Sparse Optimization
Lecture: Dual Methods, Part II

Instructor: Wotao Yin

July 2013

online discussions on piazza.com

Those who complete this lecture will know

- the alternating direction method of multipliers (ADMM)
- the variants of ADMM
- basic convergence results of ADMM
- its applications

Outline

Separable objective and coupling constraints

Consider a convex program with a *separable objective* and *coupling constraints*

$$\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \quad \text{s.t. } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b}.$$

Examples:

- $\min f(\mathbf{x}) + g(\mathbf{x}) \implies \min_{\mathbf{x},\mathbf{z}}\{f(\mathbf{x}) + g(\mathbf{z}) : \mathbf{x} - \mathbf{z} = 0\}$
- $\min f(\mathbf{x}) + g(\mathbf{A}\mathbf{x}) \implies \min_{\mathbf{x},\mathbf{z}}\{f(\mathbf{x}) + g(\mathbf{z}) : \mathbf{A}\mathbf{x} - \mathbf{z} = 0\}$
- $\min\{f(\mathbf{x}) : \mathbf{A}\mathbf{x} \in \mathcal{C}\} \implies \min_{\mathbf{x},\mathbf{z}}\{f(\mathbf{x}) + \iota_{\mathcal{C}}(\mathbf{z}) : \mathbf{A}\mathbf{x} - \mathbf{z} = 0\}$
- $\min \sum_{i=1}^N f_i(\mathbf{x}) \implies \min_{\{\mathbf{x}_i\},\mathbf{z}}\{\sum_{i=1}^N f_i(\mathbf{x}_i) : \mathbf{x}_i - \mathbf{z} = 0, \forall i\}$

  each $\mathbf{x}_i$ is **a copy** of $\mathbf{x}$ for f_i, *not* a subvector of $\mathbf{x}$.

Alternating direction method of multipliers (ADMM)

Consider

$$\begin{aligned} \min_{\mathbf{x},\mathbf{z}} \quad & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b}. \end{aligned}$$

f and g are **convex**, maybe **nonsmooth**, can take the **extended value**

Standard ADMM iteration

1. $\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{z}^k) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^k - \mathbf{b} - \mathbf{y}^k\|_2^2$,
2. $\mathbf{z}^{k+1} = \arg\min_{\mathbf{z}} f(\mathbf{x}^{k+1}) + g(\mathbf{z}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z} - \mathbf{b} - \mathbf{y}^k\|_2^2$,
3. $\mathbf{y}^{k+1} = \mathbf{y}^k - (\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{b})$.

Dates back to Douglas, Peaceman, and Rachford (50s–70s, operator splitting for PDEs); Glowinsky et al.'80s, Gabay'83; Spingarn'85; Eckstein and Bertsekes'92, He et al.'02 in variational inequality.

Alternating direction method of multipliers (ADMM)

Comments:

- $\mathbf{y}$ is the **scaled dual variable**, $\mathbf{y} = \beta \cdot$Lagrange multipliers
- $\mathbf{y}$-update can take a large step size $\gamma < \frac{1}{2}(\sqrt{5}+1)$

$$\mathbf{y}^{k+1} = \mathbf{y}^k - \gamma(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{b}).$$

- Gauss-Seidel style update is applied to $\mathbf{x}$ and $\mathbf{z}$ of either order
- If $\mathbf{x}$ and $\mathbf{z}$ are minimized jointly, it reduces to augmented Lagrangian itr:

$$(\mathbf{x}^{k+1}, \mathbf{z}^{k+1}) = \arg\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{b} - \mathbf{y}^k\|_2^2$$
$$\mathbf{y}^{k+1} = \mathbf{y}^k - (\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{b}).$$

- it extends to multiple blocks (a few questions remain open)
- it extends to Jacobian (parallel) updates with damping the update of $\mathbf{y}$

Why is ADMM liked

- Split awkward intersections and objectives to easy subproblems
  - $\mathbf{X} \succeq \mathbf{0}, \mathbf{X} \geq 0 \longrightarrow$ separate projections
  - $\|\mathbf{L}\|_* + \beta\|\mathbf{M} - \mathbf{L}\|_1 \longrightarrow$ separate subproblems with $\|\cdot\|_*$ and $\|\cdot\|_1$
  - $\|\nabla \mathbf{x}\|_1 \longrightarrow$ decouple $\|\cdot\|_1$ and ∇ to separable subproblems
  - $\sum_i \|\mathbf{x}_{[\mathcal{G}_i]}\|_2 \longrightarrow$ decouple to subproblems of individual groups
  - $\sum_{i=1}^K f_i(\mathbf{x}) \longrightarrow K$ parallel subproblems (coordinated by gather-scattering or gossiping between neighbors)
- $\#$ iterations is comparable to those of other first-order methods, so the total time can be much smaller (not always though)
- Quite easy to implement, be (nearly) state-of-the-art for a few hours' work

Outline

KKT conditions

Recall KKT conditions (omitting the complementarity part):

$$
\begin{aligned}
\text{(primal feasibility)} \quad & \mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{z}^* = \mathbf{b} \\
\text{(dual feasibility I)} \quad & 0 \in \partial f(\mathbf{x}^*) + \mathbf{A}^T \mathbf{y}^* \\
\text{(dual feasibility II)} \quad & 0 \in \partial g(\mathbf{z}^*) + \mathbf{B}^T \mathbf{y}^*
\end{aligned}
$$

Recall $\mathbf{z}^{k+1} = \arg\min_{\mathbf{z}} g(\mathbf{z}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z} - \mathbf{b} - \mathbf{y}^k\|_2^2$

$$\implies 0 \in \partial g(\mathbf{z}^{k+1}) + \mathbf{B}^T(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{b} - \mathbf{y}^k) = \partial g(\mathbf{z}^{k+1}) + \mathbf{B}^T \mathbf{y}^{k+1}$$

Therefore, dual feasibility II is maintained.

Dual feasibility I is not maintained since

$$0 \in \partial f(\mathbf{x}^{k+1}) + \mathbf{A}^T \left(\mathbf{y}^{k+1} + \mathbf{B}(\mathbf{z}^k - \mathbf{z}^{k+1})\right)$$

But, primal feasibility and dual feasibility I hold asymptotically as $k \to \infty$.

Convergence of ADMM

ADMM is neither purely-primal nor purely-dual. There is no known objective closely associated with the iterations.

Recall via the transform

$$\mathbf{y}^k = \mathbf{prox}_{\beta d_1} \mathbf{w}^k,$$

ADMM is a fixed-point iteration

$$\mathbf{w}^{k+1} = \left(\frac{1}{2} I + \frac{1}{2} \mathbf{refl}_{\beta d_1} \mathbf{refl}_{\beta d_2} \right) \mathbf{w}^k,$$

where the operator is firmly nonexpansive.

Convergence

- Assumptions: f and g convex, closed, proper, and $\exists$ KKT point
- $\mathbf{A}\mathbf{x}^k + \mathbf{B}\mathbf{z}^k \to \mathbf{b}$, $f(\mathbf{x}^k) + g(\mathbf{z}^k) \to p^*$, $\mathbf{y}^k$ converge
- In addition, if $(\mathbf{x}^k, \mathbf{y}^k)$ are bounded, they also converge

Rate of convergence

- It is on-going work
- Some existing results:
  - simplified cases, exact updates, f smooth, and ∇f Lipschitz $\longrightarrow$ objective $\sim O(1/k)$, $O(1/k^2)$
  - at least one update is exact $\longrightarrow$
    ergodic: objective error $+(\tilde{\mathbf{u}}^k - \mathbf{u}^*)^T F(\mathbf{u}^*) \sim O(1/k)$
    non-ergodic: $\|\mathbf{u}^k - \mathbf{u}^{k+1}\| \sim O(1/k)$
  - f strongly convex and ∇f Lipschitz + some full rank conditions $\longrightarrow$ both solution and objective $\sim O(1/c^k)$, $c > 1$
  - applied to LP and QP $\longrightarrow$ (asymptotic) strongly convex

Outline

Variants of ADMM

- An ADMM subproblem is easy, if it has a closed-form solution;
- If a subproblem is difficult, it may be not worth solving it exactly.

This motivates variants of ADMM.

A few approaches of inexact ADMM subproblems:

1. **Iteration limiter**: limited iterations of CG or L-BFGS applied to

$$\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{v}\|_2^2$$

   where $\mathbf{v} = \mathbf{b} - \mathbf{B}\mathbf{z}^k + \mathbf{y}^k$.

   - Applicable to quadratic f, perhaps other C^2 functions as well
   - Does not apply to nonsmooth subproblems
   - Practically efficient, but lacking theoretical guarantees for now

Variants of ADMM

2. **Cached factorization:** For quadratic subproblem $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{C}\mathbf{x} - \mathbf{d}\|_2^2$, $\mathbf{x}$-subproblem solves

$$(\mathbf{C}^T\mathbf{C} + \beta\mathbf{A}^T\mathbf{A})\mathbf{x}^{k+1} = (\cdots)$$

- cache the Cholesky or LDL^T decomposition to $(\mathbf{C}^T\mathbf{C} + \beta\mathbf{A}^T\mathbf{A})$
- later, in each iteration, solve simple triangle systems
- changing β generally requires re-factorization
- if $(\mathbf{C}^T\mathbf{C} + \beta\mathbf{A}^T\mathbf{A})$ has a (simple+low-rank) structure, apply the Woodbury matrix inversion formula

Variants of ADMM

3. **Single gradient-descent step**. Simplify $\mathbf{x}$-update from

$$\mathbf{x}^{k+1} = \arg\min f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^k - \mathbf{b} - \mathbf{y}^k\|_2^2$$

to

$$\mathbf{x}^{k+1} = \mathbf{x}^k - c^k\left(\nabla f(\mathbf{x}^k) + \beta\mathbf{A}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^k - \mathbf{b} - \mathbf{y}^k)\right)$$

- applicable to C^1 subproblems only
- convergence requires reduced update to $\mathbf{y}$
- gradient update c^k and $\mathbf{y}$-update step sizes γ depend on spectral properties of $\mathbf{A}$

Variants of ADMM

4. **Single prox-linear step**. Simplify $\mathbf{x}$-update from

$$\mathbf{x}^{k+1} = \arg\min f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^k - \mathbf{b} - \mathbf{y}^k\|_2^2$$

to

$$\mathbf{x}^{k+1} = \arg\min f(\mathbf{x}) + \langle \mathbf{g}, \mathbf{x} \rangle + \frac{1}{2t}\|\mathbf{x} - \mathbf{x}^k\|_2^2,$$

where

$$\mathbf{g} = \nabla_{\mathbf{x}} \left(\frac{\beta}{2}\|\mathbf{A}\mathbf{x}^k + \mathbf{B}\mathbf{z}^k - \mathbf{b} - \mathbf{y}^k\|_2^2 \right)$$

- similar to the prox-linear iteration
- applicable to nonsmooth f
- convergence requires reduced $\mathbf{y}$-update
- t, β, step size γ of $\mathbf{y}$-update, and spectral properties of $\mathbf{A}$ are related
- also applicable to the other subproblem simultaneously

Variants of ADMM

5. **Approximating $\mathbf{A}^T\mathbf{A}$ by nice matrix $\mathbf{D}$.** As an example, replace

$$\mathbf{x}^{k+1} = \arg\min f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y}^k - \mathbf{b} - \mathbf{z}^k\|_2^2$$

by

$$\begin{aligned}\mathbf{x}^{k+1} = \arg\min\ & f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y}^k - \mathbf{b} - \mathbf{z}^k\|_2^2 \\ & + \frac{\beta}{2}(\mathbf{x} - \mathbf{x}^k)^T(\mathbf{D} - \mathbf{A}^T\mathbf{A})(\mathbf{x} - \mathbf{x}^k)\end{aligned}$$

- also known as "optimization transfer"
- reduces to *the prox-linear step* if $\mathbf{D} = \frac{\beta}{t}I$
- useful if

  $$\min f(\mathbf{x}) + \frac{\beta}{2}\mathbf{x}^T\mathbf{D}\mathbf{x}$$

  is computationally easier than

  $$\min f(\mathbf{x}) + \frac{\beta}{2}\mathbf{x}^T(\mathbf{A}^T\mathbf{A})\mathbf{x}.$$

- applications: $\mathbf{A}$ is an off-the-grid Fourier transform

Outline

Example: total variation

Let $\mathbf{x}$ represent a 2D image.

$$\min \mathrm{TV}(\mathbf{x}) + \frac{\mu}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

Applications

- Denoising: $\mathbf{A} = I$
- Deblurring and deconvolution: $\mathbf{A}$ is circulant matrix or convolution
- MRI CS: $\mathbf{A} = \mathbf{P}\mathbf{F}$ downsampled Fourier transform; $\mathbf{P}$ is a row selector, $\mathbf{F}$ is Fourier transform
- Circulant CS: $\mathbf{A} = \mathbf{P}\mathbf{C}$ downsampled convolution; $\mathbf{P}$ is a row selector, $\mathbf{C}$ is a circulant matrix or convolution operator

Challenge: TV is the composite of ℓ_1 and ∇x, defined as

$$\mathrm{TV}(\mathbf{x}) := \|\nabla \mathbf{x}\|_1 = \sum_{\text{pixels } (i,j)} \left\| \begin{bmatrix} x_{i+1,j} - x_{i,j} \\ x_{i,j+1} - x_{i,j} \end{bmatrix} \right\|_2 .$$

Opportunity: assuming the periodic boundary condition, $\nabla\cdot$ is a convolution operator.

Example: total variation

Decouple ℓ_1 from ∇x:

$$\min \frac{\mu}{2}\|\mathbf{Ax}-\mathbf{b}\|_2^2 + \|\mathbf{z}\|_1, \quad \text{s.t. } \nabla\mathbf{x}-\mathbf{z}=\mathbf{0}$$

where $\|\mathbf{z}\|_1 = \sum_i \|\mathbf{z}_i\|_2$.

ADMM

- $\mathbf{x}$-update is quadratic in the form of

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \mathbf{x}^T(\mu\mathbf{A}^T\mathbf{A}+\beta\nabla^T\nabla)\mathbf{x} + \text{linear terms}$$

  If $\mathbf{A}$ is identity, convolution, or partial Fourier, then

$$F(\mu\mathbf{A}^T\mathbf{A}+\beta\nabla^T\nabla)F^{-1}$$

  is a diagonal matrix. So, $\mathbf{x}$-update becomes closed-form.

- $\mathbf{z}$-subproblem is soft-thresholding

This splitting approach is often faster than the splitting

$$\min \mathrm{TV}(\mathbf{x}) + \frac{\mu}{2}\|\mathbf{Az}-\mathbf{b}\|_2^2, \quad \text{s.t. } \mathbf{x}-\mathbf{z}=0$$

because the $\mathbf{x}$-update is not in closed form.

Example: transform ℓ_1 minimization

Model

$$\min \|\mathbf{L}\mathbf{x}\|_1 + \frac{\mu}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

where examples of $\mathbf{L}$ include

- anisotropic finite difference operators
- orthogonal transforms: DCT, orthogonal wavelets
- frames: curvelets, shearlets

New models

$$\min \frac{\mu}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \|\mathbf{z}\|_1, \quad \text{s.t. } \mathbf{L}\mathbf{x} - \mathbf{z} = \mathbf{0},$$

or

$$\min \|\mathbf{L}\mathbf{x}\|_1 + \frac{\mu}{2}\|\mathbf{A}\mathbf{z} - \mathbf{b}\|_2^2, \quad \text{s.t. } \mathbf{x} - \mathbf{z} = \mathbf{0}.$$

Example: ℓ_1 fitting

Model

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$$

New model

$$\min_{\mathbf{x},\mathbf{z}} \|\mathbf{z}\|_1, \quad \text{s.t. } \mathbf{A}\mathbf{x} + \mathbf{z} = \mathbf{b}.$$

ADMM

- $\mathbf{x}$-update is quadratic
- $\mathbf{z}$-update is soft-thresholding

Example: robust (Huber-function) fitting

Model

$$\min_{\mathbf{x}} H(\mathbf{A}\mathbf{x} - \mathbf{b}) = \sum_{i=1}^{m} h(\mathbf{a}_i^T \mathbf{x} - b_i)$$

where

$$h(y) = \begin{cases} \frac{y^2}{2\mu}, & 0 \le |y| \le \mu, \\ |y| - \frac{\mu}{2}, & |y| > \mu. \end{cases}$$

Original model is differentiable, amenable to gradient descent.

Split model

$$\min_{\mathbf{x},\mathbf{z}} H(\mathbf{z}), \quad \text{s.t. } \mathbf{A}\mathbf{x} + \mathbf{z} = \mathbf{b}.$$

ADMM

- $\mathbf{x}$-update is quadratic, involving $\mathbf{A}\mathbf{A}^T$
- $\mathbf{z}$-update is component-wise separable

Outline

Block separable ADMM

Suppose $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ and f is separable, i.e.,

$$f(\mathbf{x}) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) + \cdots + f_N(\mathbf{x}_N).$$

Model

$$\begin{aligned} \min_{\mathbf{x},\mathbf{z}} \quad & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{b}. \end{aligned}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & & & \mathbf{0} \\ & \mathbf{A}_2 & & \\ & & \ddots & \\ \mathbf{0} & & & \mathbf{A}_N \end{bmatrix}$$

Block separable ADMM

The $\mathbf{x}$-update

$$\mathbf{x}^{k+1} \leftarrow \min f(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y}^k - \mathbf{b} - \mathbf{z}^k\|_2^2$$

is separable to N independent subproblems

$$\begin{aligned}
\mathbf{x}_1^{k+1} &\leftarrow \min f_1(\mathbf{x}_1) \quad + \frac{\beta}{2}\|\mathbf{A}_1\mathbf{x}_1 \quad + (\mathbf{B}\mathbf{y}^k - \mathbf{b} - \mathbf{z}^k)_1 \ \|_2^2, \\
&\vdots \\
\mathbf{x}_N^{k+1} &\leftarrow \min f_N(\mathbf{x}_N) + \frac{\beta}{2}\|\mathbf{A}_N\mathbf{x}_N + (\mathbf{B}\mathbf{y}^k - \mathbf{b} - \mathbf{z}^k)_N\|_2^2.
\end{aligned}$$

No coordination is required.

Example: consensus optimization

Model

$$\min \sum_{i=1}^{N} f_i(\mathbf{x})$$

the objective is partially separable.

Introduce N *copies* $\mathbf{x}_1, \ldots, \mathbf{x}_N$ of $\mathbf{x}$. They have the same dimensions.

New model:

$$\min_{\{\mathbf{x}_i\}, \mathbf{z}} \sum_{i=1}^{N} f_i(\mathbf{x}_i), \quad \text{s.t. } \mathbf{x}_i - \mathbf{z} = \mathbf{0}, \ \forall i.$$

A more general objective with function g is $\sum_{i=1}^{N} f_i(\mathbf{x}) + g(\mathbf{z})$.

New model:

$$\min_{\{\mathbf{x}_i\}, \mathbf{y}} \sum_{i=1}^{N} f_i(\mathbf{x}_i) + g(\mathbf{z}), \quad \text{s.t. } \begin{bmatrix} I & & \\ & \ddots & \\ & & I \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{bmatrix} - \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} \mathbf{z} = \mathbf{0}.$$

Example: consensus optimization

Lagrangian

$$L(\{\mathbf{x}_i\}, \mathbf{z}; \{\mathbf{y}_i\}) = \sum_i \left(f_i(\mathbf{x}_i) + \frac{\beta}{2}\|\mathbf{x}_i - \mathbf{z} - \mathbf{y}_i\|_2^2 \right)$$

where $\mathbf{y}_i$ is the Lagrange multipliers to $\mathbf{x}_i - \mathbf{z} = 0$.

ADMM

$$\begin{aligned}
\mathbf{x}_i^{k+1} &= \arg\min_{\mathbf{x}_i} f_i(\mathbf{x}_i) + \frac{\beta}{2}\|\mathbf{x}_i - \mathbf{z}^k - \mathbf{y}_i^k\|_2, \quad i = 1, \ldots, N, \\
\mathbf{z}^{k+1} &= \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i^{k+1} - \beta^{-1}\mathbf{y}_i^k), \\
\mathbf{y}_i^{k+1} &= \mathbf{y}_i^k - (\mathbf{x}_i^{k+1} - \mathbf{z}^{k+1}), \quad i = 1, \ldots, N.
\end{aligned}$$

The exchange problem

Model $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^n$,

$$\min \sum_{i=1}^{N} f_i(\mathbf{x}_i), \quad \text{s.t. } \sum_{i=1}^{N} \mathbf{x}_i = \mathbf{0}.$$

- it is the dual of the consensus problem
- *exchanging* n goods among N parties to *minimize* a *total* cost
- our goal: to decouple $\mathbf{x}_i$-updates

An equivalent model

$$\min \sum_{i=1}^{N} f_i(\mathbf{x}_i), \quad \text{s.t. } \mathbf{x}_i - \mathbf{x}_i' = \mathbf{0},\ \forall i, \quad \sum_{i=1}^{N} \mathbf{x}_i' = \mathbf{0}.$$

The exchange problem

ADMM after consolidating the $\mathbf{x}_i'$ update:

$$\begin{aligned}
\mathbf{x}_i^{k+1} &= \arg\min_{\mathbf{x}_i} f_i(\mathbf{x}_i) + \frac{\beta}{2}\|\mathbf{x}_i - (\mathbf{x}_i^k - \mathrm{mean}\{\mathbf{x}_i^k\} - \mathbf{u}^k)\|_2^2, \\
\mathbf{u}^{k+1} &= \mathbf{u}^k + \mathrm{mean}\{\mathbf{x}_i^{k+1}\}.
\end{aligned}$$

Applications: distributed dynamic energy management

Distributed ADMM I

$$\min_{\{\mathbf{x}_i\},\mathbf{y}} \sum_{i=1}^{N} f_i(\mathbf{x}_i) + g(\mathbf{z}), \quad \text{s.t.} \begin{bmatrix} I & & \\ & \ddots & \\ & & I \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{bmatrix} - \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} \mathbf{z} = \mathbf{0}.$$

Consider N computing nodes with MPI (message passing interface).

- $\mathbf{x}_i$ are local variables; $\mathbf{x}_i$ is stored and updated on node i only
- $\mathbf{z}$ is the global variable; computed and communicated by MPI
- $\mathbf{y}_i$ are dual variables, stored and updated on node i only

At each iteration, given $\mathbf{y}^k$ and $\mathbf{z}_i^k$

- each node i computes $\mathbf{x}_i^{k+1}$
- each node i computes $\mathbf{p}_i := (\mathbf{x}_i^{k+1} - \beta^{-1}\mathbf{y}_i^k)$
- MPI gathers $\mathbf{p}_i$ and scatters its mean, $\mathbf{z}^{k+1}$, to all nodes
- each node i computes $\mathbf{y}_i^{k+1}$

Example: distributed LASSO

Model

$$\min \|\mathbf{x}\|_1 + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

Decomposition

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_N \end{bmatrix} \mathbf{x}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_N \end{bmatrix}.$$

$\Longrightarrow$

$$\frac{\beta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \sum_{i=1}^{N} \frac{\beta}{2}\|\mathbf{A}_i\mathbf{x} - \mathbf{b}_i\|_2^2 =: \sum_{i=1}^{N} f_i(\mathbf{x}).$$

LASSO has the form

$$\min \sum_{i=1}^{N} f_i(\mathbf{x}) + g(\mathbf{x})$$

and thus can be solved by distributed ADMM.

Example: dual of LASSO

LASSO

$$\min \|\mathbf{x}\|_1 + \frac{\beta}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

Lagrange dual

$$\min_{\mathbf{y}}\{\mathbf{b}^T\mathbf{y} + \frac{\mu}{2}\|\mathbf{y}\|_2^2 : \|\mathbf{A}^T\mathbf{y}\|_\infty \leq 1\}$$

equivalently,

$$\min_{\mathbf{y},\mathbf{z}}\{-\mathbf{b}^T\mathbf{y} + \frac{\mu}{2}\|\mathbf{y}\|_2^2 + \iota_{\{\|\mathbf{z}\|_\infty \leq 1\}} : \mathbf{A}^T\mathbf{y} + \mathbf{z} = \mathbf{0}\}$$

Standard ADMM:

- primal $\mathbf{x}$ is the multipliers to $\mathbf{A}^T\mathbf{y} + \mathbf{z} = \mathbf{0}$
- $\mathbf{z}$-update is projection to ℓ_∞-ball; easy and separable
- $\mathbf{y}$-update is quadratic

Example: dual of LASSO

- Dual augmented Lagrangian (the scaled form):

$$L(\mathbf{y}, \mathbf{z}; \mathbf{x}) = \mathbf{b}^T\mathbf{y} + \frac{\mu}{2}\|\mathbf{y}\|_2^2 + \iota_{\{\|\mathbf{z}\|_\infty \leq 1\}} + \frac{\beta}{2}\|\mathbf{A}^T\mathbf{y} + \mathbf{z} - \mathbf{x}\|_2^2$$

- Dual ADMM iterations:

$$\begin{aligned}
\mathbf{z}^{k+1} &= \mathrm{Proj}_{\|\cdot\|_\infty \leq 1}\left(\mathbf{x}^k - \mathbf{A}^T\mathbf{y}^k\right), \\
\mathbf{y}^{k+1} &= \left(\mu I + \beta\mathbf{A}\mathbf{A}^T\right)^{-1}\left(\beta\mathbf{A}(\mathbf{x}^k - \mathbf{z}^{k+1}) - \mathbf{b}\right), \\
\mathbf{x}^{k+1} &= \mathbf{x}^k - \gamma(\mathbf{A}^T\mathbf{y}^{k+1} + \mathbf{z}^{k+1}).
\end{aligned}$$

  and upon termination at step K, return primal solution

$$\mathbf{x}^* = \beta\mathbf{x}^K \quad \text{(de-scaling)}.$$

- Computation bottlenecks:
  - $\left(\mu I + \beta\mathbf{A}\mathbf{A}^T\right)^{-1}$, unless $\mathbf{A}\mathbf{A}^T = I$ or $\mathbf{A}\mathbf{A}^T \approx I$
  - $\mathbf{A}(\mathbf{x}^k - \mathbf{z}^{k+1})$ and $\mathbf{A}^T\mathbf{y}^k$, unless $\mathbf{A}$ is small or has structures

Example: dual of LASSO

Observe

$$\min_{\mathbf{y},\mathbf{z}}\{\mathbf{b}^T\mathbf{y} + \frac{\mu}{2}\|\mathbf{y}\|_2^2 + \iota_{\{\|\mathbf{z}\|_\infty \le 1\}} : \mathbf{A}^T\mathbf{y} + \mathbf{z} = \mathbf{0}\}$$

- All the objective terms are perfectly separable
- The constraints cause the computation bottlenecks
- We shall try to decouple the blocks of $\mathbf{A}^T$

Distributed ADMM II

A general form with *inseparable* f and *separable* g

$$\min_{\mathbf{x},\mathbf{z}} \sum_{l=1}^{L} \left(f_l(\mathbf{x}) + g_l(\mathbf{z}_l) \right), \quad \text{s.t. } \mathbf{A}\mathbf{x} + \mathbf{z} = \mathbf{b}$$

- Make L *copies* $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L$ of $\mathbf{x}$
- Decompose

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_L \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_L \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_L \end{bmatrix}$$

- Rewrite $\mathbf{A}\mathbf{x} + \mathbf{z} = \mathbf{0}$ as

$$\mathbf{A}_l \mathbf{x}_l + \mathbf{z}_l = \mathbf{b}_l, \; \mathbf{x}_l - \mathbf{x} = \mathbf{0}, \quad l = 1, \ldots, L.$$

Distributed ADMM II

New model:

$$\min_{\mathbf{x},\{\mathbf{x}_l\},\mathbf{z}} \sum_{l=1}^{L} (f_l(\mathbf{x}_l) + g_l(\mathbf{z}_l))$$

$$\text{s.t.} \quad \mathbf{A}_l\mathbf{x}_l + \mathbf{z}_l = \mathbf{b}_l, \ \mathbf{x}_l - \mathbf{x} = \mathbf{0}, \quad l = 1, \ldots, L.$$

- $\mathbf{x}_l$'s are copies of $\mathbf{x}$
- $\mathbf{z}_l$'s are sub-blocks of $\mathbf{z}$
- Group variables $\{\mathbf{x}_l\}, \mathbf{z}, \mathbf{x}$ into two *sets*
    - $\{\mathbf{x}_l\}$: given $\mathbf{z}$ and $\mathbf{x}$, the updates of $\mathbf{x}_l$ are separable
    - $(\mathbf{z}, \mathbf{x})$: given $\{\mathbf{x}_l\}$, the updates of $\mathbf{z}_l$ and $\mathbf{x}$ are separable

  Therefore, standard (2-block) ADMM applies.
- One can also add a simple regularizer $h(\mathbf{x})$

Distributed ADMM II

Consider L computing nodes with MPI.

- $\mathbf{A}_l$ is local data store on node l only
- $\mathbf{x}_l, \mathbf{z}_l$ are local variables; $\mathbf{x}_l$ is stored and updated on node l only
- $\mathbf{x}$ is the global variable; computed and dispatched by MPI
- $\mathbf{y}_l, \bar{\mathbf{y}}_l$ are Lagrange multipliers to $\mathbf{A}_l\mathbf{x}_l + \mathbf{z}_l = \mathbf{b}_l$ and $\mathbf{x}_l - \mathbf{x} = \mathbf{0}$, respectively, stored and updated on node l only

At each iteration,

- each node l computes $\mathbf{x}_l^{k+1}$, using data $\mathbf{A}_l$
- each node l computes $\mathbf{z}_l^{k+1}$, prepares $\mathbf{p}_l = (\cdots)$
- MPI gathers $\mathbf{p}_l$ and scatters its mean, $\mathbf{x}^{k+1}$, to all nodes l
- each node l computes $\mathbf{y}_l^{k+1}, \bar{\mathbf{y}}_l^{k+1}$

Example: distributed dual LASSO

Recall

$$\min_{\mathbf{y},\mathbf{z}}\{\mathbf{b}^T\mathbf{y} + \frac{\mu}{2}\|\mathbf{y}\|_2^2 + \iota_{\{\|\mathbf{z}\|_\infty \le 1\}} : \mathbf{A}^T\mathbf{y} + \mathbf{z} = \mathbf{0}\}$$

Apply distributed ADMM II

- decompose $\mathbf{A}^T$ to row blocks, equivalently, $\mathbf{A}$ to column blocks.
- make copies of $\mathbf{y}$
- parallel computing + MPI (gathering and scatting vectors of size $\dim(\mathbf{y})$)

Recall distribute ADMM I

- decompose $\mathbf{A}$ to row blocks.
- make copies of $\mathbf{x}$
- parallel computing + MPI (gathering and scatting vectors of size $\dim(\mathbf{x})$)

Between I and II, which is better?

- If $\mathbf{A}$ is fat
  - column decomposition in approach II is more efficient
  - the global variable of approach II is smaller
- If $\mathbf{A}$ is tall,
  - row decomposition in approach I is more efficient
  - the global variable of approach I is smaller

Distributed ADMM III

A formulation with *separable* f and *separable* g

$$\min \sum_{j=1}^{N} f_j(\mathbf{x}_j) + \sum_{i=1}^{M} g_i(\mathbf{z}_i), \quad \text{s.t. } \mathbf{A}\mathbf{x} + \mathbf{z} = \mathbf{b},$$

where

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N), \quad \mathbf{z} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M).$$

Decompose $\mathbf{A}$ in *both directions* as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{A}_{1N} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \cdots & \mathbf{A}_{2N} \\ & & \cdots & \\ \mathbf{A}_{M1} & \mathbf{A}_{M2} & \cdots & \mathbf{A}_{MN} \end{bmatrix}, \text{ also } \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_M \end{bmatrix}.$$

Same model:

$$\min \sum_{j=1}^{N} f_j(\mathbf{x}_j) + \sum_{i=1}^{M} g_i(\mathbf{z}_i), \quad \text{s.t. } \sum_{j=1}^{N} \mathbf{A}_{ij}\mathbf{x}_j + \mathbf{z}_i = \mathbf{b}_i, \quad i = 1, \ldots, M.$$

Distributed ADMM III

$\mathbf{A}_{ij}\mathbf{x}_j$'s are coupled in the constraints. Standard treatment:

$$\mathbf{p}_{ij} = \mathbf{A}_{ij}\mathbf{x}_j.$$

New model:

$$\min \ \sum_{j=1}^{N} f_j(\mathbf{x}_j) + \sum_{i=1}^{M} g_i(\mathbf{z}_i), \quad \text{s.t.} \quad \begin{array}{ll} \sum_{j=1}^{N} \mathbf{p}_{ij} + \mathbf{z}_i = \mathbf{b}_i, & \forall\ i, \\ \mathbf{p}_{ij} - \mathbf{A}_{ij}\mathbf{x}_j = \mathbf{0}, & \forall\ i, j. \end{array}$$

ADMM

- alternate between $\{\mathbf{p}_{ij}\}$ and $(\{\mathbf{x}_j\}, \{\mathbf{z}_i\})$
- $\mathbf{p}_{ij}$-subproblems have closed-form solutions
- $(\{\mathbf{x}_j\}, \{\mathbf{z}_i\})$-subproblem are separable over all $\mathbf{x}_j$ and $\mathbf{z}_i$
  - $\mathbf{x}_j$-update involves f_j and $\mathbf{A}_{1j}^T\mathbf{A}_{1j}, \ldots, \mathbf{A}_{Mj}^T\mathbf{A}_{Mj}$;
  - $\mathbf{z}_i$-update involves g_i.
- ready for distributed implementation

Question: how to further decouple f_j and $\mathbf{A}_{1j}^T\mathbf{A}_{1j}, \ldots, \mathbf{A}_{Mj}^T\mathbf{A}_{Mj}$?

Distributed ADMM IV

For each $\mathbf{x}_j$, make M identical copies: $\mathbf{x}_{1j}, \mathbf{x}_{2j}, \ldots, \mathbf{x}_{Mj}$.

New model:

$$\min \sum_{j=1}^{N} f_j(\mathbf{x}_j) + \sum_{i=1}^{M} g_i(\mathbf{z}_i), \quad \text{s.t.} \quad \begin{array}{ll} \sum_{j=1}^{N} \mathbf{p}_{ij} + \mathbf{z}_i = \mathbf{b}_i, & \forall\ i, \\ \mathbf{p}_{ij} - \mathbf{A}_{ij}\mathbf{x}_{ij} = \mathbf{0}, & \forall\ i, j, \\ \mathbf{x}_j - \mathbf{x}_{ij} = \mathbf{0}, & \forall\ i, j. \end{array}$$

ADMM

- alternate between $(\{\mathbf{x}_j\}, \{\mathbf{p}_{ij}\})$ and $(\{\mathbf{x}_{ij}\}, \{\mathbf{z}_i\})$
- $(\{\mathbf{x}_j\}, \{\mathbf{p}_{ij}\})$-subproblem are separable
  - $\mathbf{x}_j$-update involves f_j only; computes $\mathbf{prox}_{f_j}$
  - $\mathbf{p}_{ij}$-update is in closed form
- $(\{\mathbf{x}_{ij}\}, \{\mathbf{z}_i\})$-subproblem are separable
  - $\mathbf{x}_{ij}$-update involves $(\alpha I + \beta \mathbf{A}_{ij}^T \mathbf{A}_{ij})$;
  - $\mathbf{y}_i$-update involves g_i only; computes $\mathbf{prox}_{g_i}$.
- ready for distributed implementation

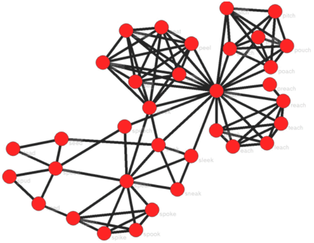
Outline

Decentralized ADMM

After making local copies $\mathbf{x}_i$ for $\mathbf{x}$, instead of imposing the consistency constraints like

$$\mathbf{x}_i - \mathbf{x} = 0, \quad i = 1, \ldots, M,$$

consider graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{\text{nodes}\}$ and $\mathcal{E} = \{\text{edges}\}$



and impose one type of the following consistency constraints

$$\begin{aligned} \mathbf{x}_i - \mathbf{x}_j = \mathbf{0}, &\quad \forall (i,j) \in \mathcal{E}, \text{ or} \\ \mathbf{x}_i - \mathbf{z}_{ij} = \mathbf{0},\ \mathbf{x}_j - \mathbf{z}_{ij} = \mathbf{0}, &\quad \forall (i,j) \in \mathcal{E}, \text{ or} \\ \text{mean}\{\mathbf{x}_j : (i,j) \in \mathcal{E}\} - \mathbf{x}_i = \mathbf{0}, &\quad \forall i \in \mathcal{V}. \end{aligned}$$

Decentralized ADMM

- Decentralized ADMM run on a *connected* network
- There is no data fusion / control center
- Applications:
    - wireless sensor networks
    - collaborative learning
- ADMM will alternative perform the followings
    - Local computation at each node
    - Communication between neighbors or broadcasting in neighborhood
- Since data is not shared or centrally store, data security is preserved
- Convergence rate depends on
    - the properties (e.g., convexity, condition number) of the objective function
    - the size, connectivity, and spectral properties of the graph

Outline

Example: latent variable graphical model selection

V. Chandrasekaran, P. Parrilo, A. Willsky

Model of regularized maximum normal likelihood

$$\min_{R,S,L} \langle R, \hat{\Sigma}_X \rangle - \log\det(R) + \alpha\|S\|_1 + \beta\mathrm{Tr}(L), \quad \text{s.t.} \quad R = S - L, R \succ 0, L \succeq 0,$$

where X are the observed variables, $\Sigma_X^{-1} \approx R = S - L$, S is spare, L is low rank. First two terms are from the log-likelihood function

$$\ell(K; \Sigma) = \log\det(K) - \mathrm{tr}(K\Sigma).$$

Introduce indicator function

$$\mathcal{I}(L \succeq 0) := \begin{cases} 0, & \text{if } L \succeq 0 \\ +\infty, & \text{otherwise.} \end{cases}$$

Obtain the 3-block formulation

$$\min_{R,S,L} \langle R, \hat{\Sigma}_X \rangle - \log\det(R) + \alpha\|S\|_1 + \beta\mathrm{Tr}(L) + \mathcal{I}(L \succeq 0), \quad \text{s.t.} \quad R - S + L = 0.$$

Example: stable principle component pursuit

Model

$$\begin{array}{ll} \min_{L,S,Z} & \|L\|_* + \rho\|S\|_1 \\ \text{s.t.} & L + S + Z = M \\ & \|Z\|_F \le \sigma, \end{array}$$

$$M = \text{low-rank} + \text{sparse} + \text{noise}.$$

For quantities such as images and videos, add $L \ge 0$ component wise.

New model:

$$\begin{array}{ll} \min_{L,S,Z,K} & \|L\|_* + \rho\|S\|_1 + \mathcal{I}(\|Z\|_F \le \sigma) + \mathcal{I}(K \ge 0) \\ \text{s.t.} & L + S + Z = M \\ & L - K = 0. \end{array}$$

Block-form constraints:

$$\begin{pmatrix} I & I \\ I & 0 \end{pmatrix} \begin{pmatrix} L \\ S \end{pmatrix} + \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix} \begin{pmatrix} Z \\ K \end{pmatrix} = \begin{pmatrix} M \\ 0 \end{pmatrix}.$$

Example: mixed TV and ℓ_1 regularization

Model

$$\min_x \mathrm{TV}(x) + \alpha\|Wx\|_1, \quad \text{s.t. } \|Rx - b\|_2 \le \sigma.$$

New model:

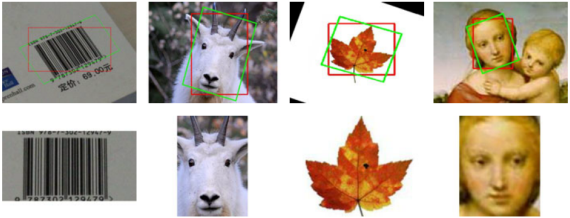
$$\begin{array}{ll} \min_x & \sum_i \|z_i\|_2 + \alpha\|Wx\|_1 + \mathcal{I}(\|y\|_2 \le \sigma) \\ \text{s.t.} & z_i = D_i x, \forall i = 1, \ldots, N \\ & y = Rx - b. \end{array}$$

If use two sets of variables, x vs $(y, \{z_i\})$

$$\begin{pmatrix} R \\ D_1 \\ \vdots \\ D_N \end{pmatrix} x - \begin{pmatrix} y \\ z_1 \\ \vdots \\ z_N \end{pmatrix} = \begin{pmatrix} b \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

x-subproblem is *not* easy to solve.

Example: alignment for linearly correlated images



Model:

$$\min_{I^0,E,\tau} \|I^0\|_* + \lambda\|E\|_1 \quad \text{subject to} \quad I \circ \tau = I^0 + E$$

Linearize the non-convex term $I \circ \tau$: $I \circ (\tau + \delta\tau) \approx I \circ \tau + \nabla I \cdot \Delta\tau$.

New model

$$\min_{I^0,E,\Delta\tau} \|I^0\|_* + \lambda\|E\|_1 \quad \text{subject to} \quad I \circ \tau + \nabla I \Delta\tau = I^0 + E$$

Two solutions to decouple variables

To solve a subproblem with coupling variables

1. apply the prox-linear inexact update, or
2. introduce bridge variables, as done in distributed ADMM.

For example, consider

$$\min_{\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}} (f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2)) + g(\mathbf{y}), \quad \text{s.t. } (\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2) + \mathbf{B}\mathbf{y} = \mathbf{b}.$$

In the ADMM $(\mathbf{x}_1, \mathbf{x}_2)$-subproblem, $\mathbf{x}_1$ and $\mathbf{x}_2$ are coupled.

However, the prox-linear update is separable

$$\begin{bmatrix} \mathbf{x}_1^{k+1} \\ \mathbf{x}_2^{k+1} \end{bmatrix} = \operatorname*{arg\,min}_{\mathbf{x}_1,\mathbf{x}_2} \;\; (f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2)) + \langle \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \rangle + \frac{1}{2t} \left\| \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{x}_1^k \\ \mathbf{x}_2^k \end{bmatrix} \right\|_2^2 .$$

Example: patient motion detection during radiation therapy

Goal: to separate different motions (machine's vs patient's)

(wmv)

- My work with Wei Deng (Rice) and group of Steve Jiang (UCSD)
- Model extending robust PCA:

$$\min_{X,P,Z} \mu_1\|X\|_* + \mu_2\|\theta\|_1 + \|Z\|_1, \quad \text{s.t. } X + D\theta + Z = \text{input video.}$$

X: static; $D\theta$: background and reg. motion, Z irreg. motion

Example: patient motion detection during radiation therapy

(avi)

Outline

Uncovered ADMM topics

- ADMM for LP, QP
- ADMM for conic programming, especially, SDP
- Multi-block ADMM schemes
- ADMM applied to non-convex problems (its convergence is open)