STANFORD UNIVERSITY
CS 229, Autumn 2014
Midterm Examination
Wednesday, November 5, 6:00pm-9:00pm

Question	Points
1 Least Squares	16 /16
2 Generative Learning	16 /16
3 Generalized Linear Models	18 /18
4 Support Vector Regression	16 /16
5 Learning Theory	0 /20
6 Short Answers	21/21
Total	87/107

≈ 81/100

Name of Student: Nguyễn Tuấn Anh

SUNetID: _____@stanford.edu

The Stanford University Honor Code:

I attest that I have not given or received aid in this examination, and that I have done my share and taken an active part in seeing to it that others as well as myself uphold the spirit and letter of the Honor Code.

Signed: _____

Được quét bằng CamScanner

1. **[16 points] Least Squares**

As described in class, in least squares regression we have a cost function:

$$J(\theta) = \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 = (X\theta - \vec{y})^T (X\theta - \vec{y})$$

The goal of least squares regression is to find θ such that we minimize $J(\theta)$ given the training data.

Let's say that we had an original set of n features, so that the training inputs were represented by the design matrix $X \in \mathbb{R}^{m \times (n+1)}$. However, we now gain access to one additional feature for every example. As a result, we now have an additional vector of features $\vec{v} \in \mathbb{R}^{m \times 1}$ for our training set that we wish to include in our regression. We can do this by creating a new design matrix: $\tilde{X} = [X\ \vec{v}] \in \mathbb{R}^{m \times (n+2)}$.

Therefore the new parameter vector is $\theta_{new} = \begin{pmatrix} \theta \\ p \end{pmatrix}$ where $p \in \mathbb{R}$ is the parameter corresponding to the new feature vector $\vec{v}$.

Note: For mathematical simplicity, throughout this problem you can assume that $X^T X = I \in \mathbb{R}^{(n+1)\times(n+1)}$ and $\tilde{X}^T \tilde{X} = I \in \mathbb{R}^{(n+2)\times(n+2)}, \vec{v}^T \vec{v} = 1$. This is called an *orthonormality* assumption – specifically, the columns of $\tilde{X}$ are orthonormal. The conclusions of the problem hold even if we do not make this assumption, but this will make your derivations easier.

(a) [2 points] Let $\hat{\theta} = \arg\min_\theta J(\theta)$ be the minimizer of the original least squares objective (using the original design matrix X). Using the orthornormality assumption, show that $J(\hat{\theta}) = (XX^T\vec{y} - \vec{y})^T(XX^T\vec{y} - \vec{y})$. I.e., show that this is the value of $\min_\theta J(\theta)$ (the value of the objective at the minimum).

- $\hat{\theta} = \arg\min_\theta J(\theta) \Rightarrow \nabla_\theta J(\theta) = 0$ (Assume $J(\theta)$ is convex)
- $\nabla_\theta J(\theta) = \nabla_\theta (X\theta - \vec{y})^T (X\theta - \vec{y})$
  $= \nabla_\theta (\theta^T X^T - \vec{y}^T)(X\theta - \vec{y})$
  $= \nabla_\theta (\theta^T X^T X\theta - \theta^T X^T \vec{y} - \vec{y}^T X\theta + \vec{y}^T\vec{y})$
  $= \nabla_\theta (\theta^T X^T X\theta - 2\vec{y}^T X\theta + \vec{y}^T\vec{y})$
  $= \nabla_\theta \theta^T X^T X\theta - 2\nabla_\theta \vec{y}^T X\theta$
  $= 2X^T X\theta - 2\vec{y}^T X$
  $= 0$ ($X^T X = I$)

$\Leftrightarrow X^T X\theta = \vec{y}^T X$

$\Leftrightarrow \theta = X^T\vec{y} \Rightarrow \hat{\theta} = \arg\min_\theta J(\theta) \Leftrightarrow \boxed{\hat{\theta} = X^T y}$

$\Rightarrow \boxed{J(\hat{\theta}) = (XX^T\vec{y} - \vec{y})^T (XX^T\vec{y} - \vec{y})}$ (Done)

Được quét bằng CamScanner

(b) [5 points] Now let $\hat{\theta}_{new}$ be the minimizer for $\tilde{J}(\theta_{new}) = (\tilde{X}\theta_{new} - \vec{y})^T(\tilde{X}\theta_{new} - \vec{y})$. Find the new minimized objective $\tilde{J}(\hat{\theta}_{new})$ and write this expression in the form: $\tilde{J}(\hat{\theta}_{new}) = J(\hat{\theta}) + f(X, \vec{v}, \vec{y})$ where $J(\hat{\theta})$ is as derived in part (a) and f is some function of $X, \vec{v}$, and $\vec{y}$.

- From part a, we have:

$$J(\hat{\theta}) = (XX^T\vec{y} - \vec{y})^T (XX^T\vec{y} - \vec{y})$$

$$= (\vec{y}^T XX^T - \vec{y}^T)(XX^T\vec{y} - \vec{y})$$

$$= \cancel{\vec{y}^T XX^T XX^T\vec{y}} \; \overset{II}{} \; \cancel{- \vec{y}^T XX^T\vec{y}} - \vec{y}^T XX^T\vec{y} + \vec{y}^T\vec{y}$$

$$= -\vec{y}^T XX^T\vec{y} + \vec{y}^T\vec{y}$$

$$\Rightarrow \boxed{J(\hat{\theta}) = -\vec{y}^T XX^T\vec{y} + \vec{y}^T\vec{y}} \quad (1)$$

- We know that (1) is TRUE with all X, also we have $\tilde{J}(\hat{\theta}_{new})$:

$$\tilde{J}(\hat{\theta}_{new}) = -\vec{y}^T \tilde{X}\tilde{X}^T\vec{y} + \vec{y}^T\vec{y}$$

$$= -\vec{y}^T [X \; \vec{w}] \begin{bmatrix} X^T \\ \vec{w}^T \end{bmatrix} \vec{y} + \vec{y}^T\vec{y}$$

$$= -\vec{y}^T \left(XX^T + \vec{w}\vec{w}^T \right) \vec{y} + \vec{y}^T\vec{y}$$

$$= \underbrace{-\vec{y}^T XX^T\vec{y} + \vec{y}^T\vec{y}}_{= J(\hat{\theta})} \underbrace{- \vec{y}^T\vec{w}\vec{w}^T\vec{y}}_{f(X, \vec{w}, \vec{y})}$$

$$\Rightarrow \boxed{\tilde{J}(\hat{\theta}_{new}) = J(\hat{\theta}) + f(X, \vec{w}, \vec{y}) \quad \text{where } f(X, \vec{w}, \vec{y}) = -\vec{y}^T\vec{w}\vec{w}^T\vec{y}}$$

(Done)

Được quét bằng CamScanner

(c) [6 points] Prove that the optimal objective value does not increase upon adding a feature to the design matrix. That is, show $\tilde{J}(\hat{\theta}_{new}) \leq J(\hat{\theta})$.

- From part b, we have $J(\hat{\theta})$ and $\tilde{J}(\hat{\theta}_{new})$ relation:

$$\begin{aligned}\tilde{J}(\hat{\theta}_{new}) &= J(\hat{\theta}) - \vec{y}^T \vec{\omega}\, \vec{\omega}^T \vec{y} \\ &= J(\hat{\theta}) - (\vec{\omega}^T \vec{y})^T \vec{\omega}^T \vec{y} \\ &= J(\hat{\theta}) - \|\vec{\omega}^T \vec{y}\|^2\end{aligned}$$

- We see that $\|\vec{\omega}^T \vec{y}\|^2 \geq 0 \Rightarrow -\|\vec{\omega}^T \vec{y}\|^2 \leq 0$, so:

$$\boxed{\tilde{J}(\hat{\theta}_{new}) \leq J(\hat{\theta})}$$

(Done)

(d) [3 points] Does the above result show that if we keep increasing the number of features, we can always get a model that generalizes better than a model with fewer features? Explain why or why not.

- NO!
- The result $\tilde{J}(\hat{\theta}_{new}) \leq J(\hat{\theta})$ just show us that if the #feature increase, then that lead to Model is more complex $\Rightarrow$ Tend to Overfitting.

$\Rightarrow$ # feature $\uparrow \Rightarrow$ Training Error $\downarrow$, that not mean Generalize Error (on Testset) is better because of noise feature.

(Done)

Được quét bằng CamScanner

2. **[16 points] Decision Boundaries for Generative Models**

(a) [8 points] Consider the *multinomial event model* of Naive Bayes. Our goal in this problem is to show that this is a linear classifier.

For a given text document x, let $c_1, \ldots, c_V$ indicate the number of times each word (out of V words) appears in the document. Thus, $c_i \in \{0, 1, 2, \ldots\}$ counts the occurrences of word i. Recall that the Naive Bayes model uses parameters $\phi_y = p(y = 1)$, $\phi_{i|y=1} = p(\text{word i appears in a specific document position} \mid y = 1)$ and $\phi_{i|y=0} = p(\text{word i appears in a specific document position} \mid y = 0)$.

We say a classifier is linear if it assigns a label $y = 1$ using a decision rule of the form

$$\sum_{i=1}^{V} w_i c_i + b \geq 0$$

I.e., the classifier predicts "$y = 1$" if $\sum_{i=1}^{V} w_i c_i + b \geq 0$, and predicts $y = 0$ otherwise.

Show that Naive Bayes is a linear classifier, and clearly state the values of w_i and b in terms of the Naive Bayes parameters. (Don't worry about whether the decision rule uses "$\geq$" or "$>$.") Hint: consider using log-probabilities.

- Apply Bayes Rule for both class 1 and 0, we have:

$$P(y=1|x) = \frac{P(x|y=1)\,P(y=1)}{P(x)} \qquad P(y=0|x) = \frac{P(x|y=0)\,P(y=0)}{P(x)}$$

- Compare $P(y=1|x)$ and $P(y=0|x)$, we have:

$$\text{Log}\left[\frac{P(y=1|x)}{P(y=0|x)}\right] = \text{Log}\left[\frac{P(x|y=1)\,P(y=1)}{P(x|y=0)\,P(y=0)}\right]$$

$$= \text{Log}\,P(x|y=1) + \text{Log}\,P(y=1) - \text{Log}\,P(x|y=0) - \text{Log}\,P(y=0)$$

$$= \text{Log}\left[\prod_{i=1}^{V} P(x_i|y=1)^{c_i}\right] + \text{Log}\,P(y=1) - \text{Log}\left[\prod_{i=1}^{V} P(x_i|y=0)^{c_i}\right] - \text{Log}\,P(y=0)$$

$$= \sum_{i=1}^{V} c_i \text{Log}(\phi_{i|y=1}) + \text{Log}\,\phi_y - \sum_{i=1}^{V} c_i \text{Log}(\phi_{i|y=0}) - \text{Log}(1-\phi_y)$$

$$= \sum_{i=1}^{V} c_i \underbrace{\text{Log}\left(\frac{\phi_{i|y=1}}{\phi_{i|y=0}}\right)}_{=w_i} + \underbrace{\text{Log}\left(\frac{\phi_y}{1-\phi_y}\right)}_{=b} \geq 0$$

So, classifier predict $y=1$ is:

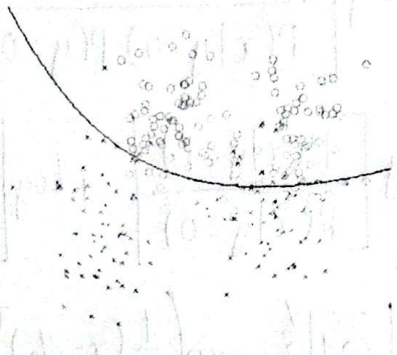
$$\sum_{i=1}^{V} w_i c_i + b \geq 0, \text{ where } \begin{cases} w_i = \text{Log}\left(\frac{\phi_{i|y=1}}{\phi_{i|y=0}}\right) \\ b = \text{Log}\left(\frac{\phi_y}{1-\phi_y}\right) \end{cases}$$

(Done)

Được quét bằng CamScanner

(b) [8 points] In Problem Set 1, you showed that Gaussian Discriminant Analysis (GDA) is a linear classifier. In this problem, we will show that a modified version of GDA has a quadratic decision boundary.

Recall that GDA models $p(x|y)$ using a multivariate normal distribution, where $(x|y=0) \sim \mathcal{N}(\mu_0, \Sigma)$ and $(x|y=1) \sim \mathcal{N}(\mu_1, \Sigma)$, where we used the same Σ for both Gaussians. For this question, we will instead use two covariance matrices Σ_0, Σ_1 for the two labels. So, $(x|y=0) \sim \mathcal{N}(\mu_0, \Sigma_0)$ and $(x|y=1) \sim \mathcal{N}(\mu_1, \Sigma_1)$.



The model distributions can now be written as:

$$p(y) = \phi^y(1-\phi)^{1-y}$$

$$p(x|y=0) = \frac{1}{(2\pi)^{n/2}|\Sigma_0|^{1/2}} \exp\Big(-\frac{1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0)\Big)$$

$$p(x|y=1) = \frac{1}{(2\pi)^{n/2}|\Sigma_1|^{1/2}} \exp\Big(-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\Big)$$

Let's follow a binary decision rule, where we predict $y=1$ if $p(y=1|x) \geq p(y=0|x)$, and $y=0$ otherwise. Show that if $\Sigma_0 \neq \Sigma_1$, then the separating hyperplane is quadratic in x.

That is, simplify the decision rule "$p(y=1|x) \geq p(y=0|x)$" to the form "$x^TAx + B^Tx + C \geq 0$" (supposing that $x \in \mathbb{R}^{n+1}$), for some $A \in \mathbb{R}^{(n+1)\times(n+1)}$, $B \in \mathbb{R}^{n+1}$, $C \in \mathbb{R}$ and $A \neq 0$. Please clearly state your values for A, B and C.

[Solve in next Page]

Được quét bằng CamScanner

- We simplify "$P(y=1|x) \geq P(y=0|x)$" like below:

$$\frac{P(y=1|x)}{P(y=0|x)} \geq 1 \Leftrightarrow \text{Log}\left[\frac{P(y=1|x)}{P(y=0|x)}\right] \geq 0 \quad (*)$$

- Now, we simplify (*) by:

$$\text{Log}\left[\frac{P(y=1|x)}{P(y=0|x)}\right] \overset{\text{Bayes Rule}}{=} \text{Log}\left[\frac{P(x|y=1)\,P(y=1)}{P(x|y=0)\,P(y=0)}\right]$$

$$= \text{Log}\left[\frac{P(x|y=1)}{P(x|y=0)}\right] + \text{Log}\left[\frac{P(y=1)}{P(y=0)}\right]$$

$$= \text{Log}\left[\left|\frac{\Sigma_1}{\Sigma_0}\right|^{\frac{1}{2}} \cdot \exp\left(\frac{1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0) - \frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right)\right] + \text{Log}\left(\frac{\phi}{1-\phi}\right)$$

$$= \frac{1}{2}\text{Log}\left|\frac{\Sigma_1}{\Sigma_0}\right| + \frac{1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0) - \frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) + \text{Log}\left(\frac{\phi}{1-\phi}\right)$$

$$= \frac{1}{2}\text{Log}\left|\frac{\Sigma_1}{\Sigma_0}\right| + \frac{1}{2}\left(x^T\Sigma_0^{-1}x - 2\mu_0^T\Sigma_0^{-1}x + \mu_0^T\Sigma_0^{-1}\mu_0\right) - \frac{1}{2}\left(x^T\Sigma_1^{-1}x - 2\mu_1^T\Sigma_1^{-1}x + \mu_1^T\Sigma_1^{-1}\mu_1\right) + \text{Log}\left(\frac{\phi}{1-\phi}\right)$$

$$= \frac{1}{2}\text{Log}\left|\frac{\Sigma_1}{\Sigma_0}\right| + \frac{1}{2}x^T\left(\Sigma_0^{-1} - \Sigma_1^{-1}\right)x + \left(\mu_1^T\Sigma_1^{-1} - \mu_0\Sigma_0^{-1}\right)x + \frac{1}{2}\mu_0^T\Sigma_0^{-1}\mu_0 - \frac{1}{2}\mu_1^T\Sigma_1^{-1}\mu_1 + \text{Log}\left(\frac{\phi}{1-\phi}\right)$$

$$= x^T \cdot \underbrace{\frac{1}{2}\left(\Sigma_0^{-1} - \Sigma_1^{-1}\right)}_{=A} \cdot x + \underbrace{\left(\mu_1^T\Sigma_1^{-1} - \mu_0\Sigma_0^{-1}\right)}_{=B^T} x + \underbrace{\frac{1}{2}\text{Log}\left|\frac{\Sigma_1}{\Sigma_0}\right| + \frac{1}{2}\mu_0^T\Sigma_0^{-1}\mu_0 - \frac{1}{2}\mu_1^T\Sigma_1^{-1}\mu_1 + \text{Log}\left(\frac{\phi}{1-\phi}\right)}_{=C} \quad \text{(Done)}$$

$$= x^T \cdot A \cdot x + B^T \cdot x + C \geq 0$$

So, $P(y=1|x) \geq P(y=0|x)$ can write as Quadratic Form $x^TAx + B^Tx + C \geq 0$,

with
$$\begin{cases} A = \frac{1}{2}\left(\Sigma_0^{-1} - \Sigma_1^{-1}\right) \\ B^T = \mu_1^T\Sigma_1^{-1} - \mu_0\Sigma_0^{-1} \\ C = \frac{1}{2}\text{Log}\left|\frac{\Sigma_1}{\Sigma_0}\right| + \frac{1}{2}\mu_0^T\Sigma_0^{-1}\mu_0 - \frac{1}{2}\mu_1^T\Sigma_1^{-1}\mu_1 + \text{Log}\left(\frac{\phi}{1-\phi}\right) \end{cases}$$

Được quét bằng CamScanner

3. **[18 points] Generalized Linear Models**
In this problem you will build a Generalized Linear Model (GLM) for a response variable y, whose distribution (parameterized by ϕ) is modeled as:

$$p(y;\phi) = (1-\phi)^{y-1}\phi$$

This distribution is known as the *geometric distribution*, and is used to model network connections and many other problems.

(a) i. [5 points] Show that the geometric distribution is an exponential family distribution. You should explicitly specify $b(y)$, η, $T(y)$, $\alpha(\eta)$. Also specify what ϕ is in terms of η.

- GLM formular w.r.t η:

$$P(y;\eta) = b(y)\cdot \exp\left[\eta^T T(y) - a(\eta)\right] \quad (1)$$

- We need to derive $p(y;\phi)$ into (1) form:

$$\begin{aligned} p(y;\phi) &= (1-\phi)^{y-1}\phi \\ &= \exp\left[\text{Log}(1-\phi)^{y-1} + \text{Log}\,\phi\right] \\ &= \exp\left[(y-1)\text{Log}(1-\phi) + \text{Log}\,\phi\right] \\ &= \exp\left[y\,\text{Log}(1-\phi) + \text{Log}\left(\frac{\phi}{1-\phi}\right)\right] \end{aligned}$$

$$\Rightarrow p(y;\phi) = \exp\left[y\,\text{log}(1-\phi) + \text{Log}\left(\frac{\phi}{1-\phi}\right)\right] \quad (2)$$

- From (1) & (2), we have:

$b(y) = 1$ | $T(y) = y$ | $\eta = \text{Log}(1-\phi)$

$\phi = 1 - e^{\eta}$

$$a(\eta) = -\text{Log}\left(\frac{\phi}{1-\phi}\right)$$

$$= -\text{Log}\left(\frac{1-e^{\eta}}{e^{\eta}}\right) \Rightarrow \boxed{a(\eta) = -\text{Log}(1-e^{\eta}) + \eta}$$

(Done)

Được quét bằng CamScanner

ii. [5 points] Suppose that we have an IID training set $\{(x^{(i)}, y^{(i)}), i = 1, ..., m\}$ and we wish to model this using a GLM based on a geometric distribution. Find the log-likelihood $\log \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}; \theta)$ defined with respect to the entire training set.

- Assumptions to Construct GLM:

① $y^{(i)}|x^{(i)}; \theta \sim$ Exponential Family (η)

② $\eta^{(i)} = \theta^T x^{(i)}$ —Part a→ $\boxed{\phi^{(i)} = 1 - e^{\theta^T x^{(i)}}}$

- From part a, we see that $p(y, \phi)$ is actually in Exponential Family (η), so:

$$\ell(\theta) = \log\left[\prod_{i=1}^{m} P(y^{(i)}|x^{(i)}; \theta)\right] = \sum_{i=1}^{m} \log P(y^{(i)}|x^{(i)}; \theta)$$

$$= \sum_{i=1}^{m} \log P(y^{(i)}; \phi^{(i)})$$

$$= \sum_{i=1}^{m} \log\left[(1-\phi^{(i)})^{y^{(i)}-1} \cdot \phi^{(i)}\right]$$

$$= \sum_{i=1}^{m} \left[(y^{(i)}-1)\log(1-\phi^{(i)}) + \log\phi^{(i)}\right]$$

$$= \sum_{i=1}^{m} \left[(y^{(i)}-1)\log(e^{\theta^T x^{(i)}}) + \log(1-e^{\theta^T x^{(i)}})\right]$$

$$\Rightarrow \boxed{\ell(\theta) = \sum_{i=1}^{m} \left[(y^{(i)}-1)\theta^T x^{(i)} + \log(1-e^{\theta^T x^{(i)}})\right]}$$ (Done)

Vectorization ↓

$$\boxed{\ell(\theta) = (\vec{y}-1)^T X\theta + \log(1-e^{X\theta})}$$

Note

$X \in \mathbb{R}^{m \times n}$

$\vec{y} \in \mathbb{R}^m$

$\theta \in \mathbb{R}^n$

Được quét bằng CamScanner

(b) [6 points] Derive the Hessian H and the gradient vector of the log likelihood, and state what one step of Newton's method for maximizing the log likelihood would be.

- From part a ii, we have Vectorization form of Log Likelihood:

$$\ell(\theta) = (\vec{y}-1)^T X\theta + \text{Log}(1-e^{X\theta})$$

- Taking Derivative of $\ell(\theta)$ w.r.t θ, we have:

$$\nabla_\theta \ell(\theta) = \nabla_\theta\left[(\vec{y}-1)^T X\theta + \text{Log}(1-e^{X\theta})\right]$$

$$= \nabla_\theta\left[\vec{y}^T X\theta - 1^T X\theta + \text{Log}(1-e^{X\theta})\right]$$

$$= \nabla_\theta \vec{y}^T X\theta - \nabla_\theta 1^T X\theta + \nabla_\theta \text{Log}(1-e^{X\theta})$$

$$= X^T\vec{y} - X^T - X^T.\frac{e^{X\theta}}{1-e^{X\theta}}$$

$$= X^T\left[(\vec{y}-1) - \frac{e^{X\theta}}{1-e^{X\theta}}\right]$$

$$\Rightarrow \boxed{\nabla_\theta \ell(\theta) = X^T\left[(\vec{y}-1) - \frac{e^{X\theta}}{1-e^{X\theta}}\right]}$$ "Gradient Vector Log-Likelihood"

- To Derive the Hessian H, we take Derivative of $\nabla_\theta \ell(\theta)$ w.r.t θ, we have:

$$H = \nabla_\theta^2 \ell(\theta) = \nabla_\theta\left[\nabla_\theta \ell(\theta)\right]$$

$$= \nabla_\theta\left[X^T\left(\vec{y} - 1 - \frac{e^{X\theta}}{1-e^{X\theta}}\right)\right]$$

$$= \nabla_\theta\left[-X^T\frac{e^{X\theta}}{1-e^{X\theta}}\right]$$

$$= -X^T.\nabla_\theta\left(\frac{e^{X\theta}}{1-e^{X\theta}}\right)$$

$$= -X^T.\frac{X^T e^{X\theta}}{(1-e^{X\theta})^T(1-e^{X\theta})}$$

$$\Rightarrow \boxed{H = -X^T.\,\text{diag}\left[\frac{e^{X\theta}}{(1-e^{X\theta})^T(1-e^{X\theta})}\right].X}$$ "Hessian Matrix"

Được quét bằng CamScanner

(c) [2 points] Show that the Hessian is negative semi-definite. This shows the optimization objective is concave, and hence Newton's method is maximizing log-likelihood.

- From part b, we have the Vectorization Form of Hessian Matrix:

$$H = -X^T \cdot \text{diag}\left[\frac{e^{X\theta}}{(1-e^{X\theta})^T(1-e^{X\theta})}\right] \cdot X$$

- The basic form of H w.r.t $x^{(i)}$ is:

$$H = -\sum_{i=1}^{m}\left[\frac{e^{\theta^T x^{(i)}}}{(1-e^{\theta^T x^{(i)}})^2}\right] x^{(i)2}$$

- H is NSD $\Leftrightarrow$ $z^T H z \leq 0$, $\forall z \in \mathbb{R}^n$:

$$z^T H z = \sum_{t=1}^{n}\sum_{k=1}^{n} H_{tk} \cdot z_k \cdot z_t$$

$$= \sum_{t=1}^{n}\sum_{k=1}^{n}\left[-\sum_{i=1}^{m}\frac{e^{\theta^T x^{(i)}}}{(1-e^{\theta^T x^{(i)}})^2} \cdot x_k^{(i)} \cdot x_t^{(i)}\right] \cdot z_k z_t$$

$$= (X^T z)^2\left[-\sum_{i=1}^{m}\frac{e^{\theta^T x^{(i)}}}{(1-e^{\theta^T x^{(i)}})^2}\right]$$

$$= -\underbrace{(X^T z)^2}_{\geq 0}\left[\sum_{i=1}^{m}\frac{\overbrace{e^{\theta^T x^{(i)}}}^{>0}}{\underbrace{(1-e^{\theta^T x^{(i)}})^2}_{\geq 0}}\right]$$

$$\leq 0, \quad \forall z \in \mathbb{R}^n$$

So, H is Negative Semi-definite

(Done)

Được quét bằng CamScanner

4. **[16 points] Support Vector Regression**

In class, we showed how the SVM can be used for classification. In this problem, we will develop a modified algorithm, called the Support Vector Regression algorithm, which can instead be used for regression, with continuous valued labels $y \in \mathbb{R}$.

Suppose we are given a training set $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$, where $x^{(i)} \in \mathbb{R}^{(n+1)}$ and $y^{(i)} \in \mathbb{R}$. We would like to find a hypothesis of the form $h_{w,b}(x) = w^T x + b$ with a small value of w. Our (convex) optimization problem is:

$$\begin{array}{lll} \min_{w,b} & \frac{1}{2}\|w\|^2 & \\ \text{s.t.} & y^{(i)} - w^T x^{(i)} - b \leq \epsilon & i = 1, \ldots, m \quad (1) \\ & w^T x^{(i)} + b - y^{(i)} \leq \epsilon & i = 1, \ldots, m \quad (2) \end{array}$$

where $\epsilon > 0$ is a given, fixed value. Notice how the original functional margin constraint has been modified to now represent the distance between the continuous y and our hypothesis' output.

(a) [3 points] Write down the Lagrangian for the optimization problem above. We suggest you use two sets of Lagrange multipliers α_i and α_i^*, corresponding to the two inequality constraints (labeled (1) and (2) above), so that the Lagrangian would be written $\mathcal{L}(w, b, \alpha, \alpha^*)$.

- Apply Lagrangian Multiplier for many conditions, we have:

Find $\min_{w,b}$ of $f(w,b) = \frac{1}{2}\|w\|^2$ w.r.t $\begin{cases} g(w,b) = y^{(i)} - w^T x^{(i)} - b - \varepsilon \leq 0 \\ h(w,b) = w^T x^{(i)} + b - y^{(i)} - \varepsilon \leq 0 \end{cases}$, $\forall i \in [1, m]$

$$\mathcal{L}(w, b, \alpha, \alpha^*) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{m} \alpha_i \left(y^{(i)} - w^T x^{(i)} - b - \varepsilon\right) + \sum_{i=1}^{m} \alpha_i^* \left(w^T x^{(i)} + b - y^{(i)} - \varepsilon\right)$$

$$\Rightarrow \boxed{\mathcal{L}(w, b, \alpha, \alpha^*) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{m} \left[(\alpha_i - \alpha_i^*) y^{(i)} + (-\alpha_i - \alpha_i^*)\varepsilon + (\alpha_i^* - \alpha_i) b + (\alpha_i^* - \alpha_i) w^T x^{(i)}\right]}$$

(Done)

Được quét bằng CamScanner

(b) [9 points] Derive the dual optimization problem. You will have to take derivatives of the Lagrangian with respect to w and b.

- From part a, we have the Lagrangian Multiplier Function:

$$\mathcal{L}(w,b,\alpha,\alpha^*) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{m}\left[(\alpha_i-\alpha_i^*)y^{(i)} + (-\alpha_i-\alpha_i^*)\varepsilon + (\alpha_i^*-\alpha_i)b + (\alpha_i^*-\alpha_i)w^T x^{(i)}\right]$$

- Take Derivative of $\mathcal{L}(w,b,\alpha,\alpha^*)$ w.r.t w, we have:

$$\nabla_w \mathcal{L}(w,b,\alpha,\alpha^*) = \nabla_w \frac{1}{2}\|w\|^2 + \nabla_w\left[\sum_{i=1}^{m}(\alpha_i^*-\alpha_i)w^T x^{(i)}\right]$$

$$= w + \sum_{i=1}^{m}(\alpha_i^*-\alpha_i)x^{(i)}$$

$$= 0$$

$$\Rightarrow \boxed{w = \sum_{i=1}^{m}(\alpha_i-\alpha_i^*)x^{(i)}} \quad (1)$$

- Take Derivative of $\mathcal{L}(w,b,\alpha,\alpha^*)$ w.r.t b, we have:

$$\nabla_b \mathcal{L}(w,b,\alpha,\alpha^*) = \nabla_b\left[\sum_{i=1}^{m}(\alpha_i^*-\alpha_i)b\right]$$

$$= \sum_{i=1}^{m}(\alpha_i^*-\alpha_i) \quad (\text{Pone})$$

$$= 0$$

$$\Rightarrow \boxed{\sum_{i=1}^{m}(\alpha_i^*-\alpha_i) = 0} \quad (2)$$

- Change (1), (2), $\|w\|^2 = w^T w$ into $\mathcal{L}(w,b,\alpha,\alpha^*)$, we have:

$$\mathcal{L}(w,b,\alpha,\alpha^*) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)x^{(i)T}x^{(j)} + \sum_{i=1}^{m}(\alpha_i-\alpha_i^*)y^{(i)} + \sum_{i=1}^{m}(-\alpha_i-\alpha_i^*)\varepsilon$$

Dual Optimization Problem:

$$\max_{\alpha,\alpha^*}\left[-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)x^{(i)T}x^{(j)} + \sum_{i=1}^{m}(\alpha_i-\alpha_i^*)y^{(i)} + \sum_{i=1}^{m}(-\alpha_i-\alpha_i^*)\varepsilon\right]$$

subject to

$$\alpha \geq 0;\ \alpha^* \geq 0;\ \sum_{i=1}^{m}(\alpha_i^*-\alpha_i) = 0$$

Được quét bằng CamScanner

[more space for problem 4 (b)]

(c) [4 points] Show that this algorithm can be kernelized. For this, you have to show that (i) the dual optimization objective can be written in terms of inner-products of training examples; and (ii) at test time, given a new x the hypothesis $h_{w,b}(x)$ can also be computed in terms of inner products.

(i)

• From part b, we re-write the Optimal Dual Problem into inner-product:

$$\max_{\alpha,\alpha^*} \left[-\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x^{(i)}, x^{(j)} \rangle + \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) y^{(i)} + \sum_{i=1}^{m} (-\alpha_i - \alpha_i^*) \varepsilon \right]$$

subject to

$$\alpha \geq 0 \; ; \; \alpha^* \geq 0 ; \; \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) = 0$$

(ii)

• At test time, the Inner Product of $h_{w,b}(x)$ is:

$$h_{w,b}(x) = g(w^T x + b)$$

$$= g\left(\sum_{i=1}^{m} (\alpha_i - \alpha_i^*) x^{(i)} \cdot x + b \right)$$

$$\Rightarrow \text{Test Time: } h_{w,b}(x) = g\left(\sum_{i=1}^{m} (\alpha_i - \alpha_i^*) \langle x^{(i)}, x \rangle + b \right) \quad \text{(Done)}$$

Được quét bằng CamScanner

5. **[20 points] Learning Theory**

Suppose you are given a hypothesis $h_0 \in \mathcal{H}$, and your goal is to determine whether h_0 has generalization error within $\eta > 0$ of the best hypothesis, $h^* = \arg\min_{h \in \mathcal{H}} \varepsilon(h)$. More specifically, we say that a hypothesis h is **η-optimal** if $\varepsilon(h) \leq \varepsilon(h^*) + \eta$. Here, we wish to answer the following question:

Given a hypothesis h_0, is h_0 η-optimal?

Let $\delta > 0$ be some fixed constant, and consider a finite hypothesis class $\mathcal{H}$ of size $|\mathcal{H}| = k$. For each $h \in \mathcal{H}$, let $\hat{\varepsilon}(h)$ denote the training error of h with respect to some training set of m IID examples, and let $\hat{h} = \arg\min_{h \in \mathcal{H}} \hat{\varepsilon}(h)$ denote the hypothesis that minimizes training error.

Now, consider the following algorithm:

1. Set
$$\gamma := \sqrt{\frac{1}{2m} \log \frac{2k}{\delta}}.$$
2. If $\hat{\varepsilon}(h_0) > \hat{\varepsilon}(\hat{h}) + \eta + 2\gamma$, then return NO.
3. If $\hat{\varepsilon}(h_0) < \hat{\varepsilon}(\hat{h}) + \eta - 2\gamma$, then return YES.
4. Otherwise, return UNSURE.

Intuitively, the algorithm works by comparing the training error of h_0 to the training error of the hypothesis $\hat{h}$ with the minimum training error, and returns NO or YES only when $\hat{\varepsilon}(h_0)$ is either significantly larger than or significantly smaller than $\hat{\varepsilon}(\hat{h}) + \eta$.

(a) [6 points] First, show that if $\varepsilon(h_0) \leq \varepsilon(h^*) + \eta$ (i.e., h_0 is η-optimal), then the probability that the algorithm returns NO is at most δ.

Được quét bằng CamScanner

- Apply Union Bound & Hoeffding's inequality for h_0, we have:

$$P[\hat{\mathcal{E}}(h_0) - \mathcal{E}(h_0) > \gamma] \leq \delta/2$$

$$\Leftrightarrow P[\hat{\mathcal{E}}(h_0) > \mathcal{E}(h_0) + \gamma] \leq \delta/2$$

Condition: $\mathcal{E}(h_0) \leq \mathcal{E}(h^*) + \eta$

$$\Rightarrow P[\hat{\mathcal{E}}(h_0) < \mathcal{E}(h_0) + \gamma] \geq 1 - \delta/2$$

Fact: $\mathcal{E}(h^*) \leq \mathcal{E}(\hat{h})$

$$\Leftrightarrow P[\hat{\mathcal{E}}(h_0) < \mathcal{E}(h^*) + \eta + \gamma] \geq 1 - \delta/2$$

$$\Leftrightarrow P[\hat{\mathcal{E}}(h_0) < \mathcal{E}(\hat{h}) + \eta + \gamma] \geq 1 - \delta/2$$

$$\Leftrightarrow P[\hat{\mathcal{E}}(h_0) > \mathcal{E}(\hat{h}) + \eta + \gamma] \leq 1 - (1 - \delta/2) = \delta/2 \quad (1)$$

- Apply Union Bound & Hoeffding's inequality for $\hat{h}$, we have:

$$P[\mathcal{E}(\hat{h}) - \hat{\mathcal{E}}(\hat{h}) > \gamma] \leq \delta/2$$

$$\Leftrightarrow P[\mathcal{E}(\hat{h}) > \hat{\mathcal{E}}(\hat{h}) + \gamma] \leq \delta/2 \quad (2)$$

- Apply Union Bound for (1) and (2), we have:

Let $X = \{\hat{\mathcal{E}}(h_0) > \mathcal{E}(\hat{h}) + \eta + \gamma\}$; $Y = \{\mathcal{E}(\hat{h}) > \hat{\mathcal{E}}(\hat{h}) + \gamma\}$

$$\Rightarrow P(X \cup Y) \leq P(X) + P(Y) \leq \delta/2 + \delta/2 = \delta$$

$$\Leftrightarrow P(X \cup Y) = P[\hat{\mathcal{E}}(h_0) > \hat{\mathcal{E}}(\hat{h}) + \eta + 2\gamma] \leq \delta$$

So, if $\mathcal{E}(h_0) \leq \mathcal{E}(h^*) + \eta$, then $P[\hat{\mathcal{E}}(h_0) > \hat{\mathcal{E}}(\hat{h}) + \eta + 2\gamma] \leq \delta$

or $P[NO]$ atmost δ

(Done)

Được quét bằng CamScanner

(b) [6 points] Second, show that if $\varepsilon(h_0) > \varepsilon(h^*) + \eta$ (i.e., h_0 is not η-optimal), then the probability that the algorithm returns YES is at most δ.

- Apply Union Bound & Hoeffding's inequality for h_0, we have:

$$P[\varepsilon(h_0) - \hat{\varepsilon}(h_0) > \gamma] \leq \delta/2$$

$$\Leftrightarrow P[\hat{\varepsilon}(h_0) < \varepsilon(h_0) - \gamma] \leq \delta/2$$

$$\Rightarrow P[\hat{\varepsilon}(h_0) > \varepsilon(h_0) - \gamma] \geq 1 - \delta/2$$

$$\Leftrightarrow P[\hat{\varepsilon}(h_0) > \varepsilon(h^*) + \eta - \gamma] \geq 1 - \delta/2$$

$$\Rightarrow P[\hat{\varepsilon}(h_0) < \varepsilon(h^*) + \eta - \gamma] \leq 1 - (1 - \delta/2) = \delta/2$$

$$\Leftrightarrow P[\hat{\varepsilon}(h_0) < \varepsilon(\hat{h}) + \eta - \gamma] \leq \delta/2 \quad (1)$$

Condition: $\varepsilon(h_0) > \varepsilon(h^*) + \eta$

Fact: $\varepsilon(h^*) \leq \varepsilon(\hat{h})$

- Apply Union Bound & Hoeffding's inequality for $\hat{h}$, we have:

$$P[\hat{\varepsilon}(\hat{h}) - \varepsilon(\hat{h}) > \gamma] \leq \delta/2$$

$$\Leftrightarrow P[\varepsilon(\hat{h}) < \hat{\varepsilon}(\hat{h}) - \gamma] \leq \delta/2 \quad (2)$$

- Apply Union Bound for (1) and (2), we have:

Let $X = \{\hat{\varepsilon}(h_0) < \varepsilon(\hat{h}) + \eta - \gamma\}$; $Y = \{\varepsilon(\hat{h}) < \hat{\varepsilon}(\hat{h}) - \gamma\}$

$$\Rightarrow P(X \cup Y) \leq P(X) + P(Y) \leq \delta/2 + \delta/2 = \delta$$

$$\Leftrightarrow P(X \cup Y) = P[\hat{\varepsilon}(h_0) < \hat{\varepsilon}(\hat{h}) + \eta - 2\gamma] \leq \delta$$

So, if $\varepsilon(h_0) > \varepsilon(h^*) + \eta$, then $P[\hat{\varepsilon}(h_0) < \hat{\varepsilon}(\hat{h}) + \eta - 2\gamma] \leq \delta$
or $P[\text{YES}]$ at most δ

(Done)

Được quét bằng CamScanner

(c) [8 points] Finally, suppose that $h_0 = h^*$, and let $\eta > 0$ and $\delta > 0$ be fixed. Show that if m is sufficiently large, then the probability that the algorithm returns YES is at least $1 - \delta$.

Hint: observe that for fixed η and δ, as $m \to \infty$, we have

$$\gamma = \sqrt{\frac{1}{2m} \log \frac{2k}{\delta}} \to 0.$$

This means that there are values of m for which $2\gamma < \eta - 2\gamma$.

- Apply Union Bound & Hoeffding's inequality for h_0, we have:

$$P[\hat{\varepsilon}(h_0) - \varepsilon(h_0) < \gamma] \geq 1 - \delta/2$$

Condition: $h_0 = h^*$

$$\Leftrightarrow P[\hat{\varepsilon}(h_0) < \varepsilon(h_0) + \gamma] \geq 1 - \delta/2$$

$$\Leftrightarrow P[\hat{\varepsilon}(h_0) < \varepsilon(h^*) + \gamma] \geq 1 - \delta/2$$

Fact: $\varepsilon(h^*) \leq \varepsilon(\hat{h})$

$$\Leftrightarrow P[\hat{\varepsilon}(h_0) < \varepsilon(\hat{h}) + \gamma] \geq 1 - \delta/2$$

$$\Rightarrow P[\hat{\varepsilon}(h_0) > \varepsilon(\hat{h}) + \gamma] \leq 1 - (1 - \delta/2) = \delta/2 \quad (1)$$

- Apply Union Bound & Hoeffding's inequality for $\hat{h}$, we have:

$$P[\varepsilon(\hat{h}) - \hat{\varepsilon}(\hat{h}) < \gamma] \geq 1 - \delta/2$$

$$\Leftrightarrow P[\varepsilon(\hat{h}) < \hat{\varepsilon}(\hat{h}) + \gamma] \geq 1 - \delta/2$$

$$\Rightarrow P[\varepsilon(\hat{h}) > \hat{\varepsilon}(\hat{h}) + \gamma] \leq \delta/2 \quad (2)$$

- Apply Union Bound for (1) & (2), we have:

Let $X = \{\hat{\varepsilon}(h_0) > \varepsilon(\hat{h}) + \gamma\}$; $Y = \{\varepsilon(\hat{h}) > \hat{\varepsilon}(\hat{h}) + \gamma\}$

$$\Rightarrow P(X \cup Y) \leq P(X) + P(Y) \leq \delta/2 + \delta/2 = \delta$$

$$\Leftrightarrow P(X \cup Y) = P[\hat{\varepsilon}(h_0) > \hat{\varepsilon}(\hat{h}) + 2\gamma] \leq \delta \quad (3)$$

Condition: $m \to \infty \Rightarrow 2\gamma < \eta - 2\gamma$

$$(3) \Leftrightarrow P[\hat{\varepsilon}(h_0) < \hat{\varepsilon}(\hat{h}) + 2\gamma] \geq 1 - \delta$$

$$\Leftrightarrow P[\hat{\varepsilon}(h_0) < \hat{\varepsilon}(\hat{h}) + \eta - 2\gamma] \geq 1 - \delta$$

So, if $(h_0 = h^*)$ & $(m \to \infty)$, then $P[\hat{\varepsilon}(h_0) < \hat{\varepsilon}(\hat{h}) + \eta - 2\gamma] \geq 1 - \delta$

or $P[\text{YES}]$ at least $1 - \delta$

Được quét bằng CamScanner

6. **[24 points] Short answers**

The following questions require a reasonably short answer (usually at most 2-3 sentences or a figure, though some questions may require longer or shorter explanations).

To discourage random guessing, one point will be deducted for a wrong answer on true/false or multiple choice questions! Also, no credit will be given for answers without a correct explanation.

(a) [3 points] You have an implementation of Newton's method and gradient descent. Suppose that one iteration of Newton's method takes twice as long as one iteration of gradient descent. Then, this implies that gradient descent will converge to the optimal objective faster. True/False?

- FALSE!
- To know which method is converge faster, we need more conditions. Newton Method need Less iterations than GD generally. Each step of Newton Method might take more long time than GD but the overall time to convergence can still be shorter by Application of Hessian Matrix H to understand the "curvature" of $J(\theta)$.

(Done)

(b) [3 points] A stochastic gradient descent algorithm for training logistic regression with a fixed learning rate will always converge to exactly the optimal setting of the parameters $\theta^* = \arg\max_\theta \prod_{i=1}^m p(y^{(i)}|x^{(i)};\theta)$, assuming a reasonable choice of the learning rate. True/False?

- FALSE!
- Because of Fixed Learning Rate, SGD never congerge to exactly optimal point. Due to SGD just base on 1 sample data point, so the update phase very ROUGH, SGD is just go around the optimal point but never touch that.

[Instead of looking all noise of the whole data set at the same time, SGD just update by 1 data point → After many time update, the noise of each datapoint will affect to SGD → SGD can NOT converge to optimal point]

(Done)

Được quét bằng CamScanner

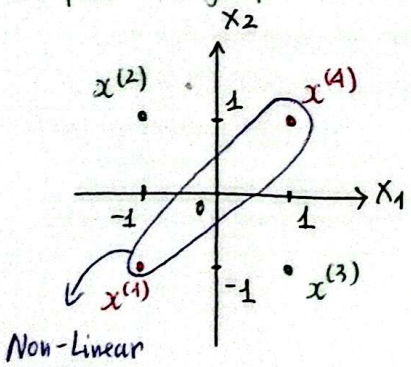
(c) [3 points] Given a valid kernel $K(x, y)$ over $\mathbb{R}^m$, is $K_{norm}(x, y) = \frac{K(x,y)}{\sqrt{K(x,x)K(y,y)}}$ a valid kernel?

- YES! $K_{norm}(x,y)$ is valid Kernel!
- We re-write $K_{norm}(x,y)$ into $K_{norm}(x^{(i)}, x^{(j)}) = \frac{\langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle}{\sqrt{\langle \phi(x^{(i)}), \phi(x^{(i)}) \rangle \cdot \langle \phi(x^{(j)}), \phi(x^{(j)}) \rangle}}$ (both inner products ≥ 0)
- Apply Mercer's Theorem, we have:

$$z^T K_{norm} z = \sum_i \sum_j z_i K_{ij} z_j$$
$$= \sum_i \sum_j z_i \cdot K_{norm}(x^{(i)}, x^{(j)}) \cdot z_j$$
$$= \sum_i \sum_j \sum_t z_i \cdot [\phi(x^{(i)})]_t [\phi(x^{(j)})]_t \cdot z_j$$
$$= \sum_t \left[\sum_i z_i \phi(x^{(i)})_t \right]^2 \geq 0 \;, \forall z \in \mathbb{R}^n \text{ (Done)}$$

(d) [3 points] Consider a 2 class classification problem with a dataset of inputs $\{x^{(1)} = (-1, -1), x^{(2)} = (-1, +1), x^{(3)} = (+1, -1), x^{(4)} = (+1, +1)\}$. Can a linear SVM (with no kernel trick) shatter this set of 4 points?

- NO! Linear SVM can NOT shatter these 4 points.
- We plot the graph below:



We assume $(x^{(1)}, x^{(4)}) \in$ class 1
$(x^{(2)}, x^{(3)}) \in$ class 0

$\Rightarrow$ To apply Linear SVM to classify this problem is impossible!

$\Rightarrow$ We need to use Kernel Trick to do that.

(Done)

Được quét bằng CamScanner

(e) [3 points] For linear hypotheses (i.e. of the form $h(x) = w^T x + b$), the vector of learned weights w is always perpendicular to the separating hyperplane. True/False? Provide a counterexample if False, or a brief explanation if True.

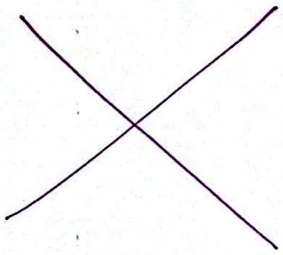
- TRUE!
- W always perpendicular to the separating hyperplane because w is normal vector of this hyperplane. W indicates the fastest direction of increase of h(x).

(Done)

(f) [3 points] Let $\mathcal{H}$ be a set of classifiers whose VC-dimension is 5. Suppose we have four training examples and labels, $\{(x^{(1)}, y^{(1)}), ..., (x^{(4)}, y^{(4)})\}$, and we select a classifier h from $\mathcal{H}$ by minimizing the classification error on the training set. In the absence of any other information about the set of classifiers $\mathcal{H}$, can we say that

i. $x^{(5)}$ will certainly be classified correctly?
ii. $x^{(5)}$ will certainly be classified incorrectly?
iii. we cannot tell?

Briefly justify your answer.



Được quét bằng CamScanner

(g) [6 points] Suppose you would like to use a linear regression model in order to predict the price of houses. In your model, you use the features $x_0 = 1$, $x_1 =$ size in square meters, $x_2 =$ height of roof in meters. Now, suppose a friend repeats the same analysis using exactly the same training set, only he represents the data instead using features $x'_0 = 1$, $x'_1 = x_1$, and $x'_2 =$ height in cm (so $x'_2 = 100x_2$).

i. [3 points] Suppose both of you run linear regression, solving for the parameters via the Normal equations. (Assume there are no degeneracies, so this gives a unique solution to the parameters.) You get parameters θ_0, θ_1, θ_2; your friend gets θ'_0, θ'_1, θ'_2. Then $\theta'_0 = \theta_0$, $\theta'_1 = \theta_1$, $\theta'_2 = \frac{1}{100}\theta_2$. True/False?

- TRUE!
- From information, we have: $X = [1 \; x_1 \; x_2]$, $X' = [1 \; x_1 \; 100x_2]$ $\Rightarrow X' = DX$ with $D = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 100 \end{bmatrix}$
- Normal Equation of Friend:

$$\theta' = (X'^T X')^{-1} X'^T \vec{y} = \left[(DX)^T (DX)\right]^{-1} (DX)^T \vec{y} = D^{-1}(X^T X)^{-1} X^T \vec{y} = D^{-1}\theta$$

$$\Rightarrow \boxed{\theta' = D^{-1}\theta} \Rightarrow \begin{cases} \theta_0' = \theta_0 \\ \theta_1' = \theta_1 \\ \theta_2' = \theta_2/100 \end{cases} \quad \text{(Done)}$$

ii. [3 points] Suppose both of you run linear regression, initializing the parameters to 0, and compare your results after running just *one* iteration of batch gradient descent. You get parameters θ_0, θ_1, θ_2; your friend gets θ'_0, θ'_1, θ'_2. Then $\theta'_0 = \theta_0$, $\theta'_1 = \theta_1$, $\theta'_2 = \frac{1}{100}\theta_2$. True/False?

- FALSE!
- BGD in Vectorization Form of Friend:

$$\theta' := \theta' - \alpha \cdot \frac{1}{m} X'^T (X'\theta' - \vec{y})$$

- With initializing parameter is 0, after 1 iteration we have:

$$\theta' = +\alpha \cdot \frac{1}{m} X'^T \vec{y} = \alpha \cdot \frac{1}{m} \cdot (DX)^T \vec{y} = D \cdot \alpha \frac{1}{m} X^T \vec{y} = D \cdot \theta$$

$$\theta = \alpha \frac{1}{m} X^T \vec{y}$$

$$\Rightarrow \boxed{\theta' = D \cdot \theta} \Rightarrow \begin{cases} \theta_0' = \theta_0 \\ \theta_1' = \theta_1 \\ \theta_2' = 100\theta_2 \end{cases} \quad \text{(Done)}$$

Được quét bằng CamScanner