Homework 3

Large-scale optimization

Jens Sjölund Sebastian Mair

Deadline: November 17th, 2023 at 17.00

Problem 7.1

Prove that if $\Omega = \{x \mid \|x\| \leq 1\}$, then

$$P_\Omega(x) = \begin{cases} x, & \text{if } \|x\| \leq 1, \\ \frac{x}{\|x\|}, & \text{otherwise.} \end{cases}$$

Problem 7.4

Let $c \in \mathbb{R}^n$ be a constant and $x \in \mathbb{R}^n$ be a variable. Consider the constrained optimization problem

$$\min_{x \in \Omega} \; c^\mathsf{T} x,$$

where Ω is either of the following sets:

(a) the unit ball $\{x \mid \|x\|_2 \leq 1\}$,

(b) the unit simplex $\{x \mid x \geq 0, \; \sum_{i=1}^n x_i = 1\}$, and

(c) a box $\{x \mid 0 \leq x_i \leq 1, \; i = 1, 2, \ldots, n\}$.

Work on the following tasks.

(i) Characterize the solution(s) $x^\star$ in terms of c for every Ω.

(ii) Implement the projected gradient method.

(iii) Visualize the optimization path for two-dimensional problems and initialize your optimization variable with a constant, i.e., $(4, 2)$. Experiment with various choices of c. Discuss your results.

Problem 8.5

For the following norm functions f over the vector space $\mathbb{R}^n$, find $\partial f(x)$ and $f'(x; v)$ for all x and v.

(a) The ℓ_1-norm $f(x) = \|x\|_1$.

(b) The ℓ_2-norm $f(x) = \|x\|_2$.

(c) The ℓ_∞-norm $f(x) = \|x\|_\infty$.

Problem ℓ_1-regularized linear regression

Consider an ℓ_1-regularized linear regression model and the following optimization problem

$$\min_x \frac{1}{2}\sum_{i=1}^{N}(b_i - a_i^\mathsf{T} x)^2 + \lambda\|x\|_1$$

with optimization variable $x \in \mathbb{R}^n$ and regularization strength $\lambda > 0$. The data set consists of N pairs (a_i, b_i), where $a_i \in \mathbb{R}^n$ is a feature vector and $b_i \in \mathbb{R}$ is the corresponding target. Specifically, we use the UCI naval[1] data set with "GT Compressor decay state coefficient" as the target.
Work on the following tasks.

(i) Characterize the individual parts of the objective function (convexity, smoothness).

(ii) Optimize the problem stated above using the subgradient method and the proximal-gradient method. Justify your choices.

ADMM for linear programming

Consider a linear programming problem of the form

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c^\mathsf{T} x \\ \text{subject to} \quad & Ax \le b \\ & \ell \le x \le u \end{aligned}$$

over the variables $x \in \mathbb{R}^n$, where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $\ell \in (\mathbb{R}\cup\{-\infty\})^n$, and $u \in (\mathbb{R}\cup\{\infty\})^n$.

(i) Find convex sets Ω_1, Ω_2, and Ω_3 such that the linear programming problem above can be written as

$$\begin{aligned} \underset{x,\tilde{x},y,\tilde{y}}{\text{minimize}} \quad & c^\mathsf{T} x + I_{\Omega_1}(x,y) + I_{\Omega_2}(\tilde{x}) + I_{\Omega_3}(\tilde{y}) \\ \text{subject to} \quad & x = \tilde{x} \\ & y = \tilde{y}, \end{aligned} \tag{LP}$$

where I_Ω denotes the indicator function of the set Ω.

(ii) State the ADMM iterations corresponding to the problem (LP) when considering (x, y) and $(\tilde{x}, \tilde{y})$ as the two sets of (primal) variables. Assume a constant α.

(iii) Show that the ADMM iterations can be simplified to

$$\begin{aligned} \begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} &= \begin{pmatrix} I & A^\mathsf{T} \\ A & -I \end{pmatrix}^{-1} \begin{pmatrix} \tilde{x}^k + \frac{\lambda_x^k}{\alpha} - \frac{c}{\alpha} \\ \tilde{y}^k + \frac{\lambda_y^k}{\alpha} \end{pmatrix} + \begin{pmatrix} 0 \\ \tilde{y}^k + \frac{\lambda_y^k}{\alpha} \end{pmatrix} \\ \tilde{x}^{k+1} &= P_{\Omega_2}\left(x^{k+1} - \frac{\lambda_x^k}{\alpha}\right) \\ \tilde{y}^{k+1} &= P_{\Omega_3}\left(y^{k+1} - \frac{\lambda_y^k}{\alpha}\right) \\ \begin{pmatrix} \lambda_x^{k+1} \\ \lambda_y^{k+1} \end{pmatrix} &= \begin{pmatrix} \lambda_x^k \\ \lambda_y^k \end{pmatrix} - \alpha \begin{pmatrix} x^{k+1} - \tilde{x}^{k+1} \\ y^{k+1} - \tilde{y}^{k+1} \end{pmatrix}. \end{aligned}$$

(iv) What is the computational bottleneck of this approach? Describe an efficient way of addressing this bottleneck.

[1] https://archive.ics.uci.edu/dataset/316/condition+based+maintenance+of+naval+propulsion+plants

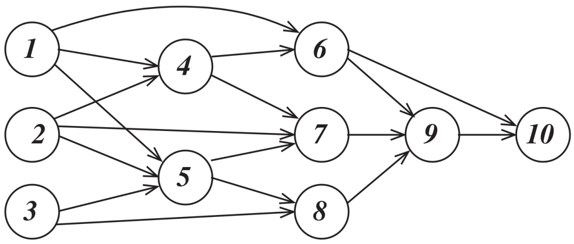

Figure 11.1: Computation graph for a function of three variables, with $N = 10$ nodes.

Problem 11.2

Let

$$f(x) = (\phi \circ \phi_l \circ \phi_{l-1} \circ \ldots \circ \phi_1)(x) = \phi(\phi_l(\phi_{l-1}(\ldots(\phi_1(x))\ldots), \tag{11.1}$$

where $\phi_1 \colon \mathbb{R}^n \to \mathbb{R}^{m_1}$, $\phi_i \colon \mathbb{R}^{m_{i-1}} \to \mathbb{R}^{m_i}$ $(i = 2, 3, \ldots, l)$, and $\phi \colon \mathbb{R}^{m_l} \to \mathbb{R}$. Furthermore, consider a partition of the variable vector x as $x = (x_1, x_2, \ldots, x_l)$, where $x_i \in \mathbb{R}^{n_i}$ for $i = 1, 2, \ldots, l$, so that $x \in \mathbb{R}^n$ with $n = \sum_{i=1}^{l} n_i$. Let

$$f(x) = \phi(\phi_l(x_l, \phi_{l-1}(x_{l-1}, \phi_{l-2}(x_{l-2}, \ldots(x_3, \phi_2(x_2, \phi_1(x_1))\ldots), \tag{11.4}$$

where $\phi_1 \colon \mathbb{R}^{n_1} \to \mathbb{R}^{m_1}$, $\phi_i \colon \mathbb{R}^{n_i} \times \mathbb{R}^{m_{i-1}} \to \mathbb{R}^{m_i}$ $(i = 2, 3, \ldots, l)$, and $\phi \colon \mathbb{R}^{m_l} \to \mathbb{R}$.

Sketch the computation graphs for the nested function (11.1) and the progressive function (11.4), in a format similar to Figure 11.1.

Submission instructions

Please submit a **single pdf document** that includes all *detailed derivations*, *justifications*, *code*, and *result figures*. Your solutions can be written on a computer, e.g., using LaTeX, or be handwritten, as long as they are readable.

All code snippets provided by us are written in Python. You can choose to implement the task in another language but you have to take care to port the given code yourself. Random states can then be neglected.

Your submission will be peer-reviewed. To pass the review, you should have **at least 70% correct**. Your reviewer will then comment on your submission and rate it according as *strong accept*, *accept*, *borderline accept*, *borderline reject*, or *reject* together with a justification.