Matthew Maya 04/09/2023

# Homework #3

1.) $X_n$ = input data sample, it's a vector
$Y_n$ = ground-truth class label of $X_n$
$\hat{Y}_n$ = output "soft-label"/confidence of a logistic regression Classifier given the input $X_n$
n = 1 to N
K = # of classes

1a.)

$$L_{Binary} = -\frac{1}{N}\sum_{k=1}^{K}(y_n\log(\hat{y}_n)+(1-y_n)\log(1-\hat{y}_n)) \qquad K=2$$

$Y_n$ = integer

$$-y_n\log(\hat{y}_n)+(1-y_n)\log(1-\hat{y}_n)$$

1b.)

$$L_{Binary} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} Y_{(n,K)}\log(\hat{Y}_{(n,K)}) \qquad K \geq 2$$

$Y_n$ = One-hot vector

1c.) 2 classes: class-0, Class-1
$X_n$ is in class-1 ($y_n = 1$)
$\hat{Y}_n = 0.8$

$$L(Y_n, \hat{Y}_n) = -(Y_n\log(\hat{Y}_n)+(1-Y_n)\log(1-\hat{Y}_n))$$

$$L(1, 0.8) = -[(1)\log(0.8)+(1-1)\log(1-0.8)] = -\log(0.8)$$

1d.) 3 classes: Class-0, class-1, class-2
$X_n$ is in class-2 ($Y_n = 2$)
$\hat{Y}_n = [0.1, 0.2, 0.7]^T$ $\qquad Y_n = [0, 0, 1]$

$$L([0,0,1],[0.1,0.2,0.7]^T) = 0 + 0 + [-(1)\log(0.7)+(1-1)\log(1-0.7)]$$
$$= -\log(0.7)$$

1e.) $f(x) = -\ln\left(\frac{1}{1+e^{-x}}\right) = -\left[\ln(1) - \ln(1+e^{-x})\right]$

$u = 1+e^{-x}$

$\frac{d}{du} = -e^{-x}$

$f'(u) = -\left[\ln(1) - \ln(u)\right] = -\left[0 - \frac{1}{u}\,du\right]$

$f'(x) = -\left[\frac{1}{1+e^{-x}}(-e^{-x})\right] = \frac{e^{-x}}{1+e^{-x}}$

$$\left(\frac{w}{v}\right)' = \frac{w'v - v'w}{v^2}$$

$w = e^{-x}$

$w' = -e^{-x}$

$v = 1+e^{-x}$

$v' = 0-e^{-x}$

$$f''(x) = \frac{-e^{-x}(1+e^{-x}) - (-e^{-x})(e^{-x})}{(1-e^{-x})^2} = -\frac{e^{-x}}{(1-e^{-x})^2}$$

2a.) MSE loss $= \frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2$

2b.) MAE loss $= \frac{1}{N}\sum_{n=1}^{N}|y_n - \hat{y}_n|$

2c.) MAPE loss $= \frac{1}{N}\sum_{n=1}^{N}\left|\frac{y_n - \hat{y}_n}{y_n}\right| * 100\%$

3a.) The output of a decision tree for regression looks like a Staircase because the tree partitions the input space into regio based on the feature values. Each split creates a new thresho for the feature, and the output remains constant within eac region.

3b.) Building a deep tree such that every leaf-node of the tree is a pure node might not always be a good strategy as it can lead to overfitting to the outliers affecting the rest of the data

3c.) total # of training samples: 60

3d.) max-depth: 2

3e.) # of 'pure' nodes: 2 (Node-2 and Node-4)

entropy of Node-0: $H(p) = -\sum_{n=1}^{N} P_n \log_2(P_n)$

samples = 60

[10, 20, 30]

$P_1 = \frac{10}{60} = \frac{1}{6}$

$P_2 = \frac{20}{60} = \frac{1}{3}$

$P_3 = \frac{30}{60} = \frac{1}{2}$

$$H = -\left[\tfrac{1}{6}(\log_2(1/6)) + 1/3(\log_2(1/3)) + \tfrac{1}{2}(\log_2(1/2))\right] \approx 1.46$$

3f.) leaf/terminal node: 3

4a.) if the outputs are i.i.d random variables or weakly correlated. High variance low bias

4b.) Bagging

5.) The goal of Boosting is to reduce Bias whereas the goal of bagging is to reduce variance. Bagging reduces the variance by creating more training data sets; Boosting decreases bias because sample data sets are weighted on their performance.

6a.) Bagging trains multiple models on bootstrap replicates of the training set and then averages the outputs from the simple models. Stacking trains various models using the outputs from many other models to make predictio

6b.) Stacking many polynomial models of the same degree might lead to overfitting of data since they'd likely capture similar patterns and make similar predictions.

6C.) Yes, stacking models of different types/structures can be useful as they're more likely to capture different aspects of the data. This is important that the base models are diverse enough and don't correlate strongly with each other as it can lead to overfitting.

7a-Overfitting.) Classiffication task #1 where the training accuracy is 100% and the testing accuracy is 50% is likely to be overfitting

7b- underfitting.) Classiffication task #2 where both the training and testing accuracy are 50% is likely to be underfitting

8a.) hyper-parameters determine the structure and complexity of the model. It's value is chosen before a learning algorithm is used.

Examples: for a polynomial model, the degree
# of epochs
learning rate
# of branches in decision tree
# of clusters

8b.) A validation set is needed to evaluate the performance of a model and tune the hyper-parameters

8C.) This can lead to overfitting as pre-selecting the optimal hyper-parameters on the training set the model can get use to the training set and perform poorly on new data

8d.) optimizing hyper-parameters using the testing set can also lead to overfitting. The test set should only be used to evaluate the final performance of the model, after hyperparameters have been selected using a seperate validation set to avoid errors in new data

9a.) Maximizing the margin in the input space will improve Classifier robustness against noises because it helps to seperate the data points of different classes with a larger distance, reducing any errors caused by noise.

9b.) Yes, the margin in the input space will be maximized by a nonlinear SVM since it uses Kernel functions to map the input space to a higher-dimensional feature space, which can better seperate data points of different Classes

9c.) SVM can potentially cause an "out-of-memory" error because it needs to compute the Kernel matrix which is memory inefficient for a large data set.

9d.) The purpose of using a Kernel function is to map the original non-linear observations into a higher-dimensional space which they become Seperable

10a.) We use weighted accuracy as compared to standard accuracy because the # of instances in the minority class can result in an error using standard accuracy. Weighted accuracy takes into account the class distribution giving more weight to that smaller class. This makes sure that the classifier is performing well on both classes instead of only the bigger class.

Standard accuracy can mislead results by predicting the majority class for all instances

10b.) Re-sampling,

- up-sampling
- down sampling

Sampling with replacement
Sampling without replacement
Generate synthetic data samples

11.) PMF for a discrete variable $X = [P_1, P_2, P_3, \ldots, P_K]$ and $\Sigma_K P_K = 1$

$$H(x) = -\Sigma_K P_K \log_2(p_K)$$

a.) non-negative

Prove $H(x) \geq 0$

when $p_K = 0$ then $H(x) = 0$

since $\Sigma_K P_K = 1$ we can use jensens inequality

$f(\Sigma_K t_K x_K) \leq \Sigma_K t_K f(x_K)$ $\quad$ $t_K$ = non negative weights such that $\Sigma K\, t_K = 1$

$$H(x) = -\Sigma_K P_K \log(P_K)$$
$$\leq -\log(\Sigma_K P_K^2)$$
$$\leq -\log(\Sigma_K PK)$$
$$= -\log(1) = 0$$

b.) reaches maximum when the PMF is a uniform distribution, i.e $P_K = 1/K$

$P_K = 1/K$ for all $K$

$$H(x) = \Sigma_K (1/K) \log(1/K)$$
$$= -(1/K)\, \Sigma K \log(1/K)$$
$$= -(1/K)\, \Sigma K (-\log(K))$$
$$= \log(K)$$

since $\log(K)$ is a constant value that doesn't depend on the specific values of $P_K$ it's maximized when $P_K = 1/K$ for all $K$. This means entropy reaches the maximum when the PMF is a uniform distribution