Algorithms: Majorization-Minimization (MM)

Prof. Daniel P. Palomar

ELEC5470/IEDA6100A - Convex Optimization
The Hong Kong University of Science and Technology (HKUST)
Fall 2020-21

Outline

1. **Majorization-Minimization Algorithm**
   - MM in a Nutshell
   - Applications
   - Surrogate Functions*
   - Algorithms derived from MM*
   - More Applications*
   - Connection to SCA
2. **Block Majorization-Minimization Algorithm**
   - Block MM
   - Algorithms derived from Block MM*
   - Applications*

Outline

1. **Majorization-Minimization Algorithm**
   - MM in a Nutshell
   - Applications
   - Surrogate Functions*
   - Algorithms derived from MM*
   - More Applications*
   - Connection to SCA
2. **Block Majorization-Minimization Algorithm**
   - Block MM
   - Algorithms derived from Block MM*
   - Applications*

Outline

Majorization-Minimization

- Consider the following presumably **difficult optimization problem**:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in \mathcal{X}, \end{array}$$

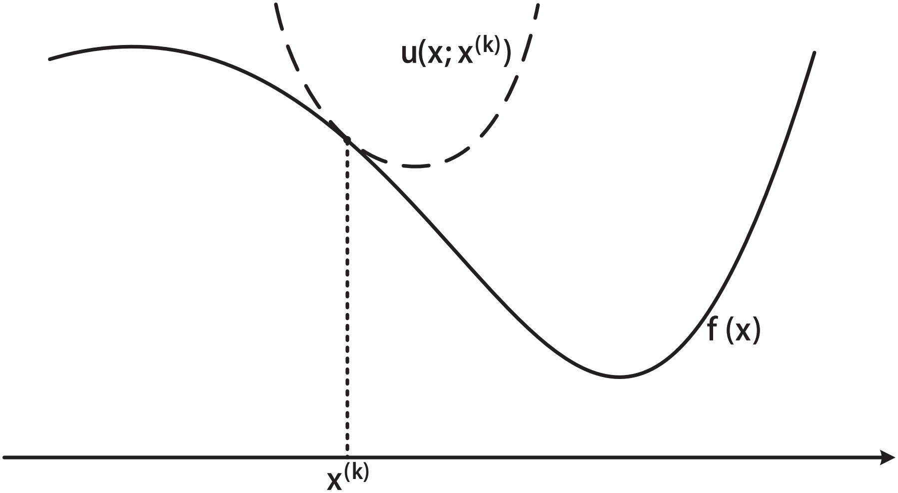
  with $\mathcal{X}$ being the feasible set and $f(\mathbf{x})$ being continuous.
- Idea: **successively minimize a more managable surrogate function** $u(\mathbf{x}, \mathbf{x}^k)$:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x} \in \mathcal{X}} u(\mathbf{x}, \mathbf{x}^k),$$

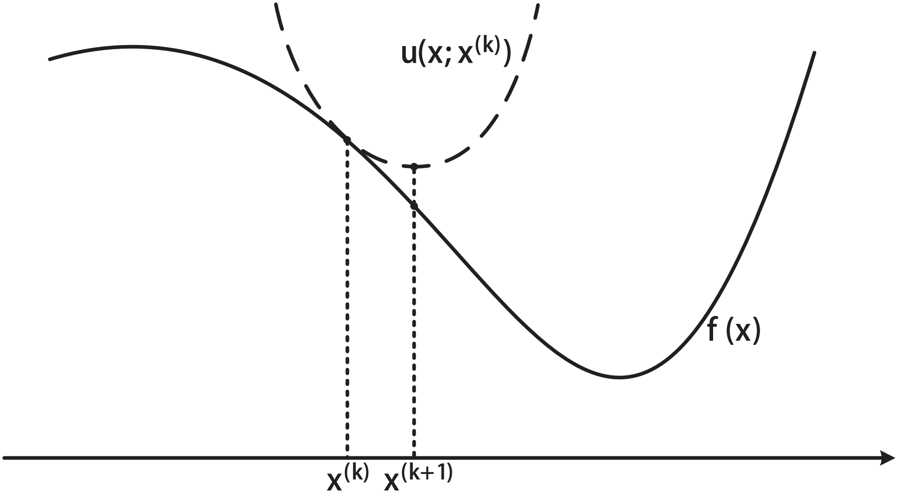
  hoping the sequence of minimizers $\{\mathbf{x}^k\}$ will converge to optimal $\mathbf{x}^\star$.
- Question: how to construct $u(\mathbf{x}, \mathbf{x}^k)$?
- Answer: that's more like an art (Sun et al. 2017)[1].

[1] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Processing*, vol. 65, no. 3, pp. 794–816, 2017.

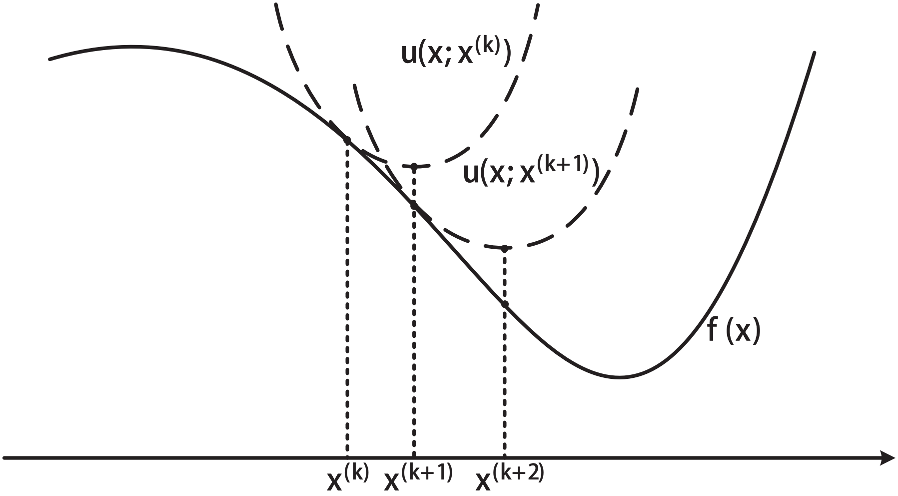
Iterative algorithm



Iterative algorithm



Iterative algorithm



Surrogate/majorizer

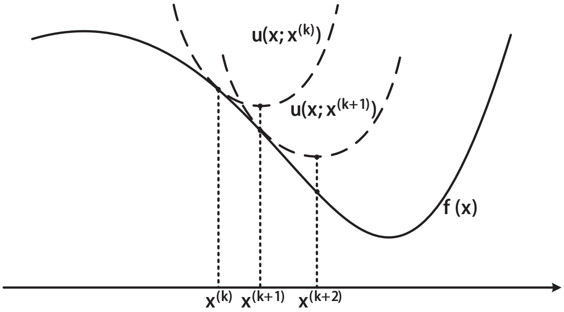
- Construction rule of the majorizing function:

$$u(\mathbf{y}, \mathbf{y}) = f(\mathbf{y}), \ \forall \mathbf{y} \in \mathcal{X} \tag{A1}$$

$$u(\mathbf{x}, \mathbf{y}) \geq f(\mathbf{x}), \ \forall \mathbf{x}, \mathbf{y} \in \mathcal{X} \tag{A2}$$

$$u'(\mathbf{x}, \mathbf{y}; \mathbf{d})\big|_{\mathbf{x}=\mathbf{y}} = f'(\mathbf{y}; \mathbf{d}), \ \forall \mathbf{d} \text{ with } \mathbf{y} + \mathbf{d} \in \mathcal{X} \tag{A3}$$

$$u(\mathbf{x}, \mathbf{y}) \text{ is continuous in } \mathbf{x} \text{ and } \mathbf{y} \tag{A4}$$



Algorithm

Algorithm MM

Set $k = 0$ and initialize with a feasible point $\mathbf{x}^0 \in \mathcal{X}$.

repeat

- $\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} u(\mathbf{x}, \mathbf{x}^k)$
- $k \leftarrow k + 1$

until convergence

return $\mathbf{x}^k$

- Property of MM: $\{f(\mathbf{x}^k)\}$ is nonincreasing, i.e., $f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k)$.
- That means that $\{f(\mathbf{x}^k)\} \to p^\star$, but what about the convergence of the iterates $\{\mathbf{x}^k\}$?

Technical preliminaries

- **Distance from a point to a set**:
$$d(\mathbf{x}, \mathcal{S}) = \inf_{\mathbf{s}\in\mathcal{S}} \|\mathbf{x} - \mathbf{s}\|.$$
- **Limit point**: $\bar{\mathbf{x}}$ is a limit point of $\{\mathbf{x}^k\}$ if there exists a subsequene of $\{\mathbf{x}^k\}$ that converges to $\bar{\mathbf{x}}$. Note that every bounded sequence in $\mathbb{R}^n$ has a limit point (or convergent subsequence).
- **Directional derivative**: Let $f: \mathcal{X} \to \mathbb{R}$ be a function, where $\mathcal{X} \subseteq \mathbb{R}^m$ is a convex set. The directional derivative of f at point $\mathbf{x}$ in the direction $\mathbf{d}$ is defined by
$$f'(\mathbf{x}; \mathbf{d}) \triangleq \liminf_{\lambda\downarrow 0} \frac{f(\mathbf{x} + \lambda\mathbf{d}) - f(\mathbf{x})}{\lambda}.$$
- **Stationary point**: $\mathbf{x} \in \mathcal{X}$ is a stationary point of f if
$$f'(\mathbf{x}; \mathbf{d}) \geq 0, \ \forall \mathbf{d} \text{ such that } \mathbf{x} + \mathbf{d} \in \mathcal{X}.$$
  - A stationary point may be a local min, a local max., or a saddle point.
  - If $\mathcal{X} = \mathbb{R}^n$ and f is differentiable, then stationarity means $\nabla f(\mathbf{x})$.

Convergence

The following gives the convergence of the MM algorithm to a stationary point (Razaviyayn et al. 2013)[2].

Theorem

Suppose $\mathcal{X}$ is convex. Under assumptions A1-A4, every limit point of the sequence $\{\mathbf{x}^k\}$ is a stationary point of the original problem.

If we further assume that the level set $\mathcal{X}^0 = \{\mathbf{x} | f(\mathbf{x}) \leq f(\mathbf{x}^0)\}$ is compact, then

$$\lim_{k\to\infty} d\left(\mathbf{x}^k, \mathcal{X}^\star\right) = 0,$$

where $\mathcal{X}^\star$ is the set of stationary points.

- The case of nonconvex $\mathcal{X}$ has to be considered on a case by case basis (and it is usually manageable).

[2] M. Razaviyayn, M. Hong, and Z. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1126–1153, 2013.

References

- Short tutorial on MM:
  D. R. Hunter and K. Lange (2004). "A tutorial on MM algorithms." *Amer. Statistician*, 58, 30–37.

- Exhaustive tutorial on MM with many applications and tricks:
  Y. Sun, P. Babu, and D. P. Palomar (2017). "Majorization-minimization algorithms in signal processing, communications, and machine learning." *IEEE Trans. Signal Processing*, 65(3), 794–816.

- Convergence of MM:
  M. Razaviyayn, M. Hong, and T. Luo. (2013). "A unified convergence analysis of block successive minimization methods for nonsmooth optimization." *SIAM J. Optim.*, 23(2), 1126–1153.

Outline

Nonnegative Least Squares

- Consider the following nonnegative LS problem:

$$\underset{\mathbf{x}\geq\mathbf{0}}{\text{minimize}} \quad \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2$$

where $\mathbf{b} \in \mathbb{R}_+^m$, $\mathbf{b} \neq \mathbf{0}$, and $\mathbf{A} \in \mathbb{R}_{++}^{m\times n}$.

- Observe that this problem cannot be solved with the conventional LS solution $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$ due to the nonnegativity constraints.
- The problem is a convex quadratic problem, so one could use some QP solver; however, we will develop a simple iterative algorithm based on MM.
- The critical step in the application of MM is to find a convenient majorizer of the function $\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2$.

Nonnegative Least Squares

- Consider the following quadratic majorizer of $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$:

$$u(\mathbf{x}, \mathbf{x}^k) = f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^k)^T \mathbf{\Phi}(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k)$$

where $\mathbf{\Phi}(\mathbf{x}^k) = \mathrm{Diag}\left(\frac{[\mathbf{A}^T\mathbf{A}\mathbf{x}^k]_1}{x_1^k}, \ldots, \frac{[\mathbf{A}^T\mathbf{A}\mathbf{x}^k]_n}{x_n^k}\right)$.

- Note that $u(\mathbf{x}, \mathbf{x}^k)$ is a valid majorizer because it's continuous, $u(\mathbf{x}^k, \mathbf{x}^k) = f(\mathbf{x}^k)$, $\nabla u(\mathbf{x}^k, \mathbf{x}^k) = \nabla f(\mathbf{x}^k)$, and it is an upper-bound $u(\mathbf{x}, \mathbf{x}^k) \geq f(\mathbf{x})$ since it has a higher curvature:

$$\mathbf{\Phi}(\mathbf{x}^k) \succeq \mathbf{A}^T\mathbf{A}.$$

- Now that we have the majorizer, we can formulate the problem to be solved at each iteration $k = 0, 1, \ldots$ as

$$\underset{\mathbf{x} \geq \mathbf{0}}{\text{minimize}} \quad u(\mathbf{x}, \mathbf{x}^k)$$

Nonnegative Least Squares

- Since this problem is convex, we can set the gradient to zero (ignoring for a moment the constraint):

$$\nabla f(\mathbf{x}^k) + \mathbf{\Phi}(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k) = \mathbf{0}$$

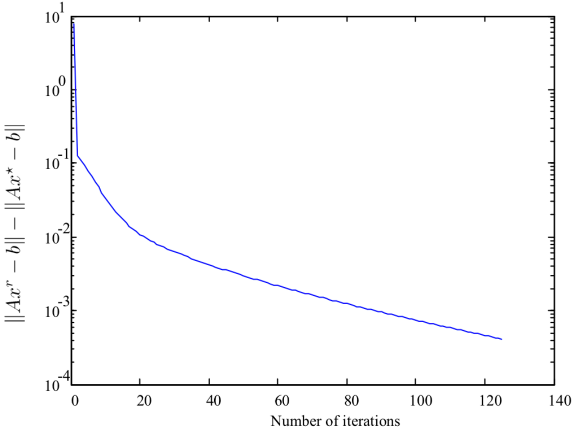
  which leads to $\mathbf{x} = \mathbf{x}^k - \mathbf{\Phi}(\mathbf{x}^k)^{-1}\nabla f(\mathbf{x}^k)$.
- Now using $\nabla f(\mathbf{x}^k) = \mathbf{A}^T\mathbf{A}\mathbf{x}^k - \mathbf{A}^T\mathbf{b}$, we can finally write the MM iterate as

$$\begin{aligned}\mathbf{x}^{k+1} &= \mathbf{x}^k - \text{Diag}\left(\frac{x_1^k}{[\mathbf{A}^T\mathbf{A}\mathbf{x}^k]_1}, \ldots, \frac{x_n^k}{[\mathbf{A}^T\mathbf{A}\mathbf{x}^k]_n}\right)(\mathbf{A}^T\mathbf{A}\mathbf{x}^k - \mathbf{A}^T\mathbf{b}) \\ &= \text{Diag}\left(\frac{x_1^k}{[\mathbf{A}^T\mathbf{A}\mathbf{x}^k]_1}, \ldots, \frac{x_n^k}{[\mathbf{A}^T\mathbf{A}\mathbf{x}^k]_n}\right)\mathbf{A}^T\mathbf{b} \\ &= \mathbf{c}^k \odot \mathbf{x}^k\end{aligned}$$

  where $c_i^k = \frac{[\mathbf{A}^T\mathbf{b}]_i}{[\mathbf{A}^T\mathbf{A}\mathbf{x}^k]_i}$.

- Example of the convergence of the MM iterative algorithm

$$\mathbf{x}^{k+1} = \mathbf{c}^{k} \odot \mathbf{x}^{k} \qquad k = 0, 1, \ldots$$



Sparse regression: Reweighted ℓ_1-norm minimization

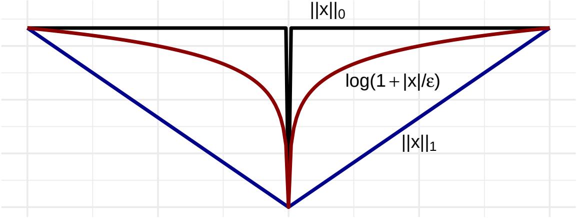
- Consider the following NP-hard sparse signal recovery problem:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & \|\mathbf{x}\|_0 \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{array}$$

- One common way to deal with it is with the ℓ_1-norm approximation:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & \|\mathbf{x}\|_1 \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{array}$$

- For a better fit to the indicator function in $\|\mathbf{x}\|_0$, consider a concave and nondecreasing penalty function $\phi(t)$. For example, $\phi(t) = \log(1 + t/\varepsilon)$:



Sparse regression: Reweighted ℓ_1-norm minimization

- However, the resulting problem with such $\phi(t)$ is nonconvex:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & \sum_{i=1}^{n} \phi(|x_i|) \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{array}$$

- We can then use MM by finding a majorizer of $\phi(t)$.
- The function $\phi(t) = \log(1 + t/\varepsilon)$, for $t \geq 0$, is concave and is majorized at $t = t_0$ by its linearization:

$$\phi(t) \leq \phi(t_0) + \phi(t_0)'(t - t_0) = \phi(t_0) + \frac{1}{\varepsilon + t_0}(t - t_0)$$

- Thus, the function $\phi(|x_i|)$ is majorized at x_i^k (up to an irrelevant constant) by $w_i^k |x_i|$ with $w_i^k = \phi'(t)|_{t=|x_i^k|} = \frac{1}{\varepsilon + |x_i^k|}$.

Sparse regression: Reweighted ℓ_1-norm minimization

- Summarizing, at each iteration $k = 1, 2, \ldots$, the problem is:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & \sum w_i^k \left| x_i \right| \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b} \end{array}$$

  where $w_i^k = \frac{1}{\varepsilon + \left| x_i^k \right|}$.

- More details in (Candes et al. 2008)[3].

[3] E. J. Candes, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5-6, pp. 877–905, 2008.

Reweighted LS for ℓ_1-norm minimization

- Consider the following convex problem:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$$

- If instead we had the ℓ_2-norm, then it would be an LS with solution $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$.
- The problem is convex and can be rewritten as a linear program (LP), so one could use some LP solver; however, we will develop a simple iterative algorithm based on MM.
- The critical step in the application of MM is to find a convenient majorizer of the function $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$, where $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$.

Reweighted LS for ℓ_1-norm minimization

- Consider the following quadratic majorizer of $f(t) = |t|$ for $t \neq 0$ (for simplicity we ignore this case):
$$u(t, t^k) = \frac{1}{2|t^k|}(t^2 + (t^k)^2).$$
- It is a valid majorizer since it is continuous, $u(t, t^k) \geq f(t)$, $u(t^k, t^k) = f(t)$, and $\frac{d}{dt}u(t^k, t^k) = \frac{d}{dt}f(t^k)$.
- Now we can apply it to the ℓ_1-norm: a quadratic majorizer of $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_1$ is
$$u(\mathbf{x}, \mathbf{x}^k) = \sum_{i=1}^{n} \frac{1}{2|[\mathbf{A}\mathbf{x}^k - \mathbf{b}]_i|}([\mathbf{A}\mathbf{x} - \mathbf{b}]_i^2 + ([\mathbf{A}\mathbf{x}^k - \mathbf{b}]_i)^2).$$
- Now that we have the majorizer, we can write the MM iterative algorithm for $k = 0, 1, \ldots$ as
$$\underset{\mathbf{x}}{\text{minimize}} \quad \|(\mathbf{A}\mathbf{x} - \mathbf{b}) \odot \mathbf{w}^k\|_2^2$$
where $w_i^k = \sqrt{\frac{1}{2|[\mathbf{A}\mathbf{x}^k - \mathbf{b}]_i|}}$.

LASSO ($\ell_2 - \ell_1$ optimization) via BCD

- Consider the problem

$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) \triangleq \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$$

- We can use BCD on each element of $\mathbf{x} = (x_1, \ldots, x_N)$.
- The optimization w.r.t. each block x_i at iteration $k = 0, 1, \ldots$ is

$$\underset{x_i}{\text{minimize}} \quad f_i(x_i) \triangleq \frac{1}{2}\|\tilde{\mathbf{y}}_i^k - \mathbf{a}_i x_i\|_2^2 + \lambda|x_i|$$

where $\tilde{\mathbf{y}}_i^k \triangleq \mathbf{y} - \sum_{j<i} \mathbf{a}_j x_j^{k+1} - \sum_{j>i} \mathbf{a}_j x_j^k$.

- This leads to the iterates for $k = 0, 1, \ldots$

$$x_i^{k+1} = \text{soft}_\lambda \left(\mathbf{a}_i^T \tilde{\mathbf{y}}_i^k\right) / \|\mathbf{a}_i\|^2, \quad i = 1, \ldots, N$$

where $\text{soft}_\lambda(u) \triangleq \text{sign}(u)\left[|u| - \lambda\right]_+$ is the **soft-thresholding** operator ($[\cdot]_+ \triangleq \max\{\cdot, 0\}$).

LASSO ($\ell_2 - \ell_1$ optimization) via MM

- The critical step in the application of MM is to find a convenient majorizer of the function $f(\mathbf{x}) \triangleq \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$.
- Consider the following majorizer of $f(\mathbf{x})$:

$$u(\mathbf{x}, \mathbf{x}^k) = f(\mathbf{x}) + \text{dist}(\mathbf{x}, \mathbf{x}^k)$$

where $\text{dist}(\mathbf{x}, \mathbf{x}^k) = \frac{c}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2 - \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{x}^k\|_2^2$ and $c > \lambda_{\max}(\mathbf{A}^T\mathbf{A})$.

- Note that $u(\mathbf{x}, \mathbf{x}^k)$ is a valid majorizer because it's continuous, it is an upper-bound $u(\mathbf{x}, \mathbf{x}^k) \geq f(\mathbf{x})$ with $u(\mathbf{x}^k, \mathbf{x}^k) = f(\mathbf{x}^k)$, and $\nabla u(\mathbf{x}^k, \mathbf{x}^k) = \nabla f(\mathbf{x}^k)$.
- The majorizer can be rewritten in a more convenient way as

$$u(\mathbf{x}, \mathbf{x}^k) = \frac{c}{2}\|\mathbf{x} - \bar{\mathbf{x}}^k\|_2^2 + \lambda\|\mathbf{x}\|_1 + \text{const.}$$

where $\bar{\mathbf{x}}^k = \frac{1}{c}\mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}^k) + \mathbf{x}^k$.

LASSO ($\ell_2 - \ell_1$ optimization) via MM

- Now that we have the majorizer, we can formulate the problem to be solved at each iteration $k = 0, 1, \ldots$

$$\underset{\mathbf{x} \geq \mathbf{0}}{\text{minimize}} \quad \frac{c}{2}\|\mathbf{x} - \bar{\mathbf{x}}^k\|_2^2 + \lambda\|\mathbf{x}\|_1$$

- This problem looks like the original one but without the matrix $\mathbf{A}$ mixing all the components.
- As a consequence, this problem decouples into an optimization for each element, which solution we already known to be given by the soft-thresholding operator, leading to the iterates for $k = 0, 1, \ldots$

$$\mathbf{x}^{k+1} = \text{soft}_\lambda\left(\bar{\mathbf{x}}^k\right),$$

  where the soft-thresholding operator is applied elementwise.
- So what's the difference between the algorithms obtained via BCD and MM?
  - BCD algorithm updates each element on a successive or cyclical way;
  - MM algorithm updates all elements simultaneously.

Outline

Construction of majorizers or surrogate functions

- The performance of MM algorithm depends crucially on the majorizer or surrogate function $u\left(\mathbf{x}, \mathbf{x}^k\right)$.
- Guideline:
  - on the one hand, $u\left(\mathbf{x}, \mathbf{x}^k\right)$ should be as close as possible to the original function $f(\mathbf{x})$;
  - on the other hand, $u\left(\mathbf{x}, \mathbf{x}^k\right)$ should be easy to minimize.
- Many tricks to obtain majorizers in (Sun et al. 2017)[4], (Beck and Pan 2018)[5].

[4] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Processing*, vol. 65, no. 3, pp. 794–816, 2017.

[5] A. Beck and D. Pan, "Convergence of an inexact majorization-minimization method for solving a class of composite optimization problems," in *Large-Scale and Distributed Optimization. Lecture Notes in Mathematics*, R. A. Giselsson P., Ed., vol. 2227, Springer, Cham, 2018.

Construction by convexity

- Suppose $\kappa(t)$ is convex, then

$$\kappa\left(\sum_i \alpha_i t_i\right) \leq \sum_i \alpha_i \kappa(t_i)$$

with $\alpha_i \geq 0$ and $\sum \alpha_i = 1$.

Construction by convexity

- For example:

$$\begin{aligned}\kappa\left(\mathbf{w}^T\mathbf{x}\right) &= \kappa\left(\mathbf{w}^T\left(\mathbf{x}-\mathbf{x}^k\right)+\mathbf{w}^T\mathbf{x}^k\right)\\ &= \kappa\left(\sum_i \alpha_i\left(\frac{w_i\left(x_i-x_i^k\right)}{\alpha_i}+\mathbf{w}^T\mathbf{x}^k\right)\right)\\ &\le \sum_i \alpha_i\kappa\left(\frac{w_i\left(x_i-x_i^k\right)}{\alpha_i}+\mathbf{w}^T\mathbf{x}^k\right)\end{aligned}$$

- If further assume that $\mathbf{w}$ and $\mathbf{x}$ are positive ($\alpha_i = w_i x_i^k/\mathbf{w}^T\mathbf{x}^k$):

$$\kappa\left(\mathbf{w}^T\mathbf{x}\right) \le \sum_i \frac{w_i x_i^k}{\mathbf{w}^T\mathbf{x}^k}\kappa\left(\frac{\mathbf{w}^T\mathbf{x}^k}{x_i^k}x_i\right)$$

- The surrogate functions are separable (parallel algorithm).

Construction by Taylor expansion

- Suppose $\kappa(\mathbf{x})$ is concave and differentiable, then

$$\kappa(\mathbf{x}) \leq \kappa\left(\mathbf{x}^k\right) + \nabla\kappa\left(\mathbf{x}^k\right)\left(\mathbf{x} - \mathbf{x}^k\right),$$

  which is a linear upper-bound.

- Suppose $\kappa(\mathbf{x})$ is convex and twice differentiable, then

$$\kappa(\mathbf{x}) \leq \kappa\left(\mathbf{x}^k\right) + \nabla\kappa\left(\mathbf{x}^k\right)^T\left(\mathbf{x} - \mathbf{x}^k\right) + \frac{1}{2}\left(\mathbf{x} - \mathbf{x}^k\right)^T\mathbf{M}\left(\mathbf{x} - \mathbf{x}^k\right)$$

  if $\mathbf{M} - \nabla^2\kappa(\mathbf{x}) \succeq \mathbf{0}, \forall\mathbf{x}$.

Construction by inequalities

- Arithmetic-Geometric Mean Inequality:

$$\left(\prod_{i=1}^{n} x_i\right)^{1/n} \leq \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Cauchy-Schwartz Inequality:

$$\|\mathbf{x}\| \geq \frac{\mathbf{x}^T\mathbf{x}^k}{\|\mathbf{x}^k\|}$$

- Jensen's Inequality:

$$\kappa\left(\mathsf{E}\mathbf{x}\right) \leq \mathsf{E}\kappa\left(\mathbf{x}\right)$$

with $\kappa\left(\cdot\right)$ being convex.

Outline

1. **Majorization-Minimization Algorithm**
   - MM in a Nutshell
   - Applications
   - Surrogate Functions*
   - Algorithms derived from MM*
   - More Applications*
   - Connection to SCA
2. **Block Majorization-Minimization Algorithm**
   - Block MM
   - Algorithms derived from Block MM*
   - Applications*

EM algorithm

- Assume the complete data set $\{\mathbf{x}, \mathbf{z}\}$ consists of observed variable $\mathbf{x}$ and latent variable $\mathbf{z}$.
- Objective: estimate parameter $\theta \in \Theta$ from $\mathbf{x}$.
- Maximum likelihood estimator: $\hat{\theta} = \arg\min_{\theta \in \Theta} -\log p(\mathbf{x}|\theta)$
- EM (Expectation Maximization) algorithm:
  - E-step: evaluate $p\left(\mathbf{z}|\mathbf{x}, \theta^k\right)$
    👉 "guess" $\mathbf{z}$ from current estimate of θ
  - M-step: update θ as $\theta^{k+1} = \arg\min_{\theta \in \Theta} u\left(\theta, \theta^k\right)$, where

$$u\left(\theta, \theta^k\right) = -\mathrm{E}_{\mathbf{z}|\mathbf{x},\theta^k} \log p\left(\mathbf{x}, \mathbf{z}|\theta\right)$$

    👉 update θ from "guessed" complete dataset.

An MM interpretation of EM

- The objective function can be written as

$$
\begin{aligned}
-\log p(\mathbf{x}|\theta) &= -\log \mathrm{E}_{\mathbf{z}|\theta} p(\mathbf{x}|\mathbf{z},\theta) \\
&= -\log \mathrm{E}_{\mathbf{z}|\theta} \left(\frac{p\left(\mathbf{z}|\mathbf{x},\theta^k\right) p(\mathbf{x}|\mathbf{z},\theta)}{p(\mathbf{z}|\mathbf{x},\theta^k)} \right) \\
&= -\log \mathrm{E}_{\mathbf{z}|\mathbf{x},\theta^k} \left(\frac{p(\mathbf{x}|\mathbf{z},\theta)}{p(\mathbf{z}|\mathbf{x},\theta^k)} p(\mathbf{z}|\theta) \right) \\
&\leq -\mathrm{E}_{\mathbf{z}|\mathbf{x},\theta^k} \log \left(\frac{p(\mathbf{x}|\mathbf{z},\theta)}{p(\mathbf{z}|\mathbf{x},\theta^k)} p(\mathbf{z}|\theta) \right) \\
&= \underbrace{-\mathrm{E}_{\mathbf{z}|\mathbf{x},\theta^k} \log p(\mathbf{x},\mathbf{z}|\theta)}_{u\left(\theta,\theta^k\right)} + \mathrm{E}_{\mathbf{z}|\mathbf{x},\theta^k} p\left(\mathbf{z}|\mathbf{x},\theta^k\right)
\end{aligned}
$$

where the inequality follows from Jensen's inequality.

Proximal minimization

- Suppose $f(\mathbf{x})$ is convex. Solve $\min_{\mathbf{x}} f(\mathbf{x})$ by instead solving the equivalent problem

$$\underset{\mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{X}}{\text{minimize}} \quad f(\mathbf{x}) + \tfrac{1}{2c}\left\|\mathbf{x}-\mathbf{y}\right\|^2.$$

- Objective function is strongly convex in both $\mathbf{x}$ and $\mathbf{y}$.
- Algorithm:

$$\begin{aligned}\mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}\in\mathcal{X}}\left\{f(\mathbf{x}) + \frac{1}{2c}\left\|\mathbf{x}-\mathbf{y}^k\right\|^2\right\}\\ \mathbf{y}^{k+1} &= \mathbf{x}^{k+1}.\end{aligned}$$

- An MM interpretation:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathcal{X}}\left\{f(\mathbf{x}) + \frac{1}{2c}\left\|\mathbf{x}-\mathbf{x}^k\right\|^2\right\}.$$

DC programming

- Consider the unconstrained problem

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) \ ,$$

where $f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x})$ with $g(\mathbf{x})$ convex and $h(\mathbf{x})$ concave.

- DC (Difference of Convex) programming generates $\left\{\mathbf{x}^k\right\}$ by solving

$$\nabla g\left(\mathbf{x}^{k+1}\right) = -\nabla h\left(\mathbf{x}^k\right).$$

- An MM interpretation:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \left\{ g(\mathbf{x}) + \nabla h\left(\mathbf{x}^k\right)^T \left(\mathbf{x} - \mathbf{x}^k\right) \right\}.$$

Outline

Sparse generalized eigenvalue problem

- The generalized eigenvalue problem (GEVP) can be formulated as

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{maximize}} & \mathbf{x}^T\mathbf{A}\mathbf{x} \\ \text{subject to} & \mathbf{x}^T\mathbf{B}\mathbf{x} = 1. \end{array}$$

- The ℓ_0-norm regularized generalized eigenvalue problem is

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{maximize}} & \mathbf{x}^T\mathbf{A}\mathbf{x} - \rho \left\|\mathbf{x}\right\|_0 \\ \text{subject to} & \mathbf{x}^T\mathbf{B}\mathbf{x} = 1. \end{array}$$

- Replace $\left\|x_i\right\|_0$ by some nicely behaved function $g_p\left(x_i\right)$:
  - $\left|x_i\right|^p, 0 < p \leq 1$
  - $\log\left(1 + \left|x_i\right|/p\right)/\log\left(1 + 1/p\right), p > 0$
  - $1 - e^{-\left|x_i\right|/p}, p > 0.$
- Take $g_p\left(x_i\right) = \left|x_i\right|^p$ for example.

Sparse generalized eigenvalue problem

- Majorize $g_p(x_i)$ at x_i^k by quadratic function $w_i^k x_i^2 + c_i^k$ (J. Song, Babu, et al. 2015a)[6].
- The surrogate function for $g_p(x_i) = |x_i|^p$ is defined as

$$u\left(x_i, x_i^k\right) = \frac{p}{2}\left|x_i^k\right|^{p-2} x_i^2 + \left(1 - \frac{p}{2}\right)\left|x_i^k\right|^p.$$

- Solve at each iteration the following GEVP:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{maximize}} & \mathbf{x}^T\mathbf{A}\mathbf{x} - \rho\mathbf{x}^T\text{diag}\left(\mathbf{w}^k\right)\mathbf{x} \\ \text{subject to} & \mathbf{x}^T\mathbf{B}\mathbf{x} = 1 \end{array}$$

- However, as $|x_i| \to 0$, $w_i \to +\infty$.

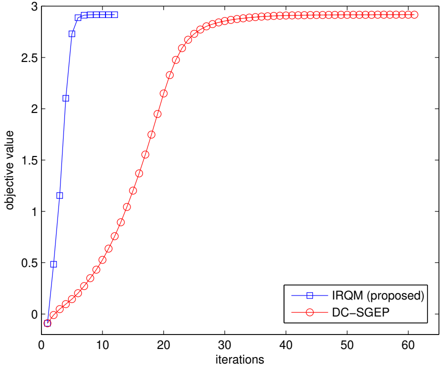
[6] J. Song, P. Babu, and D. P. Palomar, "Sparse generalized eigenvalue problem via smooth optimization," *IEEE Trans. Signal Processing*, vol. 63, no. 7, pp. 1627–1642, 2015.

Sparse generalized eigenvalue problem

- Smooth approximation of

$$g_p(x) : g_p^{\epsilon}(x) = \begin{cases} \frac{p}{2}\epsilon^{p-2}x^2, & |x| \leq \epsilon \\ |x|^p - \left(1 - \frac{p}{2}\right)\epsilon^p, & |x| > \epsilon \end{cases}$$

- When $|x| \leq \epsilon$, w remains to be a constant.



Sequence design

- Complex unimodular sequence $\{x_n \in \mathbb{C}\}_{n=1}^N$.
- Autocorrelation: $r_k = \sum_{n=k+1}^{N} x_n x_{n-k}^* = r_{-k}^*, k = 0, \ldots, N-1$.
- Integrated sidelobe level (ISL):

$$\mathrm{ISL} = \sum_{k=1}^{N-1} |r_k|^2 .$$

- Problem formulation:

$$\begin{array}{ll} \underset{\{x_n\}_{n=1}^N}{\text{minimize}} & \mathrm{ISL} \\ \text{subject to} & |x_n| = 1, \;\; n = 1, \ldots, N. \end{array}$$

Sequence design

- By Fourier transform:

$$\text{ISL} \propto \sum_{p=1}^{2N} \left[\left| \mathbf{a}_p^H \mathbf{x} \right|^2 - N \right]^2$$

with $\mathbf{x} = [x_1, \ldots, x_N]^T$, $\mathbf{a}_p = \left[1, e^{j\omega_p}, \ldots, e^{j\omega_p(N-1)}\right]^T$ and $\omega_p = \frac{2\pi}{2N}(p-1)$.

- Equivalent problem:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & \sum_{p=1}^{2N} \left(\mathbf{a}_p^H \mathbf{x} \mathbf{x}^H \mathbf{a}_p \right)^2 \\ \text{subject to} & |x_n| = 1, \ \forall n. \end{array}$$

Sequence design

- Define $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_{2N}]$, $\mathbf{p}^k = \left[|\mathbf{a}_1^H \mathbf{x}^k|^2, \ldots, |\mathbf{a}_{2N}^H \mathbf{x}^k|^2\right]^T$, $\tilde{\mathbf{A}} = \mathbf{A}\left(\text{diag}\left(\mathbf{p}^k\right) - p_{\max}^k \mathbf{I}\right)\mathbf{A}^H$.
- Quadratic surrogate function:

$$p_{\max}^k \mathbf{x}^H \mathbf{A}\mathbf{A}^H \mathbf{x} + 2\text{Re}\left(\mathbf{x}^H \left(\tilde{\mathbf{A}} - 2N^2 \mathbf{x}^k (\mathbf{x}^k)^H\right) \mathbf{x}^k\right)$$

where $p_{\max}^k \mathbf{x}^H \mathbf{A}\mathbf{A}^H \mathbf{x}$ is a constant.

- Majorized problem is (J. Song, Babu, et al. 2015b)[7]

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & \|\mathbf{x} - \mathbf{y}\|_2 \\ \text{subject to} & |x_n| = 1, \ \forall n \end{array}$$

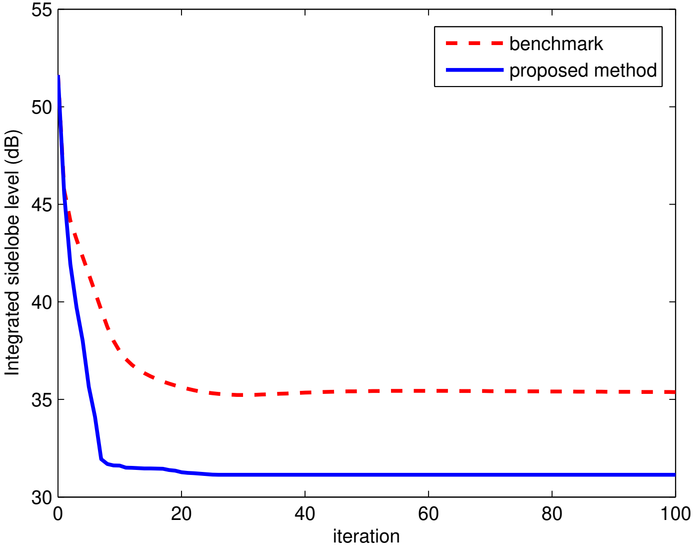
with $\mathbf{y} = -\left(\tilde{\mathbf{A}} - 2N^2 \mathbf{x}^k (\mathbf{x}^k)^H\right) \mathbf{x}^k$.

- Closed-form solution: $x_n = e^{j \arg(y_n)}$.

---

[7] J. Song, P. Babu, and D. P. Palomar, "Optimization methods for designing sequences with low autocorrelation sidelobes," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 3998–4009, 2015.

Sequence design



Covariance matrix estimation

- $\mathbf{x}_i \sim \text{elliptical}\left(\mathbf{0}, \boldsymbol{\Sigma}\right)$
- Fitting normalized sample $\mathbf{s}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|_2}$ to Angular Central Gaussian distribution

$$f(\mathbf{s}_i) \propto \det\left(\boldsymbol{\Sigma}\right)^{-1/2} \left(\mathbf{s}_i^T \boldsymbol{\Sigma}^{-1} \mathbf{s}_i\right)^{-K/2}$$

- Shrinkage penalty

$$h\left(\boldsymbol{\Sigma}\right) = \log\det\left(\boldsymbol{\Sigma}\right) + \operatorname{Tr}\left(\boldsymbol{\Sigma}^{-1}\mathbf{T}\right)$$

- Solve the following problem:

$$\begin{array}{ll} \underset{\boldsymbol{\Sigma}}{\text{minimize}} & \log\det\left(\boldsymbol{\Sigma}\right) + \frac{K}{N}\sum \log\left(\mathbf{x}_i^T \boldsymbol{\Sigma}^{-1} \mathbf{x}_i\right) + \alpha h\left(\boldsymbol{\Sigma}\right) \\ \text{subject to} & \boldsymbol{\Sigma} \succeq \mathbf{0} \end{array}$$

Covariance matrix estimation

- At $\boldsymbol{\Sigma}^k$, the objective function is majorized by (Sun et al. 2014)[8]

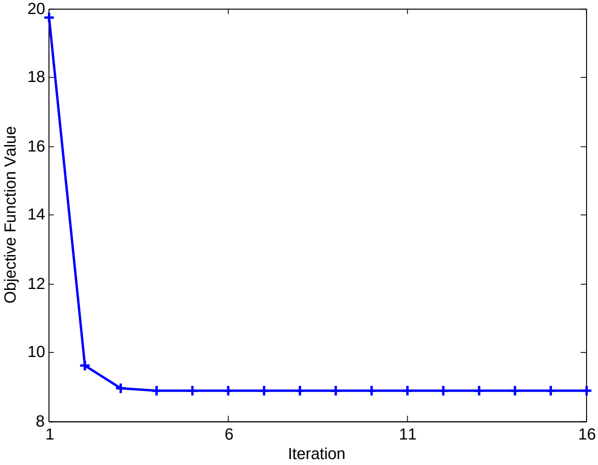
$$(1+\alpha)\log\det(\boldsymbol{\Sigma}) + \frac{K}{N}\sum_{i=1}^{N}\frac{\mathbf{x}_i^T\boldsymbol{\Sigma}^{-1}\mathbf{x}_i}{\mathbf{x}_i^T\left(\boldsymbol{\Sigma}^k\right)^{-1}\mathbf{x}_i} + \alpha\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\mathbf{T}\right)$$

- Surrogate function is convex in $\boldsymbol{\Sigma}^{-1}$.
- Setting the gradient to zero leads to the weighted sample average

$$\boldsymbol{\Sigma}^{k+1} = \frac{1}{1+\alpha}\frac{K}{N}\sum\frac{\mathbf{x}_i\mathbf{x}_i^T}{\mathbf{x}_i^T\left(\boldsymbol{\Sigma}^k\right)^{-1}\mathbf{x}_i} + \frac{\alpha}{1+\alpha}\mathbf{T}$$

[8] Y. Sun, P. Babu, and D. P. Palomar, "Regularized Tyler's scatter estimator: Existence, uniqueness, and algorithms," *IEEE Trans. Signal Processing*, vol. 62, no. 19, pp. 5143–5156, 2014.

Covariance matrix estimation



Power control by GP

- Problem: maximize system throughput. Essentially we need to solve the following problem (Chiang et al. 2007)[9]:

$$\underset{\mathbf{P}\in\mathcal{P}}{\text{minimize}} \quad \frac{\sum_{j\neq i} G_{ij}P_j + n_i}{\sum_j G_{ij}P_j + n_i}.$$

- Objective function is the ratio of two posynomials.
- Minorize a posynomial, denoted by $g(\mathbf{x}) = \sum_i m_i(\mathbf{x})$, by the mononial

$$g(\mathbf{x}) \geq \prod_i \left(\frac{m_i(\mathbf{x})}{\alpha_i}\right)^{\alpha_i}$$

where $\alpha_i = \frac{m_i(\mathbf{x}^k)}{g(\mathbf{x}^k)}$. (Arithmetic-Geometric Mean Inequality)

- Solution: approximate the denominator posynomial $\sum_j G_{ij}P_j + n_i$ by monomial.

[9] M. Chiang, C. W. Tan, D. Palomar, D. O'Neill, and D. Julian, "Power control by geometric programming," *IEEE Trans. Wireless Commun*, vol. 6, no. 7, pp. 2640–2651, 2007.

Outline

Successive Convex Approximation (SCA)

- Consider the following problem:

$$\begin{array}{ll}\underset{\mathbf{x}}{\text{minimize}} & f(\mathbf{x})\\ \text{subject to} & \mathbf{x}\in\mathcal{X}\end{array}$$

  where $\mathcal{X}$ is a closed and convex set.
- The idea of SCA is to iteratively approximate the problem by a simpler one (like in MM).
- SCA approximates f by a strongly convex function $g(\mathbf{x}\mid\mathbf{x}^k)$ satisfying the property that $\nabla g(\mathbf{x}^k\mid\mathbf{x}^k)=\nabla f(\mathbf{x}^k)$.
- At iteration $k=0,1,\ldots$ the surrogate problem is (Scutari et al. 2014)[10]

$$\begin{array}{ll}\underset{\mathbf{x}}{\text{minimize}} & g(\mathbf{x}\mid\mathbf{x}^k)+\frac{\tau}{2}(\mathbf{x}-\mathbf{x}^k)^T\mathbf{Q}(\mathbf{x}^k)(\mathbf{x}-\mathbf{x}^k)\\ \text{subject to} & \mathbf{x}\in\mathcal{X}\end{array}$$

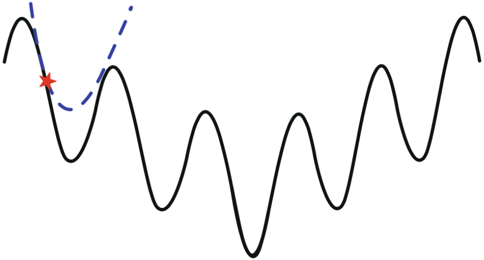
  where $\mathbf{Q}(\mathbf{x}^k)\succ\mathbf{0}$.

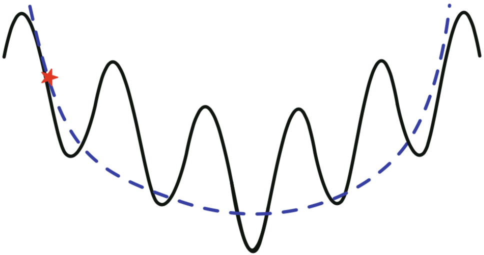
[10] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang, "Decomposition by partial linearization: Parallel optimization of multi-agent systems," *IEEE Trans. Signal Processing*, vol. 62, no. 3, pp. 641–656, 2014.

MM vs SCA

Surrogate function:

- MM requires the surrogate function to be a global upper-bound (which can be too demanding in some cases), albeit not necessarily convex.
- SCA relaxes the upper-bound condition, but it requires the surrogate to be strongly convex.





Constraint set:

- In principle, both SCA and MM require the feasible set $\mathcal{X}$ to be convex.
- MM can be easily extended to nonconvex $\mathcal{X}$ on a case by case basis; for example: (J. Song, Babu, et al. 2015a)[11], (Kumar et al. 2019)[12], (Kumar et al. 2020)[13].
- SCA can be extended to convexify the constraint functions, but cannot deal with a nonconvex $\mathcal{X}$ directly, which limits its applicability in many real-world applications.

---

[11] J. Song, P. Babu, and D. P. Palomar, "Sparse generalized eigenvalue problem via smooth optimization," *IEEE Trans. Signal Processing*, vol. 63, no. 7, pp. 1627–1642, 2015.

[12] S. Kumar, J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "Structured graph learning via laplacian spectral constraints," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, 2019.

[13] S. Kumar, J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "A unified framework for structured graph learning via spectral constraints," *Journal of Machine Learning Research (JMLR)*, pp. 1–60, 2020.

MM vs SCA

Schedule of updates:

- MM updates the whole variable $\mathbf{x}$ at each iteration (so in principle no distributed implementation).
- If the majorizer in MM happens to be block separable in $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$, then one can have a parallel update.
- Block MM updates each block of $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ sequentially.
- SCA, on the other hand, naturally has a parallel update (assuming the constraints are separable), which can be useful for distributed implementation.

Outline

Feasible Cartesian product structure

- Consider a general optimization problem

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in \mathcal{X} \end{array}$$

where the optimization variable can be separated into N blocks

$$\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$$

and the feasible set has a **Cartesian product** structure

$$\mathcal{X} = \prod_{i=1}^{N} \mathcal{X}_i.$$

- The problem can be written as

$$\begin{array}{lll} \underset{\mathbf{x}}{\text{minimize}} & f(\mathbf{x}_1, \ldots, \mathbf{x}_N) & \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i & i = 1, \ldots, N. \end{array}$$

Preliminary: Block Coordinate Descent (BCD)

- The **Block Coordinate Descent (BCD) algorithm**, also called nonlinear **Gauss-Seidel algorithm**, optimizes $f(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ sequentially.
- At iteration k, for $i = 1, \ldots, N$:

$$\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i \in \mathcal{X}_i} f\left(\mathbf{x}_1^{k+1}, \ldots, \mathbf{x}_{i-1}^{k+1}, \mathbf{x}_i, \mathbf{x}_{i+1}^{k} \ldots, \mathbf{x}_{N+1}^{k}\right)$$

- Observe that at each iteration k the blocks are optimized sequentially.
- Merits of BCD:
  1. each subproblem may be much easier to solve, or even may have a closed-form solution;
  2. the objective value is nonincreasing along the BCD updates;
  3. it allows parallel or distributed implementations.

Preliminary: Block Coordinate Descent (BCD)

Algorithm: BCD

Initialize $\mathbf{x}^0 \in \mathcal{X}$ and set $k = 0$.

repeat

1. $k \leftarrow k + 1$, $i = (k \bmod n) + 1$
2. $\mathbf{x}_i^k = \arg\min_{\mathbf{x}_i \in \mathcal{X}_i} f\left(\mathbf{x}_i, \mathbf{x}_{-i}^{k-1}\right)$
3. $\mathbf{x}_i^k \leftarrow \mathbf{x}_i^{k-1}$, $\forall k \neq i$

until convergence

return $\mathbf{x}^k$

Preliminary: Convergence of BCD

- Suppose that i) $f(\cdot)$ is continuously differentiable over $\mathcal{X}$ and ii) each block optimization is strictly convex. Then, every limit point of the sequence $\{\mathbf{x}^k\}$ is a stationary point (Bertsekas 1999)[14], (Bertsekas and Tsitsiklis 1997)[15].
- If $\mathcal{X}$ is convex, then the strict convexity of each block optimization can be relaxed to simply having a unique solution.
- Convergence generalizations: it converges in any of the following cases (Grippo and Sciandrone 2000)[16]:
  - the two-block case $N = 2$;
  - $f(\cdot)$ is component-wise strictly quasi-convex w.r.t. $N - 2$ components;
  - $f(\cdot)$ is pseudo-convex.

---

[14] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[15] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[16] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss–Seidel method under convex constraints," *Oper. Res. Lett.*, vol. 26, no. 3, pp. 127–136, 2000.

Outline

Block Majorization-Minimization

Combination of MM and BCD (Razaviyayn et al. 2013)[17].

Algorithm: Block MM

Initialize $\mathbf{x}^0 \in \mathcal{X}$ and set $k = 0$.

repeat

1. $k \leftarrow k + 1$, $i = (k \bmod N) + 1$
2. $\mathbf{x}^k$ as + ith block: $\mathbf{x}_i^k \in \arg\min_{\mathbf{x}_i \in \mathcal{X}_i} u_i\left(\mathbf{x}_i, \mathbf{x}^{k-1}\right)$ + other blocks: $\mathbf{x}_i^k \leftarrow \mathbf{x}_i^{k-1}$, $\forall k \neq i$

until convergence

return $\mathbf{x}^k$

[17] M. Razaviyayn, M. Hong, and Z. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1126–1153, 2013.

Convergence

- Suppose surrogate function $u_i(\cdot,\cdot)$ satisfies the following assumptions:

$$u_i(\mathbf{y}_i,\mathbf{y}) = f(\mathbf{y}),\ \forall \mathbf{y}\in\mathcal{X}, \forall i \tag{B1}$$

$$u_i(\mathbf{x}_i,\mathbf{y}) \geq f(\mathbf{y}_1,\ldots,\mathbf{y}_{i-1},\mathbf{x}_i,\mathbf{y}_{i+1},\ldots,\mathbf{y}_n) \quad \forall \mathbf{x}_i\in\mathcal{X}_i, \forall \mathbf{y}\in\mathcal{X}, \forall i \tag{B2}$$

$$u_i'(\mathbf{x}_i,\mathbf{y};\mathbf{d}_i)|_{\mathbf{x}_i=\mathbf{y}_i} = f'(\mathbf{y};\mathbf{d}),\quad \forall \mathbf{d}=(\mathbf{0},\ldots,\mathbf{d}_i,\ldots,\mathbf{0}) \text{ such that } \mathbf{y}_i+\mathbf{d}_i\in\mathcal{X}_i, \forall i \tag{B3}$$

$$u_i(\mathbf{x}_i,\mathbf{y}) \text{ is continuous in } (\mathbf{x}_i,\mathbf{y}),\ \forall i \tag{B4}$$

- In short, $u_i(\mathbf{x}_i,\mathbf{x}^k)$ majorizes $f(\mathbf{x})$ on the ith block.

Convergence

The following gives the convergence of the MM algorithm to a stationary point (Razaviyayn et al. 2013)[18].

Theorem

Suppose $\mathcal{X}$ is convex. Under assumptions B1-B4 (for simplicity assume that f is continuously differentiable):

- if $u_i(\mathbf{x}_i, \mathbf{y})$ is quasi-convex in $\mathbf{x}_i$, each subproblem $\min_{\mathbf{x}_i \in \mathcal{X}_i} u_i(\mathbf{x}_i, \mathbf{x}^{k-1})$ has a unique solution for any $\mathbf{x}^{k-1} \in \mathcal{X}$, then every limit point of $\{\mathbf{x}^k\}$ is a stationary point.
- if the level set $\mathcal{X}^0 = \{\mathbf{x} | f(\mathbf{x}) \leq f(\mathbf{x}^0)\}$ is compact, each subproblem $\min_{\mathbf{x}_i \in \mathcal{X}_i} u_i(\mathbf{x}_i, \mathbf{x}^{k-1})$ has a unique solution for any $\mathbf{x}^{k-1} \in \mathcal{X}$ for at least $m-1$ blocks, then $\lim_{k \to \infty} d(\mathbf{x}^k, \mathcal{X}^\star) = 0$.

[18] M. Razaviyayn, M. Hong, and Z. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1126–1153, 2013.

Outline

Alternating proximal minimization

- Consider the problem

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & f(\mathbf{x}_1, \ldots, \mathbf{x}_m) \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i, \end{array}$$

  with $f(\cdot)$ being convex in each block.
- The convergence of BCD is not easy to establish since each subproblem may have multiple solutions.
- Alternating Proximal Minimization solves

$$\begin{array}{ll} \underset{\mathbf{x}_i}{\text{minimize}} & f\left(\mathbf{x}_1^k, \ldots, \mathbf{x}_{i-1}^k, \mathbf{x}_i, \mathbf{x}_{i+1}^k, \ldots, \mathbf{x}_m^k\right) + \frac{1}{2c}\left\|\mathbf{x}_i - \mathbf{x}_i^k\right\|^2 \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i \end{array}$$

- Strictly convex objective $\rightarrow$ unique minimizer.

Proximal splitting algorithm

- Consider the following problem

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & \sum_{i=1}^{m} f_i\left(\mathbf{x}_i\right) + f_{m+1}\left(\mathbf{x}_1, \ldots, \mathbf{x}_m\right) \\ \text{subject to} & \mathbf{x}_i \in \mathcal{X}_i, \;\; i = 1, \ldots, m \end{array}$$

with f_i convex and lower semicontinuous, f_{m+1} convex and

$$\left\| \nabla f_{m+1}\left(\mathbf{x}\right) - \nabla f_{m+1}\left(\mathbf{y}\right) \right\| \leq \beta_i \left\| \mathbf{x}_i - \mathbf{y}_i \right\|.$$

- Cyclically update:

$$\mathbf{x}_i^{k+1} = \text{prox}_{\gamma f_i}\left(\mathbf{x}_i^k - \gamma \nabla_{\mathbf{x}_i} f_{m+1}\left(\mathbf{x}^k\right)\right),$$

with the proximity operator defined as

$$\text{prox}_f\left(\mathbf{x}\right) = \arg\min_{\mathbf{y} \in \mathcal{X}} f\left(\mathbf{y}\right) + \frac{1}{2}\left\|\mathbf{x} - \mathbf{y}\right\|^2.$$

Proximal splitting algorithm

- Block MM interpretation:

$$u_i\left(\mathbf{x}_i, \mathbf{x}^k\right) = f_i\left(\mathbf{x}_i\right) + \frac{1}{2\gamma}\left\|\mathbf{x}_i - \mathbf{x}_i^k\right\|^2 + \nabla_{\mathbf{x}_i} f_{m+1}\left(\mathbf{x}^k\right)^T\left(\mathbf{x}_i - \mathbf{x}_i^k\right)$$
$$+ \sum_{j \neq i} f_j\left(\mathbf{x}_j^k\right) + f_{m+1}\left(\mathbf{x}_{-i}^k, \mathbf{x}_i\right).$$

- Check:

$$f_{m+1}\left(\mathbf{x}^k\right) + \frac{1}{2\gamma}\left\|\mathbf{x}_i - \mathbf{x}_i^k\right\|^2 + \nabla_{\mathbf{x}_i} f_{m+1}\left(\mathbf{x}^k\right)^T\left(\mathbf{x}_i - \mathbf{x}_i^k\right)$$
$$\geq f_{m+1}\left(\mathbf{x}^k\right) + \frac{\beta_i}{2}\left\|\mathbf{x}_i - \mathbf{x}_i^k\right\|^2 + \nabla_{\mathbf{x}_i} f_{m+1}\left(\mathbf{x}^k\right)^T\left(\mathbf{x}_i - \mathbf{x}_i^k\right)$$
$$\geq f_{m+1}\left(\mathbf{x}_{-i}^k, \mathbf{x}_i\right)$$

with $\gamma \in [\epsilon_i, 2/\beta_i - \epsilon_i]$ and $\epsilon_i \in (0, \min\{1, 1/\beta_i\})$.

Outline

Robust estimation of mean and covariance matrix

- $\mathbf{x}_t \sim \text{elliptical}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$
- Fitting $\{\mathbf{x}_t\}$ to a Cauchy distribution with pdf (Sun et al. 2015)[19]

$$f(\mathbf{x}) \propto \det\left(\boldsymbol{\Sigma}\right)^{-1/2}\left(1 + \left(\mathbf{x}-\boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}\right)\right)^{-(N+1)/2}$$

- Solve the following problem:

$$\underset{\boldsymbol{\mu},\boldsymbol{\Sigma}\succeq\mathbf{0}}{\text{minimize}} \quad \log\det\left(\boldsymbol{\Sigma}\right) + \tfrac{N+1}{T}\textstyle\sum_{t=1}^{T}\log\left(1 + \left(\mathbf{x}_t-\boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1}\left(\mathbf{x}_t-\boldsymbol{\mu}\right)\right)$$

[19] Y. Sun, P. Babu, and D. P. Palomar, "Regularized robust estimation of mean and covariance matrix under heavy-tailed distributions," *IEEE Trans. Signal Processing*, vol. 63, no. 12, pp. 3096–3109, 2015.

Robust estimation of mean and covariance matrix

- Block MM algorithm update:

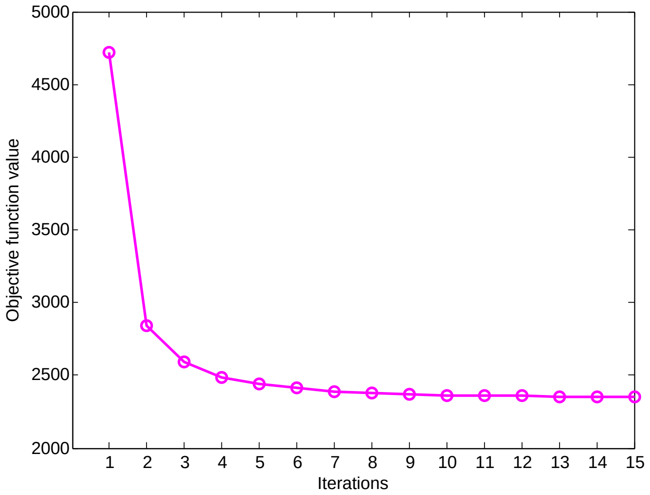
$$\boldsymbol{\mu}^{k+1} = \frac{\sum_{t=1}^{T} w_t(\boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k)\mathbf{x}_t}{\sum_{t=1}^{T} w_t(\boldsymbol{\mu}^k, \boldsymbol{\Sigma}^k)}$$

$$\boldsymbol{\Sigma}^{k+1} = \frac{N+1}{T}\sum_{t=1}^{T} w_t(\boldsymbol{\mu}^{k+1}, \boldsymbol{\Sigma}^k)(\mathbf{x}_t - \boldsymbol{\mu}^{k+1})(\mathbf{x}_t - \boldsymbol{\mu}^{k+1})^T$$

where

$$w_t(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{1 + (\mathbf{x}_t - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_t - \boldsymbol{\mu})}.$$

Robust estimation of mean and covariance matrix



Thanks

For more information visit:

https://www.danielppalomar.com

References I

Beck, A., & Pan, D. (2018). Convergence of an inexact majorization-minimization method for solving a class of composite optimization problems. In R. A. Giselsson P. (Ed.), *Large-scale and distributed optimization. Lecture notes in mathematics* (Vol. 2227). Springer, Cham.

Bertsekas, D. P. (1999). *Nonlinear programming*. Athena Scientific.

Bertsekas, D. P., & Tsitsiklis, J. N. (1997). *Parallel and distributed computation: Numerical methods*. Athena Scientific.

Candes, E. J., Wakin, M., & Boyd, S. (2008). Enhancing sparsity by reweighted l1 minimization. *J. Fourier Anal. Appl.*, *14*(5-6), 877–905.

Chiang, M., Tan, C. W., Palomar, D. P., O'Neill, D., & Julian, D. (2007). Power control by geometric programming. *IEEE Trans. Wireless Commun*, *6*(7), 2640–2651.

Grippo, L., & Sciandrone, M. (2000). On the convergence of the block nonlinear Gauss–Seidel method under convex constraints. *Oper. Res. Lett.*, *26*(3), 127–136.

References II

Kumar, S., Ying, J., M. Cardoso, J. V. de, & Palomar, D. P. (2019). Structured graph learning via laplacian spectral constraints. In *Proc. Advances in neural information processing systems (neurips)*. Vancouver, Canada.

Kumar, S., Ying, J., M. Cardoso, J. V. de, & Palomar, D. P. (2020). A unified framework for structured graph learning via spectral constraints. *Journal of Machine Learning Research (JMLR)*, 1–60.

Razaviyayn, M., Hong, M., & Luo, Z. (2013). A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM J. Optim.*, *23*(2), 1126–1153.

Scutari, G., Facchinei, F., Song, P., Palomar, D. P., & Pang, J.-S. (2014). Decomposition by partial linearization: Parallel optimization of multi-agent systems. *IEEE Trans. Signal Processing*, *62*(3), 641–656.

References III

Song, J., Babu, P., & Palomar, D. P. (2015a). Sparse generalized eigenvalue problem via smooth optimization. *IEEE Trans. Signal Processing*, *63*(7), 1627–1642.

Song, J., Babu, P., & Palomar, D. P. (2015b). Optimization methods for designing sequences with low autocorrelation sidelobes. *IEEE Trans. Signal Process.*, *63*(15), 3998–4009.

Sun, Y., Babu, P., & Palomar, D. P. (2014). Regularized Tyler's scatter estimator: Existence, uniqueness, and algorithms. *IEEE Trans. Signal Processing*, *62*(19), 5143–5156.

Sun, Y., Babu, P., & Palomar, D. P. (2015). Regularized robust estimation of mean and covariance matrix under heavy-tailed distributions. *IEEE Trans. Signal Processing*, *63*(12), 3096–3109.

Sun, Y., Babu, P., & Palomar, D. P. (2017). Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Trans. Signal Processing*, *65*(3), 794–816.