COMP5212 Machine Learning

Written Homework 2

LO, Li-yu
20997405
e-mail: lloac@connect.hkust.hk

31/Oct/2023

Problem 1

(a)

Ans: **False**. As the distribution of training data and testing data could be different, classifier A having a smaller training error than classifier B might not directly imply that it will perform better on testing dataset, i.e., smaller generalization error.

(b)

Ans: **True**. It is true that applying a model with high complexity is not always the best approach. One should select the model accordingly based on the problem, and the kind of data that is being dealt with. For instance, when doing binary classification on digit 0s and 1s, simple linear models like Linear Logistic Regression would have sufficed. On the other hand, solving the problem with convolutional neural network might not be the best means in this case.

(c)

Ans: **False**. The statement is only partially true. In fact, in a convex problem and given a relatively small learning rate, it could be assured that through gradient descent method, convergence to optima (ϵ-suboptimal in numerical calculation) could be achieved. The rationale to decrease the learning rate is that: we want to have a higher learning rate to train faster, yet, such large learning rates, or constant learning rates could lead to divergent behavior. In other words, we would want to transverse quickly from initial parameter to better ones, then explore deeper, narrower regions of the loss function, while avoiding too large learning rate that "jumps over" the optimal region.

Problem 2

Ans: **(5) none of the other choices**, as each of the option is a possible growth function. The growth function is defined when we are discussing the Theory of Generalization, and attempting to search for a limited number of the hypothesis set.

To attain a limited number of the hypothesis set, we introduce the concept of dichotomies in binary classification, which is counted by the growth function. Dichotomies in binary classification can group the infinity number of hypotheses to a limited amount, as some similar hypotheses achieve same classification.

Meanwhile, we also need to understand the concept of "shattered" and "break point". "Shattered", in short, implies that we can achieve any labelling on a set of points. In binary classification, when a set is deemed "shattered", then there exists 2^N set of labels, namely 2^N of dichotomies, or, we can say that the growth function gives 2^N, i.e., $m_{\mathcal{H}}(k) = 2^k$. As for break point k, it is defined as when the number, i.e., k, of data cannot be shattered by hypothesis set $\mathcal{H}$, then the k is the break point $\mathcal{H}$. We can also express it as $m_{\mathcal{H}}(k) < 2^k$.

The above-mentioned growth function is related to break point, and obviously, we do not hope to get an exponential $m_{\mathcal{H}}(k) = 2^k$, as it could imply that the large gap between testing error and generalization error could happen. We can further try to derive, if k exists, an upper bound for the growth function based on the number N of data point and the k break point. The upper bound is usually denoted as $\mathcal{B}(N, k)$.

Given the above premise, we then draw our focus back to the problem, we would want to identify whether such growth function exists:

- (a) 2^N. It could happen when there exists no break point k.
- (b) $2^{N^{0.5}}$. It is a polynomial, which could happen when there exists a break point k, such that $2^{k^{0.5}} < 2^k$. Note that $2^{k^{0.5}} < 2^k$ should only happen when $k > 1$.
- (c) 1. It could happen when break point $k = 1$, where we can not achieve shattered with $N = 1$.
- (d) $N^2 - N + 2$. It is a polynomial, which could happen when there exists a break point k, such that $k^2 - k + 2 < 2^k$. Note that $k^2 - k + 2 < 2^k < 2^k$ should only happen when $k > 3$.

And therefore, the answer is (5) none of the other choices.

Problem 3

(a)

The gradient of $f(x)$:

$$\nabla f(x) = w + C\sum_{i=1}^{n} \frac{exp(-y_i w^T x_i)(-y_i x_i)}{1 + exp(-y_i w^T x_i)}, where \;\; w, x_i \in \mathbb{R}^d$$

$$= \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} + C\sum_{i=1}^{n} \begin{bmatrix} \frac{exp(-y_i w^T x_i)(-y_i)}{1+exp(-y_i w^T x_i)} x_{i1} \\ \frac{exp(-y_i w^T x_i)(-y_i)}{1+exp(-y_i w^T x_i)} x_{i2} \\ \vdots \\ \frac{exp(-y_i w^T x_i)(-y_i)}{1+exp(-y_i w^T x_i)} x_{id} \end{bmatrix}$$

As for the Hessian of $f(x)$:

$$\nabla^2 f(x) = I + C\sum_{i=1}^{n} \begin{bmatrix} \left[\frac{(-y_i x_{i1})exp(-y_i w^T x_i)(-y_i)}{1+exp(-y_i w^T x_i)} x_i + \frac{(-y_i x_{i1})exp^2(-y_i w^T x_i)(-y_i)}{[1+exp(-y_i w^T x_i)]^2} x_i\right]^T \\ \left[\frac{(-y_i x_{i2})exp(-y_i w^T x_i)(-y_i)}{1+exp(-y_i w^T x_i)} x_i + \frac{(-y_i x_{i2})exp^2(-y_i w^T x_i)(-y_i)}{[1+exp(-y_i w^T x_i)]^2} x_i\right]^T \\ \vdots \\ \left[\frac{(-y_i x_{id})exp(-y_i w^T x_i)(-y_i)}{1+exp(-y_i w^T x_i)} x_i + \frac{(-y_i x_{id})exp^2(-y_i w^T x_i)(-y_i)}{[1+exp(-y_i w^T x_i)]^2} x_i\right]^T \end{bmatrix},$$

$$where\ \nabla^2 f(x) \in \mathbb{R}^{d\times d}$$

(b)

The update rule of gradient descent is as follows:

$$w \leftarrow w + \eta d$$
$$d = -(w + C\sum_{i=1}^{n} \frac{exp(-y_i w^T x_i)(-y_i x_i)}{1 + exp(-y_i w^T x_i)})$$

(c)

The time complexity of calculating one gradient descent update is $\mathcal{O}(nd)$, as a flop is carried out through dimension of d, with n features.

(d)

The time complexity of compute one Newton direction $(\nabla^2 f(w))^{-1}\nabla f(w)$ is $\mathcal{O}(\frac{4}{3}n^3)$. To calculate the Hessian matrix, as we are calculating the derivatives with respect to d dimension twice and with n features, the complexity is $\mathcal{O}(n^2 d)$. As we assumed $n \approx d$, $\mathcal{O}(n^2 d) \rightarrow \mathcal{O}(n^3)$. In addition, we also need to take the inverse of the Hessian, we need to take the flops of inverse operation into consideration. As Hessian is a symmetric matrix, we assume that Cholesky decomposition is utilized [1, p. 510]. The Cholesky decomposition complexity is $\mathcal{O}(\frac{1}{3}n^3)$, as, for a $\mathbb{R}^{n\times n}$ matrix, we need to take n square roots, $\frac{n^3-n}{6} + \frac{n(n-1)}{2}$ multiplication, and $\frac{n^3-n}{6}$ additions. Hence, $\mathcal{O}(\frac{1}{3}n^3)$.

By summarizing the two, we then infer the complexity $\mathcal{O}(\frac{4}{3}n^3)$.

(e)

From equation (3), it can be seen that it is a convex, quadratic function. Therefore, we can get the global optima (minima in this case) by setting the derivative to zero, and get the *argmin*.

$$\begin{aligned} \nabla J(d) &= \nabla^2 f(w)d + \nabla f(w), \\ &let\ \nabla J(d) = 0, \\ &i.e.,\ \nabla^2 f(w)d + \nabla f(w) = 0, \\ \therefore d &= -(\nabla^2 f(w))^{-1}\nabla f(w) \end{aligned}$$

(f)

When the Hessian matrix is in ill-condition, or non-invertible, one alternative to get the update step is by solving the Newton linear system via formulating it as least-squares problem, namely, solving it with pseudoinverse.

$$\begin{aligned} \nabla^2 f(w)d + \nabla f(w) = 0, \\ let\ A = \nabla^2 f(w), b = \nabla f(w), \\ \therefore Ad = b \end{aligned}$$

Then, we solve $Ad = b$ with the following alternative:

$$d^* = \underset{d}{\text{argmin}}\ ||Ad - b||_2,$$

where the solution is then $d^* = A^\dagger b$, where $A^\dagger = (A^T A)^{-1} A^T$.

References

[1] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.