# Machine Learning, Spring 2019

Homework 2

Due on **26th March, 11:59 PM**

---

1. Submit your solutions to Gradescope (www.gradescope.com). Homework of this week contains two part, **theoretical part** and **programming part**. So there are two assignments in gradescope, the assignment titled with programming part will require you to submit your code.
2. **Make sure each solution page is assigned to the corresponding problems** when you submit your homework.
3. The $x$-label and $y$-label must be clearly specified when plotting the results.
4. Any programming language is allowed for your code, but **make sure it is clear and readable with necesary comments.**

---

## 1 Descent direction

(a) Suppose $f$ is continuously differentiable. At a point $x \in \mathbb{R}^n$, for any direction $d$ satisfying $\nabla f(x)^T d < 0$, show that we can decrease $f$ by moving (a sufficiently small distance) along such a direction. (10 points)

(b) Given the function $f(x, y) = 3y^2 - 2y^3 - 3x^2 + 6xy$. Please judge whether the following directions are descent direction at the point $(x, y) = (1, 1)$. $d_1 = (1, 0), d_2 = (0, -1), d_3 = (1, 1)$ (5 points)

## 2 Hat Matrix

In linear regression, the in-sample predictions are given by $\hat{y} = Hy$, where $H = X(X^T X)^{-1} X^T$ (known as the hat matrix). Note that $X$ is an $N$ by $d + 1$ matrix and $X^T X$ is invertible.

(a) Show that $H$ is a projection matrix, i.e., $H^2 = H$. So $\hat{y}$ is the projection of $y$ onto some space. What is this space?(10 points)

(b) Show that $H^K = H$ for any positive integer $K$. (5 points)

(c) Show that trace$(H) = d + 1$, where the trace is the sum of diagonal elements. (5 points)

## 3 Implementation for PLA

In this problem, you are required to implement PLA to classify given data. Suppose the training data is $\mathcal{D} = (\mathbf{x}_n, \mathbf{y}_n)_{n=1}^{N}$ with $\mathbf{x}_n \in \mathbb{R}^2$, $\mathbf{y}_n \in \{+1, 0\}$, and the hypothesis function is $h(\mathbf{x})$. Then the in-sample error is defined as the ratio of the misclassified,

$$E_{\text{in}} = \frac{1}{N} \sum_{n=1}^{N} (h(\mathbf{x_n}) \neq \mathbf{y}_n) \,.$$

(a) (Linear separable case) You are given training data and test data (in file **PLA_a**), run PLA on training data and plot your hypothesis function on test data. (label of test data is not allowed to use in your algorithm.) Also, plot the in-sample error versus iteration time. (10 points)

(b) (Nonlinearly separable case) You are also given training data and test data but in this case, they are not linearly separable (in file **PLA_b**). The data will be linearly separable after the removal of several samples, which could be considered noisy samples or outliers. Run PLA on training data and plot your hypothesis function on test data. (label of test data is not allowed to use in your algorithm.) Also, plot the in-sample error versus iteration time. (10 points)

**A few constraints**: No packaged toolkit is allowed and we grade the homework according to the quality and readability of your code.

## 4 Gradient Descent Method

The perceptron learning algorithm works like this: In each iteration $t$, pick a random $\mathbf{x}(t), y(t)$ and compute the *signal* $s(t) = \mathbf{w}^T(t)\mathbf{x}(t)$. If $y(t) \cdot s(t) \leq 0$, update $\mathbf{w}$ by

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}(t) + y(t) \cdot \mathbf{x}(t) \,.$$

(a) Define error as

$$E_n(\mathbf{w}) = \max(0, 1 - y_n \mathbf{w}^T \mathbf{x}_n) \,.$$

Show that $E_n(\mathbf{w})$ is continuous and differentiable except when $y_n = \mathbf{w}^T \mathbf{x}_n$.(5 points)

(b) Show that $E_n(\mathbf{w})$ is an upper bound for $[\text{sign}(\mathbf{w}^T \mathbf{x}_n) \neq y_n]$. Hence, $\frac{1}{N}\sum_{i=1}^{N} E_n(\mathbf{w})$ is an upper bound for the in-sample classification error $E_{\text{in}}(\mathbf{w})$. (5 points)

(c) Apply gradient descent on $\frac{1}{N}\sum_{i=1}^{N} E_n(\mathbf{w})$ (ignoring the singular case $\mathbf{w}^T \mathbf{x}_n = y_n$), plot your results.(10 points)

## 5 Semi-circle toy learning task

Consider the following *toy* learning task in double semi-circles. The width and inner radius of these two semi-circles are *thk* and *rad* respectively, and they are seperated by *sep* (See in Figure 1). The samples in top semi-circle will be labeled as +1(colored as red) while the samples of the bottom semi-circle will be labeled as −1(colored as blue). In the vertical direction, the center of the top semi-circle is aligned with the middle of the edge of the bottom semi-circle.

(a) Set $rad = 10$, $thk = 5$ and $sep = 5$. Generate 2000 samples uniformly on the area. Run the PLA starting from $\mathbf{w} = 0$ utill it converges. Plot the samples and the final hypothesis. To compare, run linear regression (for classification) to obtain $\mathbf{w}$. Explain your observation. (Which algorithm will obtain better results and why). (5 points)

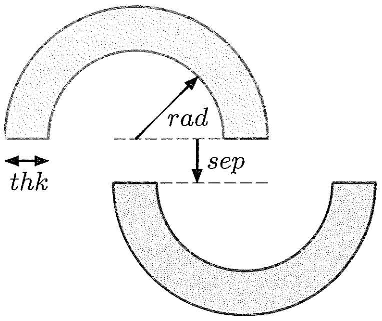

Figure 1: The double semi-circle

(b) Take the same value of $rad$ and $thk$ in (a). Generate 2000 samples for each $sep$ in $\{0.2, 0.4, \ldots, 5\}$. Then plot a figure of $sep$ versus the iteration time for PLA to converge. Explain your observations. (5 points)

(c) Take the same value of $rad$ and $thk$ in (a). Set $sep = -5$ and generate 2000 samples. What will happen if you run PLA on these examples? Run the pocket algorithm (Algorithm.1) for 100000 iterations and plot $E_{\text{in}}$ versus the iteration time $t$. Plot the data and the final hypothesis. (5 points)

**Algorithm 1** The pocket algorithm

```
Set the pocket weight vector $\hat{\mathbf{w}}$ to $\mathbf{w}(0)$ of PLA.
for $t = 0, \ldots, T-1$ do
    Run PLA for one update to obtain $\mathbf{w}(t+1)$.
    Evaluate $E_{\text{in}}(\mathbf{w}(t+1))$.
    If $\mathbf{w}(t+1)$ is better than $\hat{\mathbf{w}}$ in terms of $E_{\text{in}}$, set $\hat{\mathbf{w}}$ to $\mathbf{w}(t+1)$.
end for
Return $\hat{\mathbf{w}}$.
```

(d) In (c), since the generated samples are not linearly separable, PLA cannot converge. Find a method to successfully separate the samples. Plot $E_{\text{in}}$ versus the iteration time $t$ and plot the data and the final hypothesis. [Hint: use 3rd polynomial feature transform. See Section 3.4 Nonlinear Transformation in book *Learning from data* for more details.](10 points)