MACHINE LEARNING HOMEWORK 1 – WRITTEN PORTION

ALLEN JIANG

1) Hypothesis Class (3 points). True/False (and why): You are using an algorithm that selects a hypothesis from a class that you know contains the optimal hypothesis for a given problem. In this case, there is no benefit to using regularization.

Answer: False. Regularization is still useful because it avoids over-fitting, which is possible if the hypothesis class contains the optimal hypothesis (this only implies that under-fitting is not possible, but not over-fitting).

2) Loss Function (4 points). For each of the following, state if the function is a valid loss function. If it is, state whether it would make a suitable loss function for binary classification. If it is not a valid loss function, why not? $\hat{y}$ is the predicted label and y is the correct label.

(1) $\ell(y, \hat{y}) = y - \hat{y}$
(2) $\ell(y, \hat{y}) = \frac{1}{3}(y - \hat{y})^2$
(3) $\ell(y, \hat{y}) = |(y - \hat{y})|/\hat{y}$
(4) $\ell(y, \hat{y}) = \max(0, 1 - y \cdot \hat{y})$

Answer:
a). Not a loss function, because negative loss values are possible. Logically speaking, you cannot have negative "loss", as you can't have error better than 0. This also presents a problem in adding loss values, because summing negative and positive values will cancel each other out, leaving a smaller error than reality.

b). This is a loss function, as it is the same as the squared error used in class (just scaled differently). It is always positive and well defined, and the distance between the prediction and actual value is mapped to some positive value that scales according to the distance. It is not a good loss function for binary classification, because you are penalized for having a prediction value that goes beyond the actual value, so you end up fitting a line that minimizes distance instead of trying to classify.

c). Not a loss function, because it is undefined when $\hat{y} = 0$. As your function can include a value of 0, it has to have a defined loss for every possible value of the function.

d). This is a loss function as it is positive everywhere and well defined for all values. It can also work as loss function for binary classification specifically in differentiating between positive and negative numbers, as when y and $\hat{y}$ are the same sign, then the function is 0,

but when they are different sign, then you get some non-zero number that scales according to how far the numbers are off.

3) Ranking (7 points). Ranking is a common supervised machine learning problem, such as in ranking search engine results. In a ranking task, the predictor is given a set of instances and returns an ordering over the instances. We haven't discussed how to design or train ranking algorithm, but some of the algorithms we have learned about can be used within a ranking task.

(1) How would you use a trained regression or classification algorithm to rank a set of instances? Assume that the model has already been trained for this purpose. You only need to describe how it will be used at test time.
(2) Provide a loss function suitable for ranking. This loss function need not be related to your answer in the first part of the question.

Answer: a) If you have a trained regression, then you can simply input each instance into the regression, get some predicted value that says how likely the instance is to be well-ranked (some rank index), and then sort the instances based on how high their predicted value was and assign a ranking label accordingly. In this case, you train your regression model to output a higher value for a better ranking, and a lower value for a worse ranking.

b) A loss function that could work would simply be a least squared error, $(y - \hat{y})^2$, so the farther away your predicted ranking was from the actual ranking, the larger the value would be. You would first rank a list of instances and sort according to part (a), then use the predicted ranking value from the actual ranking to generate a loss function.

4) Regularization and Overfitting. (11 points). Statisticians love linear models because these models are very simple and interpretable. Many variants of linear models have been proposed, and most of them are formulated as a (penalized) least squares objective. Consider these three least squares objectives:

$$\hat{\beta}_0 = \operatorname*{argmin}_{\beta_0} \|y - X_1\beta_0\|_2^2, \tag{1}$$

$$(\hat{\beta}_1, \hat{\beta}_2) = \operatorname*{argmin}_{\beta_1,\beta_2} \|y - X_1\beta_1 - X_2\beta_2\|_2^2, \tag{2}$$

$$\hat{\beta}_3 = \operatorname*{argmin}_{\beta_3} \|y - X_1\beta_3\|_2^2 + \lambda\|\beta_3\|_2^2, \tag{3}$$

where $\lambda > 0$, $y \in \mathbb{R}^n$, $X_1 \in \mathbb{R}^{n\times d_1}$, and $X_2 \in \mathbb{R}^{n\times d_1}$. (3) is well known as the ridge regression. The square norm acts as a penalty function to reduce overfitting. Prove

$$\|y - X_1\hat{\beta}_3\|_2^2 \geq \|y - X_1\hat{\beta}_0\|_2^2 \geq \|y - X_1\hat{\beta}_1 - X_2\hat{\beta}_2\|_2^2. \tag{4}$$

$\|y - X_1\hat{\beta}_0\|_2^2$ can be written as $\|y - X_1\hat{\beta}_0 - X_2(0)\|_2^2$ and therefore can be considered a specific subset of $\|y - X_1\beta_1 - X_2\beta_2\|_2^2$ where $\beta_1 = \hat{\beta}_0, \beta_2 = 0$. We know for this expression, that $(\hat{\beta}_1, \hat{\beta}_2)$ are the parameters such that $\|y - X_1\beta_1 - X_2\beta_2\|_2^2$ is minimized. Therefore, $\|y - X_1\hat{\beta}_0\|_2^2$ can only result in the same or higher, but cannot possibly make the expression any lower, as $\|y - X_1\hat{\beta}_1 - X_2\hat{\beta}_2\|_2^2$ is already the minimum. So, $\|y - X_1\hat{\beta}_0\|_2^2 \geq \|y - X_1\hat{\beta}_1 - X_2\hat{\beta}_2\|_2^2$.

For the third equation, we can make a similar argument. Both $\|y - X_1\hat{\beta}_3\|_2^2, \|y - X_1\hat{\beta}_0\|_2^2$ have the same form $\|y - X_1\beta\|_2^2$, but $\hat{\beta}_0$ is the exact parameter that minimizes that form, whereas $\hat{\beta}_3$ minimizes $\|y - X_1\beta_3\|_2^2 + \lambda\|\beta_3\|_2^2$ and therefore is some other parameter. Therefore, $\|y - X_1\hat{\beta}_3\|_2^2$ results in some value that is not the minimum, unless it happens to be that $\lambda = 0$, which in that case $\hat{\beta}_3 = \hat{\beta}_0$. Therefore, $\|y - X_1\hat{\beta}_3\|_2^2 \geq \|y - X_1\hat{\beta}_0\|_2^2$