Does Alternating Direction Method of Multipliers Converge for Nonconvex Problems?

Mingyi Hong

IMSE and ECpE Department
Iowa State University

ICCOPT, Tokyo, August 2016

The Main Content

- M. Hong, Z.-Q. Luo and M. Razaviyayn, "Convergence Analysis of Alternating Direction Method of Multipliers for a Family of Nonconvex Problems", *SIAM Journal on Optimization*, Vol. 26, No. 1, 2016 (first online Oct. 2014)

Overview

- The Alternating Direction Method of Multipliers (ADMM) is a very popular method for dealing with large-scale optimization problems
- Applications to **classical problems**
  1. LP [Boyd 11], [Ye 15]
  2. SDP [Wen-Goldfarb-Yin 10], [Sun-Toh-Yang 15]
  3. QCQP [Huang-Sidiropoulos 16]
- Applications to **emerging areas**
  1. Social network inference/computing [Baingana et al 15]
  2. Training neural networks [Taylor et al 16]
  3. Smart grid [Dall'Anese et al 13], [Peng-Low 15]
  4. Bioinformatics [Forouzan-Ihler 13]

Overview

- The Alternating Direction Method of Multipliers (ADMM) is a very popular method for dealing with large-scale optimization problems
- Applications to **classical problems**
  1. LP [Boyd 11], [Ye 15]
  2. SDP [Wen-Goldfarb-Yin 10], [Sun-Toh-Yang 15]
  3. QCQP [Huang-Sidiropoulos 16]
- Applications to **emerging areas**
  1. Social network inference/computing [Baingana et al 15]
  2. Training neural networks [Taylor et al 16]
  3. Smart grid [Dall'Anese et al 13], [Peng-Low 15]
  4. Bioinformatics [Forouzan-Ihler 13]

Overview

- The Alternating Direction Method of Multipliers (ADMM) is a very popular method for dealing with large-scale optimization problems
- Applications to **classical problems**
  1. LP [Boyd 11], [Ye 15]
  2. SDP [Wen-Goldfarb-Yin 10], [Sun-Toh-Yang 15]
  3. QCQP [Huang-Sidiropoulos 16]
- Applications to **emerging areas**
  1. Social network inference/computing [Baingana et al 15]
  2. Training neural networks [Taylor et al 16]
  3. Smart grid [Dall'Anese et al 13], [Peng-Low 15]
  4. Bioinformatics [Forouzan-Ihler 13]

Research Question

- **Q**: Is ADMM convergent for nonconvex problems?

- **A**: Yes, for global consensus and sharing problems, and many more

Research Question

- **Q**: Is ADMM convergent for nonconvex problems?

- **A**: Yes, for global consensus and sharing problems, and many more

Research Question

- **Q**: Is ADMM convergent for nonconvex problems?
- **A**: Yes, for global consensus and sharing problems, and many more

Contribution

- Develop a new framework for analyzing the nonconvex ADMM
- Obtain key insights on the behavior of the algorithm
- Motivate new research in theory and applications

Contribution

- Develop a new framework for analyzing the nonconvex ADMM
- Obtain key insights on the behavior of the algorithm
- Motivate new research in theory and applications

Contribution

- Develop a new framework for analyzing the nonconvex ADMM
- Obtain key insights on the behavior of the algorithm
- Motivate new research in theory and applications

Impact

SAG
[Defazio et al 14]

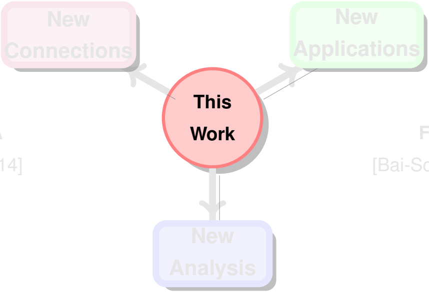
Signal Processing
[Parekh-Selesnick 15]

SAGA
[Le Roux et al 12]

IAG
[Blatt et al 07]

Learning
[Ames-H. 16]

Wireless
[Yatawatta 16]



EXTRA
[Shi et al 14]

Finance
[Bai-Scheinberg 15]

[Kumar et al 16]
[Yang-Pong-Chen 15]

[Jiang et al 16]
[Bai-Scheinberg 15]

[Wang-Yin-Zeng 16]

Impact

SAG
[Defazio et al 14]

Signal Processing
[Parekh-Selesnick 15]

SAGA
[Le Roux et al 12]

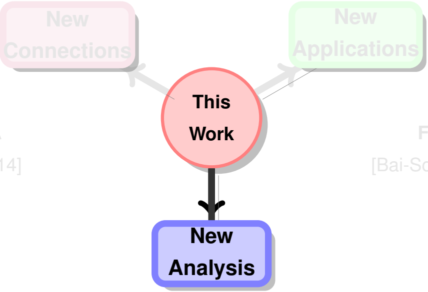
IAG
[Blatt et al 07]

Learning
[Ames-H. 16]

Wireless
[Yatawatta 16]

EXTRA
[Shi et al 14]

Finance
[Bai-Scheinberg 15]



[Kumar et al 16] [Yang-Pong-Chen 15]

[Jiang et al 16] [Bai-Scheinberg 15]

[Wang-Yin-Zeng 16]

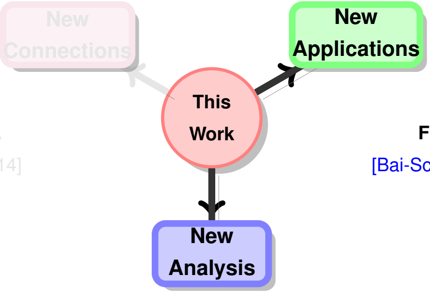
Impact

SAG
[Defazio et al 14]

SAGA
[Le Roux et al 12]

IAG
[Blatt et al 07]

EXTRA
[Shi et al 14]



Signal Processing
[Parekh-Selesnick 15]

Learning
[Ames-H. 16]

Wireless
[Yatawatta 16]

Finance
[Bai-Scheinberg 15]

[Kumar et al 16] [Yang-Pong-Chen 15]

[Jiang et al 16] [Bai-Scheinberg 15]

[Wang-Yin-Zeng 16]

Impact

SAG

[Defazio et al 14]

SAGA

[Le Roux et al 12]

IAG

[Blatt et al 07]

EXTRA

[Shi et al 14]

Signal Processing

[Parekh-Selesnick 15]

Learning

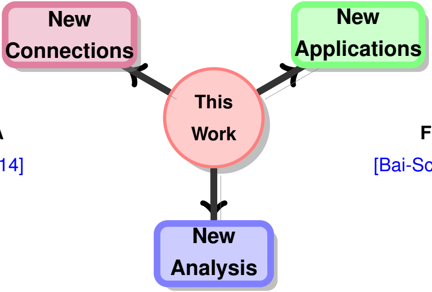
[Ames-H. 16]

Wireless

[Yatawatta 16]

Finance

[Bai-Scheinberg 15]



[Kumar et al 16]   [Yang-Pong-Chen 15]

[Jiang et al 16]   [Bai-Scheinberg 15]

[Wang-Yin-Zeng 16]

Outline

1. Overview
2. Literature Review
3. A New Analysis Framework
   - A Toy Example
   - Nonconvex Consensus Problem
   - Algorithm and Analysis
4. Recent Advances
5. Conclusion

The Basic Setup

Consider the following problem with K blocks of variables $\{x_k\}_{k=1}^K$:

$$\boxed{\begin{aligned} \min \quad & f(x) := \sum_{k=1}^{K} h_k(x_k) + g(x_1, \cdots, x_K) \qquad (P) \\ \text{s.t.} \quad & \sum_{k=1}^{K} A_k x_k = q, \; x_k \in X_k, \; \forall\, k = 1, \cdots, K \end{aligned}}$$

- $h_k(\cdot)$: a convex nonsmooth function
- $g(\cdot)$: a smooth, possibly nonconvex function
- $Ax = q$: linearly coupling constraint, $A_k \in \mathbb{R}^{M \times N_k}$, $q \in \mathbb{R}^M$
- $X_k \subseteq \mathbb{R}^{N_k}$: a closed convex set

The Basic Setup

Consider the following problem with K blocks of variables $\{x_k\}_{k=1}^{K}$:

$$\boxed{\begin{aligned} \min \quad & f(x) := \sum_{k=1}^{K} h_k(x_k) + g(x_1, \cdots, x_K) \qquad (P) \\ \text{s.t.} \quad & \sum_{k=1}^{K} A_k x_k = q, \; x_k \in X_k, \; \forall\, k = 1, \cdots, K \end{aligned}}$$

- $h_k(\cdot)$: a convex nonsmooth function
- $g(\cdot)$: a smooth, possibly nonconvex function
- $Ax = q$: linearly coupling constraint, $A_k \in \mathbb{R}^{M \times N_k}$, $q \in \mathbb{R}^M$
- $X_k \subseteq \mathbb{R}^{N_k}$: a closed convex set

The Basic Setup

Consider the following problem with K blocks of variables $\{x_k\}_{k=1}^K$:

$$\boxed{\begin{aligned} \min \quad & f(x) := \sum_{k=1}^{K} h_k(x_k) + g(x_1, \cdots, x_K) \qquad (P) \\ \text{s.t.} \quad & \sum_{k=1}^{K} A_k x_k = q, \; x_k \in X_k, \; \forall\, k = 1, \cdots, K \end{aligned}}$$

- $h_k(\cdot)$: a convex nonsmooth function
- $g(\cdot)$: a smooth, possibly nonconvex function
- $Ax = q$: linearly coupling constraint, $A_k \in \mathbb{R}^{M \times N_k}$, $q \in \mathbb{R}^M$
- $X_k \subseteq \mathbb{R}^{N_k}$: a closed convex set

The Basic Setup

Consider the following problem with K blocks of variables $\{x_k\}_{k=1}^K$:

$$\begin{aligned} \min \quad & f(x) := \sum_{k=1}^{K} h_k(x_k) + g(x_1, \cdots, x_K) \qquad (P) \\ \text{s.t.} \quad & \sum_{k=1}^{K} A_k x_k = q, \ x_k \in X_k, \ \forall\, k = 1, \cdots, K \end{aligned}$$

- $h_k(\cdot)$: a convex nonsmooth function
- $g(\cdot)$: a smooth, possibly nonconvex function
- $Ax = q$: linearly coupling constraint, $A_k \in \mathbb{R}^{M \times N_k}$, $q \in \mathbb{R}^M$
- $X_k \subseteq \mathbb{R}^{N_k}$: a closed convex set

The Basic Setup

Consider the following problem with K blocks of variables $\{x_k\}_{k=1}^K$:

$$\boxed{\begin{aligned} &\min \quad f(x) := \sum_{k=1}^K h_k(x_k) + g(x_1,\cdots,x_K) \qquad (P) \\ &\text{s.t.} \quad \sum_{k=1}^K A_k x_k = q,\ x_k \in X_k,\ \forall\, k = 1,\cdots,K \end{aligned}}$$

- $h_k(\cdot)$: a convex nonsmooth function
- $g(\cdot)$: a smooth, possibly nonconvex function
- $Ax = q$: linearly coupling constraint, $A_k \in \mathbb{R}^{M\times N_k}$, $q \in \mathbb{R}^M$
- $X_k \subseteq \mathbb{R}^{N_k}$: a closed convex set

The Basic Setup

- The augmented Lagrangian (AL) is given by

$$L(x;y) = \sum_{k=1}^{K} h_k(x_k) + g(x_1, \cdots, x_K) + \langle y, q - Ax\rangle + \frac{\rho}{2}\|q - Ax\|^2,$$

where $\rho > 0$ is the penalty parameter; y is the dual variable

The Basic Setup

- The ADMM performs a block coordinate descent (BCD) on the AL, followed by an (approximate) dual ascent
- Inexactly optimizing the AL often yields closed-form solutions

The ADMM Algorithm

At each iteration $t+1$:
Update the primal variables:

$$x_k^{t+1} = \arg\min_{x_k \in X_k} L(x_1^{t+1}, \cdots, x_{k-1}^{t+1}, x_k, x_{k+1}^t, \cdots, x_K^t; y^t), \ \forall\, k.$$

Update the dual variable:

$$y^{t+1} = y^t + \rho(q - Ax^{t+1}).$$

The Convex Case

- ADMM works for convex, separable, 2-block problems
- The $g(\cdot)$ and $h_k(\cdot)$'s convex; $g(x_1, \cdots, x_k) = \sum_{k=1}^{K} g_k(x_k)$; $K = 2$
- Many classic works on the analysis [Glowinski-Marroco 75], [Gabay-Mercier 76] [Glowinski 83]...
- Equivalence to Douglas-Rachford Splitting and PPA [Gabay 83], [Eckstein-Bertsekas 92]
- Convergence rates and iteration complexity analysis [Eckstein 89] [He-Yuan 12] [Deng-Yin 12] [Hong-Luo 12]
- Extension to multiple-blocks [Sun-Luo-Ye 14] [Chen et al 13] [Ma 12]

The Convex Case

- ADMM works for convex, separable, 2-block problems
- The $g(\cdot)$ and $h_k(\cdot)$'s convex; $g(x_1, \cdots, x_k) = \sum_{k=1}^{K} g_k(x_k)$; $K = 2$
- Many classic works on the analysis [Glowinski-Marroco 75], [Gabay-Mercier 76] [Glowinski 83]...
- Equivalence to Douglas-Rachford Splitting and PPA [Gabay 83], [Eckstein-Bertsekas 92]
- Convergence rates and iteration complexity analysis [Eckstein 89] [He-Yuan 12] [Deng-Yin 12] [Hong-Luo 12]
- Extension to multiple-blocks [Sun-Luo-Ye 14] [Chen et al 13] [Ma 12]

Solving Nonconvex Problems?

- All the works mentioned before are for convex problems
- Recently, widely (and wildly) applied to nonconvex problems as well
  - Distributed clustering [Forero-Cano-Giannakis 11]
  - Matrix separation/completion [Xu-Yin-Wen-Zhang 11]
  - Phase retrieval [Wen-Yang-Liu-Marchesini 12]
  - Distributed matrix factorization [Ling-Yin-Wen 12]
  - Manifold optimization [Lai-Osher 12]
  - Asset allocation [Wen-Peng-Liu-Bai-Sun 13]
  - Nonnegative matrix factorization [Sun-Fevotte 14]
  - Polynomial optimization/tensor decomposition [Jiang-Ma-Zhang 13, Livavas-Sidiropoulos 14]

Solving Nonconvex Problems?

- All the works mentioned before are for convex problems
- Recently, widely (and wildly) applied to nonconvex problems as well
    1. Distributed clustering [Forero-Cano-Giannakis 11]
    2. Matrix separation/completion [Xu-Yin-Wen-Zhang 11]
    3. Phase retrieval [Wen-Yang-Liu-Marchesini 12]
    4. Distributed matrix factorization [Ling-Yin-Wen 12]
    5. Manifold optimization [Lai-Osher 12]
    6. Asset allocation [Wen-Peng-Liu-Bai-Sun 13]
    7. Nonnegative matrix factorization [Sun-Fevotte 14]
    8. Polynomial optimization/tensor decomposition [Jiang-Ma-Zhang 13, Livavas-Sidiropoulos 14]

Solving Nonconvex Problems?

- All the works mentioned before are for convex problems
- Recently, widely (and wildly) applied to nonconvex problems as well
    1. Distributed clustering [Forero-Cano-Giannakis 11]
    2. Matrix separation/completion [Xu-Yin-Wen-Zhang 11]
    3. Phase retrieval [Wen-Yang-Liu-Marchesini 12]
    4. Distributed matrix factorization [Ling-Yin-Wen 12]
    5. Manifold optimization [Lai-Osher 12]
    6. Asset allocation [Wen-Peng-Liu-Bai-Sun 13]
    7. Nonnegative matrix factorization [Sun-Fevotte 14]
    8. Polynomial optimization/tensor decomposition [Jiang-Ma-Zhang 13, Livavas-Sidiropoulos 14]

Application 1: Nonnegative Tensor Factorization

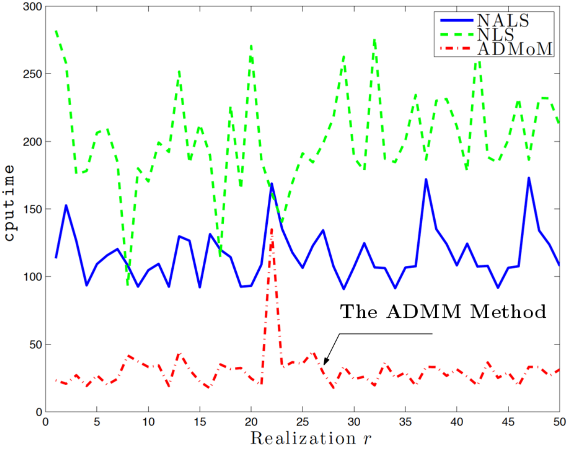


Figure: ADMM for solving tensor factorization problem [Liavas-Sidiropoulos 14]

Issues and Challenges

- **Pros:** Nonconvex ADMM achieves excellent numerical performance
- **Cons:** A general lack of global performance analysis
- Convergence claim

But this is a big "IF"!

Issues and Challenges

- **Pros:** Nonconvex ADMM achieves excellent numerical performance
- **Cons:** A general lack of global performance analysis
- **Convergence claim**
  1. **IF** the successive differences of all the primal and dual variables go to zero (e.g., $x^{t+1} - x^{t} \to 0, y^{t+1} - y^{t} \to 0$)
  2. Then any limit point is a stationary solution

But this is a big "IF"!

Issues and Challenges

- **Pros:** Nonconvex ADMM achieves excellent numerical performance
- **Cons:** A general lack of global performance analysis
- **Convergence claim**
  1. **IF** the successive differences of all the primal and dual variables go to zero (e.g., $x^{t+1} - x^t \to 0, y^{t+1} - y^t \to 0$)
  2. Then any limit point is a stationary solution

But this is a big "IF"!

Issues and Challenges

- The assumption on iterates is uncheckable *a priori*
- "Assume" (without proving) that feasibility holds in the limit
- An exception [Zhang 10]: convergence for certain special QP
  1. The AL is strongly convex
  2. Only has the linear constraint
  3. The dual stepsize is very small

Issues and Challenges

- The assumption on iterates is uncheckable *a priori*
- "Assume" (without proving) that feasibility holds in the limit
- An exception [Zhang 10]: convergence for certain special QP
  1. The AL is strongly convex
  2. Only has the linear constraint
  3. The dual stepsize is very small

Issues and Challenges

- The assumption on iterates is uncheckable *a priori*
- "Assume" (without proving) that feasibility holds in the limit
- An exception [Zhang 10]: convergence for certain special QP
    1. The AL is strongly convex
    2. Only has the linear constraint
    3. The dual stepsize is very small

Issues and Challenges

Rigorously analyzing nonconvex ADMM is challenging

- **Cases I:** Without the linear constraint, reduces to the classic BCD
- Can diverge for general nonconvex $g(\cdot)$ with $K \geq 3$ [Powell 73]
- **Cases II:** With the linear constraint and $K = 1$
- Can diverge for any fixed $\rho > 0$ [Wang-Yin-Zeng 16]

Issues and Challenges

Rigorously analyzing nonconvex ADMM is challenging

- **Cases I:** Without the linear constraint, reduces to the classic BCD
- Can diverge for general nonconvex $g(\cdot)$ with $K \geq 3$ [Powell 73]
- **Cases II:** With the linear constraint and $K = 1$
- Can diverge for any fixed $\rho > 0$ [Wang-Yin-Zeng 16]

Issues and Challenges

Rigorously analyzing nonconvex ADMM is challenging

- **Cases I:** Without the linear constraint, reduces to the classic BCD
- Can diverge for general nonconvex $g(\cdot)$ with $K \geq 3$ [Powell 73]

- **Cases II:** With the linear constraint and $K = 1$
- Can diverge for any fixed $\rho > 0$ [Wang-Yin-Zeng 16]

Outline

1. Overview
2. Literature Review
3. A New Analysis Framework
   - A Toy Example
   - Nonconvex Consensus Problem
   - Algorithm and Analysis
4. Recent Advances
5. Conclusion

A Toy Example

- First consider the following toy nonconvex example

$$\min_{x,z} \quad \frac{1}{2}x^T A x + bz, \quad \text{s.t.} \quad z \in [1,2],\ z = x$$

where A is a symmetric matrix; $x \in \mathbb{R}^N$

A Toy Example

- First consider the following toy nonconvex example

$$\min_{x,z} \quad \frac{1}{2}x^T Ax + bz, \quad \text{s.t.} \quad z \in [1,2], \; z = x$$

where A is a symmetric matrix; $x \in \mathbb{R}^N$

ADMM Convergent?

A Toy Example (cont.)

- Randomly generate the data matrices A and b with $N = 10$
- Plot the following
  1. Primal feasibility gap: $\|z - x\|$
  2. The optimality measure: $\|x - \text{proj}\left[x - (Ax + b)\right]\|$
  3. The x-feasibility gap: $\|x - \text{proj}(x)\|$
- All three quantities go to zero iff a stationary solution has been reached

First Try: $\rho = 20$

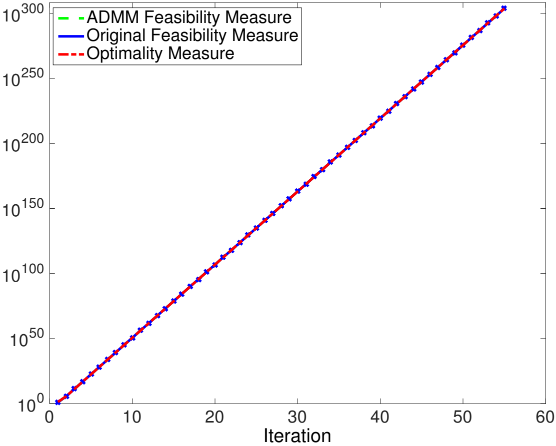


Figure: $N = 10$, $\rho = 20$

Second Try: $\rho = 200$

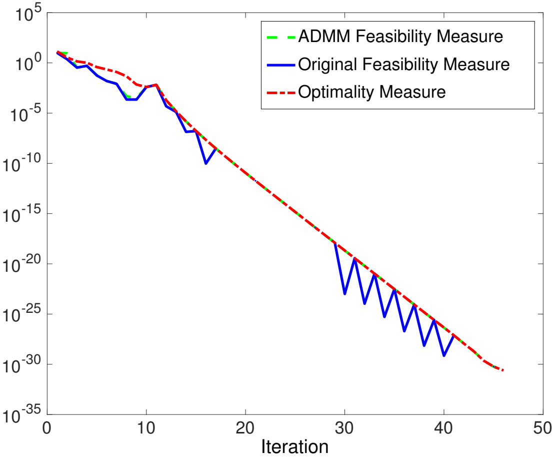


Figure: $N = 10$, $\rho = 200$

A Toy Example (cont.)

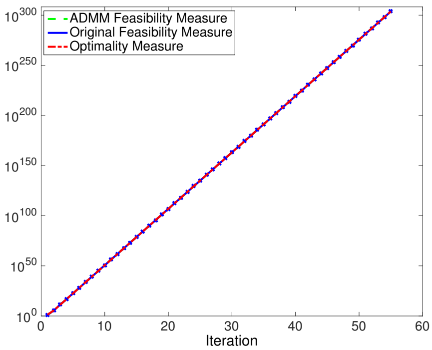


Figure: $n = 10$, $\rho = 20$

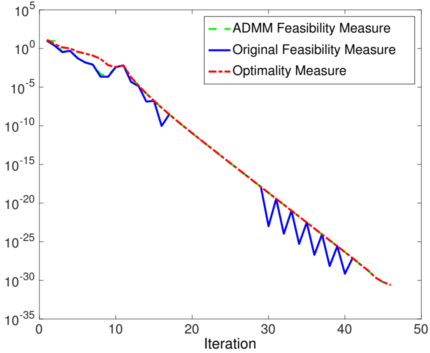


Figure: $n = 10$, $\rho = 200$

A Toy Example (cont.)

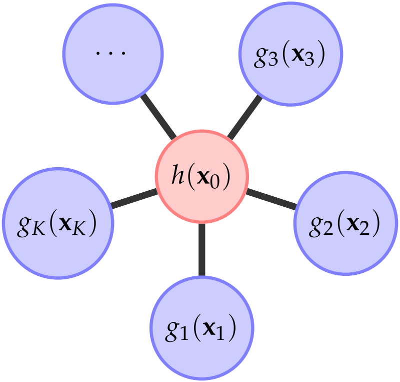
- The convergence is ρ-dependent
- When ρ is small, the algorithm fails to converge
- Different from the convex case, where any $\rho > 0$ should work
- Reminiscent to the AL method, careful choice of ρ in nonconvex case

Outline

1. Overview
2. Literature Review
3. A New Analysis Framework
   - A Toy Example
   - Nonconvex Consensus Problem
   - Algorithm and Analysis
4. Recent Advances
5. Conclusion

Problem Setup: A Nonconvex Consensus Problem

- Consider a nonconvex global consensus problem
- A distributed optimization problem defined over a network of K agents



Problem Setup: A Nonconvex Consensus Problem

- Consider a nonconvex global consensus problem
- A distributed optimization problem defined over a network of K agents
- Formally, the problem is given by

$$\min \sum_{k=1}^{K} g_k(x_k) + h(x_0), \quad \text{s.t.} \quad x_k = x_0, \; \forall\, k = 1, \cdots, K, \; x_0 \in X.$$

Problem Setup: A Nonconvex Consensus Problem

- Consider a nonconvex global consensus problem
- A distributed optimization problem defined over a network of K agents
- Formally, the problem is given by

consensus constraint

$$\min \sum_{k=1}^{K} g_k(x_k) + h(x_0), \quad \text{s.t.} \quad \boxed{x_k = x_0,} \quad \forall\, k = 1, \cdots, K,\; x_0 \in X.$$

Problem Setup: A Nonconvex Consensus Problem

- Consider a nonconvex global consensus problem
- A distributed optimization problem defined over a network of K agents
- Formally, the problem is given by

$$\min \sum_{k=1}^{K} \underbrace{\boxed{g_k(x_k)}}_{\text{nonconvex part}} + h(x_0), \quad \text{s.t.} \quad x_k = x_0, \forall\, k = 1, \cdots, K,\; x_0 \in X.$$

Problem Setup: A Nonconvex Consensus Problem

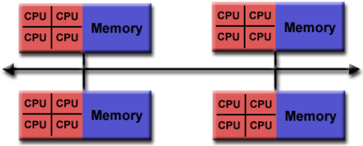
- Consider a nonconvex global consensus problem
- A distributed optimization problem defined over a network of K agents
- Formally, the problem is given by

nonsmooth part

$$\min \sum_{k=1}^{K} g_k(x_k) + \boxed{h(x_0)}, \quad \text{s.t.} \quad x_k = x_0, \forall\, k = 1, \cdots, K, \; x_0 \in X.$$

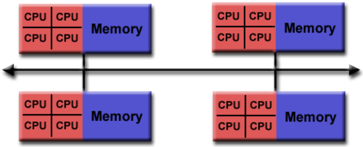
Problem Setup: A Nonconvex Consensus Problem

- Wide applications in distributed signal and information processing, parallel optimization, etc [Boyd et al 11]
- For example, in the distributed sparse PCA problem [H.-Luo-Razaviyayn 14]
  1. $g_k(x_k) = -x_k^T A_k^T A_k x_k$: $A_k^T A_k$ is the covariance matrix for local data
  2. $h(\cdot)$: some sparsity promoting nonsmooth regularizer



Problem Setup: A Nonconvex Consensus Problem

- Wide applications in distributed signal and information processing, parallel optimization, etc [Boyd et al 11]
- For example, in the distributed sparse PCA problem [H.-Luo-Razaviyayn 14]
  1. $g_k(x_k) = -x_k^T A_k^T A_k x_k$: $A_k^T A_k$ is the covariance matrix for local data
  2. $h(\cdot)$: some sparsity promoting nonsmooth regularizer



The Algorithm

- The AL function is given by

$$L(\{x_k\}, x_0; y) = \sum_{k=1}^{K} g_k(x_k) + h(x_0) + \sum_{k=1}^{K} \langle y_k, x_k - x_0 \rangle + \sum_{k=1}^{K} \frac{\rho_k}{2} \|x_k - x_0\|^2.$$

Algorithm 1. The Consensus ADMM

At each iteration $t+1$, compute:

$$x_0^{t+1} = \arg\min_{x_0 \in X} \; L(\{x_k^t\}, x_0; y^t).$$

Each node k computes x_k by solving:

$$x_k^{t+1} = \arg\min_{x_k} \; g_k(x_k) + \langle y_k^t, x_k - x_0^{t+1} \rangle + \frac{\rho_k}{2} \|x_k - x_0^{t+1}\|^2.$$

Each node k updates the dual variable:

$$y_k^{t+1} = y_k^t + \rho_k \left(x_k^{t+1} - x_0^{t+1} \right).$$

Illustration: x_0 update

x_0 solves: $x_0^{t+1} = \text{argmin}_{x_0 \in X} \; L(\{x_k^t\}, x_0; y^t)$ (often with closed-form)

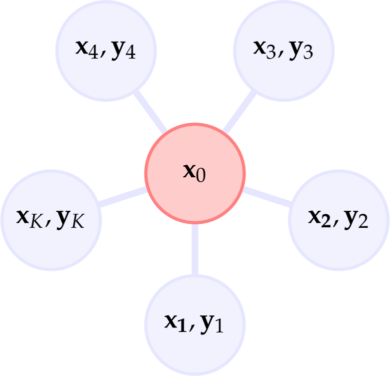


Illustration: broadcast

Broadcasts the most recent x_0

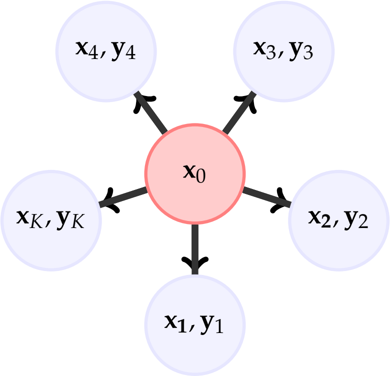


Illustration: (x_k, λ_k) update

x_k solves: $x_k^{t+1} = \arg\min_{x_k} \; g_k(x_k) + \langle y_k^t, x_k - x_0^{t+1} \rangle + \frac{\rho_k}{2} \|x_k - x_0^{t+1}\|^2.$

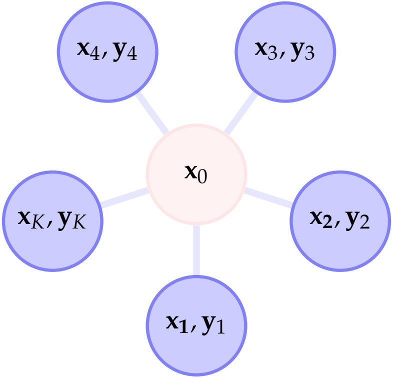
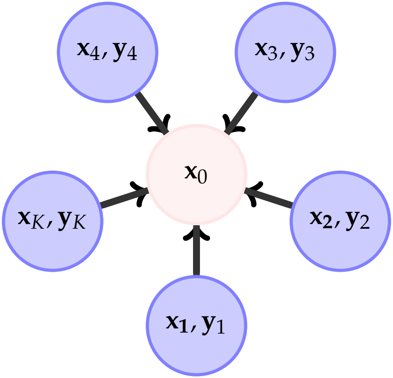


Illustration: aggregate

Aggregate (x_k, y_k) to the central node



Main Assumptions

Assumption A

A1. Each g_k has Lipschitz continuous gradient:

$$\|\nabla_k g_k(x_k) - \nabla_k g_k(z_k)\| \leq L_k \|x_k - z_k\|, \; \forall \; x_k, \; z_k, \; k = 1, \cdots, K.$$

Moreover, h is convex (possible nonsmooth); X is a closed convex set.

A2. ρ_k is large enough such that:

1. For all k, the x_k subproblem is strongly convex with modulus $\gamma_k(\rho_k)$;
2. For all k, the following is satisfied

$$\rho_k > \max\left\{\frac{2L_k^2}{\gamma_k(\rho_k)}, L_k\right\}.$$

A3. $f(x)$ is bounded from below over X.

Main Assumptions

Assumption A

A1. Each g_k has Lipschitz continuous gradient:

$$\|\nabla_k g_k(x_k) - \nabla_k g_k(z_k)\| \le L_k \|x_k - z_k\|, \; \forall\, x_k,\, z_k,\, k = 1, \cdots, K.$$

Moreover, h is convex (possible nonsmooth); X is a closed convex set.

A2. ρ_k is large enough such that:

1. For all k, the x_k subproblem is strongly convex with modulus $\gamma_k(\rho_k)$;
2. For all k, the following is satisfied

$$\rho_k > \max\left\{\frac{2L_k^2}{\gamma_k(\rho_k)}, L_k\right\}.$$

A3. $f(x)$ is bounded from below over X.

Main Assumptions

Assumption A

A1. Each g_k has Lipschitz continuous gradient:

$$\|\nabla_k g_k(x_k) - \nabla_k g_k(z_k)\| \le L_k \|x_k - z_k\|, \; \forall\, x_k,\, z_k,\, k = 1, \cdots, K.$$

Moreover, h is convex (possible nonsmooth); X is a closed convex set.

A2. ρ_k is large enough such that:

1. For all k, the x_k subproblem is strongly convex with modulus $\gamma_k(\rho_k)$;
2. For all k, the following is satisfied

$$\rho_k > \max\left\{\frac{2L_k^2}{\gamma_k(\rho_k)}, L_k\right\}.$$

A3. $f(x)$ is bounded from below over X.

Proof Ideas: Preliminary

- **Question:** Can we leverage the existing analysis for the convex case?

- Unfortunately no, because most existing analysis relies on showing

$$\|x^t - x^*\|^2 + \|y^t - y^*\|^2 \to 0$$

where (x^*, y^*) are the globally optimal primal-dual pair

How to measure the progress of the algorithm?

Proof Ideas: Preliminary

- **Question:** Can we leverage the existing analysis for the convex case?

- Unfortunately no, because most existing analysis relies on showing

$$\|\mathbf{x}^t - \mathbf{x}^*\|^2 + \|\mathbf{y}^t - \mathbf{y}^*\|^2 \to 0$$

where $(\mathbf{x}^*, \mathbf{y}^*)$ are the globally optimal primal-dual pair

How to measure the progress of the algorithm?

Proof Ideas: Preliminary

- **Question:** Can we leverage the existing analysis for the convex case?

- Unfortunately no, because most existing analysis relies on showing

$$\|\mathbf{x}^t - \mathbf{x}^*\|^2 + \|\mathbf{y}^t - \mathbf{y}^*\|^2 \to 0$$

where $(\mathbf{x}^*, \mathbf{y}^*)$ are the globally optimal primal-dual pair

How to measure the progress of the algorithm?

Proof Ideas: Preliminary

- **Question:** Can we leverage the existing analysis for the convex case?

- Unfortunately no, because most existing analysis relies on showing

$$\|\mathbf{x}^t - \mathbf{x}^*\|^2 + \|\mathbf{y}^t - \mathbf{y}^*\|^2 \to 0$$

where $(\mathbf{x}^*, \mathbf{y}^*)$ are the globally optimal primal-dual pair

How to measure the progress of the algorithm?

Proof Ideas: A Key Step

- **Solution**: Use $L(x;y)$ as the merit function to guide the progress

- **Challenge**: The behavior of $L(x;y)$ is difficult to characterize
  1. Decreases after each primal update
  2. Increases after each dual update

- **Technique**: Bound the change of the dual update by that of the primal

Proof Ideas: A Key Step

- **Solution**: Use $L(x;y)$ as the merit function to guide the progress
- **Challenge**: The behavior of $L(x;y)$ is difficult to characterize
  1. Decreases after each primal update
  2. Increases after each dual update
- **Technique**: Bound the change of the dual update by that of the primal

Proof Ideas: A Key Step

- **Solution**: Use $L(x;y)$ as the merit function to guide the progress
- **Challenge**: The behavior of $L(x;y)$ is difficult to characterize
  1. Decreases after each primal update
  2. Increases after each dual update
- **Technique**: Bound the change of the dual update by that of the primal

Proof Steps

- We develop a three-step analysis framework

- **Step 1**: Show "sufficient descent"

$$L(\{x_k^{t+1}\}, x_0^{t+1}; y^{t+1}) - L(\{x_k^t\}, x_0^t; y^t)$$
$$\leq \sum_{k=1}^{K} \left(\frac{L_k^2}{\rho_k} - \frac{\gamma_k(\rho_k)}{2} \right) \|x_k^{t+1} - x_k^t\|^2 - \frac{\sum_{k=1}^K \rho_k}{2} \|x_0^{t+1} - x_0^t\|^2$$

- **Step 2**: Show the following is "lower bounded"

$$L(x_0^{t+1}, \{x_k^{t+1}\}; y^{t+1}) \geq -\infty$$

- **Step 3**: Show convergence to the set of stationary solutions

Proof Steps

- We develop a three-step analysis framework
- **Step 1**: Show "sufficient descent"

$$L(\{x_k^{t+1}\}, x_0^{t+1}; y^{t+1}) - L(\{x_k^t\}, x_0^t; y^t)$$
$$\leq \sum_{k=1}^{K}\left(\frac{L_k^2}{\rho_k} - \frac{\gamma_k(\rho_k)}{2}\right)\|x_k^{t+1} - x_k^t\|^2 - \frac{\sum_{k=1}^K \rho_k}{2}\|x_0^{t+1} - x_0^t\|^2$$

- **Step 2**: Show the following is "lower bounded"

$$L(x_0^{t+1}, \{x_k^{t+1}\}; y^{t+1}) \geq -\infty$$

- **Step 3**: Show convergence to the set of stationary solutions

Proof Steps

- We develop a three-step analysis framework
- **Step 1**: Show "sufficient descent"

$$
\begin{aligned}
&L(\{x_k^{t+1}\}, x_0^{t+1}; y^{t+1}) - L(\{x_k^t\}, x_0^t; y^t) \\
&\le \sum_{k=1}^{K} \left(\frac{L_k^2}{\rho_k} - \frac{\gamma_k(\rho_k)}{2} \right) \|x_k^{t+1} - x_k^t\|^2 - \frac{\sum_{k=1}^K \rho_k}{2} \|x_0^{t+1} - x_0^t\|^2
\end{aligned}
$$

- **Step 2**: Show the following is "lower bounded"

$$
L(x_0^{t+1}, \{x_k^{t+1}\}; y^{t+1}) \ge -\infty
$$

- **Step 3**: Show convergence to the set of stationary solutions

Proof Steps

- We develop a three-step analysis framework
- **Step 1**: Show "sufficient descent"

$$
\begin{aligned}
& L(\{x_k^{t+1}\}, x_0^{t+1}; y^{t+1}) - L(\{x_k^t\}, x_0^t; y^t) \\
& \le \sum_{k=1}^{K} \left(\frac{L_k^2}{\rho_k} - \frac{\gamma_k(\rho_k)}{2} \right) \|x_k^{t+1} - x_k^t\|^2 - \frac{\sum_{k=1}^K \rho_k}{2} \|x_0^{t+1} - x_0^t\|^2
\end{aligned}
$$

- **Step 2**: Show the following is "lower bounded"

$$
L(x_0^{t+1}, \{x_k^{t+1}\}; y^{t+1}) \ge -\infty
$$

- **Step 3**: Show convergence to the set of stationary solutions

Proof Steps

- We develop a three-step analysis framework
- **Step 1**: Show "sufficient descent"

$$L(\{x_k^{t+1}\}, x_0^{t+1}; y^{t+1}) - L(\{x_k^t\}, x_0^t; y^t)$$
$$\leq \sum_{k=1}^{K} \boxed{\left(\frac{L_k^2}{\rho_k} - \frac{\gamma_k(\rho_k)}{2}\right)} \|x_k^{t+1} - x_k^t\|^2 - \frac{\sum_{k=1}^K \rho_k}{2} \|x_0^{t+1} - x_0^t\|^2$$

- **Step 2**: Show the following is "lower bounded"

$$L(x_0^{t+1}, \{x_k^{t+1}\}; y^{t+1}) \geq -\infty$$

- **Step 3**: Show convergence to the set of stationary solutions

Proof Steps

- We develop a three-step analysis framework
- **Step 1**: Show "sufficient descent"

$$
\begin{aligned}
&L(\{x_k^{t+1}\}, x_0^{t+1}; y^{t+1}) - L(\{x_k^t\}, x_0^t; y^t) \\
&\le \sum_{k=1}^{K} \left(\frac{L_k^2}{\rho_k} - \frac{\gamma_k(\rho_k)}{2} \right) \|x_k^{t+1} - x_k^t\|^2 - \frac{\sum_{k=1}^K \rho_k}{2} \|x_0^{t+1} - x_0^t\|^2
\end{aligned}
$$

- **Step 2**: Show the following is "lower bounded"

$$
L(x_0^{t+1}, \{x_k^{t+1}\}; y^{t+1}) \ge -\infty
$$

- **Step 3**: Show convergence to the set of stationary solutions

The Convergence Claim

Convergence of ADMM for Nonconvex Global Consensus

Claim: Suppose Assumption A is satisfied. Then we have

1. The linear constraint is satisfied eventually:

$$\lim_{t\to\infty} \|x_k^{t+1} - x_0^{t+1}\| = 0, \ \forall\, k$$

2. Any limit point of the sequence generated by Algorithm 1 is a stationary solution of the consensus problem

The Iteration Complexity Analysis

- Need new gap function to measure the gap to stationarity

$$P(x^t, y^t) := \underbrace{\boxed{\|\tilde{\nabla} L(\{x_k^t\}, x_0^t, y^t)\|^2}}_{\text{primal gap}} + \sum_{k=1}^{K} \underbrace{\boxed{\|x_k^t - x_0^t\|^2}}_{\text{dual gap}}$$

- $P(x, y) = 0 \Leftrightarrow (x, y)$ is a stationary solution

Claim: Suppose Assumption A is satisfied, $\epsilon > 0$ be some constant. Let $T(\epsilon)$ denote an iteration index which satisfies

$$T(\epsilon) := \min\left\{t \mid P(x^t, y^t) \le \epsilon, t \ge 0\right\}$$

for some $\epsilon > 0$. Then there exists some constant $C > 0$ such that

$$T(\epsilon) \le \frac{C}{\epsilon}.$$

The Iteration Complexity Analysis

- Need new gap function to measure the gap to stationarity

$$P(x^t, y^t) := \underbrace{\boxed{\|\tilde{\nabla} L(\{x_k^t\}, x_0^t, y^t)\|^2}}_{\text{primal gap}} + \sum_{k=1}^{K} \underbrace{\boxed{\|x_k^t - x_0^t\|^2}}_{\text{dual gap}}$$

- $P(x, y) = 0 \Leftrightarrow (x, y)$ is a stationary solution

Claim: Suppose Assumption A is satisfied, $\epsilon > 0$ be some constant. Let $T(\epsilon)$ denote an iteration index which satisfies

$$T(\epsilon) := \min\left\{t \mid P(x^t, y^t) \le \epsilon, t \ge 0\right\}$$

for some $\epsilon > 0$. Then there exists some constant $C > 0$ such that

$$T(\epsilon) \le \frac{C}{\epsilon}.$$

The Iteration Complexity Analysis

- Need new gap function to measure the gap to stationarity

$$P(x^t, y^t) := \underbrace{\|\tilde{\nabla} L(\{x_k^t\}, x_0^t, y^t)\|^2}_{\text{primal gap}} + \sum_{k=1}^{K} \underbrace{\|x_k^t - x_0^t\|^2}_{\text{dual gap}}$$

- $P(x, y) = 0 \Leftrightarrow (x, y)$ is a stationary solution

Claim: Suppose Assumption A is satisfied, $\epsilon > 0$ be some constant. Let $T(\epsilon)$ denote an iteration index which satisfies

$$T(\epsilon) := \min\left\{t \mid P(x^t, y^t) \le \epsilon, t \ge 0\right\}$$

for some $\epsilon > 0$. Then there exists some constant $C > 0$ such that

$$T(\epsilon) \le \frac{C}{\epsilon}.$$

Proximal Step and Flexible Updates

- Use proximal gradient to update x_k for cheap iterations
- The x_k step is replaced by

$$x_k^{t+1} = \arg\min_{x_k} \langle \nabla g_k(x_0^{t+1}), x_k - x_0^{t+1} \rangle + \langle y_k^t, x_k - x_0^{t+1} \rangle + \frac{\rho_k + L_k}{2} \|x_k - x_0^{t+1}\|^2.$$

- Gradient evaluated at the most recent x_0!
- We can also use stochastic node sampling
- **Similar convergence guarantee as Algorithm 1**

Proximal Step and Flexible Updates

- Use proximal gradient to update x_k for cheap iterations
- The x_k step is replaced by

$$x_k^{t+1} = \arg\min_{x_k} \langle \nabla g_k(x_0^{t+1}), x_k - x_0^{t+1} \rangle + \langle y_k^t, x_k - x_0^{t+1} \rangle + \frac{\rho_k + L_k}{2} \|x_k - x_0^{t+1}\|^2.$$

- Gradient evaluated at the most recent x_0!
- We can also use stochastic node sampling
- **Similar convergence guarantee as Algorithm 1**

Proximal Step and Flexible Updates

- Use proximal gradient to update x_k for cheap iterations
- The x_k step is replaced by

$$x_k^{t+1} = \arg\min_{x_k} \langle \nabla g_k(x_0^{t+1}), x_k - x_0^{t+1} \rangle + \langle y_k^t, x_k - x_0^{t+1} \rangle + \frac{\rho_k + L_k}{2} \|x_k - x_0^{t+1}\|^2.$$

- Gradient evaluated at the most recent x_0!
- We can also use stochastic node sampling
- **Similar convergence guarantee as Algorithm 1**

Proximal Step and Flexible Updates

- Use proximal gradient to update x_k for cheap iterations
- The x_k step is replaced by

$$x_k^{t+1} = \arg\min_{x_k} \; \langle \nabla g_k(x_0^{t+1}), x_k - x_0^{t+1} \rangle + \langle y_k^t, x_k - x_0^{t+1} \rangle + \frac{\rho_k + L_k}{2} \|x_k - x_0^{t+1}\|^2.$$

- Gradient evaluated at the most recent x_0!
- We can also use stochastic node sampling
- Similar convergence guarantee as Algorithm 1

Proximal Step and Flexible Updates

- Use proximal gradient to update x_k for cheap iterations
- The x_k step is replaced by

$$x_k^{t+1} = \operatorname*{argmin}_{x_k} \; \langle \nabla g_k(x_0^{t+1}), x_k - x_0^{t+1} \rangle + \langle y_k^t, x_k - x_0^{t+1} \rangle + \frac{\rho_k + L_k}{2} \|x_k - x_0^{t+1}\|^2.$$

- Gradient evaluated at the most recent x_0!
- We can also use stochastic node sampling
- **Similar convergence guarantee as Algorithm 1**

The Nonconvex Sharing Problem

- Our analysis also works for the well-known sharing problem [Boyd-Parikh-Chu-Peleato-Eckstein 11]

$$
\begin{aligned}
\min \quad & \sum_{k=1}^{K} h_k(x_k) + g\,(x_0) \\
\text{s.t.} \quad & \sum_{k=1}^{K} A_k x_k = x_0, \quad x_k \in X_k,\ k = 1, \cdots, K.
\end{aligned}
$$

- $x_k \in \mathbb{R}^{N_k}$ is the variable associated with agent k
- $(K+1)$**-block problem**, convergence unknown for the convex case
- **Apply our analysis to show convergence**

The Nonconvex Sharing Problem

- Our analysis also works for the well-known sharing problem [Boyd-Parikh-Chu-Peleato-Eckstein 11]

$$\begin{aligned} \min \quad & \sum_{k=1}^{K} h_k(x_k) + g\left(x_0\right) \\ \text{s.t.} \quad & \boxed{\sum_{k=1}^{K} A_k x_k = x_0,} \quad x_k \in X_k,\ k = 1, \cdots, K. \end{aligned}$$

- $x_k \in \mathbb{R}^{N_k}$ is the variable associated with agent k
- $(K+1)$**-block problem**, convergence unknown for the convex case
- **Apply our analysis to show convergence**

Remarks

- The first analysis framework for iteration complexity of nonconvex ADMM
- A major departure from the classic analysis for convex problems
- The AL guides the convergence of the algorithm
- The ρ_k's should be large enough, with computable lower bounds

Outline

1. Overview
2. Literature Review
3. A New Analysis Framework
   - A Toy Example
   - Nonconvex Consensus Problem
   - Algorithm and Analysis
4. Recent Advances
5. Conclusion

Recent Advances

- Many exciting recent works have been built upon our results
- **New analysis, new algorithms and new connections**

New Analysis

New analysis for weaker conditions

- Work [Li-Pong 14]: h, nonconvex, coercive; more general A_k; whole sequence convergence under Kurdyka- Lojasiewicz (KL) property
- Work [Kumar et al 16]: different update schedules
- Work [Bai-Scheinberg 15]: different characterization of iteration complexity
- Works [Jiang et al 16, Wang-Yin-Zeng 16]: both relax conditions for the K-agent sharing problem
- Work [Yang-Pong-Chen 15]: enlarges the dual stepsize by $\frac{\sqrt{5}+1}{2} \approx 1.618...$
- ...

New Analysis

New analysis for weaker conditions

- Work [Li-Pong 14]: h, nonconvex, coercive; more general A_k; whole sequence convergence under Kurdyka- Lojasiewicz (KL) property
- Work [Kumar et al 16]: different update schedules
- Work [Bai-Scheinberg 15]: different characterization of iteration complexity
- Works [Jiang et al 16, Wang-Yin-Zeng 16]: both relax conditions for the K-agent sharing problem
- Work [Yang-Pong-Chen 15]: enlarges the dual stepsize by $\frac{\sqrt{5}+1}{2} \approx 1.618...$
- ...

New Analysis

New analysis for weaker conditions

- Work [Li-Pong 14]: h, nonconvex, coercive; more general A_k; whole sequence convergence under Kurdyka- Lojasiewicz (KL) property
- Work [Kumar et al 16]: different update schedules
- Work [Bai-Scheinberg 15]: different characterization of iteration complexity
- Works [Jiang et al 16, Wang-Yin-Zeng 16]: both relax conditions for the K-agent sharing problem
- Work [Yang-Pong-Chen 15]: enlarges the dual stepsize by $\frac{\sqrt{5}+1}{2} \approx 1.618...$
- ...

New Analysis

New analysis for weaker conditions

- Work [Li-Pong 14]: h, nonconvex, coercive, more general A_k; whole sequence convergence under Kurdyka- Lojasiewicz (KL) property
- Work [Kumar et al 16]: different update schedules
- Work [Bai-Scheinberg 15]: different characterization of iteration complexity
- Works [Jiang et al 16, Wang-Yin-Zeng 16]: both relax conditions for the K-agent sharing problem
- Work [Yang-Pong-Chen 15]: enlarges the dual stepsize by $\frac{\sqrt{5}+1}{2} \approx 1.618...$
- ...

New Analysis

New analysis for weaker conditions

- Work [Li-Pong 14]: h, nonconvex, coercive, A_k's full row rank; whole sequence convergence under Kurdyka- Lojasiewicz (KL) property
- Work [Kumar et al 16]: different update schedules
- Work [Bai-Scheinberg 15]: different characterization of iteration complexity
- Works [Jiang et al 16, Wang-Yin-Zeng 16]: both relax conditions for the K-agent sharing problem
- Work [Yang-Pong-Chen 15]: enlarges the dual stepsize by $\frac{\sqrt{5}+1}{2} \approx 1.618...$
- ...

All based upon our analysis framework

New Applications

New applications in FE, SP, ML, Comm etc.

- Risk parity portfolio selection [Bai-Scheinberg 15]
- Solving certain Hamilton-Jacobi equations and differential games [Chow-Darbon-Osher-Yin-16]
- Distributed radio interference calibration [Yatawatta 16]
- Non-convex background/foreground extraction [Yang-Pong-Chen 15]
- Solving QCQP problems [Huang-Sidiropoulos 16]
- Distributed and asynchronous optimization over networks [Chang et al 16]
- Denoising using tight frame regularization [Parekh-Selesnick 15]
- Beamforming design in wireless communications [Kaleva-Tolli-Juntti 15]
- Penalized zero-variance discriminant analysis [Ames-H. 16]
- ...

New Applications

New applications in FE, SP, ML, Comm etc.

- Risk parity portfolio selection [Bai-Scheinberg 15]
- Solving certain Hamilton-Jacobi equations and differential games [Chow-Darbon-Osher-Yin-16]
- Distributed radio interference calibration [Yatawatta 16]
- Non-convex background/foreground extraction [Yang-Pong-Chen 15]
- Solving QCQP problems [Huang-Sidiropoulos 16]
- Distributed and asynchronous optimization over networks [Chang et al 16]
- Denoising using tight frame regularization [Parekh-Selesnick 15]
- Beamforming design in wireless communications [Kaleva-Tolli-Juntti 15]
- Penalized zero-variance discriminant analysis [Ames-H. 16]
- ...

New Applications

New applications in FE, SP, ML, Comm etc.

- Risk parity portfolio selection [Bai-Scheinberg 15]
- Solving certain Hamilton-Jacobi equations and differential games [Chow-Darbon-Osher-Yin-16]
- Distributed radio interference calibration [Yatawatta 16]
- Non-convex background/foreground extraction [Yang-Pong-Chen 15]
- Solving QCQP problems [Huang-Sidiropoulos 16]
- Distributed and asynchronous optimization over networks [Chang et al 16]
- Denoising using tight frame regularization [Parekh-Selesnick 15]
- Beamforming design in wireless communications [Kaleva-Tolli-Juntti 15]
- Penalized zero-variance discriminant analysis [Ames-H. 16]
- ...

New Applications

New applications in FE, SP, ML, Comm etc.

- Risk parity portfolio selection [Bai-Scheinberg 15]
- Solving certain Hamilton-Jacobi equations and differential games [Chow-Darbon-Osher-Yin-16]
- Distributed radio interference calibration [Yatawatta 16]
- Non-convex background/foreground extraction [Yang-Pong-Chen 15]
- Solving QCQP problems [Huang-Sidiropoulos 16]
- Distributed and asynchronous optimization over networks [Chang et al 16]
- Denoising using tight frame regularization [Parekh-Selesnick 15]
- Beamforming design in wireless communications [Kaleva-Tolli-Juntti 15]
- Penalized zero-variance discriminant analysis [Ames-H. 16]
- ...

New Applications

New applications in FE, SP, ML, Comm etc.

- Risk parity portfolio selection [Bai-Scheinberg 15]
- Solving certain Hamilton-Jacobi equations and differential games [Chow-Darbon-Osher-Yin-16]
- Distributed radio interference calibration [Yatawatta 16]
- Non-convex background/foreground extraction [Yang-Pong-Chen 15]
- Solving QCQP problems [Huang-Sidiropoulos 16]
- Distributed and asynchronous optimization over networks [Chang et al 16]
- Denoising using tight frame regularization [Parekh-Selesnick 15]
- Beamforming design in wireless communications [Kaleva-Tolli-Juntti 15]
- Penalized zero-variance discriminant analysis [Ames-H. 16]
- ...

New Connections

Connections of variants of ADMM with `algorithm` for convex problems

+

Nonconvex ADMM analysis

||

Generalize `algorithm` to nonconvex problems

New Connections

Connections of variants of ADMM with `algorithm` for convex problems

$+$

Nonconvex ADMM analysis

$||$

Generalize `algorithm` to nonconvex problems

New Connections

Connections of variants of ADMM with `algorithm` for convex problems

+

Nonconvex ADMM analysis

||

Generalize `algorithm` to nonconvex problems

New Connections

Connections of variants of ADMM with `algorithm` for convex problems

$+$

Nonconvex ADMM analysis

$||$

Generalize `algorithm` to nonconvex problems

New Connections

Connections of variants of ADMM with `algorithm` for convex problems

$+$

Nonconvex ADMM analysis

$||$

Generalize `algorithm` to nonconvex problems

Prox-ADMM = EXTRA

- Apply the Prox-ADMM to consensus over a general network [H. 16],

$$\min_{\mathbf{x}} \; f(\mathbf{x}) := \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad x_i = x_j \;\; \text{if} \;\; i, j \;\; \text{are neighbors}$$

Prox-ADMM = EXTRA

- Apply the Prox-ADMM to consensus over a general network [H. 16],

$$\min_{\mathbf{x}} \ f(\mathbf{x}) := \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad \boxed{x_i = x_j \ \text{ if } \ i,j \ \text{ are neighbors}}$$

Prox-ADMM = EXTRA

- The resulting algorithm is equivalent to the following primal-only iteration

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \frac{1}{2\rho} D^{-1}\left(\nabla f(\mathbf{x}^t) - \nabla f(\mathbf{x}^{t-1})\right) + W\mathbf{x}^t - \frac{1}{2}(I+W)\mathbf{x}^{t-1}$$

where D, W are some network-related matrices

- The above iteration is precisely the EXTRA algorithm [Shi-Ling-Wu-Yin 14] for convex network consensus optimization

New Claim. EXTRA converges sublinearly for nonconvex problems

Prox-ADMM = EXTRA

- The resulting algorithm is equivalent to the following primal-only iteration

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \frac{1}{2\rho} D^{-1}\left(\nabla f(\mathbf{x}^t) - \nabla f(\mathbf{x}^{t-1})\right) + W\mathbf{x}^t - \frac{1}{2}(I+W)\mathbf{x}^{t-1}$$

  where D, W are some network-related matrices

- The above iteration is precisely the EXTRA algorithm [Shi-Ling-Wu-Yin 14] for convex network consensus optimization

New Claim. EXTRA converges sublinearly for nonconvex problems

Prox-ADMM = EXTRA

- The resulting algorithm is equivalent to the following primal-only iteration

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \frac{1}{2\rho} D^{-1}\left(\nabla f(\mathbf{x}^t) - \nabla f(\mathbf{x}^{t-1})\right) + W\mathbf{x}^t - \frac{1}{2}(I+W)\mathbf{x}^{t-1}$$

  where D, W are some network-related matrices

- The above iteration is precisely the EXTRA algorithm [Shi-Ling-Wu-Yin 14] for convex network consensus optimization

New Claim. EXTRA converges sublinearly for nonconvex problems

Prox-ADMM = SAG/IAG/SAGA

- Consider the following convex finite sum problem:

$$\min_{x\in X} \quad f(x) := \frac{1}{N}\sum_{i=1}^{N} g_i(x),$$

where $g_i, \; i = 1, \cdots N$ are cost functions; N is # of data points

- Many popular fast learning algorithms, like SAG [Le Roux-Schmidt-Bach 12], IAG [Blatt et al 07], SAGA [Defazio et al 14]:
  1. Stochastically/deterministically pick one component function g_i
  2. Compute its gradient
  3. Update x^{t+1} by using an average of the past gradients

Prox-ADMM = SAG/IAG/SAGA

- Consider the following convex finite sum problem:

$$\min_{x \in X} \quad f(x) := \frac{1}{N}\sum_{i=1}^{N} g_i(x),$$

where $g_i,\ i = 1, \cdots N$ are cost functions; N is # of data points

- Many popular fast learning algorithms, like SAG [Le Roux-Schmidt-Bach 12], IAG [Blatt et al 07], SAGA [Defazio et al 14]:
    1. Stochastically/deterministically pick one component function g_i
    2. Compute its gradient
    3. Update x^{t+1} by using an average of the past gradients

Prox-ADMM = SAG/IAG/SAGA

Sample one index $i \in \{1, \cdots, N\}$, compute $\nabla g_i(x^r)$

$$z_i^r = x^r, \quad z_j^r = z_j^{r-1}, \; \forall\, j \neq i$$

$$x^{r+1} = x^r - \underbrace{\frac{1}{\beta} \sum_{j=1}^{N} \nabla g_j(z_j^{r-1})}_{\text{averaged past gradients}} + \frac{1}{\alpha} \underbrace{\left(\nabla g_i(z_i^{r-1}) - \nabla g_i(x^r)\right)}_{\text{new gradient info}}$$

- Equivalent to some variants of prox-ADMM [Hajinezhad et al 16]

New Claim. SAG/IAG/SAGA converge sublinearly for nonconvex problems

Prox-ADMM = SAG/IAG/SAGA

Sample one index $i \in \{1, \cdots, N\}$, compute $\nabla g_i(x^r)$

$$z_i^r = x^r, \quad z_j^r = z_j^{r-1}, \; \forall\, j \neq i$$

$$x^{r+1} = x^r - \underbrace{\frac{1}{\beta}\sum_{j=1}^{N} \nabla g_j(z_j^{r-1})}_{\text{averaged past gradients}} + \frac{1}{\alpha}\underbrace{\left(\nabla g_i(z_i^{r-1}) - \nabla g_i(x^r)\right)}_{\text{new gradient info}}$$

- Equivalent to some variants of prox-ADMM [Hajinezhad et al 16]

New Claim. SAG/IAG/SAGA converge sublinearly for nonconvex problems

Prox-ADMM = SAG/IAG/SAGA

$$\boxed{\begin{array}{l}\text{Sample one index } i \in \{1, \cdots, N\}, \text{compute } \nabla g_i(x^r) \\ z_i^r = x^r, \quad z_j^r = z_j^{r-1}, \; \forall\, j \neq i \\ x^{r+1} = x^r - \underbrace{\frac{1}{\beta}\sum_{j=1}^{N} \nabla g_j(z_j^{r-1})}_{\text{averaged past gradients}} + \frac{1}{\alpha}\underbrace{\left(\nabla g_i(z_i^{r-1}) - \nabla g_i(x^r)\right)}_{\text{new gradient info}}\end{array}}$$

- Equivalent to some variants of prox-ADMM [Hajinezhad et al 16]

New Claim. SAG/IAG/SAGA converge sublinearly for nonconvex problems

Outline

1 Overview

2 Literature Review

3 A New Analysis Framework
- A Toy Example
- Nonconvex Consensus Problem
- Algorithm and Analysis

4 Recent Advances

5 Conclusion

Summary

- **Quesiton**: Whether ADMM converges for nonconvex problems?
- Yes, for a class of consensus and sharing problems, and many more
- **Key insights**
  1. The penalty parameters are required to be large enough
  2. The augmented Lagrangian measures the algorithm progress
- **Key technique**: AL as merit function, leading to a three-step analysis

Summary

SAG

[Defazio et al 14]

SAGA

[Le Roux et al 12]

IAG

[Blatt et al 07]

EXTRA

[Shi et al 14]

Signal Processing

[Parekh-Selesnick 15]

Learning

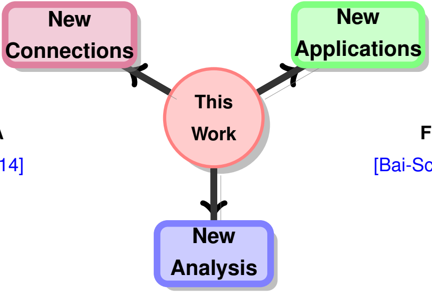
[Ames-H. 16]

Wireless

[Yatawatta 16]

Finance

[Bai-Scheinberg 15]



[Kumar et al 16] [Yang-Pong-Chen 15]

[Jiang et al 16] [Bai-Scheinberg 15]

[Wang-Yin-Zeng 16]

Thank You!