

Problem Set 1

CS 4301

Due: 9/16/2020 by 11:59pm

Note: all answers should be accompanied by explanations and code for full credit. Late homeworks will not be accepted.

Problem 1: Gradients & More (10 pts)

1. For each of the following functions: (1) Prove that they are convex, compute gradients/sub-gradients, and write the updates for (sub)gradient descent with a fixed step size.

(a) $f(w) = \sum_{m=1}^M \max(0, 1 - y_m w^T x^{(m)})^2$ for $w \in \mathbb{R}^n$ and fixed $y_1, \dots, y_M \in \mathbb{R}$ and $x^{(1)}, \dots, x^{(M)} \in \mathbb{R}^n$.

(b) $f(x) = \log [\sum_{i=1}^n \exp(a_i x_i)]$ for $x \in \mathbb{R}^n$ and constants $a \in \mathbb{R}^n$.

(c) $f(x) = \max_{m \in \{1, \dots, m\}} [a^{(m)T} x + b_m]$ for $x \in \mathbb{R}^n$ and constants $a^{(1)}, \dots, a^{(M)} \in \mathbb{R}^n$ and $b_1, \dots, b_M \in \mathbb{R}$.

Problem 2: Line Search vs. Fixed Step Size (60 pts)

1. Consider the least squares optimization problem discussed in class. That is, given a collection of data observations $(x^{(1)}, y^{(1)}), \dots, (x^{(M)}, y^{(M)}) \in \mathbb{R}^2$, consider finding the line $f(x) = ax + b$ that minimizes the squared error.

$$\min_{a,b} \frac{1}{M} \sum_{m=1}^M \left(y^{(m)} - ax^{(m)} - b \right)^2$$

- (a) In Python, implement gradient descent to find the optima a and b in the following ways:
- Standard gradient descent with a fixed step size, γ .
 - Standard gradient descent with a diminishing step size, $\frac{p}{q+t}$ for a given p and q .
 - Gradient descent with exact line search.
 - Gradient descent with backtracking line search for a given α and β .

Each different approach should be written as a single Python function that takes as input the x 's, the y 's, an initial guess for a , an initial guess for b , and any parameters that describe the method, e.g., γ for the fixed step size method.

- (b) Plot the value of the objective function versus iteration starting at $a = b = 0$ for each of the four different methods (use $\gamma = .5, p = q = 1, \alpha = .3, \beta = .1$) for 100 iterations. You should use the data observations provided in the file `linreg.data`.

2. Consider the same function fitting problem as above but with absolute error.

$$\min_{a,b} \frac{1}{M} \sum_{m=1}^M \left| y^{(m)} - ax^{(m)} - b \right|$$

- (a) In Python, implement subgradient descent to find the optima a and b in the following ways:
- Subgradient descent with a diminishing step size, $\frac{p}{q+t}$ for a given $p > 0$ and q .
 - Subgradient descent with a diminishing step size, $\frac{p}{\sqrt{t}}$ for a given $p > 0$.
- (b) Plot the value of the objective function versus iteration starting at $a = b = 0$ for each of the different methods (use $p = q = 1$) for 100 iterations. Again, you should use the data observations provided in the file `linreg.data`.

Problem 3: Numerical Gradients (30 pts)

Sometimes a function may be differentiable, but its gradients may be difficult to compute or the gradients are unavailable. In such cases, numerical methods can be used to approximate the derivative. One such strategy is to use the so-called symmetric difference quotient,

$$\frac{f(x + \epsilon v) - f(x - \epsilon v)}{2\epsilon},$$

with a small $\epsilon > 0$ to approximate the directional derivative in the direction v .

1. Consider the following univariate optimization problem.

$$\inf_a \left[\frac{\exp(a)}{1 + \exp(a)} \ln \left(\frac{\exp(a)}{1 + \exp(a)} \right) + \frac{1}{1 + \exp(a)} \ln \left(\frac{1}{1 + \exp(a)} \right) \right]$$

- (a) Is this a convex optimization problem? Prove your answer.
- (b) Use gradient descent with a fixed step size to find the global minimum.
- (c) Use gradient descent with a fixed step size and numerical approximations of the derivatives to find the global minimum.
2. Consider the convex function in `cvxnum.py`. Apply numerical gradient descent to minimize this function.