4301 - HW1

September 2020

Problem 1: Gradients & More

For each of the following functions: (1) Prove that they are convex, (2) compute gradients/subgradients, (3) write the updates for (sub)gradient descent with a fixed step size.

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

1. *Proof.* For $m \in \{1, \ldots, M\}$, define $g_m(w) = \max(0, 1 - y_m w^T x^{(m)})$.
   Based on what we have in the following, we will prove that $f(w) = \sum_{m=1}^{M} g_m(w)^2$ is convex.
   (a) Nonnegative weighted sum of convex functions is convex.
       ($f(w)$ is the summation over all $g_m(w)^2$)
   (b) The composition of two convex functions is convex.
       ($g_m(w)$ has been composited in w^2.)
       $\rightarrow$ Therefore, showing that $g_m(w)$ is convex for $m \in \{1, \ldots, M\}$, induces the desirable result.
   (c) Pointwise Maximum of convex functions is convex.
       ($h(w) = 0$ and $p(w) = 1 - y_m w^T x^{(m)}$ are both affine, hence they are convex.)

   So $f(w)$ is convex. $\square$

2. $f(w)$ is differentiable everywhere and its gradient is

$$\nabla f(w) = \sum_{m=1}^{M} \nabla g_m(w)$$

$$\nabla g_m(w) = \begin{cases} 0 & \text{if } 1 - y_m w^T x^{(m)} \leq 0 \\ 2(1 - y_m w^T x^{(m)})(-y_m x^{(m)}) & \text{if } 1 - y_m w^T x^{(m)} > 0 \end{cases}$$

3. $w^{\text{new}} = w^{\text{old}} - \lambda \nabla f(w^{\text{old}})$

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

1. *Proof.* The Hessian matrix of a multivariate convex function is positive semi-definite. The Hessian matrix of $f(x) = \log[\sum_{i=1}^{n} \exp(a_i x_i)]$ is

$$H = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix},$$

while

$$\frac{\partial^2 f(x)}{\partial x_j^2} = \frac{a_j^2 \exp(a_j x_j) \sum_{i=1}^n \exp(a_i x_i)}{(\sum_{i=1}^n \exp(a_i x_i))^2} + \frac{-(a_j \exp(a_j x_j))(a_j \exp(a_i x_i))}{(\sum_{i=1}^n \exp(a_i x_i))^2}$$

$$\frac{\partial^2 f(x)}{\partial x_k \partial x_j} = \frac{-(a_j \exp(a_j x_j))(a_k \exp(a_k x_k))}{(\sum_{i=1}^n \exp(a_i x_i))^2}.$$

The denominator is a positive number. For the ease of notation, it is negligible.

$$\frac{\partial^2 f(x)}{\partial x_j^2} \approx a_j^2 \exp(a_j x_j) \sum_{i=1}^n \exp(a_i x_i) - (a_j \exp(a_j x_j))(a_j \exp(a_j x_j))$$

$$\frac{\partial^2 f(x)}{\partial x_k \partial x_j} \approx -(a_j \exp(a_j x_j))(a_k \exp(a_k x_k)).$$

Define diagonal matrix D such that for $j \in \{1, \ldots, n\}$, $D_{j,j} = a_j^2 \exp(a_j x_j) \sum_{i=1}^n \exp(a_i x_i)$. In addition, define vector u such that $u_j = a_j \exp(a_j x_j)$ for $j \in \{1, \ldots, n\}$. By definition $H = D - uu^T$. Now, in order to prove that $f(x)$ is convex, we will show that $H = D - uu^T$ is positive semidefinite. Matrix H is positive semidefinite if for any vector $y \in \mathbb{R}^n$: $y^T H y \geq 0$. Therefore,

$$\begin{aligned}
y^T H y &= y^T (D - uu^T) y \\
&= y^T D y - y^T u u^T y \\
&= \sum_{j=1}^n \Big(y_j^2 a_j^2 \exp(a_j x_j) \sum_{i=1}^n \exp(a_i x_i) \Big) - \Big(\sum_{j=1}^n y_j a_j \exp(a_j x_j) \Big)^2 \\
&= \Big(\sum_{i=1}^n \exp(a_i x_i) \Big) \Big(\sum_{j=1}^n y_j^2 a_j^2 \exp(a_j x_j) \Big) - \Big(\sum_{j=1}^n y_j a_j \exp(a_j x_j) \Big)^2 \\
&\geq 0.
\end{aligned}$$

The last inequality is based on Cauchy Schwarz inequality: $(\sum a_i^2)(\sum b_i^2) \geq (\sum a_i b_i)^2$. As a result, the Hessian matrix of $f(x)$ is positive semidefinite, so $f(x)$ is convex.

□

2. The function $f(x)$ is differentiable everywhere. And for $j \in \{1, \ldots, n\}$

$$\nabla f(x)_j = \frac{a_j \exp(a_j x_j)}{\sum_{i=1}^n \exp(a_i x_i)}$$

3. $x^{\text{new}} = x^{\text{old}} - \lambda \nabla f(x^{\text{old}})$

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

1. *Proof.* For $m \in \{1, \ldots, M\}$ define $g_m(x) = {a^{(m)}}^T x + b_m$. Every $g_m(x)$ is affine, so they are all convex. On the other hand, the pointwise maximum of convex functions is convex. So, in conclusion $f(x) = \max_{m \in \{1, \ldots, M\}} [g_m(x)]$ is convex. □

2. $f(x)$ is not always differentiable, so we need to hire subgradient.

$$\nabla f(x)\Big|_{x^*} = \begin{cases} a^{(m^*)} : m^* \text{ is the unique answer to } \text{argmax}_m {a^{(m)}}^T x^* + b_m \\ \text{any convex combination of members of } \{a^{(m)} : m \in \{\text{argmax}_m {a^{(m)}}^T x^* + b_m\}\} \end{cases}$$

3. $x^{\text{new}} = x^{\text{old}} - \lambda \nabla f(x^{\text{old}})$

Google Colab

For the next two questions, I am sharing with you two notebooks from **Google Colab**.

YouTube Intro to Google Colab

I strongly encourage you to submit your future homeworks via Colab. Because

- It is a highly promising platform that you will find quite useful in near future.
- It makes it easier to discuss your work during office hours.
- It makes it easier for you to submit without any additional document.
- And there might be bonus points for students who submit via Colab.

For submitting your homeworks, remember to share an **editable link**.

Problem 2: Line Search vs. Fixed Step Size

Please find the answer to the second question in Colab Question 2

- You can add comments that everyone see. Please let us know if there is something wrong or confusing, and please ask all of your questions.
- You need to upload the file linreg.data
  (At the very left hand side: Files → Upload to session storage)
- My personal favorite: Runtime → Restart and run all

Problem 3: Numerical Gradients

Please find the answer to the third question in Colab Question 3

- You can add comments. Please do.
- You do not need to upload the file cvxnum.py. It has been copy pasted there.