Intro to Machine Learning Homework Assignment 2

Goktug Saatcioglu

NetId: gs2417

1. We define the distance function from the lecture notes, as given by Eq. (1.32), below using y instead of M^*.

$$D(y, x; M) = -(y \log(M(x)) + (1 - y) \log(1 - M(x))) \tag{1.32}$$

Furthermore, the distance function from Eq (1.19) from the lecture notes is given below.

$$D_{\log}(y, x; M) = \frac{1}{\log 2} \log(1 + \exp(-s(y, x; M))) \tag{1.19}$$

We wish to prove that Eq. (1.32) equals Eq. (1.19) up to a constant multiplication. To derive Eq. (1.19) we change our label set from $y \in \{0, 1\}$ to $\hat{y} \in \{-1, 1\}$. This mean that we can split the problem into two cases. If $\hat{y} = 1$, then that is the same as labelling y as 1. The distance function when y is 1 is given below.

$$\begin{aligned} D(1, x; M) &= -(\log(M(x))) \\ &= -\log(\frac{1}{1 + \exp -w^T x}) \\ &= \log(1 + \exp -w^T x) \end{aligned}$$

The distance function from Eq. (1.19) when $\hat{y} = 1$ is given below.

$$\begin{aligned} D_{\log}(1, x; M) &= \frac{1}{\log 2} \log(1 + \exp(-s(1, x; M))) \\ &= \frac{1}{\log 2} \log(1 + \exp(-w^T x)) \end{aligned}$$

Thus, we see that Eq. (1.32) is equal to Eq. (1.19) up to the constant $\frac{1}{\log 2}$ for the positive labelling case (i.e. $y = 1 \wedge \hat{y} = 1$). Now let us consider the case where $\hat{y} = -1$. This means that $y = 0$ as the distance function from Eq. (1.32) is given below.

$$\begin{aligned} D(0, x; M) &= -(\log(1 - M(x))) \\ &= -(\log(1 - \frac{1}{1 + \exp -w^T x})) \\ &= -(\log(\frac{1}{1 + \exp w^T x})) \\ &= \log(1 + \exp w^T x) \end{aligned}$$

Then the distance function from Eq. (1.19) when $\hat{y} = -1$ becomes as follows.

$$\begin{aligned} D(-1, x; M)_{\log} &= \frac{1}{\log 2} \log(1 + \exp(-s(-1, x; M))) \\ &= \frac{1}{\log 2} \log(1 + \exp(w^T x)) \end{aligned}$$

This time we see Eq. (1.32) is equal to Eq. (1.19) up to the constant $\frac{1}{\log 2}$ for the negative labelling case (i.e. $y = 0 \wedge \hat{y} = -1$). We conclude that for both cases Eq. (1.32) is equal to Eq. (1.19) up to the constant $\frac{1}{\log 2}$.
The constant $\frac{1}{\log 2}$ is necessary because for the 0-1 loss function if $w = 0$ then the loss function equals 1. Since we would like the logistic loss function to be an upper bound for the 0-1 loss function, the constant $\frac{1}{\log 2}$ is used. If we were to omit this then $w = 0$ would give us $D_{\log} = \log 2$ which would not make our logistic loss function an upper bound of the 0-1 loss function.

2. The hinge loss function is not differentiable for all values of s. Specifically, if $s = 1$ (the score function), then the derivative of the hinge loss function at that point is undefined. This means that we can't use a gradient-based optimization algorithm for finding a solution that minimizes the hinge loss. Fortunately, we can use a subgradient for the hinge loss function with respect to w. One such possible solution could look like:

$$\frac{\partial D_{hinge}}{\partial s_i} = \begin{cases} -y_i x_i & \text{if } s < 1 \\ 0 & \text{otherwise} \end{cases} \text{ where } i \text{ is an entry in a set of data.}$$

Other definitions of a subgradient can also be valid as long as at $s = 1$, the subgradient at that point is such that it is less than $-y_i x_i$. This solution is perfectly fine but other solutions have also been proposed. One alternative is the squared hinge loss and is defined as below.

$$D^2_{hinge}(y, x; M) = (\max 0, 1 - s(y, x; M))^2$$

This function smooths the hinge loss function and allows us to use define a gradient everywhere which then allows us to use gradient-based optimization algorithms. Furthermore, there are other alternatives too such as a smoothed hinge loss seen in [1] by Rennie and Srebro. Thus, even though we can't use a gradient-based optimization algorithm for finding a solution that minimizes D_{hinge}, we have solutions to overcome this problem and can use gradient-based optimization algorithms for differentiable alternatives to D_{hinge}.

3. (a) As a best model we pick an i and t such that $\tilde{R}^{(i)}_{val,t}$ has the smallest validation cost among the $2T$ trained models. Our models were trained against the training set D_{tra}. We then test these trained models against the validation set D_{val} and look for the smallest validation cost.

(b) We report the generalization error by measuring the testing cost using the best model $\tilde{R}^{(i)}_{val,t}$ we picked from part (a) and let's call this expected cost $\tilde{R}^{(i)}_{test,t}$. While the generalization error cannot be computed most of the time, we can attempt to estimate it using $\tilde{R}^{(i)}_{test,t} - \tilde{R}^{(i)}_{train,t}$ where we use $\tilde{R}^{(i)}_{test,t}$ to estimate the value of the empirical cost and $\tilde{R}^{(i)}_{train,t}$ is the expected cost. Here i, t is obtained from part (a).

4. See hw2.ipnyb.

References

[1] Jason D. M. Rennie and Nathan Srebro. *Loss Functions for Preference Levels: Regression with Discrete Ordered Labels*, Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling, 180-186, 2005.