

Assignment #10

Due: 6:00pm **Tuesday 27 April**, 2021

Upload at: <https://www.gradescope.com/courses/214271/assignments/1176031>

Assignments in COS 302 should be done individually. See the [course syllabus](#) for the collaboration policy.

Remember to append your Colab PDF as explained in the first homework, with all outputs visible.
When you print to PDF it may be helpful to scale at 95% or so to get everything on the page.

Problem 1 (12pts)

Consider the following scalar-valued function

$$f(x, y, z) = x^2y + \sin(z + 6y).$$

- (A) Compute partial derivatives with respect to x , y , and z .
- (B) We can consider f to be a function that takes a vector $\theta \in \mathbb{R}^3$ as input, where $\theta = [x, y, z]^T$. Write the gradient as a vector and evaluate it at $\theta = [3, \pi/2, 0]^T$.

Problem 2 (12pts)

The purpose of this problem is to demonstrate Clairaut's Theorem, which states that in general, the order in which you perform partial differentiation does not matter. (Technically there are assumptions that the function must satisfy, but they are almost always true for the kinds of functions we care about in machine learning.) Consider the following scalar-valued function,

$$f(x, y, z) = x \sin(xy),$$

where $x, y \in \mathbb{R}$.

(A) Compute $\frac{\partial}{\partial x} \frac{\partial}{\partial y} f(x, y)$. This means we first compute the partial derivative of f with respect to y , then compute the partial derivative of the resulting function with respect to x . This is sometimes denoted $\partial_{xy} f$.

(B) Compute $\frac{\partial}{\partial y} \frac{\partial}{\partial x} f(x, y)$.

You should have gotten the same answer for parts (A) and (B), which demonstrates Clairaut's Theorem. This holds more generally for n variables as well, that is, if $f(x_1, x_2, \dots, x_n)$ is a function of n variables. This result can be useful when differentiating functions, since it's possible that it's much more convenient computationally-wise to differentiate in an order that you choose.

Problem 3 (12pts)

Consider the following vector function from \mathbb{R}^3 to \mathbb{R}^3 :

$$f(x) = \begin{bmatrix} \sin(x_1 x_2 x_3) \\ \cos(x_2 + x_3) \\ \exp\{-\frac{1}{2}(x_3^2)\} \end{bmatrix}$$

- (A) What is the Jacobian matrix of $f(x)$?
- (B) Write the determinant of this Jacobian matrix as a function of x .
- (C) Is the Jacobian a full rank matrix for all of $x \in \mathbb{R}^3$? Explain your reasoning.

Problem 4 (16pts)

Compute the gradients for the following expressions. (You can use identities, but show your work.)

(A) $\nabla_x \text{trace}(\mathbf{x}\mathbf{x}^T + \sigma^2 \mathbf{I})$ Assume $\mathbf{x} \in \mathbb{R}^n$ and $\sigma \in \mathbb{R}$.

(B) $\nabla_x \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$ Assume $\mathbf{x}, \boldsymbol{\mu} \in \mathbb{R}^n$ and invertible symmetric $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$.

(C) $\nabla_x (\mathbf{c} - \mathbf{A}\mathbf{x})^T (\mathbf{c} - \mathbf{A}\mathbf{x})$ Assume $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{c} \in \mathbb{R}^m$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$.

(D) $\nabla_x (\mathbf{c} + \mathbf{A}\mathbf{x})^T (\mathbf{c} - \mathbf{B}\mathbf{x})$ Assume $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{c} \in \mathbb{R}^m$ and $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$.

Problem 5 (16pts)

The sigmoid function $f : \mathbb{R} \rightarrow \mathbb{R}$ (also called the *logistic function*) is defined to be:

$$f(z) = \frac{1}{1 + e^{-z}}. \quad (1)$$

(A) Compute the derivative of the sigmoid function, i.e., $f'(z)$. And verify that $f'(z) = f(z)(1 - f(z))$

(B) The cost function of a very popular machine learning model logistic regression has the following form:

$$c(\boldsymbol{\theta}, \mathbf{x}, y) = -y \log\left(\frac{1}{1 + e^{-\boldsymbol{\theta}^\top \mathbf{x}}}\right) - (1 - y) \log\left(1 - \frac{1}{1 + e^{-\boldsymbol{\theta}^\top \mathbf{x}}}\right) \quad (2)$$

with $\boldsymbol{\theta} \in \mathbb{R}^d$, $\mathbf{x} \in \mathbb{R}^d$, $y \in \mathbb{R}$. Compute the partial derivative with regards to $\boldsymbol{\theta}$, i.e. $\frac{\partial c(\boldsymbol{\theta}, \mathbf{x}, y)}{\partial \boldsymbol{\theta}}$. And verify that $\frac{\partial c(\boldsymbol{\theta}, \mathbf{x}, y)}{\partial \boldsymbol{\theta}} = (f(\boldsymbol{\theta}^\top \mathbf{x}) - y) \mathbf{x}^\top$.

Problem 6 (30pts)

In gradient descent, we attempt to minimize some function $f(x)$ by altering parameters $x \in \mathbb{R}^n$ according to the following formula:

$$x_{t+1} = x_t - \lambda(\nabla_x f(x_t))^T$$

for some small $\lambda \geq 0$ known as the *learning rate* or *step size*. We adjust x so as to move in a direction proportional to the negative gradient. We will discuss gradient descent in more detail later in the course.

Consider the simple function $f(x) = x^T A x$ for a constant positive definite matrix $A \in \mathbb{R}^{n \times n}$.

- (A) Implement a function `f(A, x)` that takes as input an $n \times n$ numpy array `A` and a 1D array `x` of length n and returns the output for the above definition.
- (B) Implement a function `grad_f(A, x)` that takes the same two arguments as above but returns $\nabla_x f(x)$ evaluated at `x`.
- (C) Now implement a third and final function `grad_descent(A, x0, lr, num_iters)` that takes the additional arguments `lr`, representing the learning rate λ above, and `num_iters` indicating the total number of iterations of gradient descent to perform. The function should print the values of x_t and $f(x_t)$ at each iteration of gradient descent.
- (D) We will perform gradient descent on f with $A = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$. Set each element of the initial x_0 to be $[10, 10]^T$. Run gradient descent for 50 iterations with learning rates of 1, 0.25, 0.1, and 0.01. What do you notice? Does x_t always converge to the same value? Does our gradient descent algorithm work every time?

Problem 7 (2pts)

Approximately how many hours did this assignment take you to complete?

My notebook URL: <https://colab.research.google.com/XXXXXXXXXXXXXXXXXXXXXXXXX>

Changelog

- 13 April 2021 – S21 initial version