Chapter 4

Majorize-minimize methods

Contents (class version)

4.0 Introduction

The **majorize-minimize** (**MM**) "algorithm" is really a family of methods for deriving algorithms for various applications. The applications are so numerous that entire books have been written on MM methods [1]. These notes focus on a couple of the most important *approaches* for designing majorizers for MM algorithms, in the context of several SIPML *applications* where **gradient descent** is inapplicable.

Approach overview

There are numerous techniques for designing MM algorithms; see [2, Ch. 14] at
http://web.eecs.umich.edu/~fessler/book/c-ox.pdf

These notes focus on two main techniques that are used especially widely:

- **quadratic** majorizers, and
- majorizers built on **convexity**.

Perhaps surprisingly, both of these design methods can be useful even for **nonconvex** problems.

Applications

These notes focus on the following concrete motivating applications. The common thread of most of these applications is that the standard **gradient descent** algorithm is inapplicable due to terms that are not **differentiable**, or that are differentiable but do not have a **Lipschitz continuous** gradient.

- More efficient **line search** for certain cost functions

$$\arg\min_{\alpha} \Psi(\boldsymbol{x} + \alpha \boldsymbol{d}) \,.$$

- Low-rank **matrix completion** (**LRMC**) (see EECS 551 notes) using the nondifferentiable rank constraint:

$$\hat{\boldsymbol{X}} = \arg\min_{\boldsymbol{X} \in \mathbb{F}^{M \times N}} \frac{1}{2} \left\| \boldsymbol{M} \odot (\boldsymbol{X} - \boldsymbol{Y}) \right\|_{\mathrm{F}}^2 + \chi_{\{\mathsf{rank}(\boldsymbol{X}) \le K\}}.$$

- **LASSO** / **sparse regression**

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \Psi(\boldsymbol{x}), \quad \Psi(\boldsymbol{x}) = \frac{1}{2} \left\| \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} \right\|_2^2 + \beta \left\| \boldsymbol{x} \right\|_1 = \sum_{i=1}^{M} h_i([\boldsymbol{A}\boldsymbol{x}]_i) + \beta \left\| \boldsymbol{x} \right\|_1$$

$$h_i(t) = \frac{1}{2} \left| t - y_i \right|^2 .$$

  The sparsity promoting term $\|\cdot\|_1$ is not differentiable, yet it is essential for problems with numerous features (machine learning) or under-sampled measurements (**compressed sensing**).

- Binary classifier design using convex loss functions and sparsity regularization:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \Psi(\boldsymbol{x}), \quad \Psi(\boldsymbol{x}) = \sum_{i=1}^{M} h([\boldsymbol{A}\boldsymbol{x}]_i) + \beta \left\|\boldsymbol{x}\right\|_1,$$

  where the ith row of $\boldsymbol{A}$ is the product of the binary label $y_i = \pm 1$ and the ith feature vector. We will consider any convex loss function h including **hinge** loss (not differentiable), **exponential** loss (does not have a **Lipschitz continuous** derivative), and **logistic** loss. Although the logistic loss is **convex** with **Lipschitz continuous** derivative, the $\|\cdot\|_1$ term is not differentiable.
- Measurements having Poisson distributions (gamma rays, X-rays, low light level optical imaging) where $\boldsymbol{y} \sim \mathsf{Poisson}\{\boldsymbol{A}\boldsymbol{x}\}$ for which the natural regularized estimator is:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \Psi(\boldsymbol{x}), \quad \Psi(\boldsymbol{x}) = \sum_{i=1}^{M} h_i([\boldsymbol{A}\boldsymbol{x}]_i) + \beta R(\boldsymbol{x}), \qquad h_i(\lambda) = \lambda - y_i \log(\lambda),$$

  where $h_i(\cdot)$ is the negative log-likelihood of the **Poisson distribution**.

This $h_i(\cdot)$ has a Lipschitz continuous derivative on $(0, \infty)$. (?)
A: True B: False **??**

The latter three applications all have data terms of the form $\sum_{i=1}^{M} h_i([\boldsymbol{A}\boldsymbol{x}]_i)$ for different convex h_i functions.

4.1 Majorize-minimize (MM) principles

To solve an optimization problem like

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \Psi(\boldsymbol{x})$$

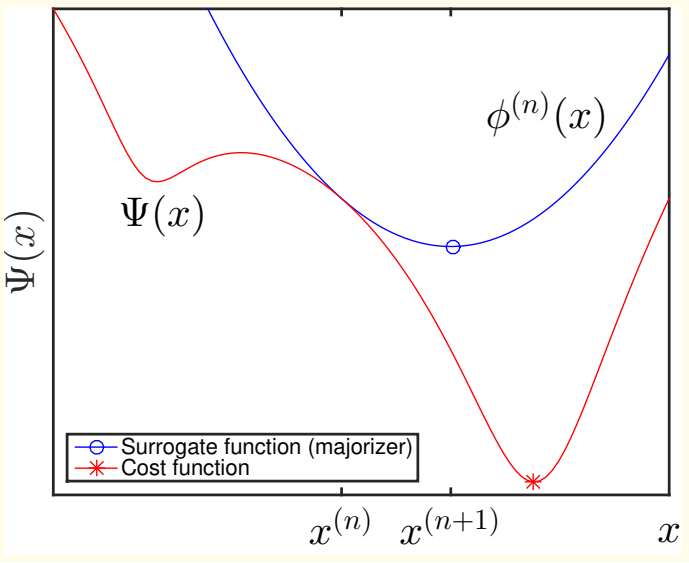
using a **majorize-minimize** (**MM**) approach, we first design a **majorizer** or **surrogate function** $\phi_k(\boldsymbol{x})$ that satisfies the following two conditions:

$$\Psi(\boldsymbol{x}_k) = \phi_k(\boldsymbol{x}_k) \tag{4.1}$$

$$\Psi(\boldsymbol{x}) \le \phi_k(\boldsymbol{x}), \; \forall \boldsymbol{x} \in \mathcal{X}. \tag{4.2}$$

Then the MM algorithm update is simply the following minimization step:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \phi_k(\boldsymbol{x}). \tag{4.3}$$



Properties of MM methods

Descent property

Any MM algorithm will monotonically decrease the cost function because of the **sandwich inequality**:

$$\Psi(\boldsymbol{x}_{k+1}) \leq \phi_k(\boldsymbol{x}_{k+1}) \leq \phi_k(\boldsymbol{x}_k) = \Psi(\boldsymbol{x}_k) \,.$$

Algebraic properties of majorizers

Affine invariance:
If ϕ_k is a majorizer for Ψ, then $a\phi_k + b$ is a majorizer for $a\,\Psi + b$ for any $a > 0$.

Linearity:
If $f_k(\boldsymbol{x})$ is a majorizer for $f(\boldsymbol{x})$ and $g_k(\boldsymbol{x})$ is a majorizer for $g(\boldsymbol{x})$, then $af_k(\boldsymbol{x}) + bg_k(\boldsymbol{x})$ is a majorizer for $af(\boldsymbol{x}) + bg(\boldsymbol{x})$ for any $a, b > 0$.

Adding a suitable nonnegative function:
If $\zeta(\boldsymbol{x}, \boldsymbol{x}_k) \geq 0$ is a (nonnegative) function for which $\zeta(\boldsymbol{x}_k, \boldsymbol{x}_k) = 0$, then the following function defines a **majorizer** for Ψ:

$$\phi_k(\boldsymbol{x}) \triangleq \Psi(\boldsymbol{x}) + \zeta(\boldsymbol{x}, \boldsymbol{x}_k). \tag{4.4}$$

These properties allow us to design majorizers for individual pieces of a composite cost function and then combine them together to form a majorizer for the entire cost function.

Quadratic majorizer for smooth cost functions

Suppose $\Psi(\boldsymbol{x}) : \mathbb{F}^N \to \mathbb{R}$ is smooth, meaning its gradient is $\boldsymbol{S}$-Lipschitz continuous per (3.14):

$$\left\| \boldsymbol{S}^{-1} \left(\nabla \Psi(\boldsymbol{x}) - \nabla \Psi(\boldsymbol{z}) \right) \right\|_2 \le \left\| \boldsymbol{S}' \left(\boldsymbol{x} - \boldsymbol{z} \right) \right\|_2 , \qquad \forall \boldsymbol{x}, \boldsymbol{z} \in \mathbb{F}^N .$$

Then a majorizer for $\Psi(\boldsymbol{x})$ (even for a **nonconvex** Ψ) is

$$\phi_k(\boldsymbol{x}) \triangleq \Psi(\boldsymbol{x}_k) + \operatorname{real}\{\langle \nabla \Psi(\boldsymbol{x}_k),\, \boldsymbol{x} - \boldsymbol{x}_k \rangle\} + \frac{1}{2} \left\| \boldsymbol{S}'(\boldsymbol{x} - \boldsymbol{x}_k) \right\|_2^2 . \tag{4.5}$$

Proof (generalizing [3, p. 22]). Clearly $\phi_k(\boldsymbol{x}_k) = \Psi(\boldsymbol{x}_k)$. By **Taylor's theorem** with remainder:

$$\begin{aligned}
\Psi(\boldsymbol{x}) &= \Psi(\boldsymbol{z}) + \operatorname{real}\left\{ \int_0^1 \langle \nabla \Psi(\boldsymbol{z} + \tau(\boldsymbol{x} - \boldsymbol{z})),\, \boldsymbol{x} - \boldsymbol{z} \rangle \, \mathrm{d}\tau \right\} \\
&= \Psi(\boldsymbol{z}) + \operatorname{real}\{\langle \nabla \Psi(\boldsymbol{z}),\, \boldsymbol{x} - \boldsymbol{z} \rangle\} + \operatorname{real}\left\{ \int_0^1 \langle \nabla \Psi(\boldsymbol{z} + \tau(\boldsymbol{x} - \boldsymbol{z})) - \nabla \Psi(\boldsymbol{z}),\, \boldsymbol{x} - \boldsymbol{z} \rangle \, \mathrm{d}\tau \right\} \\
&= \Psi(\boldsymbol{z}) + \operatorname{real}\{\langle \nabla \Psi(\boldsymbol{z}),\, \boldsymbol{x} - \boldsymbol{z} \rangle\} \\
&\qquad + \operatorname{real}\left\{ \int_0^1 \left\langle \boldsymbol{S}^{-1} \left(\nabla \Psi(\boldsymbol{z} + \tau(\boldsymbol{x} - \boldsymbol{z})) - \nabla \Psi(\boldsymbol{z}) \right),\, \boldsymbol{S}'(\boldsymbol{x} - \boldsymbol{z}) \right\rangle \mathrm{d}\tau \right\} .
\end{aligned}$$

$$
\begin{aligned}
&\implies |\Psi(\boldsymbol{x}) - \Psi(\boldsymbol{z}) - \operatorname{real}\{\langle \nabla \Psi(\boldsymbol{z}),\ \boldsymbol{x} - \boldsymbol{z} \rangle\}| \\
&\quad = \left| \operatorname{real}\left\{ \int_0^1 \left\langle \boldsymbol{S}^{-1} \left(\nabla \Psi(\boldsymbol{z} + \tau(\boldsymbol{x} - \boldsymbol{z})) - \nabla \Psi(\boldsymbol{z}) \right),\ \boldsymbol{S}'(\boldsymbol{x} - \boldsymbol{z}) \right\rangle \mathrm{d}\tau \right\} \right| \\
&\quad \le \left| \int_0^1 \left\langle \boldsymbol{S}^{-1} \left(\nabla \Psi(\boldsymbol{z} + \tau(\boldsymbol{x} - \boldsymbol{z})) - \nabla \Psi(\boldsymbol{z}) \right),\ \boldsymbol{S}'(\boldsymbol{x} - \boldsymbol{z}) \right\rangle \mathrm{d}\tau \right| \\
&\quad \le \int_0^1 \left\| \boldsymbol{S}^{-1} \left(\nabla \Psi(\boldsymbol{z} + \tau(\boldsymbol{x} - \boldsymbol{z})) - \nabla \Psi(\boldsymbol{z}) \right) \right\|_2 \left\| \boldsymbol{S}'(\boldsymbol{x} - \boldsymbol{z}) \right\|_2 \mathrm{d}\tau \\
&\quad = \left\| \boldsymbol{S}'(\boldsymbol{x} - \boldsymbol{z}) \right\|_2 \int_0^1 \left\| \boldsymbol{S}^{-1} \left(\nabla \Psi(\boldsymbol{z} + \tau(\boldsymbol{x} - \boldsymbol{z})) - \nabla \Psi(\boldsymbol{z}) \right) \right\|_2 \mathrm{d}\tau \\
&\quad \le \left\| \boldsymbol{S}'(\boldsymbol{x} - \boldsymbol{z}) \right\|_2 \int_0^1 \left\| \tau \boldsymbol{S}'(\boldsymbol{x} - \boldsymbol{z}) \right\|_2 \mathrm{d}\tau = \left\| \boldsymbol{S}'(\boldsymbol{x} - \boldsymbol{z}) \right\|_2^2 \int_0^1 \tau \, \mathrm{d}\tau = \frac{1}{2} \left\| \boldsymbol{S}'(\boldsymbol{x} - \boldsymbol{z}) \right\|_2^2,
\end{aligned}
$$

using the $\boldsymbol{S}$-Lipschitz continuity and the **Cauchy-Schwarz inequality**.

Thus

$$
\begin{aligned}
\Psi(\boldsymbol{x}) - \Psi(\boldsymbol{z}) - \operatorname{real}\{\langle \nabla \Psi(\boldsymbol{z}),\ \boldsymbol{x} - \boldsymbol{z} \rangle\} &\le \frac{1}{2} \left\| \boldsymbol{S}'(\boldsymbol{x} - \boldsymbol{z}) \right\|_2^2 \\
\implies \Psi(\boldsymbol{x}) \le \Psi(\boldsymbol{z}) + \operatorname{real}\{\langle \nabla \Psi(\boldsymbol{z}),\ \boldsymbol{x} - \boldsymbol{z} \rangle\} + \frac{1}{2} \left\| \boldsymbol{S}'(\boldsymbol{x} - \boldsymbol{z}) \right\|_2^2 &.
\end{aligned}
$$

This holds for all $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{F}^N$ and now take $\boldsymbol{z} = \boldsymbol{x}_k$ to establish (4.5). □

A possible converse is explored in a HW problem.

The MM algorithm corresponding to the majorizer (4.5) has the following minimization step:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} \phi_k(\boldsymbol{x}) \text{ (from (4.3))}$$

$$\phi_k(\boldsymbol{x}) \triangleq \Psi(\boldsymbol{x}_k) + \text{real}\{\langle \nabla \Psi(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle\} + \frac{1}{2} \|\boldsymbol{S}'(\boldsymbol{x} - \boldsymbol{x}_k)\|_2^2 \text{ (from (4.5))}$$

$$\nabla \phi_k(\boldsymbol{x}) =$$

$$\boldsymbol{x}_{k+1} =$$

This MM algorithm minimization step is the same as:

A: GD B: PGD C: PSD D: PCG E: None of these **??**

Special cases

Intuition for twice differentiable cost functions with bounded curvature. Eqn. (4.5) holds if:

$$\nabla^2 \Psi(\boldsymbol{x}) \preceq \boldsymbol{S}\boldsymbol{S}', \ \forall \boldsymbol{x} \in \mathbb{R}^N.$$

If $\boldsymbol{S} = \sqrt{L}\boldsymbol{I}$ then

$$\Psi(\boldsymbol{x}) \le \phi_k(\boldsymbol{x}) = \Psi(\boldsymbol{x}_k) + \text{real}\{\langle \nabla \Psi(\boldsymbol{x}_k), \boldsymbol{x} - \boldsymbol{x}_k \rangle\} + \frac{L}{2} \|\boldsymbol{x} - \boldsymbol{x}_k\|_2^2 .$$

4.2 Applications / MM examples

MM methods for LRMC

Problem statement (for noisy data, see EECS 551 notes):

$$Y_{ij} = \begin{cases} X_{ij} + \varepsilon_{ij}, & (i,j) \in \Omega \\ 0, & \text{otherwise}, \end{cases} \quad \underbrace{M_{ij} \triangleq \begin{cases} 1, & (i,j) \in \Omega, \\ 0, & \text{otherwise}, \end{cases}}_{\text{mask}} \quad \underbrace{\tilde{M}_{i,j} = \begin{cases} 0, & (i,j) \in \Omega \\ 1, & (i,j) \notin \Omega, \end{cases}}_{\text{complement}}$$

for a sampling set $\Omega \subset \{1, \dots, M\} \times \{1, \dots, N\}$. Equivalently: $\tilde{\boldsymbol{M}} \triangleq \boldsymbol{1}_M \boldsymbol{1}_N' - \boldsymbol{M}$.

One possible LRMC formulation uses a (non-convex) **rank constraint**:

$$\hat{\boldsymbol{X}} = \underset{\boldsymbol{X} \,:\, \mathsf{rank}(\boldsymbol{X}) \le K}{\arg\min} \; q(\boldsymbol{X}), \quad q(\boldsymbol{X}) \triangleq \|\boldsymbol{M} \odot (\boldsymbol{X} - \boldsymbol{Y})\|_{\mathrm{F}}^2. \tag{4.6}$$

Now define the following function:

$$Q(\boldsymbol{X}; \boldsymbol{Z}) = \|\boldsymbol{X} - \boldsymbol{Z} + \boldsymbol{M} \odot (\boldsymbol{Z} - \boldsymbol{Y})\|_{\mathrm{F}}^2 = \left\| \boldsymbol{X} - (\tilde{\boldsymbol{M}} \odot \boldsymbol{Z} + \boldsymbol{M} \odot \boldsymbol{Y}) \right\|_{\mathrm{F}}^2. \tag{4.7}$$

This function is a **majorizer** of $q(\boldsymbol{X})$ because

$$q(\boldsymbol{X}) = Q(\boldsymbol{X}; \boldsymbol{X})$$
$$q(\boldsymbol{X}) \le Q(\boldsymbol{X}; \boldsymbol{Z}), \; \forall \boldsymbol{Z} \in \mathbb{F}^{M \times N}.$$

Proof: Verifying $Q(\boldsymbol{X}; \boldsymbol{X}) = q(\boldsymbol{X})$ is easy. (Read)

$$\begin{aligned} Q(\boldsymbol{X}; \boldsymbol{Z}) &= \|\boldsymbol{X} - \boldsymbol{Z} + \boldsymbol{M} \odot (\boldsymbol{Z} - \boldsymbol{Y})\|_{\mathrm{F}}^2 = \left\|(\tilde{\boldsymbol{M}} + \boldsymbol{M}) \odot (\boldsymbol{X} - \boldsymbol{Z}) + \boldsymbol{M} \odot (\boldsymbol{Z} - \boldsymbol{Y})\right\|_{\mathrm{F}}^2 \\ &= \left\|\tilde{\boldsymbol{M}} \odot (\boldsymbol{X} - \boldsymbol{Z}) + \boldsymbol{M} \odot (\boldsymbol{X} - \boldsymbol{Y})\right\|_{\mathrm{F}}^2 = \left\|\tilde{\boldsymbol{M}} \odot (\boldsymbol{X} - \boldsymbol{Z})\right\|_{\mathrm{F}}^2 + \|\boldsymbol{M} \odot (\boldsymbol{X} - \boldsymbol{Y})\|_{\mathrm{F}}^2 \\ &\geq \|\boldsymbol{M} \odot (\boldsymbol{X} - \boldsymbol{Y})\|_{\mathrm{F}}^2 = q(\boldsymbol{X}), \end{aligned}$$

where the 4th equality holds because $\tilde{\boldsymbol{M}} \odot \boldsymbol{M} = \boldsymbol{0}$. □

In the earlier notation, define $q_k(\boldsymbol{X}) \triangleq Q(\boldsymbol{X}, \boldsymbol{X}_k)$ at the kth iteration. Then the MM requirements hold:

$$\begin{aligned} q_k(\boldsymbol{X}_k) &= Q(\boldsymbol{X}_k, \boldsymbol{X}_k) = q(\boldsymbol{X}_k) \\ q_k(\boldsymbol{X}) &= Q(\boldsymbol{X}, \boldsymbol{X}_k) \geq q(\boldsymbol{X}), \ \forall \boldsymbol{X}. \end{aligned}$$

I originally designed this majorizer by making a 2nd-order Taylor expansion of $q(\boldsymbol{X})$ about $\boldsymbol{Z}$ and then using the fact that the elements of $\boldsymbol{M}$ are all 0 or 1. We'll see another application of that approach later in this chapter. Here, the proof above illustrates a simpler way to design the majorizer, using the following definition:

$$Q(\boldsymbol{X}, \boldsymbol{Z}) \triangleq q(\boldsymbol{X}) + \left\|\tilde{\boldsymbol{M}} \odot (\boldsymbol{X} - \boldsymbol{Z})\right\|_{\mathrm{F}}^2, \tag{4.8}$$

because the 2nd term is nonnegative and is zero when $\boldsymbol{X} = \boldsymbol{Z}$, per (4.4).

Now we use the **quadratic majorizer** (4.7) = (4.8) to develop a MM algorithm for LRMC that is practical for problems that are not too large.

LRMC by iterative low-rank approximation

Consider LRMC using the rank constraint (4.6):

$$\hat{\boldsymbol{X}} = \underset{\boldsymbol{X} \,:\, \mathsf{rank}(\boldsymbol{X}) \le K}{\arg\min} \left\| \boldsymbol{M} \odot (\boldsymbol{X} - \boldsymbol{Y}) \right\|_{\mathrm{F}}^2.$$

The MM algorithm update for this formulation is simply

$$\begin{aligned}
\boldsymbol{X}_{k+1} &= \underset{\boldsymbol{X} \,:\, \mathsf{rank}(\boldsymbol{X}) \le K}{\arg\min} \; Q(\boldsymbol{X}; \boldsymbol{X}_k) \\
&= \underset{\boldsymbol{X} \,:\, \mathsf{rank}(\boldsymbol{X}) \le K}{\arg\min} \left\| \boldsymbol{X} - (\tilde{\boldsymbol{M}} \odot \boldsymbol{X}_k + \boldsymbol{M} \odot \boldsymbol{Y}) \right\|_{\mathrm{F}}^2 \\
&= \underset{\boldsymbol{X} \,:\, \mathsf{rank}(\boldsymbol{X}) \le K}{\arg\min} \left\| \boldsymbol{X} - \tilde{\boldsymbol{X}}_k \right\|_{\mathrm{F}}^2, \quad \tilde{\boldsymbol{X}}_k \triangleq \tilde{\boldsymbol{M}} \odot \boldsymbol{X}_k + \boldsymbol{M} \odot \boldsymbol{Y} = \begin{cases} Y_{i,j}, & (i,j) \in \Omega \\ [\boldsymbol{X}_k]_{i,j}, & (i,j) \notin \Omega. \end{cases}
\end{aligned}$$

This algorithm alternates between two steps:

- Take the current guess $\boldsymbol{X}_k$ and replace all the values at sampled locations with the measurements from $\boldsymbol{Y}$ to get $\tilde{\boldsymbol{X}}_k$. (As mentioned earlier, this can be done "in place" using `X[M] .= Y[M]`)
- Perform low-rank (rank at most K) approximation (using **SVD**) to $\tilde{\boldsymbol{X}}_k$ to get the next iterate $\boldsymbol{X}_{k+1}$.

Because this is a MM method, we know it decreases the Frobenius norm cost function monotonically.

(There is still no guarantee here of convergence of $\{\boldsymbol{X}_k\}$ to a global minimizer because the rank constraint set is nonconvex. For convex formulations see [4].)

LASSO / sparse regression / compressed sensing

These applications all involve the following **composite** cost function:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \Psi(\boldsymbol{x}), \quad \Psi(\boldsymbol{x}) = f(\boldsymbol{x}) + \beta \left\|\boldsymbol{x}\right\|_1, \quad f(\boldsymbol{x}) = \frac{1}{2} \left\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\right\|_2^2. \tag{4.9}$$

There are simple non-iterative solutions when $\boldsymbol{A}$ is identity, diagonal, or unitary, but not otherwise.

First draw inspiration from (4.4) to define a **quadric majorizer** for $f(\boldsymbol{x})$:

$$f_k(\boldsymbol{x}) \triangleq \tag{4.10}$$

$$= \tag{4.11}$$

$$=$$

$$=$$

by **completing the square**, where c_1, c_2 are constants independent of $\boldsymbol{x}$.
Here we require $\boldsymbol{H} \succeq \boldsymbol{0}$ and $\boldsymbol{A}'\boldsymbol{A} + \boldsymbol{H} \succ \boldsymbol{0}$, but we do not insist that $\boldsymbol{H} \succ \boldsymbol{0}$.

ISTA: iterative soft thresholding algorithm

$$f_k(\boldsymbol{x}) =$$

Thus a majorizer for the original cost function is

$$\begin{aligned} \phi_k(\boldsymbol{x}) &= \\ &= \end{aligned} \tag{4.12}$$

The minimization step is the **iterative soft thresholding algorithm** (**ISTA**) [5]:

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} \phi_k(\boldsymbol{x}) = \tag{4.13}$$

It is also called the **proximal gradient method** (**PGM**), especially for more general regularizers.
This MM algorithm is remarkably simple, especially given that that 1-norm is non-differentiable.

A serious drawback of ISTA is its slow convergence $O(1/k)$ and it needs $\|\boldsymbol{A}\|_2$.

The threshold above is:
A: β B: β/L C: βL D: βL^2 E: None **??**

PGM with diagonal majorizers

Recall from HW1#10:

(4.14)

With this choice of $\boldsymbol{H}$, the majorizer becomes

$$\phi_k(\boldsymbol{x}) =$$

and the MM algorithm becomes

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} \phi_k(\boldsymbol{x}) = \qquad (4.15)$$

where $\boldsymbol{D} = \mathsf{diag}\{\boldsymbol{d}\}$ and $\oslash$ denotes `./` element-wise division. So the jth element of the gradient step is thresholded by β/d_j. No Lipschitz constant or operator norm is needed.

It still has slow $O(1/k)$ convergence rate, but there is a fast version called FISTA [6] and an even faster version called the **proximal optimized gradient method** (**POGM**) [7] that both have $O(1/k^2)$ bounds.

A HW problem will compare ISTA / FISTA / POGM with the `lasso_cls` constrained least-squares formulation from Ch. 1.

Convexity approach to separability

All of the preceding examples used **quadratic majorizers** for the data term. Now we turn to non-quadratic examples.

Many of cost functions have data-fit terms of this form:

$$f(\boldsymbol{x}) =$$

where each function h_i is **convex** (but not necessarily smooth).

Examples:
- $h_i(t) = \frac{1}{2} |t - y_i|^2$
- $h_i(t) = \max(1 - t, 0)$ (**hinge loss**, the tightest convex upper bound on the 0-1 loss)
- $h_i(t) = \mathrm{e}^{-t}$ (**exponential loss**)
- $h_i(t) = t - y_i \log(t)$ (Poisson negative log-likelihood)

The commonality of all of these cost function components is convexity (only), *i.e.*,

(4.16)

The importance of this inequality is that the RHS is **additively separable**.

Here is the key technique for designing a **separable** majorizer using convexity:

$$[\boldsymbol{A}\boldsymbol{x}]_i =$$

$$h_i([\boldsymbol{A}\boldsymbol{x}]_i) =$$

$$\leq$$

$$\implies \sum_{i=1}^{M} h_i([\boldsymbol{A}\boldsymbol{x}]_i) \leq$$

Note that $\sum_{j=1}^{N} f_j^{(k)}(x_j^{(k)}) = \sum_{j=1}^{N} \sum_{i=1}^{M} \alpha_{ij}\, h_i([\boldsymbol{A}\boldsymbol{x}_k]_i) = \sum_{i=1}^{M} h_i([\boldsymbol{A}\boldsymbol{x}_k]_i)$, so (4.1) holds.

Now consider a cost function like

$$\Psi(\boldsymbol{x}) = \sum_{i=1}^{M} h_i([\boldsymbol{A}\boldsymbol{x}]_i) + \beta \left\|\boldsymbol{x}\right\|_1 = \sum_{i=1}^{M} h_i([\boldsymbol{A}\boldsymbol{x}]_i) + \beta \sum_{j=1}^{N} |x_j| \,.$$

The above inequalities establish the following separable majorizer:

$$\phi_k(\boldsymbol{x}) = \qquad (4.17)$$

The (parallelizable!) minimization step of the MM algorithm for this majorizer is

$$\boldsymbol{x}_{k+1} = \arg\min_{\boldsymbol{x}} \phi_k(\boldsymbol{x}), \quad x_j^{(k+1)} =$$

Example. Consider again the LASSO problem (4.9), where $h_i(t) = \frac{1}{2}|t - y_i|^2$.
The separable majorizer (4.17) is:

$$\begin{aligned}
\phi_j^{(k)}(x_j) &= f_j^{(k)}(x_j) + \beta |x_j| = \sum_{i=1}^M \alpha_{ij} h_i\left(\frac{a_{ij}}{\alpha_{ij}}(x_j - x_j^{(k)}) + [\boldsymbol{A}\boldsymbol{x}_k]_i\right) + \beta |x_j| \\
&= \sum_{i=1}^M \alpha_{ij} \frac{1}{2} \left| \frac{a_{ij}}{\alpha_{ij}}(x_j - x_j^{(k)}) + [\boldsymbol{A}\boldsymbol{x}_k]_i - y_i \right|^2 + \beta |x_j| \\
&= \sum_{i=1}^M \frac{1}{\alpha_{ij}} \frac{1}{2} \left| a_{ij}(x_j - x_j^{(k)}) + \alpha_{ij}([\boldsymbol{A}\boldsymbol{x}_k]_i - y_i) \right|^2 + \beta |x_j| \\
&= \frac{1}{2} d_j \left| x_j - \left[\boldsymbol{x}_k - \boldsymbol{D}^{-1}\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{y})\right]_j \right|^2 + \beta |x_j| + c_3, \quad \boldsymbol{D} = \mathsf{diag}\{d_j\}, \quad d_j \triangleq \sum_{i=1}^M |a_{ij}|^2 / \alpha_{ij},
\end{aligned}$$

where c_3 is a constant independent of $\boldsymbol{x}$.

So the minimization step of the MM algorithm is simply soft thresholding:

$$\boldsymbol{x}_{k+1} = \mathrm{soft}\big(\boldsymbol{x}_k - \boldsymbol{D}^{-1}\boldsymbol{A}'(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{y}), \beta \oslash \boldsymbol{d}\big).$$

To finalize the algorithm we must specify α_{ij} factors satisfying (4.16). A natural choice, *cf.* (4.14), is:

$$\alpha_{ij} = \frac{|a_{ij}|}{\sum_{k=1}^N |a_{ik}|} \implies d_j = \sum_{i=1}^M \frac{|a_{ij}|^2}{\alpha_{ij}} = \sum_{i=1}^M \frac{|a_{ij}|^2}{|a_{ij}| / \sum_{k=1}^N |a_{ik}|} = \sum_{i=1}^M |a_{ij}| \left(\sum_{k=1}^N |a_{ik}| \right) = [|\boldsymbol{A}|' |\boldsymbol{A}| \boldsymbol{1}]_j.$$

If $\beta = 0$, this MM algorithm is the same as:
A: GD B: PGD C: PSD D: PCG E: None of these **??**

If $\beta > 0$, this MM algorithm is the same as:
A: GD B: PGD C: ISTA/PGM (4.13) D: ISTA/PGM (4.15) E: None of these **??**

Example. Exponential loss for binary classifier design $h_i(t) = \mathrm{e}^{-t}$. (Read)
For this application the separable majorizer (4.17) is

$$\phi_j^{(k)}(x_j) = f_j^{(k)}(x_j) + \beta\,|x_j| = \sum_{i=1}^{M} \alpha_{ij}\, h_i\left(\frac{a_{ij}}{\alpha_{ij}}(x_j - x_j^{(k)}) + [\boldsymbol{A}\boldsymbol{x}_k]_i\right) + \beta\,|x_j|$$

$$= \sum_{i=1}^{M} \alpha_{ij} \exp\left(-\left(\frac{a_{ij}}{\alpha_{ij}}(x_j - x_j^{(k)}) + [\boldsymbol{A}\boldsymbol{x}_k]_i\right)\right) + \beta\,|x_j|\,.$$

There is no closed-form expression for the MM algorithm update, but one can perform parallel 1D minimization for each coordinate $j = 1, \ldots, N$:

$$x_j^{(k+1)} = \arg\min_{x_j} \phi_j^{(k)}(x_j) = \arg\min_{x_j} \sum_{i=1}^{M} \alpha_{ij} \exp\left(-\left(\frac{a_{ij}}{\alpha_{ij}}(x_j - x_j^{(k)}) + [\boldsymbol{A}\boldsymbol{x}_k]_i\right)\right) + \beta\,|x_j|\,.$$

The same principle applies to the **hinge** loss function. However, in both cases the 1D minimization itself seems challenging: the exponential does not have a Lipschitz derivative and the hinge is not differentiable.

Poisson measurements (MLEM)

For the Poisson noise problem with $h_i(t) = t - y_i \log(t)$, we must have $a_{ij} \geq 0$ and we use the convexity inequality with a slightly different approach:

$$[\boldsymbol{A}\boldsymbol{x}]_i = \sum_{j=1}^{N} a_{ij} x_j =$$

$$h_i([\boldsymbol{A}\boldsymbol{x}]_i) = h_i\left(\sum_{j=1}^{N} \left(\frac{a_{ij} x_j^{(k)}}{[\boldsymbol{A}\boldsymbol{x}_k]_i} \right) \frac{x_j}{x_j^{(k)}} [\boldsymbol{A}\boldsymbol{x}_k]_i \right) \leq \sum_{j=1}^{N} \left(\frac{a_{ij} x_j^{(k)}}{[\boldsymbol{A}\boldsymbol{x}_k]_i} \right) h_i\left(\frac{x_j}{x_j^{(k)}} [\boldsymbol{A}\boldsymbol{x}_k]_i \right)$$

$$\implies \sum_{i=1}^{M} h_i([\boldsymbol{A}\boldsymbol{x}]_i) \leq \sum_{j=1}^{N} f_j^{(k)}(x_j), \quad f_j^{(k)}(x_j) \triangleq \sum_{i=1}^{M} \left(\frac{a_{ij} x_j^{(k)}}{[\boldsymbol{A}\boldsymbol{x}_k]_i} \right) h_i\left(\frac{x_j}{x_j^{(k)}} [\boldsymbol{A}\boldsymbol{x}_k]_i \right)$$

Here we are essentially using iteration-dependent convexity parameters:

$$\alpha_{ij} =$$

Again considering 1-norm regularization, the separable majorizer is

$$\phi_k(\boldsymbol{x}) = \sum_{j=1}^{N} \phi_j^{(k)}(x_j), \quad \phi_j^{(k)}(x_j) = f_j^{(k)}(x_j) + \beta \, |x_j| \, .$$

Here we need $x_j \geq 0$ so $|x_j| = x_j$.

Differentiating to find the minimizer over x_j:

$$\begin{aligned}
0 = \frac{\partial}{\partial x_j} \phi_j^{(k)}(x_j) &= \sum_{i=1}^{M} a_{ij} \, \dot{h}_i \left(\frac{x_j}{x_j^{(k)}} [\boldsymbol{A}\boldsymbol{x}_k]_i \right) + \beta = \sum_{i=1}^{M} a_{ij} \left(1 - \frac{x_j^{(k)} y_i}{x_j [\boldsymbol{A}\boldsymbol{x}_k]_i} \right) + \beta \\
&= \sum_{i=1}^{M} a_{ij} - \frac{x_j^{(k)}}{x_j} \sum_{i=1}^{M} a_{ij} \frac{y_i}{[\boldsymbol{A}\boldsymbol{x}_k]_i} + \beta \\
\implies x_j^{(k+1)} &= \frac{x_j^{(k)}}{\sum_{i=1}^{M} a_{ij} + \beta} \sum_{i=1}^{M} a_{ij} \frac{y_i}{[\boldsymbol{A}\boldsymbol{x}_k]_i}.
\end{aligned}$$

The version with $\beta = 0$ is used in clinical PET and SPECT systems daily.

The 1-norm seems to be of much less help in enforcing sparsity here compared to the case of the 2-norm data term [8].

Line search using Huber's majorizer

Recall that for "inverse problem" cost functions of the form $\Psi(\boldsymbol{x}) = \sum_{j=1}^{J} f_j(\boldsymbol{B}_j \boldsymbol{x})$ where each component function f_j is convex and smooth, we used GD to solve the 1D **line-search** minimization problem:

$$\alpha_k = \arg\min_{\alpha} h_k(\alpha), \quad h_k(\alpha) = \Psi(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k) = \sum_{j=1}^{J} f_j(\boldsymbol{u}_j + \alpha \boldsymbol{v}_j), \quad \boldsymbol{u}_j \triangleq \boldsymbol{B}_j \boldsymbol{x}_k, \quad \boldsymbol{v}_j \triangleq \boldsymbol{B}_j \boldsymbol{d}_k.$$

To apply GD for the line search, in HW you found a Lipschitz constant of the derivative of $h_k(\alpha)$ to be

$$L_{\dot{h}_k} = \sum_{j=1}^{J} L_{\dot{f}_j} \|\boldsymbol{v}_j\|_2^2 .$$

That Lipschitz constant leads to correct but undesirably small step sizes for many functions of interest, thereby possibly requiring excess numbers of inner line search iterations.

We focus on component functions of the form (note the JULIA dot):

$$f_j(\boldsymbol{v}) = \mathbf{1}' \, \psi_j \, .(\boldsymbol{v}),$$

for which

$$L_{\dot{f}_j} = L_{\dot{\psi}}.$$

If ψ has a Lipschitz continuous derivative, then specializing (4.5):

$$ \qquad . \tag{4.18} $$

The GD-based line-search used in HW was essentially a MM method using this quadratic majorizer. I call this the **maximum curvature** majorizer because if

$$\left|\dot{\psi}(t) - \dot{\psi}(s)\right| \leq L_{\dot{\psi}} \left|t - s\right|$$

and ψ is twice differentiable at t, then

$$\ddot{\psi}(t) = $$

So the Lipschitz constant for the derivative of a function is a (tight) bound for its maximum second derivative.

Often we can find quadratic majorizers for ψ with lower curvature than $L_{\dot{\psi}}$.

Lower curvature majorizers lead to larger step sizes and hence convergence in fewer iterations.

Huber's majorizer

Theorem (Huber, 1981 [9, p. 184], [2, Ch. 14]). Suppose $\psi : \mathbb{R} \to \mathbb{R}$ satisfies the following conditions

$$\begin{aligned}
&\psi(t) \text{ is differentiable,} \\
&\psi(t) = \psi(-t), \ \forall t \text{ (symmetry),} \\
&\omega_\psi(t) \triangleq \dot{\psi}(t) \,/t \text{ is bounded and monotone nonincreasing for } t > 0.
\end{aligned} \tag{4.19}$$

Then the parabola function defined by

$$\begin{aligned}
q(t;s) &\triangleq \\
&= \left(\psi(s) - \frac{\omega_\psi(s)}{2} s^2 \right) + \frac{\omega_\psi(s)}{2} t^2,
\end{aligned} \tag{4.20}$$

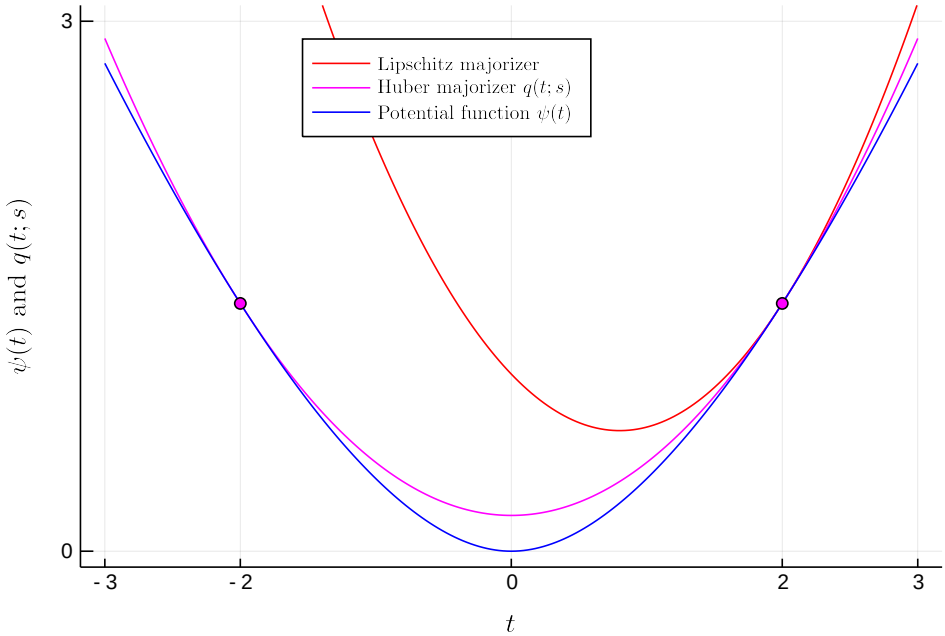
is a majorizer for ψ, *i.e.*, it satisfies conditions analogous to (4.1) and (4.2), namely

$$\begin{aligned}
\psi(s) &= q(s;s) \\
\psi(t) &\le q(t;s), \ \forall s,t.
\end{aligned}$$

The curvature ω_ψ is optimal in the sense of being the smallest value that ensures those requirements.

Challenge. Prove or disprove: Under Huber's conditions, $\dot{\psi}$ is Lipschitz continuous, and $L_{\dot{\psi}} = \omega_\psi(0)$.

Example. This figure illustrates the Huber quadratic majorizer (4.20) and the Lipschitz quadratic majorizer (4.18) for the **Fair potential** with $\delta = 3$ at $s = 2$. Clearly the Huber majorizer is tighter, allowing larger step sizes.



MM line search using Huber's majorizer

To realize these larger steps, one uses a MM approach instead of GD for the line search.
Dropping the iteration k subscript to simplify notation, the line-search problem is

$$\alpha_* = \arg\min_{\alpha} h(\alpha), \quad h(\alpha) = \Psi(\boldsymbol{x} + \alpha \boldsymbol{d}) = \sum_{j=1}^{J} f_j(\boldsymbol{u}_j + \alpha \boldsymbol{v}_j), \qquad \begin{array}{l} \boldsymbol{u}_j \triangleq \boldsymbol{B}_j \boldsymbol{x}, \\ \boldsymbol{v}_j \triangleq \boldsymbol{B}_j \boldsymbol{d}, \end{array} \qquad f_j(\boldsymbol{v}) = \mathbf{1}' \, \psi_j \, .(\boldsymbol{v}).$$

(Dropping k here also avoids confusion with the inner MM iteration for the line search.)

Assuming each ψ_j satisfies Huber's conditions, the following quadratic function is a majorizer for $h(\alpha)$ for the nth inner iteration:

$$\phi_n(\alpha) \triangleq q(\alpha; \alpha_n) = \sum_{j=1}^{J} q_j(\alpha; \alpha_n)$$

$$\begin{aligned} q_j(\alpha; \alpha_n) &= \mathbf{1}' \left(\psi_j \, .(\boldsymbol{u}_j + \alpha_n \boldsymbol{v}_j) + \dot{\psi}_j \, .(\boldsymbol{u}_j + \alpha_n \boldsymbol{v}_j) \odot \boldsymbol{v}_j (\alpha - \alpha_n) + \frac{1}{2} \omega_j \, .(\boldsymbol{u}_j + \alpha_n \boldsymbol{v}_j) \odot \boldsymbol{v}_j \, \texttt{.\^{}2} \, (\alpha - \alpha_n)^2 \right) \\ &= c_0 + c_1(\alpha_n)(\alpha - \alpha_n) + \frac{1}{2} c_2(\alpha_n)(\alpha - \alpha_n)^2 \\ c_0 &= h(\alpha_n) \end{aligned}$$

$$c_1(\alpha_n) = \sum_{j=1}^{J} \text{real}\Big\{ \boldsymbol{v}_j' \, \dot{\psi}_j \, .(\boldsymbol{u}_j + \alpha_n \boldsymbol{v}_j) \Big\} = \dot{h}(\alpha_n)$$

$$c_2(\alpha_n) = \sum_{j=1}^{J} (\boldsymbol{v}_j \texttt{ .^2 })' \, \omega_j \, .(\boldsymbol{u}_j + \alpha_n \boldsymbol{v}_j) = \sum_{j=1}^{J} \boldsymbol{v}_j' \, \text{diag}\{\omega_j \, .(\boldsymbol{u}_j + \alpha_n \boldsymbol{v}_j)\} \, \boldsymbol{v}_j \le \sum_{j=1}^{J} L_{\dot{\psi}_j} \, \|\boldsymbol{v}_j\|_2^2 \, .$$

Differentiating $q(\alpha; \alpha_n)$ w.r.t. α and setting to zero leads to the following MM update:

$$\alpha_{n+1} = \arg\min_{\alpha} \phi_n(\alpha) =$$

A HW problem will compare the speed of this to the GD-based line search used previously.

1D optimization using Huber's majorizer

Facts:

$$\arg\min_x \sum_{n=1}^N \frac{1}{2} |x - x_n|^2 = \frac{1}{N} \sum_{n=1}^N x_n \qquad \text{(sample mean)}$$

$$\arg\min_x \sum_{n=1}^N |x - x_n| = \text{median}(x_1, \ldots, x_N) \qquad \text{(sample median)}$$

What about more general 1D case:

$$\hat{x} = \arg\min_x f(x), \quad f(x) = \sum_{n=1}^N \psi(x - x_n) \,.$$

In general there is no closed-form solution for $\hat{x}$. If ψ satisfies Huber's conditions then we can form a quadratic majorizer:

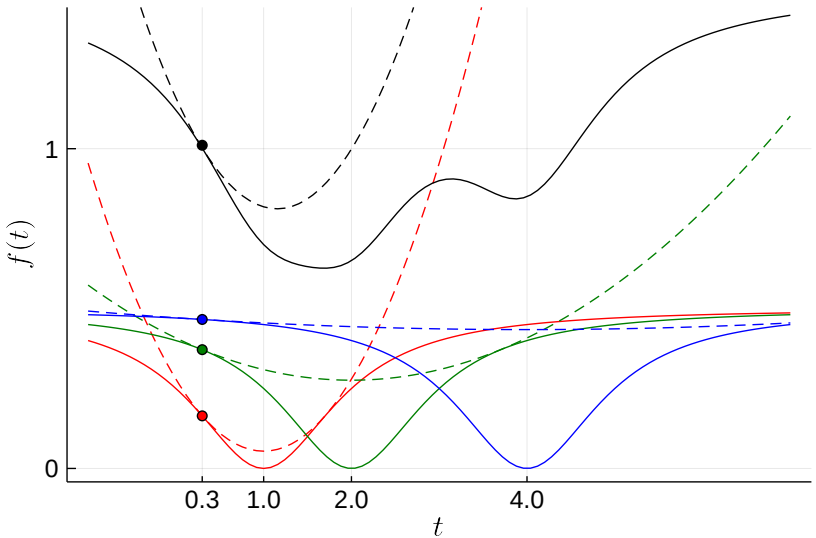
$$f(x) \le \phi^{(t)}(x) = \sum_{n=1}^N \psi\big(x^{(t)} - x_n\big) + \dot{\psi}\big(x^{(t)} - x_n\big)(x - x^{(t)}) + \omega_\psi\big(x^{(t)} - x_n\big) \frac{1}{2}(x - x^{(t)})^2$$

and update by zeroing its derivative:

$$x^{(t+1)} = x^{(t)} - \frac{\dot{\phi}^{(t)}(x^{(t)})}{\ddot{\phi}^{(t)}(x^{(t)})} = x^{(t)} - \frac{\sum_{n=1}^N \dot{\psi}\big(x^{(t)} - x_n\big)}{\sum_{n=1}^N \omega_\psi(x^{(t)} - x_n)}.$$

This update will decrease $f(x)$ monotonically, whereas Newton's method has no such guarantee in general.

Example. Consider the Geman & McClure potential $\psi(t) = \frac{t^2/2}{1+t^2}$ for which $\omega_\psi(t) = \frac{1}{(1+t^2)^2}$, and the case where $x_1 = 1$, $x_2 = 2$, $x_3 = 4$. Here the Huber quadratic majorizer works as needed, whereas the Newton parabola would fail.

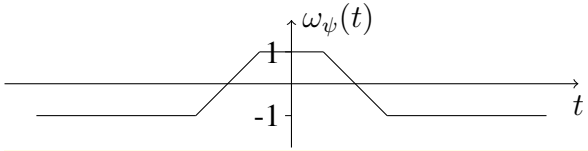


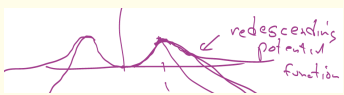
Huber's conditions and uniqueness

Do Huber's 3 conditions (4.19) ensure ψ has a unique minimizer?
ψ differentiable, symmetric and $\omega_\psi(t) = \dot{\psi}(t)\,/t$ is bounded and nonincreasing for $t > 0$.

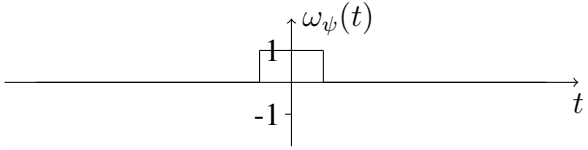
Example.
Consider this weighting function and note that

$$\psi(t) = \int_0^t \dot{\psi}(s)\,\mathrm{d}s = \int_0^t s\,\omega_\psi(s)\,\mathrm{d}s\,.$$





Now suppose we add a 4th condition:
$\omega_\psi \geq 0$
Now we have that $\dot{\psi}(t) \geq 0\ \forall t \geq 0$,
so $\psi(t)$ is non-decreasing for $t \geq 0$.



Huber hinge function

The **hinge loss** function is not differentiable, making it unsuitable for gradient-based methods.
The (continuously differentiable) **Huber hinge** loss function for classifier design is defined, for $\delta > 0$, as [10]:

$$h(t;\delta) = \begin{cases} 1 - t - \delta/2, & t \le 1 - \delta \\ \frac{1}{2\delta}(t-1)^2, & 1 - \delta \le t \le 1 \\ 0, & 1 \le t. \end{cases}$$

The Lipschitz constant of the derivative of this function is:
A: $1/\delta^2$ B: $1/\delta$ C: 1 D: δ E: δ^2 **??**

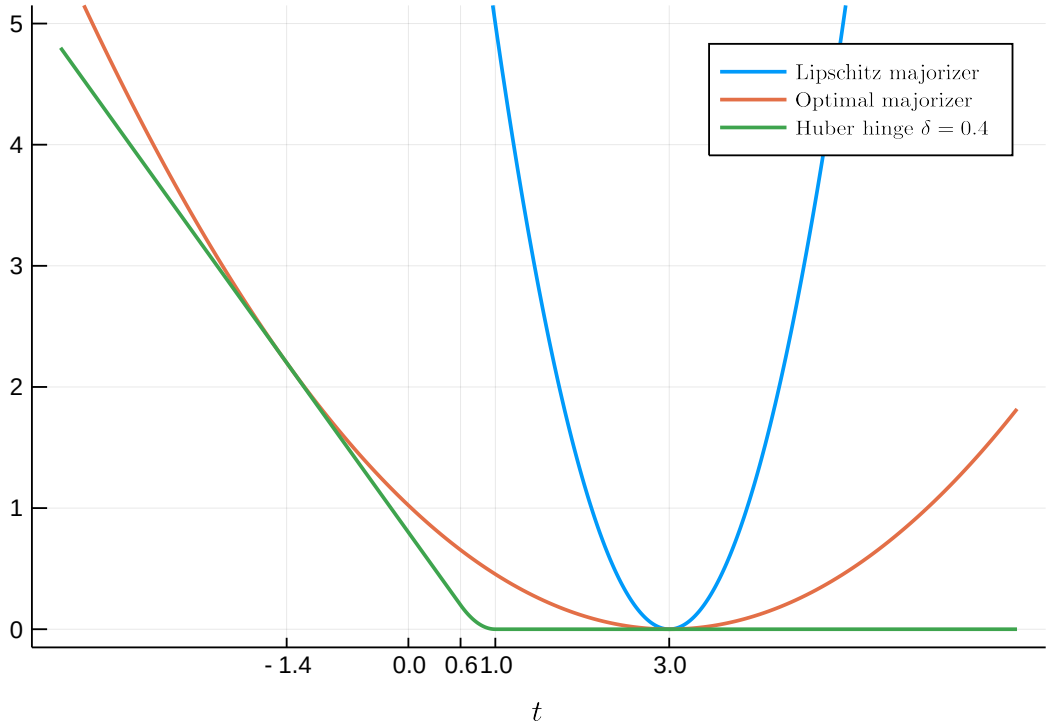
A quadratic majorizer with this curvature is used in [10, eqn. (34)] to train a **support-vector machine** (**SVM**).

For $s > 1$, find the quadratic function having *optimal*, *i.e.*, smallest possible curvature:

$$q(t;s) =$$

What is $c(s)$? (group)

Example. The following figure illustrates the case where $\delta = 2/5$ and $s = 3$.



If $1 - \delta \le s \le 1$ then the curvature of the optimal quadratic majorizer is?
A: $1/\delta^2$ B: $1/\delta$ C: 1 D: δ E: δ^2 **??**

If $1 - \delta \le s \le 1$ then the vertex of the optimal quadratic majorizer is?
A: 0 B: s C: 1 D: $1/s$ E: None of these **??**

A HW problem considers the case $s < 1 - \delta$.

The "best quadratic majorizer" for the (nondifferentiable) case $\delta = 0$ (the ordinary hinge function) is given in [11, eqn. (9)]. They call it "sharp" quadratic majorization. However, the expression there is invalid at the hinge corner point, so the authors propose an ad hoc modification that destroys the guarantee of the monotone descent property of MM methods, so is quite undesirable. Using $\delta > 0$ gives us the Huber hinge for which our quadratic majorizer is optimal.

An alternative to the Huber hinge is the "squared hinge loss" function [12] given by $h(t) = (\max(1 - t, 0))^2$. This function is convex and continuously differentiable and has a Lipschitz continuous derivative. However it increases unnecessarily rapidly (quadratically) for negative values, whereas the Huber hinge increases linearly.

Acceleration methods

For unconstrained problems, one acceleration method that retains monotonicity is the over-relaxation approach of [13]. In the MM setting, one first makes a standard MM update:

$$\tilde{\boldsymbol{x}}_{t+1} \triangleq \arg\min_{\boldsymbol{x}} \phi_t(\boldsymbol{x})$$

and then takes an extrapolated step

$$\boldsymbol{x}_{t+1} \triangleq \tilde{\boldsymbol{x}}_{t+1} + \alpha_t(\tilde{\boldsymbol{x}}^{(t+1)} - \boldsymbol{x}_t)$$

where the step size α_t is chosen to ensure that

$$\phi_t(\boldsymbol{x}_{t+1}) \leq \phi_t(\boldsymbol{x}_t),$$

which in turn ensures that $\Psi(\boldsymbol{x}_t)$ decreases monotonically. This method is viable when evaluating the majorizer ϕ_t is less expensive than a line search on Ψ.

4.3 Summary

This chapter has touched on a few of the many methods for designing MM optimization algorithms, focusing on quadratic majorizers and majorizers based on convexity.

One can combine the methods. For example in transmission tomography the data-fit term involves $h_i(t) = (b_i \exp(-t) + r_i) - y_i \log(b_i \exp(-t) + r_i)$ which is nonconvex. Nevertheless, one can design a quadratic majorizer for it [14] and then make a separable quadratic majorizer of that [15] to derive a simple MM algorithm.

For generalizations of MM, see the "relatively smooth" approach of [16] and the generalized MM (G-MM) approach, see [17].

Bibliography

[1] K. Lange. *MM optimization algorithms*. Soc. Indust. Appl. Math., 2016 (cit. on p. 4.2).

[2] J. A. Fessler. *Image reconstruction: Algorithms and analysis*. Book in preparation. ., 2006 (cit. on pp. 4.2, 4.25).

[3] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer, 2004 (cit. on p. 4.7).

[4] C. Grussler and P. Giselsson. "Low-rank inducing norms with optimality interpretations". In: *SIAM J. Optim.* 28.4 (Jan. 2018), 3057–78 (cit. on p. 4.12).

[5] I. Daubechies, M. Defrise, and C. De Mol. "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". In: *Comm. Pure Appl. Math.* 57.11 (Nov. 2004), 1413–57 (cit. on p. 4.14).

[6] A. Beck and M. Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM J. Imaging Sci.* 2.1 (2009), 183–202 (cit. on p. 4.15).

[7] A. B. Taylor, J. M. Hendrickx, and Francois Glineur. "Exact worst-case performance of first-order methods for composite convex optimization". In: *SIAM J. Optim.* 27.3 (Jan. 2017), 1283–313 (cit. on p. 4.15).

[8] D. J. Lingenfelter, J. A. Fessler, and Z. He. "Sparsity regularization for image reconstruction with Poisson data". In: *Proc. SPIE 7246 Computational Imaging VII*. 2009, 72460F (cit. on p. 4.22).

[9] P. J. Huber. *Robust statistics*. New York: Wiley, 1981 (cit. on p. 4.25).

[10] P. J. F. Groenen, G. Nalbantov, and J. C. Bioch. "SVM-Maj: a majorization approach to linear support vector machines with different hinge errors". In: *Advances in Data Analysis and Classification* 2.1 (Apr. 2008), 17–43 (cit. on p. 4.32).

[11] J. de Leeuw and K. Lange. "Sharp quadratic majorization in one dimension". In: *Comp. Stat. Data Anal.* 53.7 (May 2009), 2471–84 (cit. on p. 4.34).

[12] Z. Zhang, D. Liu, G. Dai, and M. I. Jordan. "Coherence functions with applications in large-margin classification methods". In: *J. Mach. Learning Res.* 13 (Sept. 2012), 2705–34 (cit. on p. 4.34).

[13] Y. Yu. "Monotonically overrelaxed EM algorithms". In: *J. Computational and Graphical Stat.* 21.2 (2012), 518–37 (cit. on p. 4.35).

[14] H. Erdogan and J. A. Fessler. "Monotonic algorithms for transmission tomography". In: *IEEE Trans. Med. Imag.* 18.9 (Sept. 1999), 801–14 (cit. on p. 4.36).

[15] H. Erdogan and J. A. Fessler. "Ordered subsets algorithms for transmission tomography". In: *Phys. Med. Biol.* 44.11 (Nov. 1999), 2835–51 (cit. on p. 4.36).

[16] H. Lu, R. M. Freund, and Y. Nesterov. “Relatively smooth convex optimization by first-order methods, and applications”. In: *SIAM J. Optim.* 28.1 (Jan. 2018), 333–54 (cit. on p. 4.36).

[17] S. N. Parizi, K. He, S. Sclaroff, and P. Felzenszwalb. “Generalized majorization-minimization”. In: *Proc. Intl. Conf. Mach. Learn*. Vol. 97. 2019, 5022–31 (cit. on p. 4.36).