THE CHINESE UNIVERSITY OF HONGKONG, SHENZHEN
COMPUTER AND INFORMATION ENGINEERING

Homework #1

Student name: *Fanyi Meng (223015127)*

Course: *Advanced Machine Learning (AIR 6002)* – Professor: *Prof Tongxin Li*
Due date: *January 31st, 2024*

Question 1: Basics

(*a*) What is a hypothesis set?

(*b*) What is the hypothesis set of a linear model?

(*c*) What is overfitting?

(*d*) What are two ways to prevent overfitting?

(*e*) What are training data and test data, and how are they used differently? Why should you never change your model based on information from test data?

(*f*) What are the two assumptions we make about how our dataset is sampled?

(*g*) Consider the machine learning problem of deciding whether or not an email is spam. What could X, the input space, be? What could Y, the output space, be?

(*h*) What is the k-fold cross-validation procedure?

Answer

(*a*) Hypothesis set contains all possible models on a specific dataset.

(*b*) Hypothesis set of a linear model means the model can be represented by linear model with its parametes. It can be writen as: $y = \sum_{i=1}^{n} \theta_i x_i + \theta_0$

(*c*) Overfitting means the model performs well in the train model but not as well in test model, because the model learns some irrelevant knowledge or random noise.

(*d*) Adding Regularization and using a lower learning rate can prevent overfitting.

(*e*) Training data is used to train the model and test data is used to test the model's performance. Changing model based on information from test data will improve the performance incorrectly because the model have "seen" the test data, which would cause the model can't have a matched performance in read-world data.

(*f*) Indepedent and Identically Distributed.

(*g*) **Input**:Email Address, Email Content, the network information(IP/MAC Address)
Output:spam or not

(*h*) Firstly the dataset is devided into k subsets, and then the model will be trained for k times, in each time one subset will be used to test the model and the other subsets will be used to trian. Finally the final model performance will be calculated as the average of the k models.

Question 2: Bias-Variance Tradeoff

(*a*) Derive the bias-variance decomposition for the squared error loss function. That is, show that for a model f_S trained on a dataset S to predict a target $y(x)$ for each x,

$$\mathbb{E}_S\left[E_{\text{out}}\left(f_S\right)\right]=\mathbb{E}_x[\operatorname{Bias}(x)+\operatorname{Var}(x)]$$

given the following definitions :

$$\begin{aligned} F(x) &= \mathbb{E}_S\left[f_S(x)\right] \\ E_{\text{out}}\left(f_S\right) &= \mathbb{E}_x\left[\left(f_S(x)-y(x)\right)^2\right] \\ \operatorname{Bias}(x) &= (F(x)-y(x))^2 \\ \operatorname{Var}(x) &= \mathbb{E}_S\left[\left(f_S(x)-F(x)\right)^2\right] \end{aligned}$$

(*b*) Coding Part.

Answer

(*a*)

$$\begin{aligned} E_{out}(f_S) &= \mathbb{E}_x[(f_S(x)-y(x))^2] \\ &= \mathbb{E}_x[f_S(x)^2-2f_S(x)y(x)+y(x)^2] \end{aligned}$$

By rewriting $f_S(x)$ as $F(x)+(f_S(x)-F(x))$,$E_{out}(f_S)$ can be represented as

$$\begin{aligned} E_{out}(f_S) &= \mathbb{E}_x[(F(x)+(f_S(x)-F(x)))^2-2(F(x)+(f_S(x)-F(x)))y(x)+y(x)^2] \\ &= \mathbb{E}_x[(F(x)-y(x))^2+(f_S(x)-F(x))^2+2(f_S(x)-F(x))(F(x)-y(x))] \end{aligned}$$

As $F(x)=\mathbb{E}_S\left[f_S(x)\right]$, $\mathbb{E}_S(f_S(x)-F(x))(F(x)-y(x))$ equals to zero. So:

$$\mathbb{E}_S\left[E_{\text{out}}\left(f_S\right)\right]=\mathbb{E}_X[\underbrace{(F(x)-y(x))^2}_{\operatorname{Bias}(x)}+\underbrace{(f_S(x)-F(x))^2}_{\operatorname{Var}(x)}]$$

(*b*) The code and figure is available in Here(Github Link)

Question 3

Find the closed-form solutions of the following optimization problems ($\mathbf{W}\in\mathbb{R}^{K\times D}, N\gg D>K$):

(*a*) $\min\limits_{W,b}\sum_{i=1}^N\|\mathbf{y}_i-\mathbf{W}\mathbf{x}_i-\mathbf{b}\|^2$

(*b*) $\min\limits_{W,b}\sum_{i=1}^N\|\mathbf{y}_i-\mathbf{W}\mathbf{x}_i-\mathbf{b}\|^2+\frac{\lambda}{2}\|\mathbf{W}\|_F^2$

Answer

(a) Denote $\overline{\mathbf{W}} = [b, w], \overline{\mathbf{x}_i} = [1, x_i]^\top$, then the objective function can be rewriten as:

$$\sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{W}\mathbf{x}_i - \mathbf{b}\|^2 = \|\mathbf{Y} - \overline{\mathbf{W}\mathbf{X}}\|_F^2$$

Then, the gradient on $\mathbf{W}$ is:

$$\frac{\nabla \|\mathbf{Y} - \overline{\mathbf{W}\mathbf{X}}\|_F^2}{\nabla \overline{\mathbf{W}}} = 2\overline{\mathbf{W}\mathbf{X}\mathbf{X}}^\top - 2\mathbf{Y}\overline{\mathbf{X}}^\top$$

Let the gradient equals to zero, we can get the optimal $\overline{\mathbf{W}} = \mathbf{Y}\overline{\mathbf{X}}^\top \left(\overline{\mathbf{X}\mathbf{X}}^\top\right)^{-1}$

(b) Similarly,we can get the result $\overline{\mathbf{W}} = \mathbf{Y}\overline{\mathbf{X}}^\top \left(\overline{\mathbf{X}\mathbf{X}}^\top + \lambda \mathbf{I}\right)^{-1}$

Question 4

Consider the following problem

$$\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{W}\Phi - \mathbf{Y}\|_F^2 + \frac{\lambda}{2}\|\mathbf{W}\|_F^2$$

where $\|\cdot\|_F$ denotes the Frobenius norm; $Y \in \mathbb{R}^K \times N, \Phi = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \ldots, \phi(\mathbf{x}_N)], \mathbf{x}_i \in \mathbb{R}^D$, $i = 1, 2, \ldots, N$ and ϕ is the feature map induced by a kernel function $k(\cdot, \cdot)$. Prove that for any x$\in \mathbb{R}^D$, we can make prediction as

$$\mathbf{y} = \mathbf{W}\phi(\mathbf{x}) = \mathbf{Y}(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{k}(\mathbf{x}),$$

where $\mathbf{K} = \Phi^\top\Phi$ and $\mathbf{k}(\mathbf{x}) = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \ldots, k(\mathbf{x}_N, \mathbf{x})]^\top$.

Answer

By calculating the gradient, we get:

$$W\Phi^\top\Phi - Y\Phi + \lambda W = 0$$

$$W = Y\Phi(\Phi^\top\Phi + \lambda I)^{-1}$$

Considering $\mathbf{K} = \Phi^\top\Phi$ and $\mathbf{k}(\mathbf{x}) = \Phi\phi(x)^\top$. We can predict y by:

$$\begin{aligned} y &= W\phi(x) \\ &= [Y\Phi(\Phi^\top\Phi + \lambda I)^{-1}]\phi(x) \\ &= Y(K + \lambda I)^{-1}k(x) \end{aligned}$$

Question 5

Compute the space and time complexities (in the form of big O, consider only the training stage) of the following algorithms:

(*a*) Ridge regression (Question 2(b)) with the closed-form solution

(*b*) N data points of $D-$dimension, choose d principal components)

(*c*) Neural network with architecture $D - H_1 - H_2 - K$ on a mini-batch of size B (consider only the forward process and neglect the computational costs of activation functions)

[Hint: the time complexity of $A \in \mathbb{R}^{m\times n} \times B \in \mathbb{R}^{n\times l}$ is $O(mnl)$; the time complexities of eigenvalue decomposition and inverse of an $n \times n$ matrix are both $O(n^3)$.]

Answer

(*a*) **Time complexity:** $\mathcal{O}(ND^2 + D^3)$ **Space complexity:** $\mathcal{O}(ND + D^2)$

(*b*) **Time complexity:** $\mathcal{O}(ND^2 + D^3)$ **Space complexity:** $\mathcal{O}(ND + D^2)$

(*c*) **Time complexity:** $\mathcal{O}(BDH_1 + BH_1H_2 + BH_2K)$ **Space complexity:** $\mathcal{O}(BD + BH_1 + BH_2 + BK + DH_1 + H_1H_2 + H_2K)$

Question 6

Prove the convergence of the generic gradient boosting algorithm (AnyBoost). Specifically, suppose in the algorithm of AnyBoost (page 14 of Lecture 02), the gradient of the objective function $\mathcal{L}$ is L-Lipschitz continuous, i.e., there exists $L > 0$ such that

$$\|\nabla\mathcal{L}(H) - \nabla\mathcal{L}(H')\| \leq L\|H - H'\|$$

holds for any H and H'. Suppose in the algorithm, α is computed as

$$\alpha_{t+1} = -\frac{\langle\nabla\mathcal{L}(H_t), h_{t+1}\rangle}{L\|h_{t+1}\|^2}.$$

Then the ensemble model is updated as $H_{t+1} = H_t + \alpha_{t+1}h_{t+1}$. Prove that the algorithm either terminates at round T with $\langle\nabla\mathcal{L}(H_t), h_{t+1}\rangle$ or $\mathcal{L}(H_t)$ converges to a finite value, in which case

$$\lim_{t\to\infty}\langle\nabla\mathcal{L}(H_t), h_{t+1}\rangle = 0.$$

* [Hint: Using the following result: Suppose $\mathcal{L} : \mathcal{H} \to \mathbb{R}$ and $\|\nabla\mathcal{L}(F) - \nabla\mathcal{L}(G)\| \leq L\|F - G\|$ holds for any F and G in $\mathcal{H}$, then $\mathcal{L}(F + wG) - \mathcal{L}(F) \leq w\langle\nabla\mathcal{L}(F), G\rangle + \frac{Lw^2}{2}\|G\|^2$ holds for any $w > 0$.]

Answer

By using the Hint,

$$\begin{aligned}\mathcal{L}(H_{t+1}) - \mathcal{L}(H_t) &= \mathcal{L}(H_t + \alpha_{t+1}h_{t+1}) - \mathcal{L}(H_t)l \\ &\leq \alpha_{t+1}\langle \nabla\mathcal{L}(H_t), h_{t+1}\rangle + \frac{L\alpha_{t+1}^2\|h_{t+1}\|^2}{2} \\ &= -\frac{\langle \nabla\mathcal{L}(H_t), h_{t+1}\rangle^2}{2L\|h_{t+1}\|^2}.\end{aligned}$$

Considering when $h_{T+1} = 0$, the algorithm will terminate at time T, otherwise, which means $h_{t+1} \neq 0$ for all t as $\mathcal{L}(H_{t+1}) - \mathcal{L}(H_t) \to 0$, $\lim_{t\to\infty}\langle \nabla\mathcal{L}(H_t), h_{t+1}\rangle$ will also converge to 0.

Question 7:SGD

Linear regression learns a model of the form:

$$f(x_1, x_2, \cdots, x_d) = \left(\sum_{i=1}^{d} w_i x_i\right) + b$$

(*a*) We can make our algebra and coding simpler by writing $f(x_1, x_2, \cdots, x_d) = \mathbf{w}^T\mathbf{x}$ for vectors w and x. But at first glance, this formulation seems to be missing the bias term b from the equation above. How should we define x and w such that the model includes the bias term?

Linear regression learns a model by minimizing the squared loss function L, which is the sum across all training data $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)\}$ of the squared difference between actual and predicted output values:

$$L(f) = \sum_{i=1}^{N}\left(y_i - \mathbf{w}^T\mathbf{x}_i\right)^2$$

(*b*) SGD uses the gradient of the loss function to make incremental adjustments to the weight vector w. Derive the gradient of the squared loss function with respect to w for linear regression.

(c)-(h) Coding Part.

Answer

(*a*) By defineing $\mathbf{x} = (x_1, x_2, \cdots, x_d, 1)$ and $\mathbf{w} = (w_1, w_2, \cdots, x_n, b)$

(*b*) $\nabla\mathbf{w} = -2\cdot(\mathbf{y_i} - \mathbf{x_i}\cdot\mathbf{w})\mathbf{x_i}$

(*c*) - (h) The code and figure is available in Part 1 and Part 2(Github Link)

(*1*) As strating point varies, the distance between start points and the convergent point changes and converge time varies.

(2) Dataset 1 is more regular, so easier to converge.

(*g*) For small η, Training loss decreases as the epochs grows.

(*i*) Even closed-form solution exists, SGD is still useful since SGD can work on high-dimension data, in this case calculating inverse will have a huge cost. Also it performs well when parameter needs to be updated as new data added.

(*j*) By setting a threshold of loss function.

(*k*) Perceptron converges faster than SGD algorithms for linear model.

Question 8

True or False? If False, then explain shortly.

(*a*) The inequality $G(\mathcal{F}, n) \leq n^2$ holds for any model class $\mathcal{F}$.

(*b*) The VC dimension of an axis-aligned rectangle in a 2D space is 4.

(*c*) The VC dimension of a circle in a 2D space is 4.

(*d*) The VC dimension of 1-nearest neighbor classifier in d-dimensional space is $d+1$.

(*e*) Let d be the VC dimension of $\mathcal{F}$. Then the inequality $G(\mathcal{F}, n) \leq \left(\frac{en}{d}\right)^d$ always holds.

Answer

(*a*) **False**. When n=1, $G(\mathcal{F}, n)$ can take the value 2 for f_1 predict 1 and f_2 predict -1, but n^2 equals to 1.

(*b*) **True**. 4 points can be scattered by 4 sides of rectangle.

(*c*) **False**. VC dimension of a circle is 3.

(*d*) **False**. VC dimension of 1-nearest neighbor classifier is infinity.

(*e*) **False**. For $n = 1$,$G(\mathcal{F}, n)$ is 2 but $\left(\frac{en}{d}\right)^d < 1$ for $d > e$

Question 9

In LASSO, the model class is defined as $\mathcal{F} = \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle : \|\mathbf{w}\|_1 \leq \alpha\}$. Suppose $\mathbf{x} \in \mathbb{R}^d$, $y \in \{-1, +1\}$, the training data are $S = \{(\mathbf{x}_i, y_i)\}_i = 1^n$, and $\max_{1 \leq i \leq n} \|\mathbf{x}_i\|_\infty \leq \beta$, where $\| \cdot \|_\infty$ denotes the largest absolute element of a vector.

(a) Find an upper bound of the empirical Rademacher complexity, where σ_i are the Rademacher variables.

(b) Suppose the loss function is the absolute loss. Use the inequality (highlighted in blue) on page 30 and Lemma 5 on page 35 (i.e., (i.e.,$\mathcal{R}(\ell \circ \mathcal{F}) \leq \eta \mathcal{R}(\mathcal{F})$)) of Lecture 03 to derive a generalization error bound for LASSO.

* Hint: For question (a), please use the inequality $\langle \mathbf{a}, \mathbf{b} \rangle \leq \|\mathbf{a}\|_1 \|\mathbf{b}\|_\infty$ and the following lemma:

Lemma 1. Let $A \subseteq \mathbb{R}^n$ be a finite set of points with $r = \max_{\mathbf{x} \in A} \|\mathbf{x}\|_2$ and denote $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. Then

$$\mathbb{E}_\sigma \left[\max_{\mathbf{x} \in A} \sum_{i=1}^n x_i \sigma_i \right] \leq r\sqrt{2 \log |A|}$$

where $|A|$ denotes the cardinality of set A and σ_i are the Rademacher variables.

Answer

(a) The empirical Rademacher complexity can be written as :

$$\begin{aligned}
\mathcal{R}(\mathcal{F}) &= \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i f(x_i) \right] \\
&= \mathbb{E}_\sigma \left[\sup_{\substack{w_i \in \mathcal{W} \\ x_i \in \mathcal{X}}} \frac{1}{n} \sum_{i=1}^n \sigma_i (\langle w_i, x_i \rangle + \|\mathbf{w}\|_1) \right]
\end{aligned}$$

By using the inequality of $\langle \mathbf{a}, \mathbf{b} \rangle \leq \|\mathbf{a}\|_1 \|\mathbf{b}\|_\infty$

$$\mathcal{R}(\mathcal{F}) \leq \mathbb{E}_\sigma \left[\sup_{\substack{w_i \in \mathcal{W} \\ x_i \in \mathcal{X}}} \frac{1}{n} \sum_{i=1}^n \sigma_i (\|\mathbf{w_i}\|_1 \|\mathbf{x}_i\|_\infty + \|\mathbf{w}\|_1) \right]$$

Then by using **Lemma 1**, $\|\mathbf{w}\|_1 \leq \alpha$and $max_{1\leq i\leq n}\|\mathbf{x}_i\|_\infty \leq \beta$ we can get the upper bound:

$$
\begin{aligned}
\mathcal{R}(\mathcal{F}) &\leq \frac{1}{n}\mathbb{E}_\sigma\left[\sup_{\substack{w_i\in\mathcal{W}\\ x_i\in\mathcal{X}}}\sum_{i=1}^{n}(\sigma_i\|\mathbf{w_i}\|_1\|\mathbf{x}_i\|_\infty + \|\mathbf{w}\|_1)\right] \\
&\leq \frac{1}{n}\mathbb{E}_\sigma\left[\sup_{\substack{w_i\in\mathcal{W}\\ x_i\in\mathcal{X}}}\sum_{i=1}^{n}\sigma_i(\alpha\beta+\alpha)\right] \\
&\leq (\alpha\beta+\alpha)\sqrt{\frac{2\log n}{n}}
\end{aligned}
$$

(b) By using the Lemma, we have

$$
\begin{aligned}
&\sup_{f\in\mathcal{F}}\left\{\mathbb{E}[\ell(f(x),y)] - \frac{1}{n}\sum_{t=1}^{n}\ell\left(f\left(x_t\right),y_t\right)\right\} \\
&\leq 2\mathcal{R}_S(\ell\circ\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2n}} \\
&\leq 2\eta\mathcal{R}_S(\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2n}} \\
&\leq 2(\alpha\beta+\alpha)\sqrt{\frac{2\log n}{n}} + 3\sqrt{\frac{\log(2/\delta)}{2n}}
\end{aligned}
$$