The Proximal Primal-Dual Approach for Nonconvex Linearly Constrained Problems

Presenter: Mingyi Hong

Joint work with Davood Hajinezhad

University of Minnesota
ECE Department

DIMACS Workshop on Distributed Opt., Information Process., and Learning
August, 2017

Agenda

- We consider the following problem

$$\begin{aligned} \min \quad & f(x) + h(x) \qquad \text{(P)} \\ \text{s.t.} \quad & Ax = b,\ x \in X \end{aligned}$$

- $f(x) : \mathbb{R}^N \to \mathbb{R}$ is a smooth non-convex function
- $h(x) : \mathbb{R}^N \to \mathbb{R}$ is a nonsmooth non-convex regularizer
- X is a compact convex set, and $\{x \mid Ax = b\} \cap X \neq \emptyset$.

The Plan

1. Design an efficient decomposition scheme decoupling the variables
2. Analyze convergence/rate of convergence
3. Discuss convergence to first/second-order stationary solutions
4. Explore different variants of the algorithms; obtain useful insights
5. Evaluate practical performance

App 1: Distributed optimization

- Consider a network consists of N agents, who collectively optimize

$$\boxed{\min_{y\in X}\ f(y) := \sum_{i=1}^{N} f_i(y) + h_i(y),}$$

where $f_i(y), h_i(y) : X \to \mathbb{R}$ is cost/regularizer for local to agent i

- Each f_i, h_i is only known to agent i (e.g., through local measurements)
- y is assumed to be scalar for ease of presentation
- Agents are connected by a network defined by an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, with $|\mathcal{V}| = N$ vertices and $|\mathcal{E}| = E$ edges

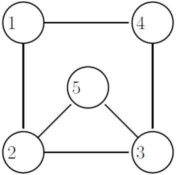
App 1: Distributed optimization

- Introduce local variables $\{x_i\}$, reformulate to the consensus problem

$$
\boxed{
\begin{aligned}
\min_{\{x_i\}} \quad & \sum_{i=1}^{N} f_i(x_i) + h_i(x_i) \\
\text{s.t.} \quad & Ax = 0 \qquad \text{(consensus constraint)}
\end{aligned}
}
$$

where $A \in \mathbb{R}^{E\times N}$ is the edge-node incidence matrix; $x := [x_1, \cdots, x_N]^T$

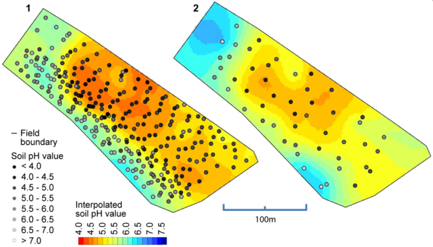
- If $e \in \mathcal{E}$ and it connects vertex i and j with $i > j$, then $A_{ev} = 1$ if $v = i$, $A_{ev} = -1$ if $v = j$ and $A_{ev} = 0$ otherwise.



$$
A = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix}.
$$

App 2: Partial consensus

- "Strict consensus" may not be practical; often not required [Koppel et al 16]
  1. Due to noises in local communication
  2. The variables to be estimated has spatial variability
  3. ....



App 2: Partial consensus

- **Relax** the consensus requirement

$$\min_{i} \quad \sum_{i=1}^{N} f_i(x_i) + h_i(x_i)$$
$$\text{s.t.} \quad \|x_i - x_j\|^2 \le b_{ij}, \quad \forall (i,j) \in E.$$

- Introduce "link variable" $\{z_{ij} = x_i - x_j\}$; Equivalent reformulation

$$\min_{i} \quad \sum_{i=1}^{N} f_i(x_i) + h_i(x_i)$$
$$\text{s.t.} \quad Ax - z = 0, \quad z \in Z$$

App 2: Partial consensus

- The local cost functions can be non-convex in a number of situations
  1. The use of non-convex regularizers, e.g., SCAD/MCP [Fan-Li 01, Zhang 10]
  2. Non-convex quadratic functions, e.g., high-dimensional regression with missing data [Loh-Wainwright 12], sparse PCA
  3. Sigmoid loss function (approximating 0-1 loss) [Shalev-Shwartz et al 11]
  4. Loss function for training neural nets [Allen-Zhu-Hazan 16]

App 3: Non-convex subspace estimation

- Let $\Sigma \in \mathbb{R}^{p\times p}$ be an unknown covariance matrix, with eigen-decomposition

$$\Sigma = \sum_{i=1}^{p} \lambda_i \mathbf{u}_i \mathbf{u}_i^T$$

where $\lambda_1 \geq \cdots \geq \lambda_p$ are eigenvalues; $\mathbf{u}_1, \cdots, \mathbf{u}_p$ are eigenvectors

- The k-dimensional principal subspace of Σ

$$\Pi^* = \sum_{i=1}^{k} \lambda_i \mathbf{u}_i \mathbf{u}_i^T = UU^T$$

- **Principal subspace estimation.** Given i.i.d samples $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, estimate Π^*, based on sample covariance matrix $\widehat{\Sigma}$

App 3: Non-convex subspace estimation

- Problem formulation [Gu et al 14]

$$\widehat{\Pi} = \arg\min_{\Pi} \quad -\langle \widehat{\Sigma}, \Pi \rangle + P_\alpha(\Pi)$$
$$\text{s.t.} \quad 0 \preceq \Pi \preceq I, \; \mathrm{Tr}(\Pi) = k. \quad \text{(Fantope set)}$$

where $P_\alpha(\Pi)$ is a non-convex regularizer (such as MCP/SCAD)

- **Estimation result.** [Gu et al 14] Under certain condition on α, every first-order stationary solution is "good", with high probability:

$$\|\widehat{\Pi} - \Pi^*\|_F \le s_1 \sqrt{\frac{s}{n}} + s_2 \sqrt{\frac{\log(p)}{n}}$$

- $s = |\mathrm{supp}(\mathrm{diag}(\Pi^*))|$ is the subspace sparsity [Vu et al 13]

App 3: Non-convex subspace estimation

- **Question.** How to find first-order stationary solution?
- Need to deal with both the Fantope and non-convex regularizer $P_\alpha(\Pi)$
- A heuristic approach proposed in [Gu et al 14]
  1. Introduce linear constraint $X = \Pi$
  2. Impose non-convex regularizer on X, Fantope constraint on Π

$$\begin{aligned} \widehat{\Pi} = \arg\min_{\Pi} \quad & -\langle \widehat{\Sigma}, \Pi \rangle + P_\alpha(X) \\ \text{s.t.} \quad & 0 \preceq \Pi \preceq I, \ \mathrm{Tr}(\Pi) = k. \quad \text{(Fantope set)} \\ & \Pi - X = 0 \end{aligned}$$

  3. Same formulation as (P), only heuristic algorithm without any guarantee

The literature

Literature

- The Augmented Lagrangian (AL) methods [Hestenes 69, Powell 69], is a classical algorithm for solving nonlinear non-convex constrained problems
- Many existing packages (e.g., LANCELOT)
- Recent developments [Curtis et al 16] [Friedlander 05], and many more
- Convex problem + linear constraints, [Lan-Monterio 15] [Liu et al 16] analyzed the iteration complexity for the AL method
- Requires double-loop
- In the non-convex setting difficult to handle non-smooth regularizers
- Difficult to be implemented in a distributed manner

Literature

- Recent works consider AL-type methods for linearly constrained problems
- Nonconvex problem + linear constraints, [Artina-Fornasier-Solombrino 13]
    1. Approximate the Augmented Lagrangian using proximal point (make it convex)
    2. Solve the linearly constrained convex approximation with increasing accuracy
- AL based methods for smooth non-convex objective + linearly coupling constraints [Houska-Frasch-Diehl 16]
    1. AL based Alternating Direction Inexact Newton (ALADIN)
    2. Combines SQP and AL, global line search, Hessian computation, etc.
- Still requires double-loop
- No global rate analysis

Literature

- Dual decomposition [Bertsekas 99]
  1. Gradient/subgradient applied to the dual
  2. Convex separable objective + convex coupling constraints
  3. Lots of application, e.g., in wireless communications [Palomar-Chiang 06]
- Arrow-Hurwicz-Uzawa primal-dual algorithm [Arrow-Hurwicz-Uzawa 58]
  1. Applied to study saddle point problems [Gol'shtein 74][Nedić-Ozdaglar 07]
  2. Primal-dual hybrid gradient [Zhu-Chan 08]
  3. ...
- Do not to work for non-convex problem (difficult to use the dual structure)

Literature

- ADMM is popular in solving linearly constrained problems
- Some theoretical results for applying ADMM for non-convex problems
  1. [Hong-Luo-Razaviyayn 14]: non-convex consensus and sharing
  2. [Li-Pong 14], [Wang-Yin-Zeng 15], [Melo-Monterio 17] with more relaxed conditions, or faster rates
  3. [Pang-Tao 17] for non-convex DC program with sharp stationary solutions
- Block-wise structure, but requires a special block
- Does not apply to problem (P)

The plan of the talk

- First consider the simpler problem (unconstrained, smooth)

$$\min_{x\in\mathbb{R}^N} f(x), \quad \text{s.t. } Ax = b \qquad \text{(Q)}$$

- Algorithm, analysis and discussion

- First-/second order stationarity

- Then generalize

- Applications and numerical results

The proposed algorithms

The proposed algorithm

- We draw elements form AL and Uzawa methods
- The augmented Lagrangian for problem (P) is given by

$$L_\beta(x,\mu) = f(x) + \langle \mu, Ax - b\rangle + \frac{\beta}{2}\|Ax - b\|^2$$

where $\mu \in \mathbb{R}^M$ dual variable; $\beta > 0$ penalty parameter

- One primal gradient-type step + one dual gradient-type step

The proposed algorithm

- Let $B \in \mathbb{R}^{M\times N}$ be some arbitrary matrix to be defined later
- The proposed Proximal Primal Dual Algorithm is given below

Algorithm 1. The Proximal Primal Dual Algorithm (Prox-PDA)

At iteration 0, initialize μ^0 and $x^0 \in \mathbb{R}^N$.
At each iteration $r+1$, update variables by:

$$x^{r+1} = \arg\min_{x\in\mathbb{R}^n} \langle \nabla f(x^r), x - x^r\rangle + \langle \mu^r, Ax - b\rangle + \frac{\beta}{2}\|Ax-b\|^2 + \frac{\beta}{2}\|x - x^r\|^2_{B^TB}; \tag{1a}$$

$$\mu^{r+1} = \mu^r + \beta(Ax^{r+1} - b). \tag{1b}$$

Comments

- The primal iteration has to choose the proximal term

$$\frac{\beta}{2}\|x - x^r\|^2_{B^T B}$$

- Choose B appropriately to ensure the following key properties:
    1. The primal problem is strongly convex, hence easily solvable;
    2. The primal problem is decomposable over different variable blocks.

- We illustrate this point using the distributed optimization problem
- A network consists of 3 users: $1 \leftrightarrow 2 \leftrightarrow 3$
- Define the signed graph Laplacian as $L_- = A^T A \in \mathbb{R}^{N\times N}$
- Its (i,i)th diagonal entry is the degree of node i, and its (i,j)th entry is -1 if $e=(i,j)\in\mathcal{E}$, and 0 otherwise.

$$L_- = \begin{bmatrix} 1 & -1 & 0\\ -1 & 2 & -1\\ 0 & -1 & 1\end{bmatrix}, \qquad L_+ = \begin{bmatrix} 1 & 1 & 0\\ 1 & 2 & 1\\ 0 & 1 & 1\end{bmatrix}$$

- Define the signless incidence matrix $B := |A|$
- Using this choice of B, we have $B^T B = L_+ \in \mathbb{R}^{N\times N}$, which is the signless graph Laplacian

Comments

- We illustrate this point using the distributed optimization problem
- A network consists of 3 users: $1 \leftrightarrow 2 \leftrightarrow 3$
- Define the signed graph Laplacian as $L_{-} = A^T A \in \mathbb{R}^{N \times N}$
- Its (i,i)th diagonal entry is the degree of node i, and its (i,j)th entry is -1 if $e = (i,j) \in \mathcal{E}$, and 0 otherwise.

$$L_{-} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \qquad L_{+} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

- Define the signless incidence matrix $B := |A|$
- Using this choice of B, we have $B^T B = L_{+} \in \mathbb{R}^{N \times N}$, which is the signless graph Laplacian

Comments

- We illustrate this point using the distributed optimization problem
- A network consists of 3 users: $1 \leftrightarrow 2 \leftrightarrow 3$
- Define the signed graph Laplacian as $L_- = A^T A \in \mathbb{R}^{N \times N}$
- Its (i,i)th diagonal entry is the degree of node i, and its (i,j)th entry is -1 if $e = (i,j) \in \mathcal{E}$, and 0 otherwise.

$$L_- = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \qquad L_+ = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

- Define the signless incidence matrix $B := |A|$
- Using this choice of B, we have $B^T B = L_+ \in \mathbb{R}^{N \times N}$, which is the signless graph Laplacian

Comments

- We illustrate this point using the distributed optimization problem
- A network consists of 3 users: $1 \leftrightarrow 2 \leftrightarrow 3$
- Define the signed graph Laplacian as $L_- = A^T A \in \mathbb{R}^{N\times N}$
- Its (i,i)th diagonal entry is the degree of node i, and its (i,j)th entry is -1 if $e = (i,j) \in \mathcal{E}$, and 0 otherwise.

$$L_- = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \qquad L_+ = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

- Define the signless incidence matrix $B := |A|$
- Using this choice of B, we have $B^T B = L_+ \in \mathbb{R}^{N\times N}$, which is the signless graph Laplacian

Comments

- We illustrate this point using the distributed optimization problem
- A network consists of 3 users: $1 \leftrightarrow 2 \leftrightarrow 3$
- Define the signed graph Laplacian as $L_{-} = A^T A \in \mathbb{R}^{N\times N}$
- Its (i,i)th diagonal entry is the degree of node i, and its (i,j)th entry is -1 if $e=(i,j)\in\mathcal{E}$, and 0 otherwise.

$$L_{-} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \qquad L_{+} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

- Define the signless incidence matrix $B := |A|$
- Using this choice of B, we have $B^T B = L_{+} \in \mathbb{R}^{N\times N}$, which is the signless graph Laplacian

Comments

- We illustrate this point using the distributed optimization problem
- A network consists of 3 users: $1 \leftrightarrow 2 \leftrightarrow 3$
- Define the signed graph Laplacian as $L_- = A^T A \in \mathbb{R}^{N\times N}$
- Its (i,i)th diagonal entry is the degree of node i, and its (i,j)th entry is -1 if $e=(i,j)\in\mathcal{E}$, and 0 otherwise.

$$L_- = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \qquad L_+ = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

- Define the signless incidence matrix $B := |A|$
- Using this choice of B, we have $B^T B = L_+ \in \mathbb{R}^{N\times N}$, which is the signless graph Laplacian

Comments

- Then x-update step becomes

$$
\begin{aligned}
x^{r+1} &= \arg\min_x \sum_{i=1}^N \langle \nabla f_i(x_i^r), x_i\rangle + \langle \mu^r, Ax-b\rangle + \frac{\beta}{2}x^T L_- x + \underbrace{\frac{\beta}{2}(x-x^r)^T L_+ (x-x^r)}_{\text{proximal term}} \\
&= \arg\min_x \sum_{i=1}^N \langle \nabla f_i(x_i^r), x_i\rangle + \langle \mu^r, Ax-b\rangle + \frac{\beta}{2}x^T (L_- + L_+) x - \beta x^T L_+ x^r \\
&= \arg\min_x \underbrace{\sum_{i=1}^N \langle \nabla f_i(x_i^r), x_i\rangle + \langle \mu^r, Ax-b\rangle - \beta x^T L_+ x^r}_{\text{linear in } x} + \beta x^T D x
\end{aligned}
$$

- $D = \text{diag}[d_1, \cdots, d_N] \in \mathbb{R}^{N\times N}$ is the degree matrix
- The problem is separable over the nodes, and strongly convex.

The analysis steps

Main assumptions

A1. $f(x)$ differentiable and has Lipschitz continuous gradient, i.e.,

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall\ x, y \in \mathbb{R}^N.$$

Further assume that $A^T A + B^T B \succeq I_N$.

A2. There exists a constant $\delta > 0$ such that

$$\exists \underline{f} > -\infty, \quad \text{s.t. } f(x) + \frac{\delta}{2}\|Ax - b\|^2 \geq \underline{f},\ \forall\ x \in \mathbb{R}^N.$$

A3. The constraint $Ax = b$ is feasible over $x \in \mathbb{R}^N$.

Functions satisfying the assumptions

- **The sigmoid function**. The sigmoid function is given by

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}} \in [-1,\ 1].$$

- **The** arctan **function**. $\arctan(x) \in [-1, 1]$ so [A2] is ok. $\arctan'(x) = \frac{1}{x^2+1} \in [0,\ 1]$ so it is bounded, which implies that [A1] is true.
- **The** tanh **function**. Note that we have

$$\tanh(x) \in [-1, 1], \quad \tanh'(x) = 1 - \tanh(x)^2 \in [0, 1].$$

- **The** logit **function**. The logistic function is related to the tanh as

$$2\text{logit}(x) = \frac{2e^x}{e^x+1} = 1 + \tanh(x/2).$$

- **The quadratic function $x^T Q x$.** Suppose Q is symmetric but not necessarily positive semidefinite, and $x^T Q x$ is strongly convex in the null space of $A^T A$.

Optimality Conditions

- The first and second order necessary condition for local min is given as

$$\nabla f(x^*) + \langle \mu^*, A \rangle = 0, \quad Ax^* = b. \tag{2a}$$

$$\langle y, \nabla^2 f(x^*) y \rangle \geq 0, \quad \forall\ y \in \{y \mid Ay = 0\}. \tag{2b}$$

- The second-order necessary condition is equivalent to the condition that $\nabla^2 f(x^*)$ is positive semi-definite in the null space of A

- Sufficient condition for strict/strong local minimizer is given by

$$\begin{aligned} &\nabla f(x^*) + \langle \mu^*, A \rangle = 0, \quad Ax^* = b. \\ &\langle y, \nabla^2 f(x^*) y \rangle > 0, \quad \forall\ y \in \{y \mid Ay = 0\}. \end{aligned} \tag{3}$$

Optimality Conditions

- Define a strict saddle point to be the solution (x^*, μ^*) such that

$$\begin{aligned} &\nabla f(x^*) + \langle \mu^*, A\rangle = 0, \quad Ax^* = b, \\ &\exists\, y \in \{y \mid Ay = 0\}, \text{ and } \sigma > 0 \text{ such that } \langle y, \nabla^2 f(x^*) y\rangle < 0. \end{aligned} \tag{4}$$

- Has strictly negative "eigenvalue" in the null space of A.
- Issues related to strict saddles have been brought up recently in ML communities; see recent works [Ge *et al* 15] [Sun-Qu-Wright 15]
- GD-type algorithms have been developed, but mostly in unconstrained and smooth setting [Lee et al 16] [Jin et al 17]
- **Question.** Prox-PDA converges to strict saddle, 2nd-order stationary sols?

The Analysis: Step 1

- Our first step bounds the descent of the augmented Lagrangian

- **Observation**. Dual variable is given as

$$A^T\mu^{r+1} = -\nabla f(x^r) - \beta B^T B(x^{r+1} - x^r)$$

- Change of dual can be bounded by change of primal

The Analysis: Step 1

- Let $\sigma_{\min}(A^T A)$ be the smallest non-zero eigenvalue for $A^T A$

Lemma

Suppose Assumptions [A1] and [A3] are satisfied. Then the following is true

$$
\begin{aligned}
L_\beta(x^{r+1},\mu^{r+1}) - L_\beta(x^r,\mu^r) \le & -\left(\frac{\beta - L}{2} - \frac{2L^2}{\beta\sigma_{\min}(A^T A)}\right)\|x^{r+1}-x^r\|^2 \\
& + \frac{2\beta\|B^T B\|}{\sigma_{\min}(A^T A)}\left\|(x^{r+1}-x^r)-(x^r-x^{r-1})\right\|^2_{B^T B}.
\end{aligned}
$$

Comments

- The rhs cannot be made negative
- The AL alone does not descend
- Need a new object that is decreasing in the order of

$$\beta \left\| (x^{r+1} - x^r) - (x^r - x^{r-1}) \right\|^2_{B^T B}$$

- The change of the sum of the constraint violation $\|Ax^{r+1} - b\|^2$ and the proximal term $\|x^{r+1} - x^r\|^2_{B^T B}$ has the desired term.

The Analysis: Step 2

Lemma

Suppose Assumption [A1] is satisfied. Then the following is true

$$
\begin{aligned}
&\frac{\beta}{2}\left(\|Ax^{r+1}-b\|^2+\|x^{r+1}-x^r\|^2_{B^TB}\right)\\
&\le \frac{\beta}{2}\left(\|x^r-x^{r-1}\|^2_{B^TB}+\|Ax^r-b\|^2\right)+L\|x^{r+1}-x^r\|^2\\
&\quad -\frac{\beta}{2}\left(\|(x^r-x^{r-1})-(x^{r+1}-x^r)\|^2_{B^TB}+\|A(x^{r+1}-x^r)\|^2\right).
\end{aligned}
$$

- **Observation.** The new object, $\beta/2\left(\|Ax^{r+1}-b\|^2+\|x^{r+1}-x^r\|^2_{B^TB}\right)$, increases in $\|x^{r+1}-x^r\|^2$ and decreases in $\|(x^r-x^{r-1})-(x^{r+1}-x^r)\|^2_{B^TB}$
- The change of AL behaves in an opposite manner
- **Good news.** A conic combination of the two decreases at every iteration.

The Analysis: Step 2

Lemma

Suppose Assumption [A1] is satisfied. Then the following is true

$$
\begin{aligned}
&\frac{\beta}{2}\left(\|Ax^{r+1}-b\|^2+\|x^{r+1}-x^r\|^2_{B^TB}\right)\\
&\le \frac{\beta}{2}\left(\|x^r-x^{r-1}\|^2_{B^TB}+\|Ax^r-b\|^2\right)+L\|x^{r+1}-x^r\|^2\\
&\quad -\frac{\beta}{2}\left(\|(x^r-x^{r-1})-(x^{r+1}-x^r)\|^2_{B^TB}+\|A(x^{r+1}-x^r)\|^2\right).
\end{aligned}
$$

- **Observation.** The new object, $\beta/2\left(\|Ax^{r+1}-b\|^2+\|x^{r+1}-x^r\|^2_{B^TB}\right)$, increases in $\|x^{r+1}-x^r\|^2$ and decreases in $\|(x^r-x^{r-1})-(x^{r+1}-x^r)\|^2_{B^TB}$
- The change of AL behaves in an opposite manner
- **Good news.** A conic combination of the two decreases at every iteration.

The Analysis: Step 2

Lemma

Suppose Assumption [A1] is satisfied. Then the following is true

$$
\begin{aligned}
&\frac{\beta}{2}\left(\|Ax^{r+1}-b\|^2+\|x^{r+1}-x^r\|^2_{B^TB}\right)\\
&\le \frac{\beta}{2}\left(\|x^r-x^{r-1}\|^2_{B^TB}+\|Ax^r-b\|^2\right)+L\|x^{r+1}-x^r\|^2\\
&\quad -\frac{\beta}{2}\left(\|(x^r-x^{r-1})-(x^{r+1}-x^r)\|^2_{B^TB}+\|A(x^{r+1}-x^r)\|^2\right).
\end{aligned}
$$

- **Observation.** The new object, $\beta/2\left(\|Ax^{r+1}-b\|^2+\|x^{r+1}-x^r\|^2_{B^TB}\right)$, increases in $\|x^{r+1}-x^r\|^2$ and decreases in $\|(x^r-x^{r-1})-(x^{r+1}-x^r)\|^2_{B^TB}$
- The change of AL behaves in an opposite manner
- **Good news.** A conic combination of the two decreases at every iteration.

Step 3: Constructing the potential function

- Let us define the potential function for Algorithm 1 as

$$P_{c,\beta}(x^{r+1}, x^r, \mu^{r+1}) = L_\beta(x^{r+1}, \mu^{r+1}) + \frac{c\beta}{2}\left(\|Ax^{r+1} - b\|^2 + \|x^{r+1} - x^r\|^2_{B^T B}\right)$$

where $c > 0$ is some constant to be determined later.

Lemma

Suppose the assumptions made in Lemma 2 are satisfied. Then we have

$$\begin{aligned} P_{c,\beta}(x^{r+1}, x^r, \mu^{r+1}) \le P_{c,\beta}(x^r, x^{r-1}, \mu^r) &- \left(\frac{\beta - L}{2} - \frac{2L^2}{\beta\sigma_{\min}(A^T A)} - cL\right)\|x^{r+1} - x^r\|^2 \\ &- \left(\frac{c\beta}{2} - \frac{2\beta\|B^T B\|}{\sigma_{\min}(A^T A)}\right)\left\|(x^{r+1} - x^r) - (x^r - x^{r-1})\right\|^2_{B^T B}. \end{aligned}$$

The choice of parameters

- As long as c and β are chosen appropriately, the function $P_{c,\beta}$ decreases at each iteration of Prox-PDA
- The following choices of parameters are sufficient for ensuring descent

$$c \geq \max\left\{\frac{\delta}{L}, \frac{4\|B^T B\|}{\sigma_{\min}(A^T A)}\right\}. \tag{5}$$

- The β satisfies

$$\beta > \frac{L}{2}\left(2c+1+\sqrt{(2c+1)^2+\frac{16L^2}{\sigma_{\min}(A^T A)}}\right). \tag{6}$$

Step 4: main result

- Now we are ready to present the main result
- Define $Q(x^{r+1}, \mu^{r+1})$ as the 'stationarity gap' of problem (P)

$$Q(x^{r+1}, \mu^r) := \underbrace{\|\nabla_x L_\beta(x^{r+1}, \mu^r)\|^2}_{\text{primal gap}} + \underbrace{\|Ax^{r+1} - b\|^2}_{\text{dual gap}}.$$

- $Q(x^{r+1}, \mu^r) \to 0$ implies that the limit point (x^*, μ^*) is a 1st order sol of (P)

$$0 = \nabla f(x^*) + A^T \mu^*, \quad Ax^* = b.$$

The main result

Claim (H. - 16)

Suppose Assumption A is satisfied. Further suppose that the conditions on β and c in (5) and (6) are satisfied. Then

1. **(Eventual Feasibility).** *The constraint is satisfied in the limit, i.e.,*

$$\lim_{r\to\infty} \mu^{r+1} - \mu^r \to 0,\ \lim_{r\to\infty} Ax^r \to b, \text{ and } \lim_{r\to\infty} x^{r+1} - x^r = 0.$$

2. **(Convergence to KKT).** *Every limit point of $\{x^r, \mu^r\}$ converges to a KKT point of problem (P). Further, $Q(x^{r+1}, \mu^r) \to 0$.*

3. **(Sublinear Convergence Rate).** *For any given $\varphi > 0$, let us define T to be the first time that the optimality gap reaches below φ, i.e.,*

$$T := \arg\min_r Q(x^{r+1}, \mu^r) \leq \varphi.$$

Then there exists a constant $\nu > 0$ such that $\varphi \leq \frac{\nu}{T-1}$.

Extension: Increasing the proximal parameter

- The previous algorithm requires to explicitly compute the bound for β
- Requires global information; Alternatives?

Algorithm 2. The Prox-PDA with Increasing Proximal (Prox-PDA-IP)

At iteration 0, initialize μ^0 and $x^0 \in \mathbb{R}^N$.
At each iteration $r+1$, update variables by:

$$
\begin{aligned}
x^{r+1} &= \arg\min_{x\in\mathbb{R}^n} \langle \nabla f(x^r), x^r\rangle + \langle \mu^r, Ax-b\rangle \\
&\quad + \frac{\beta^{r+1}}{2}\|Ax-b\|^2 + \frac{\beta^{r+1}}{2}\|x-x^r\|^2_{B^TB} \\
\mu^{r+1} &= \mu^r + \beta^{r+1}(Ax^{r+1}-b).
\end{aligned}
$$

Extension: Increasing the proximal parameter

- Primal step similar to the classic GD with diminishing primal stepsize $1/\beta^r$ [Bertsekas-Tsitsiklis 96]

- The term β^r should satisfy the following conditions

$$\frac{1}{\beta^r} \to 0, \quad \sum_{r=1}^{\infty} \frac{1}{\beta^r} = \infty.$$

- Proof requires construction of a new potential function

$$L_{\beta^{r+1}}(x^{r+1}, \mu^{r+1}) + \frac{c\beta^{r+1}\beta^r}{2}\|Ax^{r+1} - b\|^2 + \frac{c\beta^{r+1}\beta^r}{2}\|x^r - x^{r+1}\|^2_{B^TB}.$$

- Similar convergence as Claim 1. (1)-(2); The rate (for a randomized version)

$$\mathbb{E}[Q(x^T, \mu^T)] \in \mathcal{O}\left(T^{-1/3}\right).$$

Second order stationary solutions?

- So far we have been focused on convergence (rate) on the 1st order solutions
- Will prox-PDA stuck at strict saddle points?
- We can show that with probability 1 this will not happen.

Claim (H.-Razaviyayn-Lee 17)

Under the same assumption as in the previous claim, and further suppose that (x^0, λ^0) are initialized randomly. Then with probability one, the iterates $\{(x^{r+1}, \mu^{r+1})\}$ generated by the Prox-PDA algorithm converges to a second-order stationary solution satisfying (2b).

Proof steps

- First represent the iterates using a linear system

$$\begin{bmatrix} x^{r+1} \\ x^r \end{bmatrix} = \begin{bmatrix} 2I - \frac{1}{\beta}H - 2A^TA - \frac{1}{\beta}\Delta^r & -I + \frac{1}{\beta}H + A^TA + \frac{1}{\beta}\Delta^{r-1} \\ I & 0 \end{bmatrix} \begin{bmatrix} x^r \\ x^{r-1} \end{bmatrix} + \begin{bmatrix} A^Tb + \frac{1}{\beta}(\Delta^r - \Delta^{r-1})x^* \\ 0 \end{bmatrix}.$$

where

$$H := \nabla^2 f(x^*), \quad d^{r+1} := -x^* + x^{r+1}$$
$$\Delta^{r+1} := \int_0^1 (\nabla^2 f(x^* + td^{r+1}) - H)dt.$$

- Then show that the above mapping is a diffeomorphism; apply Stable Manifold Theorem to argue that strict saddle point is not stable [Shub 87]

Generalize to (P)?

Generalize to (P)?

- **Can we generalize the Prox-PDA to the following problem?**

$$\begin{aligned} \min \quad & f(x) + h(x) \qquad \text{(P)} \\ \text{s.t.} \quad & Ax = b,\ x \in X \end{aligned}$$

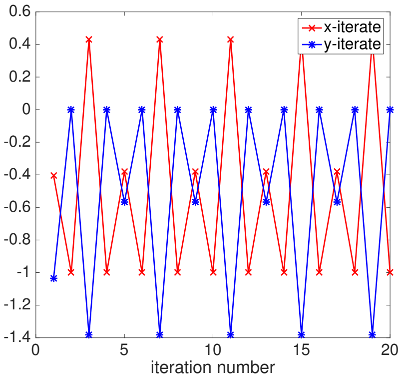
- With the following assumptions

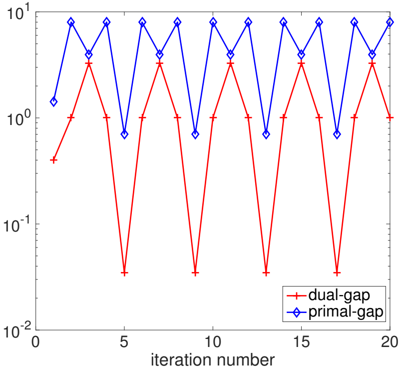
  B1 $h(x) = g_0(x) + h_0(x)$ a non-convex regularizer; g_0 is smooth non-convex, $h_0(x)$ is nonsmooth convex (such as the MCP/SCAD regularizer)

  B2 X is a closed compact convex set

An example

- Consider the following problem (adapted from [Wang-Yin-Zeng 16])
$$\min \; x^2 - y^2, \quad \text{s.t.} \;\; x = y, \; x \in [-1, 1], \; y \in [-2, 0]$$
- Any point in the set $[-1, 0]$ is optimal
- Apply Prox-PDA (with $x^0 = 1$, $y^0 = \mu^0 = 0$, $\beta = 5$)





Generalization to (P)?

- **What went wrong?**
- One can on longer establish the relationship

$$A^T \mu^{r+1} = -\nabla f(x^r) - \beta B^T B(x^{r+1} - x^r)$$

- Change of dual cannot be bounded by change of primal
- **How to proceed?**

Generalization to (P)?

- **What went wrong?**
- One can on longer establish the relationship

$$A^T\mu^{r+1} = -\nabla f(x^r) - \beta B^T B(x^{r+1} - x^r)$$

- Change of dual cannot be bounded by change of primal
- **How to proceed?**

Generalization to (P)?

- **What went wrong?**
- One can on longer establish the relationship

$$A^T\mu^{r+1} = -\nabla f(x^r) - \beta B^T B(x^{r+1} - x^r)$$

- Change of dual cannot be bounded by change of primal
- **How to proceed?**

Adding perturbation

- The key idea is to perturb the primal-dual iteration
- We perturb the dual update by

$$\mu^{r+1} = \mu^r + \rho^{r+1}\left(Ax^{r+1} - b - \gamma^{r+1}\mu^r\right)$$

- Perturb the primal by multiplying $(1 - \rho^{r+1}\gamma^{r+1})$ in front of $\langle \mu^r, Ax - b\rangle$
- Gradually reduce the size of the perturbation constant γ
- **Note:** perturbing dual ascent- type methods has been considered for convex problems [Koshal- Nedić-Shanbhag 11]; not perturbing primal

The Perturbed Prox-PDA

Algorithm 3. The Perturbed Prox-PDA (P-Prox-PDA)

At iteration 0, initialize μ^0 and $x^0 \in \mathbb{R}^N$.
At each iteration $r+1$, update variables by:

$$x^{r+1} = \arg\min_{x\in X} \langle \nabla f(x^r), x - x^r\rangle + (1-\rho^{r+1}\gamma^{r+1})\langle \mu^r, Ax-b\rangle + h(x) + \frac{\rho^{r+1}}{2}\|Ax-b\|^2 + \frac{\beta^{r+1}}{2}\|x-x^r\|^2_{B^TB}; \tag{7a}$$

$$\lambda^{r+1} = \lambda^r + \rho^{r+1}\left(Ax^{r+1} - b - \gamma^{r+1}\lambda^r\right) \tag{7b}$$

- **Intuition.** Adding dual perturbation results in the decent

$$-\rho^{r+1}\gamma^{r+1}\|\lambda^{r+1}-\lambda^r\|^2$$

Conditions on the sequences

- We need the following conditions on the penalty parameter

$$\frac{1}{\rho^r} \to 0, \quad \sum_{r=1}^{\infty} \frac{1}{\rho^r} = \infty, \quad \sum_{r=1}^{\infty} \frac{1}{(\rho^r)^2} < \infty$$

- We need the following conditions on the perturbation

$$\rho^{r+1}\gamma^{r+1} = \tau \in (0,1), \quad \text{for some constant } \tau.$$

- This implies the perturbation on the "dual gradient" goes to zero

Outline of convergence result for P-Prox-PDA

- Suppose Assumption A and B are satisfied

- The conditions on $\{\rho^r, \beta^r\}$ and $\{\gamma^r\}$ given above are satisfied; Then

$$\lim_{r\to\infty} \mu^{r+1} - \mu^r \to 0, \ \lim_{r\to\infty} Ax^r \to b, \text{ and } \ \lim_{r\to\infty} x^{r+1} - x^r = 0$$

- Every limit point of $\{x^r, \mu^r\}$ converges to a first order stationary point of (P) [Hong.-Hajinezhad 17]

- A randomized version of the algorithm converges with a rate

$$\mathbb{E}[Q(x^T, \mu^T)] \in \mathcal{O}\left(T^{-1/3}\right).$$

Remarks

- In our perturbation scheme, increasing penalty parameters and proximal terms are used together with decreasing dual gradient perturbation
- **Question.** Will the algorithm work if all parameters are kept constant?
- Yes, converge to a ϵ-stationary solution
- In particular, for fixed (ρ, β) we need to choose $\rho\gamma = \mathcal{O}(1)$, and $\gamma = \mathcal{O}(\epsilon)$

Definition

ϵ-**stationary solution.** A solution (x^*, λ^*) is called an ϵ-stationary solution if

$$\|Ax^* - b\|^2 \leq \epsilon, \quad \langle \nabla f(x^*) + A^T\lambda^* + \xi^*, x^* - x\rangle \leq 0, \ \forall\ x \in X. \tag{8}$$

where $\xi^* \in \partial h(x^*)$.

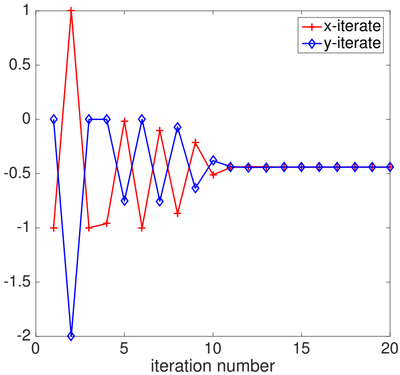
Applications

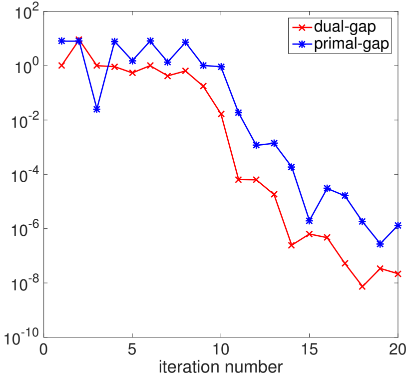
A toy example

- Apply the perturbed version of Prox-PDA to the example

$$\min\ x^2 - y^2, \quad \text{s.t. } x = y,\ x \in [-1, 1],\ y \in [-2, 0]$$

- With $\rho^r = r$, $\gamma^r = 0.001/\rho^r$, $\beta = 5$





Application to distributed non-convex optimization

- Application of Prox-PDA type method to distributed non-convex optimization

$$\boxed{\min_{i} \quad \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad Ax = 0}$$

- Here A is the incidence matrix, $B = |A|$

- Provide explicit update rules for each distributed node [H.- 16]

Application to distributed non-convex optimization

- The system update rule is given by

$$x^{r+1} = x^r - \frac{1}{2\beta} D^{-1}\left(\nabla f(x^r) - \nabla f(x^{r-1})\right) + Wx^r - \frac{1}{2}(I+W)x^{r-1}$$

where in the last equality we have defined the weight matrix $W := \frac{1}{2}D^{-1}(L_+ - L_-)$, which is a row stochastic matrix.

- Each agent updates by

$$\begin{aligned} x_i^{r+1} &= x_i^r - \frac{1}{2\beta d_i}\left(\nabla f_i(x_i^r) - \nabla f_i(x_i^{r-1})\right) \\ &\quad + \sum_{j\in\mathcal{N}(i)} \frac{1}{d_i} x_j^r - \frac{1}{2}\left(\sum_{j\in\mathcal{N}(i)} \frac{1}{d_i} x_j^{r-1} + x_i^{r-1}\right) \end{aligned}$$

- Completely decoupled, new update based on the most recent two iterates

Application to distributed non-convex optimization

- Interestingly, such iteration has the same form as the EXTRA [Shi et al 14], developed for convex consensus problem
- The same observation has also been made in [Mokhtari-Ribeiro 16] (in the convex case)
- By appealing to our analysis, the EXTRA works for the non-convex distributed optimization problem as well (with appropriate β)
- Converges (with sublinear rate) to both 1st and 2nd order stationary solutions
- Different proof techniques
- Other variants of Prox-PDA also can be specialized in this case

Numerical result for distributed non-convex optimization

- We consider a distributed non-negative PCA problem

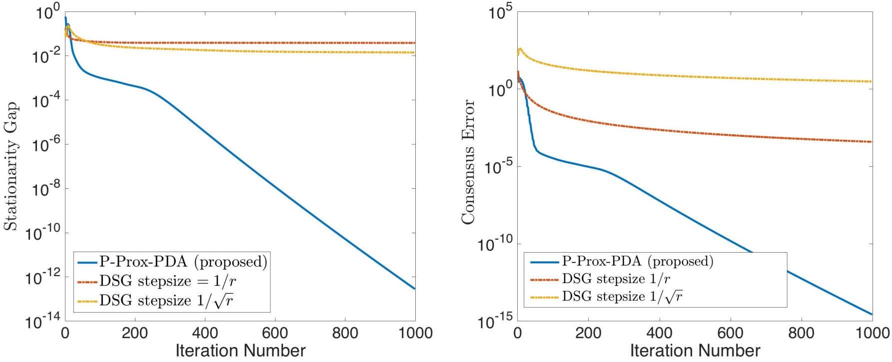
$$\begin{aligned} \min_{z} \quad & \sum_{i=1}^{N} -z^\top D_i^\top D_i z + h(z) \\ \text{s.t.} \quad & \|z\|_2^2 \le 1, \quad z \ge 0. \end{aligned}$$

- $h(z)$ is the MCP regularizer
- Divide the agents randomly into three different sets: $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$
- Consider the following reformulation

$$\begin{aligned} \min_{x} \quad & \sum_{i=1}^{N} -x_i^\top D_i^\top D_i x_i + \frac{1}{|\mathcal{S}_1|} \sum_{i \in \mathcal{S}_1} h(x_i) \\ \text{s.t.} \quad & \|x_i\|_2^2 \le 1 \quad i \in \mathcal{S}_2 \\ & x_i \ge 0 \qquad i \in \mathcal{S}_3 \\ & Ax = 0, \qquad \text{(the consensus constraint)} \end{aligned}$$

Numerical result for distributed non-convex optimization

- Compare with the DSG algorithm proposed in [Nedić-Ozdaglar-Parrilo 10]
- The DSG is designed for convex problems with per-agent local constraint
- We generate the network according to [Yildiz-Scaglione 08]



Numerical result for distributed non-convex optimization

- Compare average performance over 100 random network generation
- Both algorithms stop at 2000 iterations

Table: Comparison of perturbed prox-PDA and DSG

	Stat-Gap		Cons-Vio	
N	P-Prox-PDA	DSG	P-Prox-PDA	DSG
5	$2.1e-19$	0.1	$1.4e-18$	$4.5e-5$
10	$1.4e-19$	0.48	$1.1e-18$	$4.5e-5$
20	$6.7e-18$	0.05	$2.7e-16$	$1.7e-4$
40	$2.19e-13$	0.02	$3.1e-15$	$6.9e-4$

Application to sparse subspace estimation

- We consider the following sparse subspace estimation (with MCP regularizer) [Gu et al 14]

$$\begin{aligned} \widehat{\Pi} = \arg\min_{\Pi} \quad & -\langle \widehat{\Sigma}, \Pi \rangle + P_\alpha(Y) \\ \text{s.t.} \quad & 0 \preceq \Pi \preceq I, \ \mathrm{Tr}(\Pi) = k. \quad \text{(Fantope set)} \\ & \Pi - Y = 0 \end{aligned}$$

  where $P_\alpha(\Pi)$ is chosen to be MCP

- Choose the following for P-Prox-PDA

$$X := [Y; \Pi], \quad A^T A = \begin{bmatrix} I & -I \\ -I & I \end{bmatrix}, \quad B^T B = \begin{bmatrix} I & I \\ I & I \end{bmatrix}$$

- We choose $\alpha^r = r$, $\gamma^r = 10^{-3}/r$

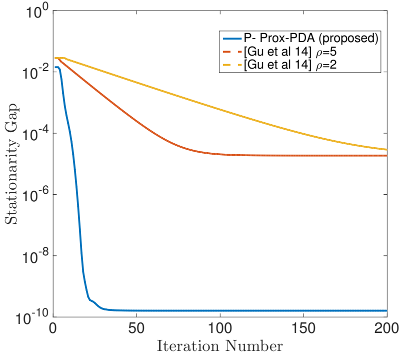
Application to sparse subspace estimation

- Experiment setup following [Gu et al 14] [1]
  1. Construct Σ by eigen-decomposition
  2. **First data set.** $s = 5, k = 1$, $\lambda_1 = 100$; $\lambda_k = 1$, $\forall\ k \neq 1$
  3. $\mathbf{x}_1$ has 5 non-zeros entries, with magnitude $1/\sqrt{5}$
  4. **Second data set.** $s = 10, k = 5$; Top-5 $\lambda_k = 100$, $k = 1, \cdots, 4$, $\lambda_5 = 10$
  5. Eigenvectors are generated by orthnormalizing a 10-sparse Gaussian vectors
  6. SCAD regularizer, $b = 3$

[1] We would like to thank Q. Gu and Z. Wang for providing the codes.

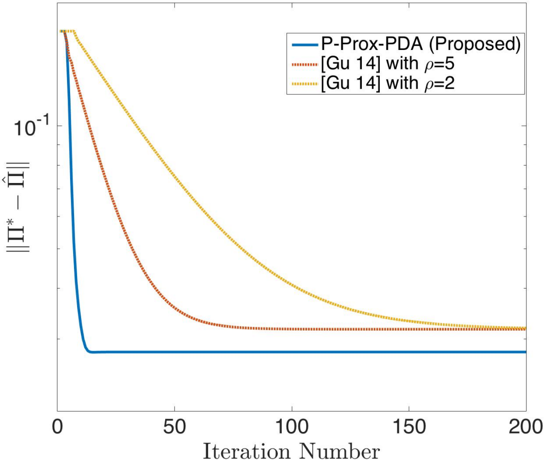
Application to sparse subspace estimation

- We show one realization of P-Prox-PDA and the algorithm in [Gu et al 14]
- Consider the scenario where $n = 80$, $p = 128$, $k = 1$, $s = 5$



Application to sparse subspace estimation

- Compare the recovery error



Application to sparse subspace estimation

- Compare the averaged performance of different algorithms
- Generate 100 true covariance matrices Σ; for each Σ, generate 100 samples

Table: Subspace Estimation Error

	$\|\hat{\Pi} - \Pi^*\|$	
Parameters	PPD	[Gu et al 14]
$n = 80, p = 128, k = 1, s = 5$	0.031 ± 0.01	0.033 ± 0.01
$n = 150, p = 200, k = 1, s = 5$	0.022 ± 0.07	0.025 ± 0.08
$n = 80, p = 128, k = 1, s = 10$	0.047 ± 0.01	0.063 ± 0.01
$n = 80, p = 128, k = 5, s = 10$	0.24 ± 0.05	0.31 ± 0.02
$n = 70, p = 128, k = 5, s = 10$	0.23 ± 0.03	0.33 ± 0.03
$n = 128, p = 128, k = 5, s = 10$	0.17 ± 0.02	0.25 ± 0.02

Application to sparse subspace estimation

- Compare the support recovery performance
- Use True Positive Rate (TPR) and False Positive Rate (FPR)

Table: Support Recovery Results

	TPR		FPR	
Parameters	PPD	[Gu 14]	PPD	[Gu 14]
$n = 80, p = 128, k = 1, s = 5$	1 ± 0	1 ± 0	0 ± 0	0 ± 0
$n = 150, p = 200, k = 1, s = 5$	1 ± 0	1 ± 0	0 ± 0	0 ± 0
$n = 80, p = 128, k = 1, s = 10$	1 ± 0	1 ± 0	0 ± 0	0 ± 0
$n = 80, p = 128, k = 5, s = 10$	1 ± 0	1 ± 0	0.53 ± 0.03	0.56 ± 0.04
$n = 70, p = 128, k = 5, s = 10$	1 ± 0	1 ± 0	0.57 ± 0.01	0.59 ± 0.02
$n = 128, p = 128, k = 5, s = 10$	1 ± 0	1 ± 0	0.53 ± 0.05	0.54 ± 0.01

Conclusion

- In this work we consider solving the following non-convex problem

$$\boxed{\begin{aligned} \min \quad & f(x) + h(x) \qquad \text{(P)} \\ \text{s.t.} \quad & Ax = b, \; x \in X \end{aligned}}$$

- A number of primal-dual based algorithms
- For smooth problems, convergence to first and second order stationary solutions, with global rate
- For nonsmooth problems, primal-dual perturbation scheme
- Compact representation for distributed consensus problem

Future Works

- How about 2nd-order stationarity for non-smooth, constrained problems?
- Preliminary results reported in [Chang-H.-Pang 17], use (single-sided) second order directional derivative to characterize
- The resulting condition is much more complicated than that for the unconstrained linearly constrained case; checking those conditions could be NP-hard; Efficient algorithms?
- Stochasticity? What if objective/gradient is only known through a noisy first/zeroth order oracle?
- **More applications:** Mumford-Shah regularization for image processing (e.g., inpainting) [Möllenhoff et al 14]; Topic modeling [Fu et al 16]; etc.

Thank You!

The randomized algorithm

- Let $B \in \mathbb{R}^{M\times N}$ be some arbitrary matrix to be defined later
- The proposed Proximal Primal Dual Algorithm is given below

Algorithm 1. The Proximal Primal Dual Algorithm (Prox-PDA)

At iteration 0, initialize μ^0 and $x^0 \in \mathbb{R}^N$, fixed T.
For $r = 1, \cdots, T$

$$x^{r+1} = \arg\min_{x\in\mathbb{R}^n} \langle \nabla f(x^r), x - x^r\rangle + \langle \mu^r, Ax - b\rangle + \frac{\beta}{2}\|Ax-b\|^2 + \frac{\beta}{2}\|x - x^r\|^2_{B^TB}; \tag{9a}$$

$$\mu^{r+1} = \mu^r + \beta(Ax^{r+1} - b). \tag{9b}$$

Output (x^t, μ^t), where t is uniformly randomly generated from $[1, 2, \cdots, T]$