Pattern Recognition and Machine Intelligence: Assignment 2

Due on **April 14, 2024** at 23:59 pm

Dr. Xiaojuan Qi

TA: *Runyu Ding* *Haozheng Wan*

Problem 1

[30 pts] Please select all of the choices that apply. Please note that there may be more than one correct answer.

AC (a) **[5 pts]** Which of the following statements describing logistic regression are correct?

A. Logistic regression belongs to discriminative models for classification.

B. Logistic regression is an unsupervised clustering algorithm that groups similar data points together.

C. Logistic regression determines a linear decision boundary.

C (b) **[5 pts]** Which of the following statements correctly describes the effect of the learning rate in gradient descent?

A. The learning rate does not affect the convergence of gradient descent.

B. A higher learning rate will always result in faster convergence.

C. The optimal learning rate depends on the specific problem, and setting it too high or too low can lead to slow convergence or non-convergence.

B (c) **[5 pts]** Suppose we have a function $f(x) = x^2 - 6x + 9$ with x as a parameter for optimization. We want to find the minimum value of this function using gradient descent. We start at an initial guess of $x_0 = 4$, with a learning rate of 0.1. Which of the following options correctly computes the gradient of this function, i.e., $\nabla f(x)$, and the updated value of x, i.e., x_1 by taking one-step gradient descent?

A. $\nabla f(x)$=1, x_1=3.8

B. $\nabla f(x)$=2, x_1=3.8

C. $\nabla f(x)$=1, x_1=4.2

D. $\nabla f(x)$=2, x_1=4.2

$\nabla f(x) = 2x - 6$

$\nabla f(x_0) = 2$

$x_{(1)} = x_{(0)} - 0.1 \cdot \nabla f(x_0)$

$= 4 - 2 \cdot 0.1$

C (d) **[5 pts]** What is the true about the SVM?

A. The primary goal of SVM is to minimize classification error.

B. The primary goal of SVM is to maximize the margin between classes. margin between support vectors and decision boundary

C. The support vectors in SVM stand for data points closest to the decision boundary.

AC (e) **[5 pts]** What is true about using the kernel function in SVM?

A. More efficient than computing $\phi()$. ✓

B. Computations is not feasible if h is very high.

C. No Need to explicitly compute $\phi()$. ✓

A (f) **[5 pts]** Consider a Support Vector Machine and the given training data from two classes. Determine the optimal hyperplane. Additionally, discuss whether the removal of x_2 would affect the optimal hyperplane.

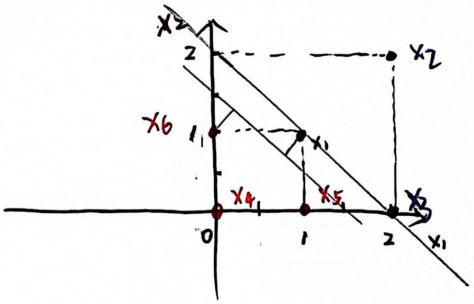
depends on whether x_2 is support vector

A. The optimal hyperplane is $x_1 + x_2 = 3/2$. Removing x_2 will not change the optimal hyperplane.

B. The optimal hyperplane is $x_1 + x_2 = 3/2$. Removing x_2 will change the optimal hyperplane.

C. The optimal hyperplane is $x_1 + x_2 = 2$. Removing x_2 will not change the optimal hyperplane.

D. The optimal hyper-plane is $x_1 + x_2 = 2$. Removing x_2 will change the optimal hyperplane.



category	$\mathbf{x} = (x_1, x_2)^T$
ω_1	$\mathbf{x}_1 = (1,1)^T$
ω_1	$\mathbf{x}_2 = (2,2)^T$
ω_1	$\mathbf{x}_3 = (2,0)^T$
ω_2	$\mathbf{x}_4 = (0,0)^T$
ω_2	$\mathbf{x}_5 = (1,0)^T$
ω_2	$\mathbf{x}_6 = (0,1)^T$

Problem 2

ω_1, ω_2 . Initial $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \Rightarrow x = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$. $\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{pmatrix}$

[35 pts] Suppose you have a dataset of customers who have either purchased a product (represented by ω_1: purchased; ω_2: not purchased). You want to use logistic regression to predict whether a new customer will purchase the product based on their age (x_1) and income (x_2). You fit a logistic regression model with the input $\mathbf{x} = [1\ x_1\ x_2]^T$ and the classifier $\theta = [\theta_0\ \theta_1\ \theta_2]^T = [-2.5\ 0.05\ 0.001]^T$.

(a) **[15 pts]** Please write down the logistic regression function to predict the probability of a customer purchasing the product $P(\omega_1 \mid \mathbf{x})$ as a function of x_1 and x_2.

(b) **[10 pts]** Interpret the meaning of coefficient θ_0 in the logistic regression model.

coefficient n. 系数

(c) **[10 pts]** Suppose a new customer is 30 years old and has an income of 5,000. Use the logistic regression equation to predict the probability that this customer will purchase the product.

a).

$$P(\omega_1|x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

$$= \frac{1}{1+e^{-\sum_{i=1}^{3}\theta_i \cdot x_i}}$$

$$= \frac{1}{1+\exp(+2.5-0.05x_1-0.001x_2)}$$

b)

θ_0 is a coefficient which multiplies $x_0\ (\equiv 1)$ to shape the linear decision boundary.

Owing to constant x_0, which equals to 1, θ_0 in $\theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2$ can be seen as a bias of this hyperplane. Meanwhile, thanks to introducing θ_0, the bias of this hyperplane could not be "0" and it will no longer pass the orgin constantly.

c)

$\because x = \begin{pmatrix} 1 \\ 30 \\ 5000 \end{pmatrix}$

$\therefore \theta^T x = -2.5 + 30 \cdot 0.05 + 0.001 \cdot 5000$

$= -2.5 + 1.5 + 5$

$= 4$

$\because \theta^T x > 0$

$\therefore g(\theta^T x) > 0.5$

$\therefore P(\omega_1|x) > P(\omega_2|x)$

$\therefore$ This customer will purchase this product

Problem 3

[35 pts] Gradient Descent: Consider a two-category classifier with a linear discriminant function $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$ where $\mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = (w_1, w_2)^T$ is the weight vector and $\mathbf{x} = (x_1, x_2)^T$ is the two-dimensional feature vector, we write this linear discriminant function as

$$\begin{aligned} g(\mathbf{x}) &= \mathbf{w}^T\mathbf{x} + w_0 \\ &= \mathbf{W}^T\mathbf{X}, \end{aligned} \qquad \mathbf{W} = \begin{pmatrix} w_1 \\ w_2 \\ w_0 \end{pmatrix} \quad \mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

where, $\mathbf{W} = (w_1, w_2, w_0)^T$ and $\mathbf{X} = (x_1, x_2, 1)^T$.
Given the collected training data denoted as $D = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, the corresponding labels $\{z_1, z_2, ..., z_n\}$ where $z_i = -1$ if x_i belongs to category ω_1 and $z_i = 1$ if x_i belongs to category ω_2. In order to obtain $\mathbf{W}$, we minimize the following loss $\mathcal{J}(\mathbf{W})$ over the training samples:

$$\mathcal{J}(\mathbf{W}) = (\sum_{i=1}^{n} l_i) + \lambda\mathbf{W}^2, \text{where} \quad l_i = \max(0, 1 - z_i g(\mathbf{x}_i)) \qquad \text{Hinge Loss}$$

where l_i is the hinge loss to penalize incorrect predictions or predictions inside the specified margin, and λ is the hyper-parameter to regularize the norm of the weights $\mathbf{W}^2$ to avoid over-fitting (in practice) and ensure uniqueness (in our case). We employ gradient descent to minimize $\mathcal{J}(\mathbf{W})$ and obtain the weights $\mathbf{W} = (w_1, w_2, w_0)^T$. The equation for updating $\mathbf{W}$ at one step k is as below,

$$\mathbf{W}_k = \mathbf{W}_{k-1} - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{W}} |_{\mathbf{W}=\mathbf{W}_{k-1}},$$

where η is the learning rate, $\mathbf{W}_k$ are the weights obtained after the k-th update, and $\frac{\partial \mathcal{J}}{\partial \mathbf{W}}$ is the gradient.
Hint 1: This hinge loss term l_i of a sample i incurs a loss only when "$1 - z_i g(\mathbf{x}_i) > 0$". You should check this condition and discuss different situations when computing the gradient.

Hint 2: If you are not familiar with matrix computations, you can consider expand the loss function:

$$\mathcal{J}(\mathbf{W}) = (\sum_{i=1}^{n} l_i) + \lambda(w_1^2 + w_2^2 + w_0^2), \text{where} \quad l_i = \max(0, 1 - z_i(w_1 x_1^i + w_2 x_2^i + w_0)), \mathbf{x}_i = (x_1^i, x_2^i)^T.$$

And compute the gradient for w_1, w_2, w_0 respectively.

(a) **[15 pts]** Find the gradient $\frac{\partial l_i}{\partial \mathbf{W}}$ with l_i specified as above. And write down $\frac{\partial \mathcal{J}}{\partial \mathbf{W}}$ based on $\frac{\partial l_i}{\partial \mathbf{W}}$.

(b) **[20 pts]** Consider the training data in Problem 1 (a total of six samples for two categories), initial weights $\mathbf{W} = (-1.95, -1.84, 3.20)$, learning rate $\eta = 0.05$ and the regularizer $\lambda = 0.0001$, please try to write down the gradient descent process for several iterations until $|\mathcal{J}(\mathbf{W}_{k-1}) - \mathcal{J}(\mathbf{W}_k)| < 0.06$, and plot the loss curve (you can write a program or draw it by yourself). Compare the final results with the hyper-plane estimated in Problem 1.

(Hint: When performing one gradient descent step, you should finish the following: (1) Calculate the gradient $\frac{\partial l_i}{\partial \mathbf{W}}$ for each sample i. (2) Calculate $\frac{\partial \mathcal{J}}{\partial \mathbf{W}}$ considering all training samples and the regularization term. (3) Use the learning rate and the computed gradient to update the weights of the linear discriminant function. Notice that only a few iterations are needed for convergence.)

ɔion.

a)

$J(w) = \sum_{i=1}^{n} l_i + \lambda \cdot w^2$, where $l_i = \max(0, 1 - z_i \cdot g(x_i))$

Step 1 Compute $\frac{\partial l_i}{\partial w}$

1.1 $\Leftrightarrow$ $1 - z_i \cdot g(x_i) > 0$

$\therefore l_i = 1 - z_i \cdot g(x_i) = 1 - z_i \cdot w^T \cdot x_i$

$\therefore \frac{\partial l_i}{\partial w} = -z_i \cdot x_i$

1.2 $\Leftrightarrow$ $1 - z_i \cdot g(x_i) \leq 0$

$\therefore l_i = 0$

$\therefore \frac{\partial l_i}{\partial w} = 0$

To sum up,

$$\frac{\partial l_i}{\partial w} = \begin{cases} -z_i \cdot x_i, & 1 - z_i \cdot w^T x_i > 0 \\ 0, & \text{otherwise} \end{cases}$$

Step 2 compute $\frac{\partial J}{\partial w}$

$$\frac{\partial J}{\partial w} = \frac{\partial \left(\sum_{i=1}^{n} l_i + \lambda \cdot w^2 \right)}{\partial w} = \sum_{i=1}^{n} \frac{\partial l_i}{\partial w} + \lambda \cdot \frac{\partial (w^T w)}{\partial w}$$

$$= \sum_{i=1}^{n} \frac{\partial l_i}{\partial w} + 2\lambda \cdot w, \quad \text{where } \frac{\partial l_i}{\partial w} \text{ specified above}$$

b)

Initial $D = \left\{ \left(\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, -1 \right), \left(\begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}, -1 \right), \left(\begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}, -1 \right), \left(\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, 1 \right), \left(\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, 1 \right), \left(\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, 1 \right) \right\}$

$W_0 = (-1.95, -1.84, 3.20)$

$\eta = 0.05$, $\lambda = 0.0001$

▲ iteration 1

$\frac{\partial l_i}{\partial w} = \{1, 1, 1, 2, 0, 1\}$, $i \in [6]$

$\frac{\partial J}{\partial w} = \sum_{i=1}^{n} \frac{\partial l_i}{\partial w} + 2\lambda \cdot w = \begin{pmatrix} 2.99961 \\ 0.99963 \\ 2.00064 \end{pmatrix}$

$w = w - \eta \cdot \frac{\partial J}{\partial w} = \begin{pmatrix} -2.09999 \\ -1.88999 \\ 3.09997 \end{pmatrix}$

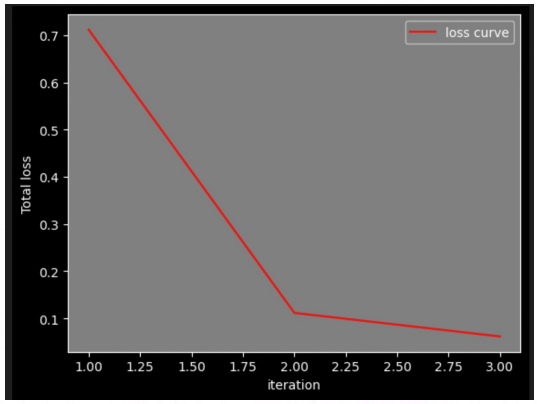
▲ iteration 2

$\frac{\partial l_i}{\partial w} = \{1, 1, 1, -1, 0, -1\}$, $i \in [6]$

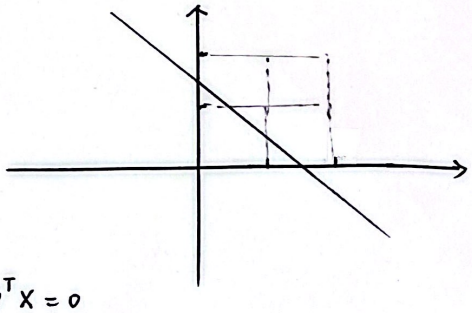
$\frac{\partial J}{\partial w} = \sum_{i=1}^{n} \frac{\partial l_i}{\partial w} + 2\lambda \cdot w = \begin{pmatrix} -4.19961 \times 10^{-4} \\ 9.99622 \times 10^{-1} \\ 6.19993 \times 10^{-4} \end{pmatrix}$

$w = w - \eta \cdot \frac{\partial J}{\partial w} = \begin{pmatrix} -2.09996 \\ -1.93996 \\ 3.09994 \end{pmatrix}$

▲ image of loss curve.



▲



$\because w^T x = 0$

$\therefore$ "$2.09996 x_1 + 1.93996 x_2 = 3.09994$"

approximates "$x_1 + x_2 = \frac{3}{2}$"

$\Rightarrow$ The result fits initial data well.

Problem 4

shared link:

https://colab.research.google.com/drive/1vUwleiLEIK9lQI7pO4jRw2Pl5leeUqw8?usp=sharing