THE CHINESE UNIVERSITY OF HONGKONG, SHENZHEN
COMPUTER AND INFORMATION ENGINEERING

Homework #2

Student name: *Fanyi Meng (223015127)*

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

Question 1: Adaboost

(*a*) Let $h_t : \mathbb{R}^m \to \{-1, 1\}$ be the weak classifier obtained at step t, and let α_t be its weight. Recall that the final classifier is

$$H(x) = \operatorname{sign}(f(x)) = \operatorname{sign}\left(\sum_{i=1}^{T} \alpha_t h_t(x)\right).$$

Suppose $\{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathbb{R}^m \times \{-1, 1\}$ is our training dataset. Show that the training set error of the final classifier can be bounded from above if an exponential loss function is used:

$$E = \frac{1}{N}\sum_{i=1}^{N} \exp\left(-y_i f\left(x_i\right)\right) \geq \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(H\left(x_i\right) \neq y_i\right)$$

where $\mathbb{1}$ is the indicator function.

(*b*) Find $D_{T+1}(i)$ in terms of Z_t, α_t, x_i, y_i, and the classifier h_t, where T is the last timestep and $t \in \{1, \ldots, T\}$. Recall that Z_t is the normalization factor for distribution D_{t+1}:

$$Z_t = \sum_{i=1}^{N} D_t(i) \exp\left(-\alpha_t y_i h_t\left(x_i\right)\right).$$

(*c*) Show that $E = \sum_{i=1}^{N} \frac{1}{N} e^{\sum_{t=1}^{T} -\alpha_t y_i h_t(x_i)}$.

(*d*) Show that

$$E = \prod_{t=1}^{T} Z_t.$$

(*e*) Show that the normalizer Z_t can be written as

$$Z_t = \left(1 - \epsilon_t\right) \exp\left(-\alpha_t\right) + \epsilon_t \exp\left(\alpha_t\right)$$

where ϵ_t is the training set error of weak classifier h_t for the weighted dataset:

$$\epsilon_t = \sum_{i=1}^{N} D_t(i) \mathbb{1}\left(h_t\left(x_i\right) \neq y_i\right).$$

(*f*) We derived all of this because it is hard to directly minimize the training set error, but we can greedily minimize the upper bound E on this error. Show that choosing α_t greedily to minimize Z_t at each iteration leads to the choices in AdaBoost:

$$\alpha_t^* = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right).$$

(*f*) **AIR6002** Implement the `GradientBoosting.fit()` and `AdaBoost.fit()` methods in the notebook "Q1_AdaBoost.ipynb" provided for you.

Some important notes and guidelines:

- For both methods, make sure to work with the class attributes provided to you. Namely, after `GradientBoosting.fit()` is called, `self.clfs` should be appropriately filled with the `self.n_clfs` trained weak hypotheses. Similarly, after `AdaBoost.fit()` is called, `self.clfs` and `self.coefs` should be appropriately filled with the `self.n_clfs` trained weak hypotheses and their coefficients, respectively.
- `AdaBoost.fit()` should additionally return an (N, T) shaped numpy array D such that `D[:,t]` contains D_{t+1} for each $t \in \{0, \dots,$ `self.n_clfs`$\}$.
- For the `AdaBoost.fit()` method, use the $0/1$ loss instead of the exponential loss.

(*f*) **AIR6002** Plot the loss curves for gradient boosting and for AdaBoost. Describe and explain the behaviour of the two loss curves you plot. You should consider two kinds of behaviours: the smoothness of the curves and the final values that the curves approach.

(*f*) **AIR6002** Compare the final loss values of the two models. Which performed better on the classification dataset?

(*f*) **AIR6002**For AdaBoost, where are the dataset weights the largest, and where are they the smallest?

Hint: *Watch how the dataset weights change across time in the animation.*

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: Recommendation Systems

(*a*) What are the differences between collaborative filtering and content-based methods?

(*b*) Suppose we have m items and n users. Let Ω be the set of indices of observed ratings given by the users on the items. Provide the objective function of content-based recommendation for users, where the model class is a neural network with two hidden layers. You need to define the notations clearly. In addition, explain how to make recommendations for a new user using your model.

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:

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

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

Question 3:Spectral Clustering

Prove $\frac{\mathbf{u}^\top \mathbf{L}\mathbf{u}}{\mathbf{u}^\top \mathbf{D}\mathbf{u}} = \text{Ncut}(A, B)$. See the definitions on page 23 of Lecture 05-I.

Answer

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

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

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

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

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

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

Question 4:Semi-supervised Learning

(*a*) Suppose $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^k$ and the training data are $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{l} \bigcup \{\mathbf{x}_i\}_{i=l+1}^{n}$. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be an affinity matrix constructed from the training data.

i) Show the objective function of linear regression with square loss and graph regularization. For simplicity, do not consider the bias term.

ii) Compute the gradient of the objective function with respect to the parameters.

iii) Is there a closed-form solution? If yes, find it.

(*b*) In the label propagation algorithm, why do we need to use $\mathbf{D}^{-1}\mathbf{W}$ instead of $\mathbf{W}$? Why do we set $\hat{\mathbf{Y}}_l^{(t+1)} \leftarrow \mathbf{Y}_l$? If there are more than 2 classes, how do we set $\mathbf{Y}^{(0)}$?

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: Graph Neural Networks

(*a*) In GNN, given $\hat{\mathbf{A}}$, how to compute $\hat{\mathbf{A}}^c := \underbrace{\hat{\mathbf{A}}\hat{\mathbf{A}} \cdots \hat{\mathbf{A}}}_{c}$ efficiently when c is large?

(*b*) Show the loss functions of node classification, graph classification, and link prediction. Explain your notations.

(*c*) Provide two examples of algorithms for each learning types or methods:

- parametric method
- nonparametric method
- inductive learning
- transductive learning
- semi-supervised learning

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: Nonlinear Dimensionality Reduction

(*a*) Show the algorithm of multidimensional scaling here.

(*b*) Present a method for out-of-sample extension of t-SNE. In other words, reduce the dimension of new samples using the training result of t-SNE.

(*c*) Given a dataset, suppose most of the samples are normal or good data, and there are a few outliers. Design a method or strategy to detect these outliers bassed on the methods learned in Lecture 07.

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: Generative Models

(*a*) Derive the evidence lower bound (ELBO) for the VAE model.

(*b*) Derive the objective functions for the generator and discriminator in a GAN, respectively.

(*c*) Briefly explain the relationship between diffusion models and Langevin dynamics.

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: Graph Convolutional Network

(*a*) **AIR6002** Implement the `GCNLayer.forward()` in the notebook “Q8_GCN.ipynb" provided for you.

(*b*) **AIR6002** Define the adjacency matrix of the graph in the notebook with self-connections.

(*c*) **AIR6002** Apply the GCN layer defined in (1) on the graph in (2), and write down the output features for all nodes.

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 $d > e$