Optimization with Online and Massive Data

Yinyu Ye

K.T. Li Chair Professor of Engineering
Department of Management Science and Engineering
Stanford University
(Guanghua School of Management, Peking University)

2014 Workshop on Optimization for Modern Computation

September 4, 2014

Outline

We present optimization models and/or computational algorithms dealing with online/streamline, structured, and/or massively distributed data:

- **Online Linear Programming**
- Least Squares with Nonconvex Regularization
- The ADMM Method with Multiple Blocks

Background

Consider a store that sells a number of goods/products

- There is a fixed selling period

Background

Consider a store that sells a number of goods/products

- There is a fixed selling period
- There is a fixed inventory of goods

Background

Consider a store that sells a number of goods/products

- There is a fixed selling period
- There is a fixed inventory of goods
- Customers come and require a bundle of goods and bid for certain prices

Background

Consider a store that sells a number of goods/products

- There is a fixed selling period
- There is a fixed inventory of goods
- Customers come and require a bundle of goods and bid for certain prices
- Objective: Maximize the revenue

Background

Consider a store that sells a number of goods/products

- There is a fixed selling period
- There is a fixed inventory of goods
- Customers come and require a bundle of goods and bid for certain prices
- Objective: Maximize the revenue
- Decision: Accept or not?

An Example

	order 1($t=1$)	order 2($t=2$)	.....	Inventory($\mathbf{b}$)
Price(π_t)	\$100	\$30	...	
Decision	x_1	x_2	...	
Pants	1	0	...	100
Shoes	1	0	...	50
T-shirts	0	1	...	500
Jackets	0	0	...	200
Hats	1	1	...	1000

Online Linear Programming Model

The classical offline version of the above program can be formulated as a linear (integer) program as all data would have arrived:

$$\begin{array}{lll} \text{maximize}_{\mathbf{x}} & \sum_{t=1}^{n} \pi_t x_t & \\ \text{subject to} & \sum_{t=1}^{n} a_{it} x_t \leq b_i, & \forall i = 1, ..., m \\ & 0 \leq x_t \leq 1, & \forall t = 1, ..., n \end{array}$$

Online Linear Programming Model

The classical offline version of the above program can be formulated as a linear (integer) program as all data would have arrived:

$$\begin{array}{lll} \text{maximize}_{\mathbf{x}} & \sum_{t=1}^{n} \pi_t x_t & \\ \text{subject to} & \sum_{t=1}^{n} a_{it} x_t \leq b_i, & \forall i = 1, ..., m \\ & 0 \leq x_t \leq 1, & \forall t = 1, ..., n \end{array}$$

Now we consider the online or streamline and data-driven version of this problem:

- We only know $\mathbf{b}$ and n at the start

Online Linear Programming Model

The classical offline version of the above program can be formulated as a linear (integer) program as all data would have arrived:

$$\begin{array}{lll} \text{maximize}_{\mathbf{x}} & \sum_{t=1}^{n} \pi_t x_t & \\ \text{subject to} & \sum_{t=1}^{n} a_{it} x_t \leq b_i, & \forall i = 1, ..., m \\ & 0 \leq x_t \leq 1, & \forall t = 1, ..., n \end{array}$$

Now we consider the online or streamline and data-driven version of this problem:

- We only know $\mathbf{b}$ and n at the start
- the constraint matrix is revealed column by column sequentially along with the corresponding objective coefficient.

Online Linear Programming Model

The classical offline version of the above program can be formulated as a linear (integer) program as all data would have arrived:

$$\begin{array}{lll} \text{maximize}_{\mathbf{x}} & \sum_{t=1}^{n} \pi_t x_t & \\ \text{subject to} & \sum_{t=1}^{n} a_{it} x_t \leq b_i, & \forall i = 1, \ldots, m \\ & 0 \leq x_t \leq 1, & \forall t = 1, \ldots, n \end{array}$$

Now we consider the online or streamline and data-driven version of this problem:

- We only know $\mathbf{b}$ and n at the start
- the constraint matrix is revealed column by column sequentially along with the corresponding objective coefficient.
- an irrevocable decision must be made as soon as an order arrives without observing or knowing the future data.

Application Overview

- Revenue management problems: Airline tickets booking, hotel booking;
- Online network routing on an edge-capacitated network;
- Combinatorial auction;
- Online adwords allocation

Model Assumptions

Main Assumptions

- The columns $\mathbf{a}_t$ arrive in a random order.
- $0 \leq a_{it} \leq 1$, for all (i, t);
- $\pi_t \geq 0$ for all t

Model Assumptions

Main Assumptions

- The columns $\mathbf{a}_t$ arrive in a random order.
- $0 \leq a_{it} \leq 1$, for all (i, t);
- $\pi_t \geq 0$ for all t

Denote the offline maximal value by $OPT(A, \pi)$. We call an online algorithm $\mathcal{A}$ to be c-competitive if and only if

$$E_\sigma \left[\sum_{t=1}^{n} \pi_t x_t(\sigma, \mathcal{A}) \right] \geq c \cdot OPT(A, \pi),$$

where σ is the permutation of arriving order.

A Learning Algorithm is Needed

- There is no distribution known so that any type of stochastic optimization models is not applicable.

A Learning Algorithm is Needed

- There is no distribution known so that any type of stochastic optimization models is not applicable.
- Unlike dynamic programming, the decision maker does not have full information/data so that a backward recursion can not be carried out to find an optimal sequential decision policy.

A Learning Algorithm is Needed

- There is no distribution known so that any type of stochastic optimization models is not applicable.
- Unlike dynamic programming, the decision maker does not have full information/data so that a backward recursion can not be carried out to find an optimal sequential decision policy.
- Thus, the online algorithm needs to be learning-based, in particular, learning-while-doing.

Sufficient and Necessary Results

Theorem

For any fixed $\epsilon > 0$, there is a $1 - \epsilon$ competitive online algorithm for the problem on all inputs when

$$B = \min_i b_i \geq \Omega\left(\frac{m \log (n/\epsilon)}{\epsilon^2}\right)$$

Sufficient and Necessary Results

Theorem

For any fixed $\epsilon > 0$, there is a $1-\epsilon$ competitive online algorithm for the problem on all inputs when

$$B = \min_i b_i \geq \Omega\left(\frac{m \log(n/\epsilon)}{\epsilon^2}\right)$$

Theorem

For any online algorithm for the online linear program in random order model, there exists an instance such that the competitive ratio is less than $1-\epsilon$ if

$$B = \min_i b_i \leq \frac{\log(m)}{\epsilon^2}.$$

Sufficient and Necessary Results

Theorem

For any fixed $\epsilon > 0$, there is a $1 - \epsilon$ competitive online algorithm for the problem on all inputs when

$$B = \min_i b_i \geq \Omega\left(\frac{m \log(n/\epsilon)}{\epsilon^2}\right)$$

Theorem

For any online algorithm for the online linear program in random order model, there exists an instance such that the competitive ratio is less than $1 - \epsilon$ if

$$B = \min_i b_i \leq \frac{\log(m)}{\epsilon^2}.$$

Agrawal, Wang and Y [Operations Research, to appear 2014]

Key Observation and Idea of the Online Algorithm I

The problem would be easy if there is a "fair and optimal price" vector:

	order 1($t = 1$)	order 2($t = 2$)	.....	Inventory($\mathbf{b}$)	$\mathbf{p}^*$
Bid(π_t)	\$100	\$30	...		
Decision	x_1	x_2	...		
Pants	1	0	...	100	\$45
Shoes	1	0	...	50	\$45
T-shirts	0	1	...	500	\$10
Jackets	0	0	...	200	\$55
Hats	1	1	...	1000	\$15

Key Observation and Idea of the Online Algorithm II

- Pricing the bid: The optimal dual price vector $\mathbf{p}^*$ of the offline problem can play such a role, that is $x_t^* = 1$ if $\pi_t > \mathbf{a}_t^T \mathbf{p}^*$ and $x_t^* = 0$ otherwise, yields a near-optimal solution as long as (m/n) is sufficiently small.

Key Observation and Idea of the Online Algorithm II

- Pricing the bid: The optimal dual price vector $\mathbf{p}^*$ of the offline problem can play such a role, that is $x_t^* = 1$ if $\pi_t > \mathbf{a}_t^T \mathbf{p}^*$ and $x_t^* = 0$ otherwise, yields a near-optimal solution as long as (m/n) is sufficiently small.
- Based on this observation, our online algorithm works by learning a threshold price vector $\hat{\mathbf{p}}$ and use $\hat{\mathbf{p}}$ to price the bids.

Key Observation and Idea of the Online Algorithm II

- Pricing the bid: The optimal dual price vector $\mathbf{p}^*$ of the offline problem can play such a role, that is $x_t^* = 1$ if $\pi_t > \mathbf{a}_t^T \mathbf{p}^*$ and $x_t^* = 0$ otherwise, yields a near-optimal solution as long as (m/n) is sufficiently small.
- Based on this observation, our online algorithm works by learning a threshold price vector $\hat{\mathbf{p}}$ and use $\hat{\mathbf{p}}$ to price the bids.
- One-time learning algorithm: learns the price vector once using the initial ϵn input $(1/\epsilon^3)$:

$$\max_{\mathbf{x}} \sum_{t=1}^{\epsilon n} \pi_t x_t \text{ s.t. } \sum_{t=1}^{\epsilon n} a_{it} x_t \leq (1-\epsilon)\epsilon b_i, \; 0 \leq x_t \leq 1, \; \forall i, t.$$

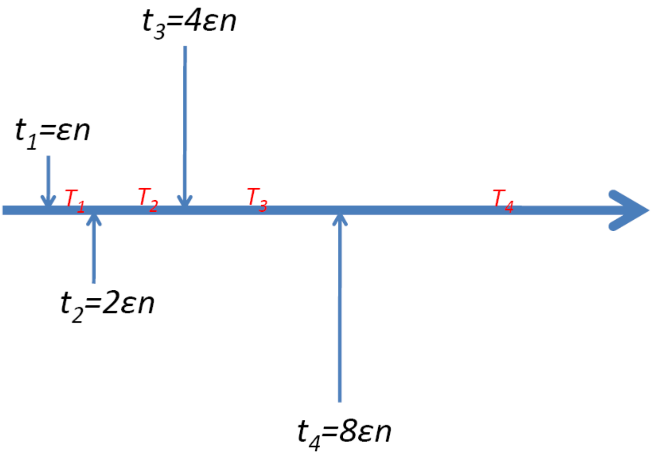
Key Observation and Idea of the Online Algorithm II

- Pricing the bid: The optimal dual price vector $\mathbf{p}^*$ of the offline problem can play such a role, that is $x_t^* = 1$ if $\pi_t > \mathbf{a}_t^T \mathbf{p}^*$ and $x_t^* = 0$ otherwise, yields a near-optimal solution as long as (m/n) is sufficiently small.
- Based on this observation, our online algorithm works by learning a threshold price vector $\hat{\mathbf{p}}$ and use $\hat{\mathbf{p}}$ to price the bids.
- One-time learning algorithm: learns the price vector once using the initial ϵn input $(1/\epsilon^3)$:

$$\max_{\mathbf{x}} \sum_{t=1}^{\epsilon n} \pi_t x_t \text{ s.t. } \sum_{t=1}^{\epsilon n} a_{it} x_t \le (1-\epsilon)\epsilon b_i, \; 0 \le x_t \le 1, \; \forall i, t.$$

- Dynamic learning algorithm: dynamically updates the price vector at a carefully chosen pace $(1/\epsilon^2)$.

Geometric Pace of Price Updating



Related Work on Random-Permutation

	Sufficient Condition	Learning
Kleinberg [2005]	$B \geq \frac{1}{\epsilon^2}$, for $m = 1$	Dynamic
Devanur et al [2009]	$\mathrm{OPT} \geq \frac{m^2 \log(n)}{\epsilon^3}$	One-time
Feldman et al [2010]	$B \geq \frac{m \log n}{\epsilon^3}$ **and** $\mathrm{OPT} \geq \frac{m \log n}{\epsilon}$	One-time
Agrawal et al [2010]	$B \geq \frac{m \log n}{\epsilon^2}$ **or** $\mathrm{OPT} \geq \frac{m^2 \log n}{\epsilon^2}$	Dynamic
Molinaro and Ravi [2013]	$B \geq \frac{m^2 \log m}{\epsilon^2}$	Dynamic
Kesselheim et al [2014]	$B \geq \frac{\log m}{\epsilon^2}$	Dynamic*
Gupta and Molinaro [2014]	$B \geq \frac{\log m}{\epsilon^2}$	Dynamic*

Table: Comparison of several existing results

Summary and Future Questions on OLP

- We have designed a dynamic near-optimal online algorithm for a very general class of online linear programming problems.

Summary and Future Questions on OLP

- We have designed a dynamic near-optimal online algorithm for a very general class of online linear programming problems.
- The algorithm is distribution-free, thus is robust to distribution/data uncertainty.

Summary and Future Questions on OLP

- We have designed a dynamic near-optimal online algorithm for a very general class of online linear programming problems.
- The algorithm is distribution-free, thus is robust to distribution/data uncertainty.
- The dynamic learning algorithm has the feature of "learning-while-doing", and the pace the price is updated is neither too fast nor too slow...

Summary and Future Questions on OLP

- We have designed a dynamic near-optimal online algorithm for a very general class of online linear programming problems.
- The algorithm is distribution-free, thus is robust to distribution/data uncertainty.
- The dynamic learning algorithm has the feature of "learning-while-doing", and the pace the price is updated is neither too fast nor too slow...
- Buy-and-sell model?

Summary and Future Questions on OLP

- We have designed a dynamic near-optimal online algorithm for a very general class of online linear programming problems.
- The algorithm is distribution-free, thus is robust to distribution/data uncertainty.
- The dynamic learning algorithm has the feature of "learning-while-doing", and the pace the price is updated is neither too fast nor too slow...
- Buy-and-sell model?
- Multi-product price-posting market?

Outline

- Online Linear Programming
- **Least Squares with Nonconvex Regularization**
- The ADMM Method with Multiple Blocks

Unconstrained L_2+L_p Minimization

Consider the convex quadratic optimization problem with L_p quasi-norm regularization:

$$\text{Minimize}_x \quad f_p(\mathbf{x}) := \|A\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_p^p, \ \mathbf{x} \in \mathcal{X} \tag{1}$$

where $\mathcal{X}$ is a convex set, data $A \in R^{m \times n}, \mathbf{b} \in R^m$, parameter $0 \leq p < 1$, and

$$\|\mathbf{x}\|_p^p = \sum_j \|x_j\|^p.$$

Unconstrained L_2+L_p Minimization

Consider the convex quadratic optimization problem with L_p quasi-norm regularization:

$$\text{Minimize}_x \quad f_p(\mathbf{x}) := \|A\mathbf{x}-\mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_p^p, \ \mathbf{x} \in \mathcal{X} \tag{1}$$

where $\mathcal{X}$ is a convex set, data $A \in R^{m\times n}, \mathbf{b} \in R^m$, parameter $0 \le p < 1$, and

$$\|\mathbf{x}\|_p^p = \sum_j \|x_j\|^p.$$

When $p=0$: $\|\mathbf{x}\|_0^0 := \|\mathbf{x}\|_0 := |\{j : \ x_j \neq 0\}|$ that is, the number of nonzero entries in $\mathbf{x}$.

Application and Motivation

The original goal is to control $\|\mathbf{x}\|_0^0 = |\{j : \ x_j \neq 0\}|$, the size of the support set of $\mathbf{x}$, for

- Cardinality constrained portfolio management
- Sparse image reconstruction
- Sparse signal recovering
- Compressed sensing – reweighed L_1 seems more effective

Application and Motivation

The original goal is to control $\|\mathbf{x}\|_0^0 = |\{j : x_j \neq 0\}|$, the size of the support set of $\mathbf{x}$, for

- Cardinality constrained portfolio management
- Sparse image reconstruction
- Sparse signal recovering
- Compressed sensing – reweighed L_1 seems more effective

But $L_2 + L_0$ is known to be an NP-Hard problem, and hope $L_2 + L_p$ could be easier...

Modern Portfolio Theory

A case $p = 1$ does not help:

$$\text{Minimize}_x \quad \|A\mathbf{x} - \mathbf{b}\|_2^2, \; \mathbf{e}^T\mathbf{x} = 1, \; \mathbf{x} \geq \mathbf{0};$$

or "short" is allowed:

$$\text{Minimize}_x \quad \|A\mathbf{x} - \mathbf{b}\|_2^2, \; \mathbf{e}^T\mathbf{x} = 1.$$

Modern Portfolio Theory

A case $p = 1$ does not help:

$$\text{Minimize}_x \quad \|A\mathbf{x} - \mathbf{b}\|_2^2,\ \mathbf{e}^T\mathbf{x} = 1,\ \mathbf{x} \geq \mathbf{0};$$

or "short" is allowed:

$$\text{Minimize}_x \quad \|A\mathbf{x} - \mathbf{b}\|_2^2,\ \mathbf{e}^T\mathbf{x} = 1.$$

Let $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-,\ (\mathbf{x}^+, \mathbf{x}^-) \geq \mathbf{0}$. Then,

$$\mathbf{e}^T\mathbf{x}^+ - \mathbf{e}^T\mathbf{x}^- = 1,$$

so that

$$\|\mathbf{x}\|_1 = \mathbf{e}^T\mathbf{x}^+ + \mathbf{e}^T\mathbf{x}^- = 1 + 2\mathbf{e}^T\mathbf{x}^-.$$

Modern Portfolio Theory

A case $p = 1$ does not help:

$$\text{Minimize}_x \quad \|A\mathbf{x} - \mathbf{b}\|_2^2, \; \mathbf{e}^T\mathbf{x} = 1, \; \mathbf{x} \geq \mathbf{0};$$

or "short" is allowed:

$$\text{Minimize}_x \quad \|A\mathbf{x} - \mathbf{b}\|_2^2, \; \mathbf{e}^T\mathbf{x} = 1.$$

Let $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-, \; (\mathbf{x}^+, \mathbf{x}^-) \geq \mathbf{0}$. Then,

$$\mathbf{e}^T\mathbf{x}^+ - \mathbf{e}^T\mathbf{x}^- = 1,$$

so that

$$\|\mathbf{x}\|_1 = \mathbf{e}^T\mathbf{x}^+ + \mathbf{e}^T\mathbf{x}^- = 1 + 2\mathbf{e}^T\mathbf{x}^-.$$

Minimizing $\|\mathbf{x}\|_1$ is about to control the debt exposure, not about the cardinality.

The Hardness Result

Question: Is $L_2 + L_p$ minimization easier than $L_2 + L_0$ minimization?

The Hardness Result

Question: Is $L_2 + L_p$ minimization easier than $L_2 + L_0$ minimization?

Theorem

Deciding the global minimal objective value of $L_2 + L_p$ minimization is strongly NP-hard for any given $0 \leq p < 1$ and $\lambda > 0$.

The Hardness Result

Question: Is $L_2 + L_p$ minimization easier than $L_2 + L_0$ minimization?

Theorem

Deciding the global minimal objective value of $L_2 + L_p$ minimization is strongly NP-hard for any given $0 \leq p < 1$ and $\lambda > 0$.

Chen, Ge, Jian, Wang and Y [*Math Programming* 2011 and 2014]

The Easiness Result

However,

Theorem

There are FPTAS algorithms that provably compute a (second-order) ϵ-KKT point of $L_2 + L_p$ minimization.

The Easiness Result

However,

Theorem

There are FPTAS algorithms that provably compute a (second-order) ϵ-KKT point of $L_2 + L_p$ minimization.

Bian, Chen, Ge, Jian, and Y [*Math Programming* 2011 and 2014]

The Easiness Result

However,

Theorem

There are FPTAS algorithms that provably compute a (second-order) ϵ-KKT point of $L_2 + L_p$ minimization.

Bian, Chen, Ge, Jian, and Y [*Math Programming* 2011 and 2014]

Question: Does any (second-order) KKT point or solution possess predictable sparse properties?

Theory of Constrained L_2+L_p: First-Order Bound

Theorem

Let $\mathbf{x}^*$ *be any first-order KKT point and let*

$$L_i = \left(\frac{\lambda p}{2\|\mathbf{a}_i\| \sqrt{f(\mathbf{x}^*)}} \right)^{\frac{1}{1-p}} .$$

Then, for any i, *either* $x_i^* = 0$ *or* $|x_i^*| \geq L_i$.

Theory of Constrained L_2+L_p: Second-Order Bound

Theorem

Let $\mathbf{x}^*$ be any KKT point that satisfies the second-order necessary conditions and let

$$L_i = \left(\frac{\lambda p(1-p)}{2\|\mathbf{a}_i\|^2}\right)^{\frac{1}{2-p}}.$$

Then, for any i, either $x_i^* = 0$ or $|x_i^*| \geq L_i$. Moreover, the support columns of $\mathbf{x}^*$ are linearly independent.

Theory of Constrained L_2+L_p: Second-Order Bound

Theorem

Let $\mathbf{x}^*$ be any KKT point that satisfies the second-order necessary conditions and let

$$L_i = \left(\frac{\lambda p(1-p)}{2\|\mathbf{a}_i\|^2} \right)^{\frac{1}{2-p}} .$$

Then, for any i, either $x_i^* = 0$ or $|x_i^*| \geq L_i$. Moreover, the support columns of $\mathbf{x}^*$ are linearly independent.

Chen, Xu and Y [*SIAM Journal on Scientific Computing* 2010]

Extension to other Regularizations

Consider the Least Squares problem with any non-convex regularization:

$$\text{Minimize}_x \quad f_p(\mathbf{x}) := \|A\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \textstyle\sum_i \phi(|x_i|)$$

where $\phi(\cdot)$ is a concave increasing function.

Extension to other Regularizations

Consider the Least Squares problem with any non-convex regularization:

$$\text{Minimize}_x \quad f_p(\mathbf{x}) := \|A\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \textstyle\sum_i \phi(|x_i|)$$

where $\phi(\cdot)$ is a concave increasing function.

First-order bound: either $x_i^* = 0$ or $2\|\mathbf{a}_i\|\sqrt{f(\mathbf{x}^*)} \geq \lambda|\phi'(x_i^*)|$.

Extension to other Regularizations

Consider the Least Squares problem with any non-convex regularization:

$$\text{Minimize}_x \quad f_p(\mathbf{x}) := \|A\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \sum_i \phi(|x_i|)$$

where $\phi(\cdot)$ is a concave increasing function.

First-order bound: either $x_i^* = 0$ or $2\|\mathbf{a}_i\|\sqrt{f(\mathbf{x}^*)} \geq \lambda|\phi'(x_i^*)|$.

Second-order bound: either $x_i^* = 0$ or $2\|\mathbf{a}_i\|^2 \geq \lambda|\phi''(x_i^*)|$.

Summary and Future Questions on LSNR

- Unfortunately, finding the global minimizer of LSNR problems is (strongly) NP-hard;

Summary and Future Questions on LSNR

- Unfortunately, finding the global minimizer of LSNR problems is (strongly) NP-hard;
- but fortunately finding an KKT point is easy!

Summary and Future Questions on LSNR

- Unfortunately, finding the global minimizer of LSNR problems is (strongly) NP-hard;
- but fortunately finding an KKT point is easy!
- There are desired structure properties of any KKT point of LSNR problems.

Summary and Future Questions on LSNR

- Unfortunately, finding the global minimizer of LSNR problems is (strongly) NP-hard;
- but fortunately finding an KKT point is easy!
- There are desired structure properties of any KKT point of LSNR problems.
- Could one apply statistical analyses to a local minimizers or KKT points of LSNR?

Summary and Future Questions on LSNR

- Unfortunately, finding the global minimizer of LSNR problems is (strongly) NP-hard;
- but fortunately finding an KKT point is easy!
- There are desired structure properties of any KKT point of LSNR problems.
- Could one apply statistical analyses to a local minimizers or KKT points of LSNR?
- When is a local minimizer of LSNR also global?

Summary and Future Questions on LSNR

- Unfortunately, finding the global minimizer of LSNR problems is (strongly) NP-hard;
- but fortunately finding an KKT point is easy!
- There are desired structure properties of any KKT point of LSNR problems.
- Could one apply statistical analyses to a local minimizers or KKT points of LSNR?
- When is a local minimizer of LSNR also global?
- Faster algorithms for solving LSNR, such as ADMM convergence for two blocks:

Summary and Future Questions on LSNR

- Unfortunately, finding the global minimizer of LSNR problems is (strongly) NP-hard;
- but fortunately finding an KKT point is easy!
- There are desired structure properties of any KKT point of LSNR problems.
- Could one apply statistical analyses to a local minimizers or KKT points of LSNR?
- When is a local minimizer of LSNR also global?
- Faster algorithms for solving LSNR, such as ADMM convergence for two blocks:

$$\min \ f(\mathbf{x}) + r(\mathbf{y}), \ \text{s.t. } \mathbf{x} - \mathbf{y} = \mathbf{0}, \ \mathbf{x} \in X?$$

Outline

- Distributionally Robust Optimization
- Online Linear Programming
- Least Squares with Nonconvex Regularization
- **The ADMM Method with Multiple Blocks**

Alternating Direction Method of Multipliers I

$$\boxed{\min\left\{\theta_1(\mathbf{x}_1)+\theta_2(\mathbf{x}_2) \mid A_1\mathbf{x}_1+A_2\mathbf{x}_2=\mathbf{b},\ \mathbf{x}_1\in\mathcal{X}_1,\mathbf{x}_2\in\mathcal{X}_2\right\}}$$

- $\theta_1(\mathbf{x}_1)$ and $\theta_2(\mathbf{x}_2)$ are convex closed proper functions;
- $\mathcal{X}_1$ and $\mathcal{X}_2$ are convex sets.

Alternating Direction Method of Multipliers I

$$\boxed{\min \{\theta_1(\mathbf{x}_1) + \theta_2(\mathbf{x}_2) \mid A_1\mathbf{x}_1 + A_2\mathbf{x}_2 = \mathbf{b},\ \mathbf{x}_1 \in \mathcal{X}_1, \mathbf{x}_2 \in \mathcal{X}_2\}}$$

- $\theta_1(\mathbf{x}_1)$ and $\theta_2(\mathbf{x}_2)$ are convex closed proper functions;
- $\mathcal{X}_1$ and $\mathcal{X}_2$ are convex sets.

Original ADMM (Glowinski & Marrocco '75, Gabay & Mercier '76):

$$\begin{cases} \mathbf{x}_1^{k+1} = \arg\min\{\mathcal{L}_{\mathcal{A}}(\mathbf{x}_1, \mathbf{x}_2^k, \lambda^k) \mid \mathbf{x}_1 \in \mathcal{X}_1\}, \\ \mathbf{x}_2^{k+1} = \arg\min\{\mathcal{L}_{\mathcal{A}}(\mathbf{x}_1^{k+1}, \mathbf{x}_2, \lambda^k) \mid \mathbf{x}_2 \in \mathcal{X}_2\}, \\ \lambda^{k+1} = \lambda^k - \beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} - \mathbf{b}), \end{cases}$$

where the augmented Lagrangian function $\mathcal{L}_{\mathcal{A}}$ is defined as

$$\mathcal{L}_{\mathcal{A}}(\mathbf{x}_1, \mathbf{x}_2, \lambda) = \sum_{i=1}^{2} \theta_i(\mathbf{x}_i) - \lambda^T\Big(\sum_{i=1}^{2} A_i\mathbf{x}_i - \mathbf{b}\Big) + \frac{\beta}{2}\Big\|\sum_{i=1}^{2} A_i\mathbf{x}_i - \mathbf{b}\Big\|^2.$$

ADMM for Multi-block Convex Minimization Problems

Convex minimization problems with three blocks:

$$\boxed{\begin{array}{ll} \min & \theta_1(\mathbf{x}_1) + \theta_2(\mathbf{x}_2) + \theta_3(\mathbf{x}_3) \\ \text{s.t.} & A_1\mathbf{x}_1 + A_2\mathbf{x}_2 + A_3\mathbf{x}_3 = \mathbf{b} \\ & \mathbf{x}_1 \in \mathcal{X}_1,\ \mathbf{x}_2 \in \mathcal{X}_2,\ \mathbf{x}_3 \in \mathcal{X}_3 \end{array}}$$

ADMM for Multi-block Convex Minimization Problems

Convex minimization problems with three blocks:

$$\boxed{\begin{array}{ll} \min & \theta_1(\mathbf{x}_1) + \theta_2(\mathbf{x}_2) + \theta_3(\mathbf{x}_3) \\ \text{s.t.} & A_1\mathbf{x}_1 + A_2\mathbf{x}_2 + A_3\mathbf{x}_3 = \mathbf{b} \\ & \mathbf{x}_1 \in \mathcal{X}_1,\ \mathbf{x}_2 \in \mathcal{X}_2,\ \mathbf{x}_3 \in \mathcal{X}_3 \end{array}}$$

The direct and natural extension of ADMM:

$$\left\{ \begin{array}{l} \mathbf{x}_1^{k+1} = \arg\min\{\mathcal{L}_{\mathcal{A}}(\mathbf{x}_1, \mathbf{x}_2^k, \mathbf{x}_3^k, \lambda^k) \,|\, \mathbf{x}_1 \in \mathcal{X}_1\} \\ \mathbf{x}_2^{k+1} = \arg\min\{\mathcal{L}_{\mathcal{A}}(\mathbf{x}_1^{k+1}, \mathbf{x}_2, \mathbf{x}_3^k, \lambda^k) \,|\, \mathbf{x}_2 \in \mathcal{X}_2\} \\ \mathbf{x}_3^{k+1} = \arg\min\{\mathcal{L}_{\mathcal{A}}(\mathbf{x}_1^{k+1}, \mathbf{x}_2^{k+1}, \mathbf{x}_3, \lambda^k) \,|\, \mathbf{x}_3 \in \mathcal{X}_3\} \\ \lambda^{k+1} = \lambda^k - \beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} + A_3\mathbf{x}_3^{k+1} - \mathbf{b}) \end{array} \right.$$

$$\mathcal{L}_{\mathcal{A}}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \lambda) = \sum_{i=1}^{3} \theta_i(\mathbf{x}_i) - \lambda^T\Big(\sum_{i=1}^{3} A_i\mathbf{x}_i - \mathbf{b}\Big) + \frac{\beta}{2}\Big\| \sum_{i=1}^{3} A_i\mathbf{x}_i - \mathbf{b}\Big\|^2$$

Existing Theoretical Results of the Extended ADMM

Not easy to analyze the convergence: the operator theory for the ADMM cannot be directly extended to the ADMM with three blocks. Big difference between the ADMM with two blocks and with three blocks.

Existing Theoretical Results of the Extended ADMM

Not easy to analyze the convergence: the operator theory for the ADMM cannot be directly extended to the ADMM with three blocks. Big difference between the ADMM with two blocks and with three blocks. Existing results for global convergence:

- Strong convexity; plus β in a specific range (Han & Yuan '12).
- Certain conditions on the problem; then take a sufficiently small stepsize γ (Hong & Luo '12)

$$\lambda^{k+1} = \lambda^k - \gamma\beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} + A_3\mathbf{x}_3^{k+1} - \mathbf{b}).$$

- A correction term (He et al. '12, He et al. -IMA, Deng at al. '14, Ma et al. '14...)

Existing Theoretical Results of the Extended ADMM

Not easy to analyze the convergence: the operator theory for the ADMM cannot be directly extended to the ADMM with three blocks. Big difference between the ADMM with two blocks and with three blocks. Existing results for global convergence:

- Strong convexity; plus β in a specific range (Han & Yuan '12).
- Certain conditions on the problem; then take a sufficiently small stepsize γ (Hong & Luo '12)

$$\lambda^{k+1} = \lambda^k - \gamma\beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} + A_3\mathbf{x}_3^{k+1} - \mathbf{b}).$$

- A correction term (He et al. '12, He et al. -IMA, Deng at al. '14, Ma et al. '14...)

But, these did not answer the open question whether or not the direct extension of ADMM converges under the simple convexity assumption.

Divergent Example of the Extended ADMM I

We simply consider the system of homogeneous linear equations with three variables:

$A_1 x_1 + A_2 x_2 + A_3 x_3 = \mathbf{0}$, where

Divergent Example of the Extended ADMM I

We simply consider the system of homogeneous linear equations with three variables:

$$A_1x_1 + A_2x_2 + A_3x_3 = \mathbf{0}, \text{ where } A = (A_1, A_2, A_3) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{pmatrix}.$$

Divergent Example of the Extended ADMM I

We simply consider the system of homogeneous linear equations with three variables:

$$A_1x_1 + A_2x_2 + A_3x_3 = \mathbf{0}, \text{ where } A = (A_1, A_2, A_3) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{pmatrix}.$$

Then the extended ADMM with $\beta = 1$ can be specified as a linear map

$$\begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 4 & 6 & 0 & 0 & 0 & 0 \\ 5 & 7 & 9 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 2 & 0 & 1 & 0 \\ 1 & 2 & 2 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1^{k+1} \\ x_2^{k+1} \\ x_3^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} 0 & -4 & -5 & 1 & 1 & 1 \\ 0 & 0 & -7 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 & 2 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1^{k} \\ x_2^{k} \\ x_3^{k} \\ \lambda^{k} \end{pmatrix}.$$

Divergent Example of the Extended ADMM II

Or equivalently,

$$\begin{pmatrix} x_2^{k+1} \\ x_3^{k+1} \\ \lambda^{k+1} \end{pmatrix} = M \begin{pmatrix} x_2^{k} \\ x_3^{k} \\ \lambda^{k} \end{pmatrix},$$

where

$$M = \frac{1}{162} \begin{pmatrix} 144 & -9 & -9 & -9 & 18 \\ 8 & 157 & -5 & 13 & -8 \\ 64 & 122 & 122 & -58 & -64 \\ 56 & -35 & -35 & 91 & -56 \\ -88 & -26 & -26 & -62 & 88 \end{pmatrix}.$$

Divergent Example of the Extended ADMM III

The matrix $M = V\mathrm{Diag}(\mathrm{d})\mathrm{V}^{-1}$, where

$$d = \begin{pmatrix} 0.9836 + 0.2984i \\ 0.9836 - 0.2984i \\ 0.8744 + 0.2310i \\ 0.8744 - 0.2310i \\ 0 \end{pmatrix}.$$

Note that $\rho(M) = |d_1| = |d_2| > 1$.

Theorem

There exist an example where the direct extension of ADMM of three blocks with any real number initial point in a subspace is not convergent for any choice of β.

Chen, He, Y, and Yuan [*Manuscript* 2013]

Divergent Example of the Extended ADMM III

The matrix $M = V\mathrm{Diag}(\mathrm{d})\mathrm{V}^{-1}$, where

$$d = \begin{pmatrix} 0.9836 + 0.2984i \\ 0.9836 - 0.2984i \\ 0.8744 + 0.2310i \\ 0.8744 - 0.2310i \\ 0 \end{pmatrix}.$$

Note that $\rho(M) = |d_1| = |d_2| > 1$.

Theorem

There exist an example where the direct extension of ADMM of three blocks with any real number initial point in a subspace is not convergent for any choice of β.

Chen, He, Y, and Yuan [*Manuscript* 2013]

Corollary

When starting from a random point, there exist an example the direct extension of ADMM of three blocks is not convergent with probability one for any choice of β.

Strong Convexity Helps?

Consider the following example

$$\begin{array}{ll} \min & 0.05x_1^2 + 0.05x_2^2 + 0.05x_3^2 \\ \text{s.t.} & \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0. \end{array}$$

Strong Convexity Helps?

Consider the following example

$$\begin{array}{ll} \min & 0.05x_1^2 + 0.05x_2^2 + 0.05x_3^2 \\ \text{s.t.} & \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0. \end{array}$$

- Then, the linear mapping matrix M in the extended ADMM ($\beta = 1$) has $\rho(M) = 1.0087 > 1$

Strong Convexity Helps?

Consider the following example

$$\begin{array}{ll} \min & 0.05x_1^2 + 0.05x_2^2 + 0.05x_3^2 \\ \text{s.t.} & \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0. \end{array}$$

- Then, the linear mapping matrix M in the extended ADMM ($\beta = 1$) has $\rho(M) = 1.0087 > 1$
- Able to find a proper initial point such that the extended ADMM diverges

Strong Convexity Helps?

Consider the following example

$$\begin{array}{ll} \min & 0.05x_1^2 + 0.05x_2^2 + 0.05x_3^2 \\ \text{s.t.} & \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0. \end{array}$$

- Then, the linear mapping matrix M in the extended ADMM ($\beta = 1$) has $\rho(M) = 1.0087 > 1$
- Able to find a proper initial point such that the extended ADMM diverges
- even for strongly convex programming, the extended ADMM is not necessarily convergent for a certain $\beta > 0$.

The Small-Stepsized ADMM

Recall that, In the small stepsized ADMM, the Lagrangian multiplier is updated by

$$\lambda^{k+1} := \lambda^k - \gamma\beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} + \ldots + A_3\mathbf{x}_3^{k+1}).$$

The Small-Stepsized ADMM

Recall that, In the small stepsized ADMM, the Lagrangian multiplier is updated by

$$\lambda^{k+1} := \lambda^k - \gamma\beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} + \ldots + A_3\mathbf{x}_3^{k+1}).$$

Convergence is proved:

The Small-Stepsized ADMM

Recall that, In the small stepsized ADMM, the Lagrangian multiplier is updated by

$$\lambda^{k+1} := \lambda^k - \gamma\beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} + \ldots + A_3\mathbf{x}_3^{k+1}).$$

Convergence is proved:

- One block (Augmented Lagrangian Method): $\gamma \in (0, 2)$, (Hestenes '69, Powell '69).

The Small-Stepsized ADMM

Recall that, In the small stepsized ADMM, the Lagrangian multiplier is updated by

$$\lambda^{k+1} := \lambda^k - \gamma\beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} + \ldots + A_3\mathbf{x}_3^{k+1}).$$

Convergence is proved:

- One block (Augmented Lagrangian Method): $\gamma \in (0, 2)$, (Hestenes '69, Powell '69).
- Two blocks (Alternating Direction Method of Multipliers: $\gamma \in (0, \frac{1+\sqrt{5}}{2})$, (Glowinski, '84).

The Small-Stepsized ADMM

Recall that, In the small stepsized ADMM, the Lagrangian multiplier is updated by

$$\lambda^{k+1} := \lambda^k - \gamma\beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} + \ldots + A_3\mathbf{x}_3^{k+1}).$$

Convergence is proved:

- One block (Augmented Lagrangian Method): $\gamma \in (0, 2)$, (Hestenes '69, Powell '69).
- Two blocks (Alternating Direction Method of Multipliers: $\gamma \in (0, \frac{1+\sqrt{5}}{2})$, (Glowinski, '84).
- Three blocks: for γ sufficiently small provided additional conditions on the problem, (Hong & Luo '12).

The Small-Stepsized ADMM

Recall that, In the small stepsized ADMM, the Lagrangian multiplier is updated by

$$\lambda^{k+1} := \lambda^k - \gamma\beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} + \ldots + A_3\mathbf{x}_3^{k+1}).$$

Convergence is proved:

- One block (Augmented Lagrangian Method): $\gamma \in (0, 2)$, (Hestenes '69, Powell '69).
- Two blocks (Alternating Direction Method of Multipliers: $\gamma \in (0, \frac{1+\sqrt{5}}{2})$, (Glowinski, '84).
- Three blocks: for γ sufficiently small provided additional conditions on the problem, (Hong & Luo '12).

Question: Is there a problem-data-independent γ such that the method converges?

A Numerical Study

For any given $\gamma > 0$, consider the linear system

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1+\gamma \\ 1 & 1+\gamma & 1+\gamma \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0.$$

A Numerical Study

For any given $\gamma > 0$, consider the linear system

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1+\gamma \\ 1 & 1+\gamma & 1+\gamma \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0.$$

Table: The radius of M

γ	1	0.1	1e-2	1e-3	1e-4	1e-5	1e-6	1e-7
$\rho(M)$	1.0278	1.0026	1.0001	>1	>1	>1	>1	>1

A Numerical Study

For any given $\gamma > 0$, consider the linear system

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1+\gamma \\ 1 & 1+\gamma & 1+\gamma \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0.$$

Table: The radius of M

γ	1	0.1	1e-2	1e-3	1e-4	1e-5	1e-6	1e-7
$\rho(M)$	1.0278	1.0026	1.0001	> 1	> 1	> 1	> 1	> 1

Thus, there seems no practical problem-data-independent γ such that the small-stepsized ADMM variant works.

Summary and Future Questions on ADMM

- We construct examples to show that the direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent for any given algorithm parameter β.

Summary and Future Questions on ADMM

- We construct examples to show that the direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent for any given algorithm parameter β.
- Even in the case where the objective function is strongly convex, the direct extension of ADMM loses its convergence for certain βs.

Summary and Future Questions on ADMM

- We construct examples to show that the direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent for any given algorithm parameter β.
- Even in the case where the objective function is strongly convex, the direct extension of ADMM loses its convergence for certain βs.
- There doesn't exist a problem-data-independent stepsize γ such that the small-stepsized variant of ADMM would work.

Summary and Future Questions on ADMM

- We construct examples to show that the direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent for any given algorithm parameter β.
- Even in the case where the objective function is strongly convex, the direct extension of ADMM loses its convergence for certain βs.
- There doesn't exist a problem-data-independent stepsize γ such that the small-stepsized variant of ADMM would work.
- Is there a cyclic non-converging example?

Summary and Future Questions on ADMM

- We construct examples to show that the direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent for any given algorithm parameter β.
- Even in the case where the objective function is strongly convex, the direct extension of ADMM loses its convergence for certain βs.
- There doesn't exist a problem-data-independent stepsize γ such that the small-stepsized variant of ADMM would work.
- Is there a cyclic non-converging example?
- Our results support the need of a correction step in the ADMM-type method (He&Tao&Yuan 12', He&Tao&Yuan-IMA,...).

Summary and Future Questions on ADMM

- We construct examples to show that the direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent for any given algorithm parameter β.
- Even in the case where the objective function is strongly convex, the direct extension of ADMM loses its convergence for certain βs.
- There doesn't exist a problem-data-independent stepsize γ such that the small-stepsized variant of ADMM would work.
- Is there a cyclic non-converging example?
- Our results support the need of a correction step in the ADMM-type method (He&Tao&Yuan 12', He&Tao&Yuan-IMA,...).
- Question: Is there a "simple correction" of the ADMM for the multi-block convex minimization problems? Or how to treat the multi blocks "equally"?

How to Treat All Blocks Equally?

Answer: Independent random permutation in each iteration!

How to Treat All Blocks Equally?

Answer: Independent random permutation in each iteration!

- Select the block-update order in the uniformly random fashion – this equivalently reduces the ADMM algorithm to one block.

How to Treat All Blocks Equally?

Answer: Independent random permutation in each iteration!

- Select the block-update order in the uniformly random fashion – this equivalently reduces the ADMM algorithm to one block.
- Or fix the first block, and then select the rest block order in the uniformly random fashion – this equivalently reduces the ADMM algorithm to two blocks.

How to Treat All Blocks Equally?

Answer: Independent random permutation in each iteration!

- Select the block-update order in the uniformly random fashion – this equivalently reduces the ADMM algorithm to one block.
- Or fix the first block, and then select the rest block order in the uniformly random fashion – this equivalently reduces the ADMM algorithm to two blocks.
- It works for the example – the expected $\rho(M)$ equals 0.9723!

How to Treat All Blocks Equally?

Answer: Independent random permutation in each iteration!

- Select the block-update order in the uniformly random fashion – this equivalently reduces the ADMM algorithm to one block.
- Or fix the first block, and then select the rest block order in the uniformly random fashion – this equivalently reduces the ADMM algorithm to two blocks.
- It works for the example – the expected $\rho(M)$ equals 0.9723!
- It works in general?