{bmatrix} + \underbrace{\begin{bmatrix} \partial g & \mathbf{L} \\ -\mathbf{L} & 0 \end{bmatrix}}_{:=M} \begin{bmatrix} x \\ y \end{bmatrix}$$

- can be rewritten as ($\tau = 1/\gamma, y'^k = \mathbf{L}y^k$)

$$\begin{aligned} x^{k+1} &= \mathbf{prox}_{\gamma g}\left(\mathbf{W}x^k - \gamma \nabla f(x^k) - \gamma y'^k\right) \\ y'^{k+1} &= y'^k - \tau \mathbf{L}^2 x^{k+1} \end{aligned}$$

Primal-dual distributed gradient method

ID-FBBS Algorithm

$$\mathbf{x}_{k+1} = \mathbf{prox}_{\gamma g}\left(\mathbf{W}\mathbf{x}_k - \gamma(\nabla f(\mathbf{x}_k) + \mathbf{y}_k)\right)$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \frac{1}{\gamma}(\mathbf{I} - \mathbf{W})\mathbf{x}_{k+1},$$

– $\mathbf{y}_k$ is the dual variable whose sum is **maintained at zero**.

1. **Initialization**: $\forall$ agent $i \in \mathcal{V}$: $x_{i,0}$ randomly assigned; $\sum_{i\in\mathcal{V}} y_{i,0} = 0$.
2. **Primal Update**: $\forall$ agent $i \in \mathcal{V}$, computes:

$$x_{i,k+1} = \mathbf{prox}_{\gamma g_i}\left(\sum_{j\in\mathcal{N}_i} w_{ij} x_{j,k} - \gamma(\nabla f_i(x_{i,k}) + y_{i,k})\right)$$

3. **Dual Update**: $\forall$ agent $i \in \mathcal{V}$, computes:

$$y_{i,k+1} = y_{j,k} + \frac{1}{\gamma}\sum_{j\in\mathcal{N}_i} w_{ij}(x_{i,k+1} - x_{j,k+1})$$

4. Set $k \to k+1$ and go to Step 2.

Connections to Existing Algorithms

- Recalling the ID-FBBS Algorithm

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \gamma(\nabla f(\mathbf{x}_k) + \mathbf{y}_k) \quad \text{(a)}$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \frac{1}{\gamma}(\mathbf{I} - \mathbf{W})\mathbf{x}_{k+1}, \quad \text{(b)}$$

- Let $\gamma \mathbf{y}_k = \sqrt{\mathbf{I} - \mathbf{W}}\mathbf{y}'_k$, the above algorithm can be rewritten as

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \gamma\nabla f(\mathbf{x}_k) - \sqrt{\mathbf{I} - \mathbf{W}}\mathbf{y}'_k$$

$$\mathbf{y}'_{k+1} = \mathbf{y}'_k + \sqrt{\mathbf{I} - \mathbf{W}}\mathbf{x}_{k+1}$$

- Equivalent to applying the Arrow-Hurwicz-Uzawa Method[2]

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k - \gamma\nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{y}'_k) \\ \mathbf{y}'_{k+1} = \mathbf{y}'_k + \gamma\nabla_{\mathbf{y}'} L(\mathbf{x}_{k+1}, \mathbf{y}') \end{cases}$$

  - where $L(\mathbf{x}, \mathbf{y}') = f(\mathbf{x}) + \frac{1}{\gamma}\mathbf{x}^T\sqrt{\mathbf{I} - \mathbf{W}}\mathbf{y}' + \frac{1}{2\gamma}\mathbf{x}^T(\mathbf{I} - \mathbf{W})\mathbf{x}$

[2] K.J. Arrow, L. Hurwicz, and H. Uzawa, Stanford University Press, 1958

Connections to Existing Algorithms

- Taking the augmented Lagrangian as follows:

$$L(\mathbf{x}, \mathbf{y}') = f(\mathbf{x}) + \frac{1}{\gamma}\mathbf{x}^T(\mathbf{I} - \mathbf{W})\mathbf{y}' + \frac{1}{2\gamma}\mathbf{x}^T(\mathbf{I} - \mathbf{W}^2)\mathbf{x},$$

Applying the Arrow-Hurwicz-Uzawa Method leads to

$$\mathbf{x}_{k+1} = \mathbf{W}^2\mathbf{x}_k - \gamma\nabla f(\mathbf{x}_k) - (\mathbf{I} - \mathbf{W})\mathbf{y}'_k \quad \text{(c)}$$

$$\mathbf{y}'_{k+1} = \mathbf{y}'_k + (\mathbf{I} - \mathbf{W})\mathbf{x}_{k+1} \quad \text{(d)}$$

- Evaluating (c) at $k+1$ and k, respectively and eliminating $\mathbf{y}'$ using (d), simple calculation gives

$$\mathbf{x}_{k+2} - \mathbf{W}\mathbf{x}_{k+1} = \mathbf{W}(\mathbf{x}_{k+1} - \mathbf{W}\mathbf{x}_k) + \gamma(\mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{g}(\mathbf{x}_k))$$

Let $\gamma\mathbf{y}_{k+1} = \mathbf{x}_{k+2} - \mathbf{W}\mathbf{x}_{k+1}$. Then, we recover

$$\text{the original AugDGM} \begin{cases} \mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k - \gamma\mathbf{y}_k \\ \mathbf{y}_{k+1} = \mathbf{W}\mathbf{y}_k + \mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{g}(\mathbf{x}_k). \end{cases}$$

A Unified Primal-Dual Framework

- Design a proper augmented Lagrangian:

$$L(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + \frac{1}{\gamma}\mathbf{x}^T\mathbf{A}\mathbf{y} + \frac{1}{2\gamma}\|\mathbf{x}\|_{\mathbf{B}}^2,$$

- Applying the Arrow-Hurwicz-Uzawa Method leads to

$$\mathbf{x}_{k+1} = (\mathbf{I} - \mathbf{B})\mathbf{x}_k - \gamma\nabla f(\mathbf{x}_k) - \mathbf{A}\mathbf{y}_k$$
$$\mathbf{y}_{k+1} = \mathbf{y}_k + \mathbf{A}\mathbf{x}_{k+1}$$

- Properly choose $\mathbf{A}$ and $\mathbf{B}$ such that consensus can be ensured, we can easily come up with new distributed algorithms
- What conditions on $\mathbf{A}, \mathbf{B}$ leads to convergence?

A Unified Algorithmic Framework

A unified ABC algorithm[3]

$$\mathbf{x}^{k+1} = \mathbf{A}\mathbf{x}^k - \gamma \mathbf{B} \nabla f(\mathbf{x}^k) - \mathbf{y}^k,$$
$$\mathbf{y}^{k+1} = \mathbf{y}^k + \mathbf{C}\mathbf{x}^{k+1},$$

– where $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are three weight matrices to be properly defined.

The above unified algorithm subsumes many existing algorithms.

Algorithm	$\mathbf{A}$	$\mathbf{B}$	$\mathbf{C}$
ID-FBBS/EXTRA	$\frac{1}{2}(\mathbf{I}+\mathbf{W})$	$\mathbf{I}$	$\frac{1}{2}(\mathbf{I}-\mathbf{W})$
NIDS/Exact Diffusion	$\frac{1}{2}(\mathbf{I}+\mathbf{W})$	$\frac{1}{2}(\mathbf{I}+\mathbf{W})$	$\frac{1}{2}(\mathbf{I}-\mathbf{W})$
AugDGM/NEXT	$\mathbf{W}^2$	$\mathbf{W}^2$	$(\mathbf{I}-\mathbf{W})^2$
DIGing/Harnessing	$\mathbf{W}^2$	$\mathbf{I}$	$(\mathbf{I}-\mathbf{W})^2$

[3] [Xu et al, IEEE TSP'21]

Sublinear Convergence Rate

Let $\mathbb{S}^m$ be the set of $m \times m$ symmetric matrices.

- Assumptions
  - Cost function $\{f_i\}$: L-smooth;
  - Weight Matrix:
    - i) $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{S}^m$ and $\mathbf{C} \succeq 0$,
    - ii) $\mathbf{A} = \mathbf{B}$, $\mathbf{BC} = \mathbf{CB}$, $0 \preceq \mathbf{A} \preceq \mathbf{I}$,
    - iii) $\mathbf{span}\{\mathbf{1}\} = \mathbf{null}\{\mathbf{C}\} \subseteq \mathbf{null}\{\mathbf{I} - \mathbf{A}\}$.

Theorem (Sublinear rate for the unified algorithm)

Let $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k \geq 0}$ be the iterates generated by the above algorithm with $\mathbf{1}^T \mathbf{y}_0 = 0$. Suppose the above Assumptions hold. Then, if $\gamma = \frac{1}{L}$, the algorithm converges at a sublinear rate of

$$\max \left\{ \frac{L \left\| \mathbf{x}^0 - \mathbf{x}^\star \right\|^2}{k+1}, \frac{1}{\sqrt{\eta(\mathbf{C})}} \frac{\left\| \mathbf{x}^0 - \mathbf{x}^\star \right\| \left\| \nabla f(\mathbf{x}^\star) \right\|}{k+1} \right\},$$

where $\eta(\mathbf{C}) := \frac{\lambda_{\min}(\mathbf{C})}{\lambda_{\max}(\mathbf{C})}$ denotes the eigengap of the matrix $\mathbf{C}$.

Some Observations

The convergence rate has the following structure[4]

$$\max\left\{\underbrace{\frac{L\left\|\mathbf{x}^0-\mathbf{x}^\star\right\|^2}{k+1}}_{\text{computation}}, \underbrace{\frac{1}{\sqrt{\eta(\mathbf{C})}}\frac{\left\|\mathbf{x}^0-\mathbf{x}^\star\right\|\left\|\nabla f(\mathbf{x}^\star)\right\|}{k+1}}_{\text{communication}}\right\} \overset{\mathbf{g}(\mathbf{x}^\star)=0}{\Rightarrow} \underbrace{\mathcal{O}\left(\frac{L\left\|\mathbf{x}^0-\mathbf{x}^\star\right\|^2}{k+1}\right)}_{\text{centralized rate}}.$$

- $1/\sqrt{\eta} \approx$ the diameter of the network for simple networks, e.g., line graphs
- $\|\nabla f(\mathbf{x}^\star)\|$ encodes the "heterogeneity" of functions; $\mathbf{g}(\mathbf{x}^\star)=0$ implies
  - **Case 1**: When all agents share common solution, e.g., the distribution of all local data sets are similar.
  - **Case 2**: When a spanning tree algorithm is employed, e.g, exact average of local data, e.g., local gradients.
- The algorithm reduces to the centralized one!

[4] Refer to [Xu et al, AISTATS'20; TSP'21] for more details.

Linear Convergence Rate

Let $\mathbb{S}^m$ be the set of $m \times m$ symmetric matrices.

- Assumptions
  - Cost function $\{f_i\}$: L-smooth and μ-strongly convex;
  - Weight Matrix:
    - i) $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{S}^m$ and $\mathbf{C} \succeq 0$,
    - ii) $\mathbf{A} = \mathbf{B}$, $\mathbf{BC} = \mathbf{CB}$, $\mathbf{B}^2 \preceq \mathbf{I} - \mathbf{C}$,
    - iii) $\mathbf{span}\{\mathbf{1}\} = \mathbf{null}\{\mathbf{C}\} \subseteq \mathbf{null}\{\mathbf{I} - \mathbf{A}\}$.

Theorem (Linear rate for the unified algorithm)

Let $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k \geq 0}$ be the iterates generated by the above algorithm with $\mathbf{1}^T \mathbf{y}_0 = 0$. Suppose the above Assumptions hold. Then, if $\gamma = \frac{2}{L+\mu}$, the algorithm converges at a linear rate of $\mathcal{O}(\sigma^k)$ with

$$\sigma = \max\left\{\frac{\kappa - 1}{\kappa + 1}, 1 - \lambda_{\min}(\mathbf{C})\right\},$$

where $\lambda_{\min}(\mathbf{C})$ denotes the connectivity of the graph.

Simulation Setting

A Canonical Example of Distributed Estimation

- Overall loss function

$$F = \sum_{i=1}^{m} \left(\|z_i - M_i\theta\|^2 + \lambda_i \|\theta\|_1 \right)$$

  - $M_i \in \mathcal{R}^{s\times d}$: measurement matrix
  - z_i: noisy observation of agent i
  - λ_i: regularization parameter.

- Metropolis-Hastings protocol[5]

$$w_{ij} = \begin{cases} \frac{1}{2\cdot\max\{d_i, d_j\}}, & \text{if } (i,j) \in \mathcal{E} \\ 1 - \sum_{j\in\mathcal{N}_i} w_{ij}, & \text{if } i = j \\ 0, & \text{otherwise,} \end{cases}$$

  - d_i: the degree of agent i.

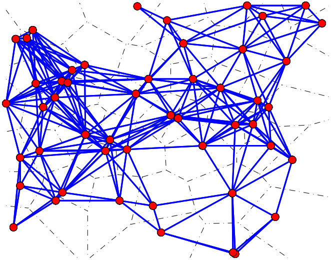


Figure: A random network of 50 nodes

[5] slightly modified to ensure the positivity.

Performance Evaluation

Parameter Setting: $d = 10, s = 1, m = 50, \lambda_i = 0.02,\ \forall i \in \mathcal{V}$;
$M_i \in \mathcal{R}^{r \times d}$: a uniform distribution; Gaussian Noise: $\mathcal{N}(0, 0.1)$

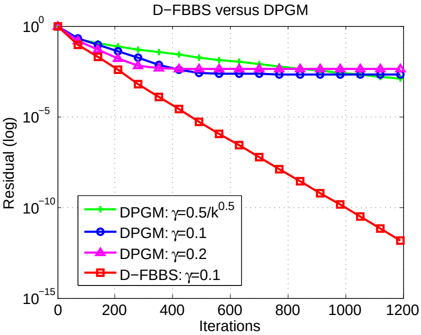


Figure: FPR ($e = \frac{\|x_k - x^*\|^2}{\|x_0 - x^*\|^2}$) Versus Iterations

References

- Boyd, Stephen, and Lieven Vandenberghe. *Convex optimization*. Cambridge University press, 2004.
- Dimitri P., Bertsekas, Angelia, Nedich and Asuman E., Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.
- Angelia, Nedich. *Lecture Notes for Convex Optimization*. University of Illinois Urbana-Champaign, 2008.
- Ryan Tibshirani, *Lecture Notes for Convex Optimzation*. Carnegie Mellon University, 2018.
- Yuxin Chen, *Lecture Notes for Large-Scale Optimization for Data Science*. Princeton University, 2018.
- Bubeck, Sébastien. (2014). Theory of Convex Optimization for Machine Learning.
- Emmanuel Candes. (2015). *Lecture Notes for Advanced Topics in Convex Optimizationg*. Stanford University, 2015.