Homework 4: Optimization and Root-Finding

Due November 21, 2023 (100 points)

Problem 1 (30 points)**.** In this problem, we consider the convergence of a fixed point iteration $\mathbf{x}_{k+1} := g(\mathbf{x}_k)$ on a *multivariable* function $g : \mathbb{R}^n \to \mathbb{R}^n$.

(a) Suppose $g \in C^1$ and $\mathbf{x}_k$ is within a small neighborhood of a fixed point $\mathbf{x}^*$ of g. Suggest a condition on the Jacobian Dg of g that guarantees g is 1-Lipschitz in this neighborhood.

(b) Using the previous result, derive a bound for the error of $\mathbf{x}_{k+1}$ in terms of the error of $\mathbf{x}_k$ and the Jacobian of g.

(c) Propose a condition on the singular values of Dg that guarantees convergence of multivariable fixed point iteration.

(d) How does the convergence rate change if $Dg(\mathbf{x}^*) = 0$?

Problem 2 (50 points)**.** In this problem, we implement a method for computing a set of n points $\mathbf{x}_i \in \mathbb{R}^d$ whose pairwise distances $\|\mathbf{x}_i - \mathbf{x}_j\|_2$ approximate a given set of distances D_{ij} and compare its performance to `scikit-learn`'s implementation. The "SMACOF" algorithm proposes the following optimization problem:

$$\min_{X \in \mathbb{R}^{d \times n}} \sum_{i,j=1}^{n} \left(D_{ij} - \|\mathbf{x}_i - \mathbf{x}_j\|_2\right)^2,$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is the i-th column of X. The objective $f(X) = \sum_{i,j=1}^{n} \left(D_{ij} - \|\mathbf{x}_i - \mathbf{x}_j\|_2\right)^2$ is called the *stress function*. Note that f is *non-convex*, so the methods we have studied so far are not guaranteed to find a global minimum to this problem.

(a) By expanding the square in $f(X)$, show that minimizing f is equivalent to minimizing $\mathrm{tr}(XVX^\top) - 4\mathrm{tr}(XB(X)X^\top)$, where $V = 2n(I_{n\times n} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)$ and

$$B_{ij}(X) = \begin{cases} -\frac{Dij}{\|\mathbf{x}_i - \mathbf{x}_j\|_2} & \text{if } \mathbf{x}_i \neq \mathbf{x}_j \text{ and } i \neq j \\ 0 & \text{if } \mathbf{x}_i = \mathbf{x}_j \text{ and } i \neq j \\ -\sum_{j \neq i} B_{ij} & \text{if } i = j. \end{cases}$$

(b) Now define $\tau(X, Z) := \mathrm{tr}(XVX^\top) - 4\mathrm{tr}(XB(Z)Z^\top)$. Show that for any matrix $Z \in \mathbb{R}^{d\times n}$, we have $\tau(X, X) \leq \tau(X, Z)$.

Hint: The Cauchy-Schwarz inequality tells us that $(\mathbf{x}_i - \mathbf{x}_j) \cdot (\mathbf{z}_i - \mathbf{z}_j) \leq \|\mathbf{x}_i - \mathbf{x}_j\|_2 \|\mathbf{z}_i - \mathbf{z}_j\|_2$.

(c) In light of the previous result, we say that $\tau(X, Z)$ *majorizes* $\tau(X, X)$. This allows us to use the *majorization-minimization* (MM) algorithm, which iterates $X^{k+1} = \arg\min_X \tau(X, X^k)$. Show that $f(X^{k+1}) \leq f(X^k)$ for each $k \geq 0$.

(d) Our surrogate objective $\tau(X, Z)$ is convex in X, so we can compute its global minimum by setting $\nabla_X \tau(X, Z) = 0$ and solving for X. Using this method, derive an explicit expression for X^{k+1} in terms of X^k.

(e) Complete the code for the SMACOF algorithm in `hw4.py`. Using the pairwise distance data you have been provided in `dists.npy`, compute a SMACOF embedding, and plot it using e.g. `matplotlib`. Also, plot the value of $f(X^k)$ for each iteration and verify that it decreases monotonically. Include screenshots of these plots in your submission.

(f) Compare your SMACOF implementation to the `scikit-learn` implementation. Which implementation decreases the objective value $f(X^k)$ more quickly as a function of iteration number k? Also, plot the objective value $f(X^k)$ against the *wall-clock time* required to compute the first k iterations. Which implementation decreases the objective value $f(X^k)$ as a function of wall-clock time? Include relevant plots to illustrate your comparison.

Problem 3 (20 points)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex and continuous, but *not necessarily differentiable* function. We define a *subgradient* of f at $\mathbf{x}_0 \in \mathbb{R}^n$ to be a vector $\mathbf{s} \in \mathbb{R}^n$ such that $f(\mathbf{x}) - f(\mathbf{x}_0) \geq \mathbf{s} \cdot (\mathbf{x} - \mathbf{x}_0)$ for all $\mathbf{x} \in \mathbb{R}^n$. This is a useful generalization of the gradient of f at points where f is not differentiable. We define the *subdifferential* $\partial f(\mathbf{x}_0)$ to be the set of all subgradients of f at $\mathbf{x}_0$.

(a) What is $\partial f(0)$ for the absolute value function $f(x) = |x|$?

(b) Suppose we wish to minimize a convex and continuous function $f : \mathbb{R}^n \to \mathbb{R}$ that may not be differentiable everywhere. Propose an optimality condition involving $\partial f(\mathbf{x}^*)$ for $\mathbf{x}^*$ to be a minimizer of f. Show that your condition holds if and only if $\mathbf{x}^*$ globally minimizes f.