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

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
Consider the following presumably difficult optimization problem:

\[ \begin{align*}
\text{minimize} \quad & f(x) \\
\text{subject to} \quad & x \in \mathcal{X},
\end{align*} \]

with \( \mathcal{X} \) being the feasible set and \( f(x) \) being continuous.

Idea: successively minimize a more manageable surrogate function \( u(x, x^k) \):

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

hoping the sequence of minimizers \( \{x^k\} \) will converge to optimal \( x^* \).

Question: how to construct \( u(x, x^k) \)?

Answer: that’s more like an art (Sun et al. 2017)\(^1\).

---

Iterative algorithm

\[ u(x; x^{(k)}) \]

\[ f(x) \]

\[ x^{(k)} \]
Iterative algorithm

\[ u(x; x^{(k)}) \]

\[ f(x) \]

\[ x^{(k)} \]

\[ x^{(k+1)} \]
Iterative algorithm
**Surrogate/majorizer**

- **Construction rule of the majorizing function:**

\[ u(y, y) = f(y), \quad \forall y \in \mathcal{X} \quad (A1) \]
\[ u(x, y) \geq f(x), \quad \forall x, y \in \mathcal{X} \quad (A2) \]
\[ u'(x, y; d)_{|x=y} = f'(y; d), \quad \forall d \text{ with } y + d \in \mathcal{X} \quad (A3) \]
\[ u(x, y) \text{ is continuous in } x \text{ and } y \quad (A4) \]
Algorithm MM

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

repeat
  $x^{k+1} = \arg \min_{x \in \mathcal{X}} u(x, x^k)$
  $k \leftarrow k + 1$
until convergence
return $x^k$

Property of MM: $\{f(x^k)\}$ is nonincreasing, i.e., $f(x^{k+1}) \leq f(x^k)$.

That means that $\{f(x^k)\} \to p^*$, but what about the convergence of the iterates $\{x^k\}$?
Technical preliminaries

- **Distance from a point to a set:**
  \[ d(x, S) = \inf_{s \in S} \| x - s \|. \]

- **Limit point:** \( \bar{x} \) is a limit point of \( \{x^k\} \) if there exists a subsequence of \( \{x^k\} \) that converges to \( \bar{x} \). 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 \( x \) in the direction \( d \) is defined by
  \[ f'(x; d) \triangleq \lim_{\lambda \downarrow 0} \inf \frac{f(x + \lambda d) - f(x)}{\lambda}. \]

- **Stationary point:** \( x \in \mathcal{X} \) is a stationary point of \( f \) if
  \[ f'(x; d) \geq 0, \quad \forall d \text{ such that } x + d \in \mathcal{X}. \]

  - 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 \( \nabla f(x) \).
The following gives the convergence of the MM algorithm to a stationary point (Razaviyayn et al. 2013)\(^2\).

**Theorem**

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

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

\[
\lim_{k \to \infty} d(x^k, X^\*) = 0,
\]

where \(X^*\) is the set of stationary points.

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

References

- **Short tutorial on MM:**

- **Exhaustive tutorial on MM with many applications and tricks:**

- **Convergence of MM:**
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
Consider the following nonnegative LS problem:

\[
\minimize_{x \geq 0} \|Ax - b\|_2^2
\]

where \(b \in \mathbb{R}^m\), \(b \neq 0\), and \(A \in \mathbb{R}^{m \times n}\).

Observe that this problem cannot be solved with the conventional LS solution \(x = (A^T A)^{-1} A^T b\) 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 \(\|Ax - b\|_2^2\).
Consider the following quadratic majorizer of $f(x) = \frac{1}{2} \|Ax - b\|_2^2$:

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

where $\Phi(x^k) = \text{Diag} \left( \frac{[A^T Ax^k]^1}{x^k_1}, \ldots, \frac{[A^T Ax^k]^n}{x^k_n} \right)$.

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

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

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

$$\min_{x \geq 0} u(x, x^k)$$
Since this problem is convex, we can set the gradient to zero (ignoring for a moment the constraint):

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

which leads to $x = x^k - \Phi(x^k)^{-1}\nabla f(x^k)$.

Now using $\nabla f(x^k) = A^T Ax^k - A^T b$, we can finally write the MM iterate as

$$x^{k+1} = x^k - \text{Diag} \left( \frac{x_1^k}{[A^T Ax^k)_1}, \ldots, \frac{x_n^k}{[A^T Ax^k]_n} \right) (A^T Ax^k - A^T b)$$

$$= \text{Diag} \left( \frac{x_1^k}{[A^T Ax^k)_1}, \ldots, \frac{x_n^k}{[A^T Ax^k]_n} \right) A^T b$$

$$= c^k \odot x^k$$

where $c_i^k = \frac{[A^T b]_i}{[A^T Ax^k]_i}$.
Nonnegative Least Squares

- Example of the convergence of the MM iterative algorithm

\[ x^{k+1} = c^k \odot x^k \quad k = 0, 1, \ldots \]
Sparse regression: Reweighted $\ell_1$-norm minimization

Consider the following NP-hard sparse signal recovery problem:

$$\begin{align*}
\text{minimize} \quad & \|x\|_0 \\
\text{subject to} \quad & Ax = b.
\end{align*}$$

One common way to deal with it is with the $\ell_1$-norm approximation:

$$\begin{align*}
\text{minimize} \quad & \|x\|_1 \\
\text{subject to} \quad & Ax = b.
\end{align*}$$

For a better fit to the indicator function in $\|x\|_0$, consider a concave and nondecreasing penalty function $\phi(t)$. For example, $\phi(t) = \log(1 + t/\varepsilon)$:
Sparse regression: Reweighted $\ell_1$-norm minimization

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

$$\minimize_{x} \sum_{i=1}^{n} \phi(|x_i|)$$
subject to $Ax = b$.

- 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 $\ell_1$-norm minimization

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

\[
\begin{align*}
\text{minimize} \quad & \sum w_i^k |x_i| \\
\text{subject to} \quad & Ax = b
\end{align*}
\]

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

- More details in (Candes et al. 2008)\(^3\).

---

Consider the following convex problem:

\[
\min_{x} \| Ax - b \|_1
\]

If instead we had the \( \ell_2 \)-norm, then it would be an LS with solution \( x = (A^T A)^{-1} A^T b \).

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 \( \| Ax - b \|_1 \), where \( \| x \|_1 = \sum_{i=1}^{n} |x_i| \).
Reweighted LS for $\ell_1$-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 $\ell_1$-norm: a quadratic majorizer of $f(x) = \|Ax - b\|_1$ is

$$u(x, x^k) = \sum_{i=1}^{n} \frac{1}{2\|Ax^k - b\|_i}(|Ax^k - b|_i^2 + (|Ax^k - b|_i)^2).$$

- Now that we have the majorizer, we can write the MM iterative algorithm for $k = 0, 1, \ldots$ as

$$\min_{x} \| (Ax - b) \odot w^k \|_2^2$$

where $w^k_i = \sqrt{\frac{1}{2\|Ax^k - b\|_i}}$. 

D. Palomar (HKUST) Algorithms: MM
LASSO ($\ell_2 - \ell_1$ optimization) via BCD

- Consider the problem

\[
\min_x f(x) \triangleq \frac{1}{2} \| y - Ax \|_2^2 + \lambda \| x \|_1
\]

- We can use BCD on each element of $x = (x_1, \ldots, x_N)$.

- The optimization w.r.t. each block $x_i$ at iteration $k = 0, 1, \ldots$ is

\[
\min_{x_i} f_i(x_i) \triangleq \frac{1}{2} \| \tilde{y}_i^k - a_i x_i \|_2^2 + \lambda | x_i |
\]

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

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

\[
\begin{aligned}
    x_i^{k+1} = \operatorname{soft}_\lambda \left( a_i^T \tilde{y}_i^k \right) / \| a_i \|^2, \\
    i = 1, \ldots, N
\end{aligned}
\]

where $\operatorname{soft}_\lambda(u) \triangleq \operatorname{sign}(u) \left[ |u| - \lambda \right]_+$ is the \textbf{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 $f(x) \triangleq \frac{1}{2} \|y - Ax\|_2^2 + \lambda \|x\|_1$.
- Consider the following majorizer of $f(x)$:

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

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

- Note that $u(x, x^k)$ is a valid majorizer because it’s continuous, it is an upper-bound $u(x, x^k) \geq f(x)$ with $u(x^k, x^k) = f(x^k)$, and $\nabla u(x^k, x^k) = \nabla f(x^k)$.

- The majorizer can be rewritten in a more convenient way as

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

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

D. Palomar (HKUST)
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 $k = 0, 1, \ldots$

$$\min_{x \geq 0} \frac{c}{2} \|x - \bar{x}^k\|_2^2 + \lambda \|x\|_1$$

- This problem looks like the original one but without the matrix $A$ 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 $k = 0, 1, \ldots$

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

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

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*
Construction of majorizers or surrogate functions

- The performance of MM algorithm depends crucially on the majorizer or surrogate function $u(x, x^k)$.

- Guideline:
  - on the one hand, $u(x, x^k)$ should be as close as possible to the original function $f(x)$;
  - on the other hand, $u(x, x^k)$ should be easy to minimize.

- Many tricks to obtain majorizers in (Sun et al. 2017)$^4$, (Beck and Pan 2018)$^5$.

---


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:
  \[ \kappa \left( w^T x \right) = \kappa \left( w^T \left( x - x^k \right) + w^T x^k \right) \]
  \[ = \kappa \left( \sum_i \alpha_i \left( \frac{w_i (x_i - x_i^k)}{\alpha_i} + w^T x^k \right) \right) \]
  \[ \leq \sum_i \alpha_i \kappa \left( \frac{w_i (x_i - x_i^k)}{\alpha_i} + w^T x^k \right) \]

- If further assume that \( w \) and \( x \) are positive (\( \alpha_i = w_i x_i^k / w^T x^k \)):
  \[ \kappa \left( w^T x \right) \leq \sum_i \frac{w_i x_i^k}{w^T x^k} \kappa \left( \frac{w^T x^k}{x_i^k} x_i \right) \]

- The surrogate functions are separable (parallel algorithm).
Construction by Taylor expansion

- Suppose $\kappa(x)$ is concave and differentiable, then
  $$\kappa(x) \leq \kappa(x^k) + \nabla \kappa(x^k)(x - x^k),$$
  which is a linear upper-bound.

- Suppose $\kappa(x)$ is convex and twice differentiable, then
  $$\kappa(x) \leq \kappa(x^k) + \nabla \kappa(x^k)^T(x - x^k) + \frac{1}{2}(x - x^k)^T M(x - x^k)$$
  if $M - \nabla^2 \kappa(x) \succeq 0, \forall x$. 

D. Palomar (HKUST) Algorithms: MM
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:

\[
\|x\| \geq \frac{x^T x}{\|x^k\|}
\]

- Jensen’s Inequality:

\[
\kappa(E x) \leq E \kappa(x)
\]

with \(\kappa(\cdot)\) 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 \( \{x, z\} \) consists of observed variable \( x \) and latent variable \( z \).
- Objective: estimate parameter \( \theta \in \Theta \) from \( x \).
- Maximum likelihood estimator: \( \hat{\theta} = \arg \min_{\theta \in \Theta} - \log p(x | \theta) \)
- EM (Expectation Maximization) algorithm:
  - E-step: evaluate \( p(z | x, \theta^k) \)
    - “guess” \( z \) from current estimate of \( \theta \)
  - M-step: update \( \theta \) as \( \theta^{k+1} = \arg \min_{\theta \in \Theta} u(\theta, \theta^k) \), where
    \[
    u(\theta, \theta^k) = -E_{z|x,\theta^k} \log p(x, z | \theta)
    \]
    update \( \theta \) from “guessed” complete dataset.
The objective function can be written as

\[- \log p(x|\theta) = - \log E_{z|\theta} p(x|z, \theta)\]

\[= - \log E_{z|\theta} \left( \frac{p(z|x, \theta^k) p(x|z, \theta)}{p(z|x, \theta^k)} \right)\]

\[= - \log E_{z|x, \theta^k} \left( \frac{p(x|z, \theta)}{p(z|x, \theta^k)} p(z|\theta) \right)\]

\[\leq - E_{z|x, \theta^k} \log \left( \frac{p(x|z, \theta)}{p(z|x, \theta^k)} p(z|\theta) \right)\]

\[= - E_{z|x, \theta^k} \log p(x, z|\theta) + E_{z|x, \theta^k} p(z|x, \theta^k)\]

\[u(\theta, \theta^k)\]

where the inequality follows from Jensen’s inequality.
Suppose $f(x)$ is convex. Solve $\min_x f(x)$ by instead solving the equivalent problem

$$\minimize_{x \in \mathcal{X}, y \in \mathcal{X}} f(x) + \frac{1}{2c} \|x - y\|^2.$$

Objective function is strongly convex in both $x$ and $y$.

Algorithm:

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

$$y^{k+1} = x^{k+1}.$$

An MM interpretation:

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

\[
\minimize_{x \in \mathbb{R}^n} f(x),
\]

where \( f(x) = g(x) + h(x) \) with \( g(x) \) convex and \( h(x) \) concave.

DC (Difference of Convex) programming generates \( \{x^k\} \) by solving

\[
\nabla g(x^{k+1}) = -\nabla h(x^k).
\]

An MM interpretation:

\[
x^{k+1} = \arg \min_x \left\{ g(x) + \nabla h(x^k)^T (x - x^k) \right\}.
\]
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*
Sparse generalized eigenvalue problem

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

  $$\begin{align*}
  \text{maximize} & \quad \mathbf{x}^T \mathbf{A} \mathbf{x} \\
  \text{subject to} & \quad \mathbf{x}^T \mathbf{B} \mathbf{x} = 1.
  \end{align*}$$

- The $\ell_0$-norm regularized generalized eigenvalue problem is

  $$\begin{align*}
  \text{maximize} & \quad \mathbf{x}^T \mathbf{A} \mathbf{x} - \rho \| \mathbf{x} \|_0 \\
  \text{subject to} & \quad \mathbf{x}^T \mathbf{B} \mathbf{x} = 1.
  \end{align*}$$

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

- Take $g_p(x_i) = |x_i|^p$ 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
  \[
  u(x_i, x_i^k) = \frac{p}{2} |x_i^k|^{p-2} x_i^2 + \left(1 - \frac{p}{2}\right) |x_i|^p.
  \]
- Solve at each iteration the following GEVP:
  \[
  \begin{aligned}
  \text{maximize} & \quad x^T A x - \rho x^T \text{diag}(w^k) x \\
  \text{subject to} & \quad x^T B x = 1
  \end{aligned}
  \]
- However, as $|x_i| \to 0$, $w_i \to +\infty$.

---

Sparse generalized eigenvalue problem

- Smooth approximation of

\[ g_p(x) : g^\varepsilon_p(x) = \begin{cases} \frac{p}{2} \varepsilon^{p-2} x^2, & |x| \leq \varepsilon \\ |x|^p - \left(1 - \frac{p}{2}\right) \varepsilon^p, & |x| > \varepsilon \end{cases} \]

- When \(|x| \leq \varepsilon\), \(w\) remains to be a constant.
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):

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

Problem formulation:

$$\minimize \{x_n\}_{n=1}^{N} \text{ISL}$$

subject to $|x_n| = 1$, $n = 1, \ldots, N$. 

D. Palomar (HKUST)
Sequence design

- By Fourier transform:

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

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

- Equivalent problem:

\[
\text{minimize} \quad \sum_{p=1}^{2N} \left( a_p^H x x^H a_p \right)^2
\]

subject to \( |x_n| = 1, \forall n \).
Define $A = [\mathbf{a}_1, \ldots, \mathbf{a}_{2N}]$, $p^k = [|\mathbf{a}_1^H x^k|^2, \ldots, |\mathbf{a}_{2N}^H x^k|^2]^T$, $\tilde{A} = A \left( \text{diag} \left( p^k \right) - p^k_{\max} \mathbf{I} \right) A^H$.

- Quadratic surrogate function:
  
  $p^k_{\max} x^H A A^H x + 2 \text{Re} \left( x^H \left( \tilde{A} - 2N^2 x^k (x^k)^H \right) x^k \right)$

  where $p^k_{\max} x^H A A^H x$ is a constant.

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

  $$\min_x \| x - y \|_2$$

  subject to $|x_n| = 1, \ \forall n$

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

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

---

Sequence design

![Graph showing the comparison between benchmark and proposed method for integrated sidelobe level (dB) over iterations. The graph plots the integrated sidelobe level in decibels on the y-axis against iteration on the x-axis. The benchmark is represented by a dashed red line, while the proposed method is represented by a solid blue line. The graph indicates a significant improvement in the proposed method compared to the benchmark.]
Covariance matrix estimation

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

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

- Shrinkage penalty

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

- Solve the following problem:

$$\begin{align*}
\text{minimize} & \quad \log \det(\Sigma) + \frac{K}{N} \sum \log \left(\mathbf{x}_i^T \Sigma^{-1} \mathbf{x}_i\right) + \alpha h(\Sigma) \\
\text{subject to} & \quad \Sigma \succeq 0
\end{align*}$$
Covariance matrix estimation

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

$$
(1 + \alpha) \log \det(\Sigma) + \frac{K}{N} \sum_{i=1}^{N} x_i^T \Sigma^{-1} x_i + \alpha \text{Tr} \left( \Sigma^{-1} T \right)
$$

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

$$
\Sigma^{k+1} = \frac{1}{1 + \alpha} \frac{K}{N} \sum_{i=1}^{N} x_i x_i^T \left( \Sigma^k \right)^{-1} x_i + \frac{\alpha}{1 + \alpha} T
$$

---

Covariance matrix estimation

![Graph showing objective function value vs iteration number.](image-url)

D. Palomar (HKUST)
Problem: maximize system throughput. Essentially we need to solve the following problem (Chiang et al. 2007)\textsuperscript{9}:

$$\minimize_{P \in \mathcal{P}} \frac{\sum_{j \neq i} G_{ij} P_j + n_i}{\sum_j G_{ij} P_j + n_i}.$$ 

Objective function is the ratio of two posynomials.

Minorize a posynomial, denoted by $g(x) = \sum_i m_i(x)$, by the monomial

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

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

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

Successive Convex Approximation (SCA)

Consider the following problem:

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad x \in \mathcal{X}
\end{align*}
\]

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 \( g(x \mid x^k) \) satisfying the property that \( \nabla g(x^k \mid x^k) = \nabla f(x^k) \).

At iteration \( k = 0, 1, \ldots \) the surrogate problem is (Scutari et al. 2014)\(^\text{10}\)

\[
\begin{align*}
\text{minimize} & \quad g(x \mid x^k) + \frac{\tau}{2}(x - x^k)^T Q(x^k)(x - x^k) \\
\text{subject to} & \quad x \in \mathcal{X}
\end{align*}
\]

where \( Q(x^k) \succ 0 \).

---

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.
**MM vs SCA**

**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.

---


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 $\mathbf{x} = (x_1, \ldots, x_N)$, then one can have a parallel update.
- Block MM updates each block of $\mathbf{x} = (x_1, \ldots, x_N)$ sequentially.
- SCA, on the other hand, naturally has a parallel update (assuming the constraints are separable), which can be useful for distributed implementation.
Outline

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
Consider a general optimization problem

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad x \in \mathcal{X}
\end{align*}
\]

where the optimization variable can be separated into \( N \) blocks

\[x = (x_1, \ldots, x_N)\]

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

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

The problem can be written as

\[
\begin{align*}
\text{minimize} & \quad f(x_1, \ldots, x_N) \\
\text{subject to} & \quad x_i \in \mathcal{X}_i, \quad i = 1, \ldots, N.
\end{align*}
\]
Preliminary: Block Coordinate Descent (BCD)

- The **Block Coordinate Descent (BCD) algorithm**, also called nonlinear **Gauss-Seidel algorithm**, optimizes $f(x_1, \ldots, x_N)$ sequentially.

  At iteration $k$, for $i = 1, \ldots, N$:

  $$x_i^{k+1} = \arg \min_{x_i \in \mathcal{X}_i} f(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_N^k)$$

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

repeat

1. $k \leftarrow k + 1$, $i = (k \mod n) + 1$
2. $x^k_i = \arg \min_{x_i \in \mathcal{X}_i} f(x_i, x_{-i}^{k-1})$
3. $x^k_i \leftarrow x_i^{k-1}$, $\forall k \neq i$

until convergence

return $x^k$
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 \( \{x^k\} \) is a stationary point (Bertsekas 1999){\textsuperscript{14}}, (Bertsekas and Tsitsiklis 1997){\textsuperscript{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){\textsuperscript{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.

---


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*
Combination of MM and BCD (Razaviyayn et al. 2013).\(^{17}\)

**Algorithm: Block MM**

Initialize \(x^0 \in \mathcal{X}\) and set \(k = 0\).

repeat

1. \(k \leftarrow k + 1, \ i = (k \mod N) + 1\)
2. \(x^k\) as \(i\)th block: \(x^k_i \in \arg \min_{x_i \in \mathcal{X}_i} u_i(x_i, x^{k-1}) + \) other blocks: \(x^k_i \leftarrow x^{k-1}_i, \ \forall k \neq i\)

until convergence

return \(x^k\)

---

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

\begin{align}
  u_i(y_i, y) &= f(y), \forall y \in \mathcal{X}, \forall i \tag{B1} \\
  u_i(x_i, y) &\geq f(y_1, \ldots, y_{i-1}, x_i, y_{i+1}, \ldots, y_n) \tag{B2} \\
  u'_i(x_i, y; d_i)|_{x_i=y_i} &= f'(y; d), \\
  &\forall d = (0, \ldots, d_i, \ldots, 0) \text{ such that } y_i + d_i \in X_i, \forall i \tag{B3} \\
  u_i(x_i, y) &\text{ is continuous in } (x_i, y), \forall i \tag{B4}
\end{align}

In short, $u_i(x_i, x^k)$ majorizes $f(x)$ on the $i$th block.
Convergence

The following gives the convergence of the MM algorithm to a stationary point (Razaviyayn et al. 2013)\(^1\).

**Theorem**

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

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

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*
Consider the problem

\[
\begin{align*}
\minimize \quad & f(x_1, \ldots, x_m) \\
\text{subject to} \quad & x_i \in X_i,
\end{align*}
\]

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

\[
\begin{align*}
\minimize \quad & f(x_1^k, \ldots, x_{i-1}^k, x_i, x_{i+1}^k, \ldots, x_m^k) + \frac{1}{2c} \| x_i - x_i^k \|^2 \\
\text{subject to} \quad & x_i \in X_i
\end{align*}
\]

Strictly convex objective \( \rightarrow \) unique minimizer.
Consider the following problem

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{m} f_i(x_i) + f_{m+1}(x_1, \ldots, x_m) \\
\text{subject to} & \quad x_i \in \mathcal{X}_i, \quad i = 1, \ldots, m
\end{align*}
\]

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

\[
\|\nabla f_{m+1}(x) - \nabla f_{m+1}(y)\| \leq \beta_i \|x_i - y_i\|.
\]

Cyclically update:

\[
x_{i}^{k+1} = \text{prox}_{\gamma f_i} \left( x_i^k - \gamma \nabla x_i f_{m+1}(x^k) \right),
\]

with the proximity operator defined as

\[
\text{prox}_f(x) = \arg \min_{y \in \mathcal{X}} f(y) + \frac{1}{2} \|x - y\|^2.
\]
Proximal splitting algorithm

- Block MM interpretation:

\[
u_i(x_i, x^k) = f_i(x_i) + \frac{1}{2\gamma} \|x_i - x_i^k\|^2 + \nabla_{x_i} f_{m+1}(x^k)^T (x_i - x_i^k)
+ \sum_{j \neq i} f_j(x_j^k) + f_{m+1}(x_{-i}^k, x_i).
\]

- Check:

\[
f_{m+1}(x^k) + \frac{1}{2\gamma} \|x_i - x_i^k\|^2 + \nabla_{x_i} f_{m+1}(x^k)^T (x_i - x_i^k)
\geq f_{m+1}(x^k) + \frac{\beta_i}{2} \|x_i - x_i^k\|^2 + \nabla_{x_i} f_{m+1}(x^k)^T (x_i - x_i^k)
\geq f_{m+1}(x_{-i}^k, x_i)
\]

with \(\gamma \in [\epsilon_i, 2/\beta_i - \epsilon_i]\) and \(\epsilon_i \in (0, \min \{1, 1/\beta_i\})\).
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
Robust estimation of mean and covariance matrix

- $x_t \sim \text{elliptical} \left( \mu, \Sigma \right)$
- Fitting $\{x_t\}$ to a Cauchy distribution with pdf (Sun et al. 2015)$^{19}$

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

- Solve the following problem:

$$\min_{\mu, \Sigma \succeq 0} \log \det(\Sigma) + \frac{N+1}{T} \sum_{t=1}^{T} \log \left( 1 + (x_t - \mu)^T \Sigma^{-1} (x_t - \mu) \right)$$

---

Robust estimation of mean and covariance matrix

- Block MM algorithm update:

\[
\mu^{k+1} = \frac{\sum_{t=1}^{T} w_t(\mu^k, \Sigma^k) x_t}{\sum_{t=1}^{T} w_t(\mu^k, \Sigma^k)}
\]

\[
\Sigma^{k+1} = \frac{N + 1}{T} \sum_{t=1}^{T} w_t(\mu^{k+1}, \Sigma^k) (x_t - \mu^{k+1})(x_t - \mu^{k+1})^T
\]

where

\[
w_t(\mu, \Sigma) = \frac{1}{1 + (x_t - \mu)^T \Sigma^{-1} (x_t - \mu)}.
\]
Robust estimation of mean and covariance matrix
Thanks

For more information visit:

https://www.danielppalomar.com


References III


