Homework Assignment 3

COGS 118A: Introduction to Machine Learning I

Due: 11:59pm, Sunday, October 21st, 2018 (Pacific Time).

Instructions: Use **Jupyter Notebook** to answer the questions below, include all your code and figures in the notebook. Export the notebook as PDF file and submit it on Gradescope. You may look up the information on the Internet, but you must write the final homework solutions by yourself.

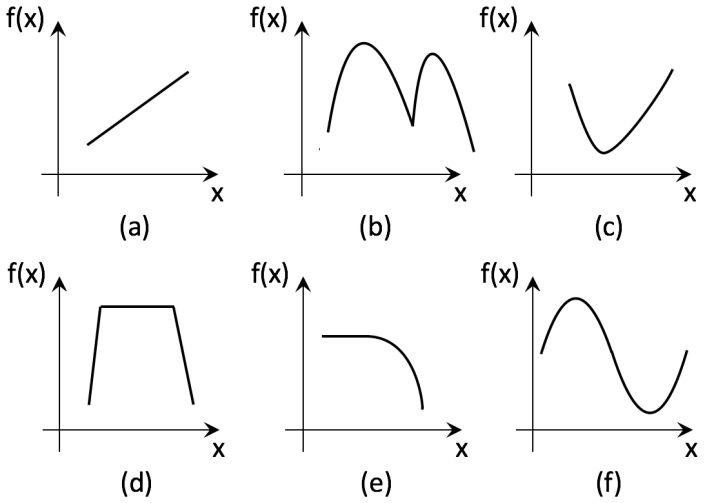
Please Note: Writing homework without using Jupyter Notebook will lead to point deduction.

Late Policy: 5% of the total points will be deducted on the first day past due. Every 10% of the total points will be deducted for every extra day past due.

Grade: ____ out of 100 points

1 (12 points) Convex

Identify the convexity for the following six functions (a-f) (Write down whether the function is convex or non-convex).



2 (40 points) Least Square Estimation

We are given the data $S = \{(x_i, y_i), i = 1, \ldots, n\}$. Here, $x_i, i = 1, \ldots, n$ are one dimensional scalars. We will try to fit the data with a line and a parabola, i.e. $y_i = w_1 x_i + w_2$ and $y_i = w_1 x_i + w_2 x_i^2 + w_3$. To solve this problem with matrix manipulation, we represent the data as matrices $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$ and $Y = [y_1, y_2, \ldots, y_n]^T$, where $\mathbf{x}_i$ is a feature vector corresponding to the data x_i: $\mathbf{x}_i$ is either $[1, x_i]^T$ or $[1, x_i, x_i^2]^T$ in this problem. We are finding W that minimizes the sum-of-squares error function $g(W)$.

$$g(W) = \|XW - Y\|_2^2 = (XW - Y)^T (XW - Y). \tag{1}$$

(a) Compute the gradient of $g(W)$ with respect to W.
(b) By setting the answer of part (a) to $\mathbf{0}$, prove the following:

$$W^* = \arg\min_W g(W) = (X^T X)^{-1} X^T Y. \tag{2}$$

(c) **Python:** Download the file **HW3.ipynb** from the course website. Then, complete the TODO blocks in the Jupyter notebook.

3 (25 points) Least Square Estimation via Gradient Descent

We are given the data $S = \{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$. Here, $\mathbf{x}_i, i = 1, \ldots, n$ contains two features, i.e. two-dimensional vectors. We will try to fit the data with a hyperplane, i.e. $y_i = w_1 x_{i1} + w_2 x_{i2} + w_3$. $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$ and $Y = [y_1, y_2, \ldots, y_n]^T$, where $\mathbf{x}_i$ is a feature vector corresponding to the data $\mathbf{x}_i$: $\mathbf{x}_i$ is $[1, x_{i1}, x_{i2}]^T$ in this problem.We are finding W that minimizes the sum-of-squares error function $g(W)$.

$$g(W) = \|XW - Y\|_2^2 = (XW - Y)^T (XW - Y). \tag{3}$$

Python: Please continue complete the TODO blocks in the Jupyter notebook.

For gradient descent, please: a) Try some different values for initialization, making sure your algorithm converge no matter what value you set. For example, you can start with all zeros.

b) Stop the iteration if the sum of the absolute values of the differences between the new W and old W is less than 0.0000001, or exceeding 10000 iterations (to avoid an infinite loop).

c) For learning rate, try a few of them for experiments. You can try the following values and observe whether your algorithm converge, if so, how fast it converges: 0.00002, 0.00001, 0.000005. For the first one, you can run a few iterations (e.g. 10 iterations) see whether it works. It is possible that the values are overflow in python.

You can write down your observations of these experiments, but it is not required. Make sure to include a working version of code and parameters in your submission.

4 Concepts

Select the correct option(s). Note that there might be multiple correct options.

1. What are the most significant difference between regression and classification?
   A. unsupervised learning vs. supervised learning
   B. prediction of continuous values vs. prediction of class labels
   C. least square estimation vs. gradient descent
   D. convex vs. non-convex problem
   E. higher vs. lower error

2. What are true about solving regression problem with gradient descent compared to closed-form solution?
   A. matrix inverse could be expensive when the dataset is large
   B. gradient descent is slower
   C. gradient descent will give you the exactly the same result as closed-form solution
   D. it's hard to set a good learning rate for gradient descent

3. Is gradient descent guaranteed to find the global optimal in a convex problem? What about non-convex problem?
   A. yes for a convex problem
   B. no for a convex problem
   C. yes for a non-convex problem
   D. no for a non-convex problem

4. What are true about local optimal and global optimal?
   A. local optimal is better
   B. There can exist multiple local optimal
   C. gradient descent is able to find the global optimal
   D. least square solution finds the global optimal