Majorization-Minimization Algorithms in Signal Processing, Communications, and Machine Learning

MATH 5472: Final Project

Neel Kanth Kundu (20525393)

December 2020

1 Overview

Majorization-Minimization (MM) is a class of optimization algorithms that can be used to solve hard optimization problems in a systematic way. MM can also be used to solve NP-hard non-convex problems approximately in a computationally efficient way. Thus, MM algorithms have been successfully used to solve problems in machine learning, signal processing and communications system design. MM algorithm has two main steps: first it forms a surrogate function in the majorization step that can locally approximate the objective function, and then in the minimization step it minimizes the surrogate function. The key lies in finding a good surrogate function that can upper bound the objective function and it should be easy to minimize it (ideally a closed form solution is expected). A similar procedure can be followed for the maximization problems where in the minorizer step a surrogate function that lower bounds the objective function is found then it is maximized in the next step. This is called the minorization-maximization algorithm. MM algorithm was discovered long ago and it is closely related to the EM algorithm used in machine learning for maximum-likelihood (ML) problems with missing data or with latent variables. The E-step of the EM algorithm where the conditional expectation of the log-likelihood of the complete data is calculated is generalized by the first step of MM algorithm which finds a surrogate function. The second step is the same where the surrogate function is maximized. Thus, MM is a more general algorithm that has a wider scope of applications. Moreover, MM shares most of the convergence properties of EM algorithm.

The success of MM algorithm lies in the constructing a surrogate function with the following desired properties: (i) Separability in variables for parallel computation, (ii) Convexity and smoothness for easy optimization, and (iii) Existence of a closed form minimizer for fast computation. These properties ensure that the surrogate function is efficient, scalable and easy to implement. Since finding the right surrogate function that gives rise to a computationally efficient algorithm is a challenging task, this paper gives a series of techniques which can be used to find the surrogate functions and also presents many illustrative example problems in machine learning and signal processing applications.

1.1 Algorithmic Framework and Convergence

Consider the following general optimization problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in \mathscr{X} \end{aligned} \tag{1}$$

where $\mathscr{X}$ is a closed convex set and $f(\mathbf{x})$ is continuous function. For the cases when $f(\mathbf{x})$ is complicated then MM algorithm can be used to solve it efficiently. The idea in MM is to generate a sequence of feasible points $\mathbf{x}_{t+1}$ that successively minimizes a surrogate function $g\left(\mathbf{x} \mid \mathbf{x}_t\right)$ as

$$\mathbf{x}_{t+1} \in \arg\min_{\mathbf{x}\in\mathcal{X}} g\left(\mathbf{x} \mid \mathbf{x}_t\right) \tag{2}$$

The sequence of minimizers $\{\mathbf{x}_t\}$ converges to the global optimum $\mathbf{x}^*$ or to a stationary point depending upon some assumptions as stated below. The conditions that the surrogate function should satisfy are:

$$g(\mathbf{x} \mid \mathbf{x}) = f(\mathbf{x}), \forall \mathbf{x} \in \mathscr{X} \tag{3a}$$

$$g(\mathbf{x} \mid \mathbf{y}) \geq f(\mathbf{x}), \forall \mathbf{x}, \mathbf{y} \in \mathscr{X} \tag{3b}$$

$$g'(\mathbf{x} \mid \mathbf{y}; \mathbf{d})\big|_{\mathbf{x}=\mathbf{y}} = f'(\mathbf{y}; \mathbf{d}), \forall \mathbf{d} \text{ with } \mathbf{y}+\mathbf{d} \in \mathscr{X} \text{ (Directional Derivative)} \tag{3c}$$

$$g(\mathbf{x} \mid \mathbf{y}) \text{ is continuous in } \mathbf{x} \text{ and } \mathbf{y} \tag{3d}$$

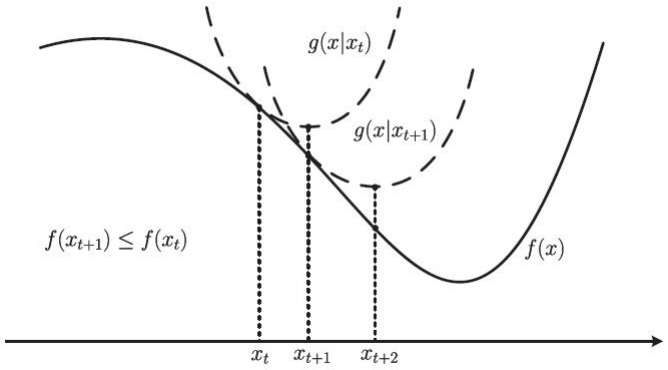
Then the general MM algorithm is given by

1. Find a feasible point $\mathbf{x}_0 \in \mathscr{X}$ and set $t = 0$
2. repeat:
3. $\quad \mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathscr{X}} g\left(\mathbf{x} \mid \mathbf{x}_t\right)$
4. $\quad t \leftarrow t+1$
5. until some convergence criterion is met

Under the assumptions (3a)-(3d), every limit point of the sequence $\{\mathbf{x}_t\}$ is a stationary point of the original problem. Further, if the sub-level sets $\mathscr{X}^0 = \left\{\mathbf{x} \mid f(\mathbf{x}) \leq f\left(\mathbf{x}^0\right)\right\}$ is compact then

$$\lim_{t\to\infty} d\left(\mathbf{x}_t, \mathscr{X}^\star\right) = 0\,, \tag{4}$$

where $\mathscr{X}^\star$ is the set of stationary points and $d(\mathbf{x}, \mathscr{S}) = \inf_{\mathbf{s}\in\mathscr{S}} \|\mathbf{x}-\mathbf{s}\|$. Pictorially the MM procedure can be visualized as shown in the figure below [1, Fig. 1].



A drawback of MM algorithm is that it can suffer from slow convergence speed due to the restrictive upper bound conditions. To overcome this problem various acceleration schemes for MM algorithm have been proposed like conjugate gradient acceleration, Newton and quasi-Newton type acceleration [1].

1.2 Surrogate Function Construction Techniques and Example Problems

1.2.1 First Order Taylor Expansion

Suppose that $f(\mathbf{x})$ can be decomposed as

$$f(\mathbf{x}) = f_0(\mathbf{x}) + f_{\text{ccv}}(\mathbf{x}) \tag{5}$$

where $f_{\text{ccv}}(\mathbf{x})$ is a differentiable concave function then linearizing $f_{\text{ccv}}(\mathbf{x})$ at $\mathbf{x} = \mathbf{x}_t$ gives the inequality: $f_{\text{ccv}}(\mathbf{x}) \leq f_{\text{ccv}}(\mathbf{x}_t) + \nabla f_{\text{ccv}}(\mathbf{x}_t)^T (\mathbf{x} - \mathbf{x}_t)$. Thus $f(\mathbf{x})$ can be upper bounded as

$$f(\mathbf{x}) \leq f_0(\mathbf{x}) + \nabla f_{\text{ccv}}(\mathbf{x}_t)^T \mathbf{x} + \text{ const.} \tag{6}$$

Example 1: Function $\log(x)$ can be upperbounded as

$$\log(x) \leq \log(x_t) + \frac{1}{x_t}(x - x_t) \tag{7}$$

with equality at $x = x_t$. This can be used to solve the following re-weighted l_1-norm minimization problem. In this we need to find a sparse solution of an under-determined system of equations $\mathbf{y} = \mathbf{A}\mathbf{x}$, and the problem is formulated as

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & \sum_{i=1}^{n} \log(\epsilon + |x_i|) \\ \text{subject to} & \mathbf{y} = \mathbf{A}\mathbf{x} \end{array} \tag{8}$$

where the objective is an approximation of the l_0-norm with $\epsilon > 0$. The re-weighted l_1-norm minimization algorithm solves problem (8) by solving

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & \sum_{i=1}^{n} \frac{|x_i|}{\epsilon + |x_i^t|} \\ \text{subject to} & \mathbf{y} = \mathbf{A}\mathbf{x} \end{array} \tag{9}$$

at the $t-$th iterate. The objective of problem (9) is obtained by applying the inequality (7) in (8). *Example 2:* Function $|x|^p$, $0 < p \leq 2$, can be upperbounded as

$$|x|^p \leq \frac{p}{2} |x_t|^{p-2} x^2 + \text{ const.} \tag{10}$$

provided $x_t \neq 0$. This inequality is used in iterative re-weighted least squares algorithm where the quadratic upper-bound is more tight and the minimization step admits an easy solution. This can be used for solving l_p-norm minimization problems. The idea is to use inequality (10) to majorize $\|\mathbf{x}\|_p^p$ as

$$\|\mathbf{x}\|_p^p \leq \|\mathbf{x}\|_{\mathbf{W}_t}^2 + \text{ const.} \tag{11}$$

where $\mathbf{W}_t$ is a diagonal matrix with the i -th diagonal element being $w_i^t = \frac{p}{2} |x_i^t|^{p-2}$ and $\|\mathbf{y}\|_{\mathbf{A}}^2 := \mathbf{y}^T \mathbf{A} \mathbf{y}$. Let us consider the following robust regression problem:

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_p^p, \tag{12}$$

where $\mathbf{b} \in \mathbb{R}^m$. Using (10) the surrogate function is given by

$$g(\mathbf{x} \mid \mathbf{x}_t) = \sum_{i=1}^{m} w_i^t (b_i - \mathbf{A}_{i,:}\mathbf{x})^2 \tag{13}$$

where $w_i^t = |b_i - \mathbf{A}_{i,:}\mathbf{x}_t|^{p-2}$, and $g(\mathbf{x} \mid \mathbf{x}_t)$ admits a closed-form minimizer given by

$$\mathbf{x}^{t+1} = \left(\mathbf{A}^T \mathbf{W}_t \mathbf{A}\right)^{-1} \mathbf{A}^T \mathbf{W}_t \mathbf{b} \tag{14}$$

Similar ideas have been used to solve sparse representation and compressed sensing problems [1].

1.2.2 Second Order Taylor Expansion

When the Hessian matrix of a function $f(\mathbf{x})$ is uniformly bounded i.e., $\mathbf{M} \succeq \nabla^2 f(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}$ then a quadratic surrogate function can be constructed using the following inequality

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \nabla f(\mathbf{y})^T(\mathbf{x}-\mathbf{y}) + \frac{1}{2}(\mathbf{x}-\mathbf{y})^T\mathbf{M}(\mathbf{x}-\mathbf{y}) \tag{15}$$

Example 3: Multi-class Logistic Regression. In this problem data pairs $(\mathbf{x}_n, \mathbf{t}_n)_{1\leq n\leq N}$, where $\mathbf{x}_n \in \mathbb{R}^m$ is a feature vector and $\mathbf{t}_n$ is a $(K+1)$ -dimensional encoding vector with $(\mathbf{t}_n)_i = 1$ if $\mathbf{x}$ belongs to the i -th category and $(\mathrm{t}_n)_i = 0$ otherwise are provided. We need to train a statistical model that can predict t based on x. Without loss of generality assume that there is only one training sample $(\mathbf{x}, \mathbf{t})$ for notational simplicity. The problem involves finding a $\mathbf{w}$, defined as $\mathbf{w} = \left[\mathbf{w}_1^T, \ldots, \mathbf{w}_K^T\right]^T$, that minimizes the negative log likelihood function:

$$L(\mathbf{w}) = \sum_{j=1}^{K} -t_j\mathbf{w}_j^T\mathbf{x} + \log\left(1 + \sum_{j=1}^{K}\exp\left(\mathbf{w}_j^T\mathbf{x}\right)\right) \tag{16}$$

The Hessian of $L(\mathbf{w})$ is uniformly upperbounded by matrix [1]

$$\mathbf{M} = \frac{1}{2}\left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{K+1}\right) \otimes \left(\mathbf{x}\mathbf{x}^T\right) \tag{17}$$

Therefore, inequality (15) implies that $L(\mathbf{w})$ can be upperbounded by

$$g\left(\mathbf{w} \mid \mathbf{w}^t\right) = \left(\left(\tilde{\mathbf{t}} - \mathbf{p}\left(\mathbf{w}^t\right)\right) \otimes \mathbf{x}\right)^T\left(\mathbf{w} - \mathbf{w}^t\right) + \frac{1}{2}\left(\mathbf{w} - \mathbf{w}^t\right)^T\mathbf{M}\left(\mathbf{w} - \mathbf{w}^t\right) \tag{18}$$

where $\tilde{\mathbf{t}} := [t_1; \ldots; t_K]$ and $\mathbf{p}(\mathbf{w}) := [p_1(\mathbf{w}); \ldots; p_K(\mathbf{w})]$ with

$$p_j(\mathbf{w}) = \frac{\exp\left(\mathbf{w}_j^T\mathbf{x}\right)}{1 + \sum_{j=1}^{K}\exp\left(\mathbf{w}_j^T\mathbf{x}\right)} \tag{19}$$

The update $\mathbf{w}$ is given by [1]

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \mathbf{M}^{-1}\left(\left(\tilde{\mathbf{t}} - \mathbf{p}\left(\mathbf{w}^t\right)\right) \otimes \mathbf{x}\right) \tag{20}$$

Thus, compared to Newton's algorithm, MM algorithm is computationally efficient as the Hessian matrix can be pre-computed once only whereas the Newton's algorithm requires to compute the Hessian matrix of L at each iteration. Similarly, MM can also be used for regularized logistic regression [1, eq. 76].

Example 4: The quadratic form $\mathbf{x}^H\mathbf{L}\mathbf{x}$, where $\mathbf{L}$ is a Hermitian matrix is upper-bounded as [1, eq. 26]

$$\mathbf{x}^H\mathbf{L}\mathbf{x} \leq \mathbf{x}^H\mathbf{M}\mathbf{x} + 2\,\mathrm{Re}\left(\mathbf{x}^H(\mathbf{L}-\mathbf{M})\mathbf{x}_t\right) + \mathbf{x}_t^H(\mathbf{M}-\mathbf{L})\mathbf{x}_t \tag{21}$$

where $\mathbf{M} \succeq \mathbf{L}$ and equality is achieved at $\mathbf{x} = \mathbf{x}_t$. This inequality can be used to solve the sparse linear regression problem which is formulated as

$$\min_{\mathbf{x}} \quad \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \rho\sum_{i=1}^{n}|x_i| \tag{22}$$

The problem is decoupled so that optimizing $\mathbf{x}$ can be done element-wise. Using the inequality (21) the first term of (22) is upperbound as [1, eq. 87]

$$\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 \leq \lambda \mathbf{x}^T\mathbf{x} - 2\mathbf{y}_t^T\mathbf{x} + \text{ const.} \tag{23}$$

where $\lambda = \lambda_{\max}\left(\mathbf{A}^T\mathbf{A}\right)$, and $\mathbf{y}_t = \mathbf{A}^T\mathbf{b} - \left(\mathbf{A}^T\mathbf{A} - \lambda\mathbf{I}\right)\mathbf{x}_t$. Then for each x_i, the problem boils down to finding a minimizer of

$$g\left(x_i \mid \mathbf{x}_t\right) = \lambda x_i^2 - 2y_i x_i + \rho\left|x_i\right| \tag{24}$$

Finally, the MM update is given by the soft-thresholding operator as: [1, eq. 89]

$$x_i^{t+1} = \begin{cases} \frac{y_i}{\lambda} - \frac{\rho}{2\lambda}, & \frac{y_i}{\lambda} > \frac{\rho}{2\lambda} \\ \frac{y_i}{\lambda} + \frac{\rho}{2\lambda}, & \frac{y_i}{\lambda} < \frac{\rho}{2\lambda} \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

Thus, the Lasso method can also be interpreted as an MM algorithm.

1.2.3 Cauchy-Schwartz Inequality

The Cauchy-Schwartz Inequality given by

$$\mathbf{x}^T\mathbf{y} \leq \|\mathbf{x}\|_2\|\mathbf{y}\|_2 \tag{26}$$

with equality when $\mathbf{x}$ and $\mathbf{y}$ are collinear can be used to form surrogate function as shown below.
Example 5: Function $|\mathbf{a}^H\mathbf{x}|$ is lowerbounded as [1, eq. 31]

$$\left|\mathbf{a}^H\mathbf{x}\right| \geq \text{Re}\left(\mathbf{x}_t^H\mathbf{a}\mathbf{a}^H\mathbf{x}\right) / \left|\mathbf{a}^H\mathbf{x}_t\right| \tag{27}$$

given that $\left|\mathbf{a}^H\mathbf{x}_t\right| \neq 0$ and equality is achieved at $\mathbf{x} = \mathbf{x}_t$. This inequality can be used in the phase retrieval problem that solves the following problem

$$\min_{\mathbf{x}} \left\|\sqrt{\mathbf{y}} - \left|\mathbf{A}^H\mathbf{x}\right|\right\|_2^2 \tag{28}$$

with $\sqrt{\cdot}$ applied element-wise. After expanding the squares and applying the inequality (27), the surrogate function is given by

$$g\left(\mathbf{x} \mid \mathbf{x}_t\right) = \left\|\mathbf{C}_t\sqrt{\mathbf{y}} - \mathbf{A}^H\mathbf{x}\right\|_2^2 \tag{29}$$

where $\mathbf{C}_t = \text{diag}\left(e^{j\arg\left(\mathbf{A}^H\mathbf{x}_t\right)}\right)$. It has a closed form minimizer given by [1, eq. 105]

$$\mathbf{x}_{t+1} = \left(\mathbf{A}\mathbf{A}^H\right)^{-1}\mathbf{A}\mathbf{C}_t\sqrt{\mathbf{y}} \tag{30}$$

1.2.4 Arithmetic-Geometric Mean Inequality

The AM-GM inequality is given by

$$\prod_{i=1}^n z_i^{\alpha_i} \leq \sum_{i=1}^n \frac{\alpha_i}{\|\boldsymbol{\alpha}\|_1} z_i^{\|\boldsymbol{\alpha}\|_1} \tag{31}$$

with z_i, α_i being non-negative scalars and equality is achieved when z_i's are equal. Let $z_i = x_i/x_i^t$ for $\alpha_i > 0$ and $z_i = x_i^t/x_i$ for $\alpha_i < 0$, then we have the following inequality
Example 6: The monomial $\prod_{i=1}^n x_i^{\alpha_i}$ is upper-bounded by[1, eq. 28]

$$\prod_{i=1}^n x_i^{\alpha_i} \leq \left(\prod_{i=1}^n \left(x_i^t\right)^{\alpha_i}\right) \sum_{i=1}^n \frac{|\alpha_i|}{\|\boldsymbol{\alpha}\|_1} \left(\frac{x_i}{x_i^t}\right)^{\|\boldsymbol{\alpha}\|_1 \,\text{sgn}(\alpha_i)} \tag{32}$$

with equality at $x_i = x_i^t$, $\forall\, i = 1, \ldots, n$. This type of inequality can be used to solve signomial and geometric programming problems that arise in wireless communications [1].

1.2.5 Schur-Complement

The Schur complement condition for $\mathbf{C} \succ \mathbf{0}$ states that

$$\mathbf{X} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \succeq 0 \tag{33}$$

if and only if the Schur complement of $\mathbf{C}$ is positive semi-definite matrix, i.e.,

$$\mathbf{S} := \mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T \succeq \mathbf{0} \tag{34}$$

This inequality can be used to upperbound the inverse of a matrix as shown below.
Example 7: Let $\mathbf{P} \succ \mathbf{0}$, then the matrix $\left(\mathbf{A}\mathbf{P}\mathbf{A}^H\right)^{-1}$ is upper-bounded by [1, eq. 34]

$$\mathbf{R}_t^{-1}\mathbf{A}\mathbf{P}_t\mathbf{P}^{-1}\mathbf{P}_t\mathbf{A}^H\mathbf{R}_t^{-1} \succeq \left(\mathbf{A}\mathbf{P}\mathbf{A}^H\right)^{-1} \tag{35}$$

where $\mathbf{R}_t = \mathbf{A}\mathbf{P}_t\mathbf{A}^H$ and equality is achieved at $\mathbf{P} = \mathbf{P}_t$.

1.2.6 Convexity Inequality

From convexity definition we have for a convex function f_{cvx}

$$f_{\text{cvx}}\left(\sum_{i=1}^{n} w_i \mathbf{x}_i\right) \leq \sum_{i=1}^{n} w_i f_{\text{cvx}}\left(\mathbf{x}_i\right) \tag{36}$$

where $\sum_{i=1}^{n} w_i = 1, \quad w_i \geq 0 . \forall i = 1, \ldots, n$. Equality is achieved if the $\mathbf{x}_i$'s are equal, or for different $\mathbf{x}_i$'s if f_{cvx} is not strictly convex. This inequality is used to prove the Jensen's inequality.
Example 8: The convex function $f\left(\mathbf{w}^T\mathbf{x}\right)$ can be upper bounded as

$$\begin{aligned} f\left(\mathbf{w}^T\mathbf{x}\right) = f\left(\mathbf{w}^T\left(\mathbf{x} - \mathbf{x}^t\right) + \mathbf{w}^T\mathbf{x}^t\right) &= f\left(\sum_i \alpha_i \left(\frac{w_i\left(x_i - x_i^t\right)}{\alpha_i} + \mathbf{w}^T\mathbf{x}^t\right)\right) \\ &\leq \sum_i \alpha_i f\left(\frac{w_i\left(x_i - x_i^t\right)}{\alpha_i} + \mathbf{w}^T\mathbf{x}^t\right) \end{aligned} \tag{37}$$

Further if we assume assume that $\mathbf{w}$ and $\mathbf{x}$ are positive then we obtain: $\left(\alpha_i = w_i x_i^t / \mathbf{w}^T\mathbf{x}^t\right)$

$$f\left(\mathbf{w}^T\mathbf{x}\right) \leq \sum_i \frac{w_i x_i^t}{\mathbf{w}^T\mathbf{x}^t} f\left(\frac{\mathbf{w}^T\mathbf{x}^t}{x_i^t} x_i\right) \tag{38}$$

The surrogate functions in (37) and (38) are separable and thus parallel algorithms can be used to solve them at a large scale.

Apart from the techniques and examples reviewed above, they can be used to upperbound more complicated functions by majorizing f more than once.

1.3 Relationship of MM Algorithm with Other Algorithms

The MM algorithm is a more general algorithm and it is closely related to other well known algorithms as reviewed below.

1.3.1 MM vs EM Algorithm

Let $\mathbf{x}$ be the observed variable and $\mathbf{z}$ be the latent variable then the MLE of parameter θ is obtained by maximizing the log-likelihood function

$$L(\boldsymbol{\theta}) = \log p(\mathbf{x} \mid \boldsymbol{\theta}) = \log \mathbb{E}_{\mathbf{z}|\boldsymbol{\theta}} p(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\theta}) \tag{39}$$

In the E-step the expectation w.r.t posterior distribution is calculated as

$$g\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}_t\right) = \mathbb{E}_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}_t} \log p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\theta}) \tag{40}$$

where $g\left(\theta \mid \theta_t\right)$ is the expected log-likelihood of the complete data set. Then in the M-step, the new estimate θ_{t+1} is defined as

$$\boldsymbol{\theta}_{t+1} \in \arg\max_{\boldsymbol{\theta} \in \Theta} g\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}_t\right) \tag{41}$$

By using the Jensen's inequality we obtain

$$\begin{aligned} L(\boldsymbol{\theta}) = \log \mathbb{E}_{\mathbf{z}|\boldsymbol{\theta}} p(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\theta}) &= \log \mathbb{E}_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}_t} \frac{p(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\theta}) p(\mathbf{z} \mid \boldsymbol{\theta})}{p\left(\mathbf{z} \mid \mathbf{x}, \boldsymbol{\theta}_t\right)} \\ &\geq \mathbb{E}_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}_t} \log \left(\frac{p(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\theta}) p(\mathbf{z} \mid \boldsymbol{\theta})}{p\left(\mathbf{z} \mid \mathbf{x}, \boldsymbol{\theta}_t\right)} \right) = g\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}_t\right) + \text{ const.} \end{aligned} \tag{42}$$

which shows that $g\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}_t\right)$ is a lower bound of $L(\theta)$. Thus, EM is a special case of MM algorithm [2].

1.3.2 MM vs Proximal Minimization

In proximal minimization we solve $\min_{\mathbf{x}} f(\mathbf{x})$ by solving the equivalent problem

$$\min_{\mathbf{x} \in \mathscr{X}, \mathbf{y} \in \mathscr{X}} g\left(\mathbf{x}, \mathbf{y}\right) = f(\mathbf{x}) + \frac{1}{2c} \|\mathbf{x} - \mathbf{y}\|^2 \tag{43}$$

where $f(\mathbf{x})$ being convex and further the objective function is strongly convex in both $\mathbf{x}$ and $\mathbf{y}$. The proximal algorithm is given by

$$\begin{aligned} \mathbf{x}^{t+1} &= \arg\min\nolimits_{\mathbf{x} \in \mathscr{X}} \left\{ f(\mathbf{x}) + \tfrac{1}{2c} \left\|\mathbf{x} - \mathbf{y}^t\right\|^2 \right\} \\ \mathbf{y}^{t+1} &= \mathbf{x}^{t+1} \end{aligned} \tag{44}$$

This algorithm can be interpreted as an MM algorithm as

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x} \in \mathscr{X}} \left\{ f(\mathbf{x}) + \frac{1}{2c} \left\|\mathbf{x} - \mathbf{x}^t\right\|^2 \right\} \tag{45}$$

1.3.3 MM vs DC (Difference of Convex) Programming

Consider the following unconstrained optimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \tag{46}$$

where $f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x})$ with $g(\mathbf{x})$ convex and $h(\mathbf{x})$ concave. DC programming generates $\left\{\mathbf{x}^t\right\}$ by solving

$$\nabla g\left(\mathbf{x}^{t+1}\right) = -\nabla h\left(\mathbf{x}^t\right) \tag{47}$$

The DC programming can also be interpreted as a special case of MM algorithm [1, eq. 26]

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}} \left\{ g(\mathbf{x}) + \nabla h\left(\mathbf{x}^t\right)^T \left(\mathbf{x} - \mathbf{x}^t\right) \right\} \tag{48}$$

2 Simulation Results

In this section we present simulation results to approximately solve a non-convex problem using MM algorithm that arises in wireless communications applications. The optimization problem to be solved is given by

$$\begin{aligned} \min_{\boldsymbol{\Phi}} \quad & \operatorname{tr}\left(\left(\boldsymbol{C}^{-1} + (\boldsymbol{\Phi} \otimes \boldsymbol{I}_M)^H \boldsymbol{W}^{-1} (\boldsymbol{\Phi} \otimes \boldsymbol{I}_M)\right)^{-1}\right) \\ \text{s.t} \quad & |\boldsymbol{\Phi}_{j,k}| = 1, \ \forall \ j,k = 1, \dots K \ . \end{aligned} \tag{49}$$

where $\boldsymbol{C} \in \mathbb{C}^{MK \times MK}$ is a given symmetric positive definite matrix representing covariance matrix of the signal, $\boldsymbol{W} = \sigma^2 \boldsymbol{I}_{MK}$ is the covariance matrix of the additive white Gaussian noise and $\boldsymbol{\Phi} \in \mathbb{C}^{K \times K}$ is the unknown variable to be optimized that has unit modulus entries. It should be noted that the optimization problem in (49) is non-convex due to the unit modulus constraint on elements of the phase shift matrix $\boldsymbol{\Phi}$. However, problems which do not have the unit-modulus constraints can be solved easily by using water filling approaches[3]. The problem in (49) can be solved using a MM algorithm by following the steps presented in [4]. We note that this objective function is similar to the objective function considered in [4, Eq. 40]. The difference lies in the constraints of our optimization variable $\boldsymbol{\Phi}$ and the swapping of matrices in the Kronecker operation. Using a similar approach as in [4], a surrogate function or majorizer function of the objective function can be constructed and finally the overall MM algorithm is summarized in Algorithm 1. Note that $\boldsymbol{B}_{i:M:i+(K-1)M, i:M:i+(K-1)M}$ denotes the submatrix of $\boldsymbol{B}$ extracted from the K rows and columns of $\boldsymbol{B}$ with indices $[i, i+M, \dots, i+M(K-1)]$, $\angle$ extracts the elementwise argument of the complex valued matrix and $||\cdot||_1$ denotes the maximum absolute column sum matrix norm. It has been shown that the MM based algorithm converges to a stationary point for bounded objective functions [5]. For this problem, the generated sequence of points $\{\boldsymbol{\Phi}_t\}, t = 0, 1, \dots$ monotonically decreases the objective function of (49) and the algorithm converges to a stationary point as shown in [4, Thm. 1].

We consider a simulation scenario with $M = 5, K = 10$ and the covariance matrix is modelled as $[\boldsymbol{C}]_{i,j} = \rho^{|i-j|}$. We use $\rho = 0.9$, a SNR of $\frac{1}{\sigma^2} = 0$ dB and set $\epsilon = 10^{-6}$. In order to observe the convergence of Algorithm 1, it is run 100 times with different random initialization points. The evolution of the objective function in (49) is plotted for the 100 different runs in Fig. 1(a). The distribution of final value of the objective function for the 100 different runs is shown in Fig. 1(b). It can be observed that all the different runs converge to a stationary point and attain almost the same value of the final objective function (with a slight difference in the order of 10^{-3}). Thus, MM algorithm can efficiently solve the optimization problem defined in (49) by generating a sequence of iterates $\boldsymbol{\Phi}_t$ with the final solution converging to a stationary point. [1]

3 Discussion

MM is a general optimization framework that can be used to solve difficult optimization problems by generating a sequence of iterates where the computation of each iterate is computationally very efficient. In this report we summarized the general MM framework, its convergence results and also its relations with other algorithms. MM algorithm is closely related to the EM algorithm that is widely used in machine learning applications with missing data or with latent variables. Specifically, it can be shown that EM algorithm, proximal minimization and DC programming all are special

[1] Note that the simulation results presented here are generated using my own implementation and the MATLAB code is available here: `https://github.com/nkkundu/MATH5472-Final-Project`

Algorithm 1: MM Algorithm for solving (49).

Input: $\epsilon, \boldsymbol{W}, \boldsymbol{C}$

Output: $\boldsymbol{\Phi}$

Set $t = 0$ and initialize $[\boldsymbol{\Phi}_0]_{i,j} = e^{j\theta_{i,j}}$ with $\theta_{i,j} \sim U[0, 2\pi]\ \forall\, i, j = 1, \ldots, K$ and $\tilde{\boldsymbol{\Phi}}_0 = \boldsymbol{\Phi}_0 \otimes \boldsymbol{I}_M$

$\text{MSE}_0 = \text{tr}\left(\left(\boldsymbol{C}^{-1} + \tilde{\boldsymbol{\Phi}}_0^H \boldsymbol{W}^{-1} \tilde{\boldsymbol{\Phi}}_0\right)^{-1}\right)$

repeat

$\tilde{\boldsymbol{\Phi}}_t = \boldsymbol{\Phi}_t \otimes \boldsymbol{I}_M$

$\boldsymbol{A}_t = \left(\tilde{\boldsymbol{\Phi}}_t \boldsymbol{C} \tilde{\boldsymbol{\Phi}}_t^H + \boldsymbol{W}\right)^{-1} \tilde{\boldsymbol{\Phi}}_t \boldsymbol{C}$

$\lambda_t = ||\boldsymbol{C}||_1 ||\boldsymbol{A}_t \boldsymbol{A}_t^H||_1$

$\boldsymbol{B} = \lambda_t \tilde{\boldsymbol{\Phi}}_t - \boldsymbol{A}_t \boldsymbol{A}_t^H \tilde{\boldsymbol{\Phi}}_t \boldsymbol{C} + \boldsymbol{A}_t \boldsymbol{C}$

$\tilde{\boldsymbol{B}} = \sum_{i=1}^{M} \boldsymbol{B}_{i:M:i+(K-1)M, i:M:i+(K-1)M}$

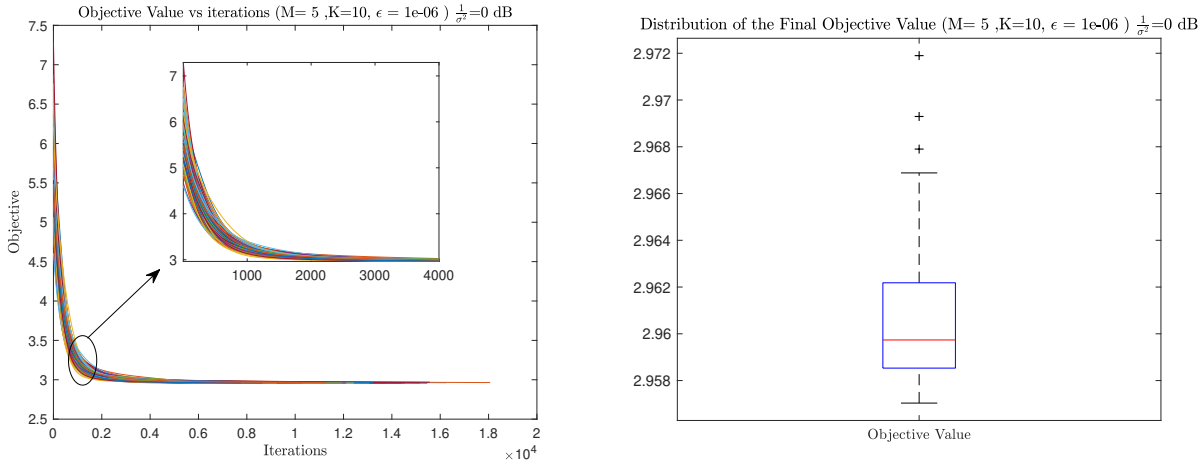
$\boldsymbol{\Phi}_{t+1} = \exp\{j \angle \tilde{\boldsymbol{B}}\}$

$\tilde{\boldsymbol{\Phi}}_{t+1} = \boldsymbol{\Phi}_{t+1} \otimes \boldsymbol{I}_M$

$\text{MSE}_{t+1} = \text{tr}\left(\left(\boldsymbol{C}^{-1} + \tilde{\boldsymbol{\Phi}}_{t+1}^H \boldsymbol{W}^{-1} \tilde{\boldsymbol{\Phi}}_{t+1}\right)^{-1}\right)$

$t = t + 1$

until $\text{MSE}_{t-1} - \text{MSE}_t \leq \epsilon$



(a) (b)

Figure 1: The plots in (a) show the objective value as the iterations of the MM algorithm increase, with $M = 5, K = 10$ at an SNR of 0 dB for 100 different random initialization points. The plot in (b) show the distribution of the final objective value for the 100 different runs of the MM Algorithm.

cases of the general MM algorithm. The success in using the MM algorithm lies in the construction of the surrogate function that can locally approximate the original objective function according to certain criteria and it should be easy to optimize at each iterate (preferably a closed form solution is expected). Thus, in this report we summarize the main techniques and inequalities which can be used to construct efficient surrogate functions. Various example problems with applications in machine learning and signal processing are also summarized to show the effectiveness of the MM algorithm. At the end we present a MM algorithm inspired from [4] to solve a non-convex problem that has a complicated objective function and also has unit modulus constraints on the optimization variable which makes it hard to solve. The MM algorithm can efficiently generate the iterates which converge to a stationary point as shown in the simulation results.

References

[1] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 794–816, 2016.

[2] W. J. Heiser, "Convergent computation by iterative majorization: Theory and applications in multidimensional data analysis," *Recent Advances in Descriptive Multivariate Analysis*, pp. 157–189, 1995.

[3] J. H. Kotecha and A. M. Sayeed, "Transmit signal design for optimal estimation of correlated MIMO channels," *IEEE Transactions on Signal Processing*, vol. 52, pp. 546–557, Feb. 2004.

[4] Z. Wang, P. Babu, and D. P. Palomar, "Design of PAR-constrained sequences for MIMO channel estimation via majorization–minimization," *IEEE Transactions on Signal Processing*, vol. 64, pp. 6132–6144, Dec. 2016.

[5] J. Song, P. Babu, and D. P. Palomar, "Optimization methods for designing sequences with low autocorrelation sidelobes," *IEEE Transactions on Signal Processing*, vol. 63, pp. 3998–4009, Aug. 2015.