Homework 4

Spring 2021
(Due: Friday, Mar 19, 2021, 11:59 pm Eastern Time)

Please submit your homework through **gradescope**. You can write, scan, type, etc. But for the convenience of grading, please merge everything into a **single PDF**.

Objective

There are three things you will learn in this homework:

(a) Understand some theoretical properties about logistic regression.

(b) Implement a logistic regression in CVX, and visualize the decision boundary.

(c) Apply kernel trick to logistic regression.

You will be asked some of these questions in Quiz 4. The Quiz will be open on Mar 20, 8am Eastern Time, and close on Mar 21, 8am Eastern Time. The Quiz is 30 minutes long.

Exercise 1. Logistic regression + gradient descent
We analyze the convergence behavior of the logistic regression when the data is **linearly separable**.

Recall that the logistic regression tries to minimize the cross-entropy loss:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\operatorname{argmin}} -\sum_{n=1}^{N}\Big\{y_n \log h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) + (1-y_n)\log(1-h_{\boldsymbol{\theta}}(\boldsymbol{x}_n))\Big\}, \tag{1}$$

where $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1+\exp\{-\boldsymbol{\theta}^T\boldsymbol{x}\}}$ is the sigmoid function. As usual, we assume that $\boldsymbol{\theta} = [\boldsymbol{w},\ w_0]$. We consider the gradient descent algorithm. We know that the objective function is convex, and so we consider the gradient descent algorithm. The iterations are (if you take the derivative of $J(\boldsymbol{\theta})$ and rearrange the terms):

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \alpha_k\left(\sum_{n=1}^{N}(h_{\boldsymbol{\theta}^{(k)}}(\boldsymbol{x}_n) - y_n)\boldsymbol{x}_n\right) \tag{2}$$

for some choices of the step size α_k.

(i) Prove that if two classes of data in $\mathbb{R}^d$ are linearly separable, then the magnitude of the slope $\|\boldsymbol{w}\|_2$ and the magnitude of the intercept $|w_0|$ would tend to ∞.

(ii) Prove that the gradient descent iterates in (2) would not converge in a finite number of steps, if we allowed the algorithm to run forever (i.e. only let it stop when $\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}\|_2 = 0$).

(iii) What happens if we restrict $\|\boldsymbol{w}\|_2 \le c_1$ and $|w_0| < c_2$ for some $c_1, c_2 > 0$? What other ways can you come up with to counter the nonconvergence issue?

(iv) Does linear separability of data cause nonconvergence for the other linear classifiers that we have studied? Why?

Exercise 2. Logistic regression loss is convex
Define the logistic loss function

$$J(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N}\Big\{y_n \log h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) + (1-y_n)\log(1-h_{\boldsymbol{\theta}}(\boldsymbol{x}_n))\Big\}, \tag{3}$$

where $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1+\exp\{-\boldsymbol{\theta}^T\boldsymbol{x}\}}$ is the sigmoid function. Show that $J(\boldsymbol{\theta})$ is convex by computing its Hessian and showing that it is positive semi-definite.

Exercise 3. Implement logistic regression
Download the dataset from the course website. There are two classes with class labels $y_n = 1$ and $y_n = 0$.

(a) Show that the logistic regression loss is given by

$$J(\boldsymbol{\theta}) = -\frac{1}{N}\left\{\left(\sum_{n=1}^{N} y_n\boldsymbol{x}_n\right)^T\boldsymbol{\theta} - \sum_{n=1}^{N}\log\left(1+e^{\boldsymbol{\theta}^T\boldsymbol{x}_n}\right)\right\}.$$

(b) Introduce a regularization term $\lambda\|\boldsymbol{\theta}\|^2$ so that the loss becomes

$$J(\boldsymbol{\theta}) = -\frac{1}{N}\left\{\left(\sum_{n=1}^{N} y_n\boldsymbol{x}_n\right)^T\boldsymbol{\theta} - \sum_{n=1}^{N}\log\left(1+e^{\boldsymbol{\theta}^T\boldsymbol{x}_n}\right)\right\} + \lambda\|\boldsymbol{\theta}\|^2. \tag{4}$$

Use CVXPY to minimize this loss function for the dataset I provided. Your $\boldsymbol{\theta}$ should be $\boldsymbol{\theta} = [\theta_2, \theta_1, \theta_0]^T$. Please use $\lambda = 0.0001$.

(c) Scatter plot the data points by marking the two classes in two colors. Then plot the decision boundary.

(d) Repeat (c), but this time using the Bayeisn decision rule. Note that since the covariance matrices are not identical, the decision boundary is not a straight line. To plot the decision boundary, you can create a grid of testing sites in the range of $[-5, 10] \times [-5, 10]$ (with 100 points along each dimension). Evaluate the decision on these testing sites. And then plot the decision using `plt.contour`.

Exercise 4. Kernel trick
Let us continue to use the dataset in Exercise 3. Our goal here is to implement the kernel trick.

(a) In Python, construct the kernel matrix $\boldsymbol{K}$, where

$$[\boldsymbol{K}]_{m,n} = \exp\left\{-\|\boldsymbol{x}_m - \boldsymbol{x}_n\|^2/h\right\}, \tag{5}$$

where $h = 1$. Print `K[47:52,47:52]`.

(b) Let us assume that that $\boldsymbol{\theta} = \sum_{n=1}^{N}\alpha_n\boldsymbol{x}_n$ for some α_n's. Then

$$\boldsymbol{\theta}^T\boldsymbol{x} = \sum_{n=1}^{N}\alpha_n\langle\boldsymbol{x}_n, \boldsymbol{x}\rangle.$$

The kernel trick says that we can replace $\langle\boldsymbol{x}_n, \boldsymbol{x}\rangle$ by a kernel $K(\boldsymbol{x}_n, \boldsymbol{x})$. Apply the kernel trick to the loss function in (4). Show that the new loss is

$$J(\boldsymbol{\alpha}) = -\frac{1}{N}\left\{\boldsymbol{y}\boldsymbol{K}\boldsymbol{\alpha} - \mathbf{1}^T\log(e^{\mathbf{0}} + e^{\boldsymbol{K}\boldsymbol{\alpha}})\right\} + \lambda\boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha}.$$

(c) Implement the kernel logistic regression training in CVXPY. Report the first two elements of the regression coefficients $\boldsymbol{\alpha}$.

(d) Scatter plot the data points and plot the decision boundary, just like what you did in Exercise 3(d).

Exercise 5. Project Check Point # 4

At this checkpoint, I assume that you have been playing the existing code and trying various things. I can foresee that many of you are struggling to implement your idea. As I have said in the previous checkpoint, this is completely normal but you need to overcome it. If you have not yet started doing anything, please do it immediately.

To keep track of your progress, I am asking for three specific deliverables in this homework.

1. You should more or less have a hypothesis in mind. State the hypothesis in your report, with some descriptions. For example, you can say, "noise2noise does not apply to independent but non-zero mean noise", or "if we adversarially train a classifier with a uniform distribution of the attack strengths, we lose performance in the easy cases." These are just some simple examples to demonstrate what I am looking for. I am looking for **one** statement like these and a description of what makes you have this speculation. Please do not be too ambitious. Focus on one hypothesis. Of course, please do not say something trivial.

2. Based on your experience in playing the code, list out the things you need to do to verify the hypothesis. If you have already done something, please report your preliminary results. So I am looking for one of the two things: (1) A list of actions you need to do to verify your hypothesis. Give a detailed plan of how to do it. What dataset to use, how to generate the testing configurations, and what kind of **quantitative results** you are expecting to get? If possible, hand draw your speculated plots. (2) If you have already done some preliminary studies, describe how you conducted the experiments. Explain why these experiments make sense. Show your results. It is perfectly okay to be incomplete at this stage, but you need to show some progress.

3. Draw me a timeline from now until the end of the semester. State what needs to be completed by when. You need to have at least two milestones before the final report. What are they? Do you have enough time to accomplish what you want to do? Do you need to revise the scope of your project? Write a brief paragraph summarizing your plan.

Since this is Homework 4, I expect you to have some progress in the project. Therefore, I will tell the TA to read this homework. If we see that your progress is unsatisfactory, we will flag you by taking some points off this homework. Please take this feedback seriously. The grading for the final project report will be straight.