ECO481_Assignment2

Alexander Tran - 1006314089

3/19/2023

Exercise 1

1. It is more likely that (c) is true, as we would expect more training data to make the model less prone to overfitting and consequently better at generalizing. (a) is not true because a classifier trained on less training data is more likely to overfit. (b) is not true because when the feature space is larger, overfitting is more likely. (d) is not true because as the number of training examples goes to infinity, the bias of your model trained on that data will not significantly change.

2. We do not have enough information in either case, as $P(Z|X,Y) = \frac{P(X,Y|Z)P(Z)}{P(X,Y)} = \frac{P(X \cap Y \cap Z)}{P(X \cap Y)}$, and cannot obtain the necessary quantities with the given information. For example, we do not have any way to obtain $P(X \cap Y)$, without extra information such as whether or not X and Y are independant, or $P(X|Y)$.

3. The most correct option out of the four is (b): in both classifiers, a pair of features is assumed to be dependent. For Naive Bayes classifier, we assume that feature values are only independent given the label, and are not necessarily independant otherwise. In decision trees we do not assume independence between a pair of features either. While neither assume strict dependance, options (a), (c), and (d) are almost entirely incorrect.

Exercise 2

1. The Naive Bayes classifier is given by maximizing $P(Y=y|X=x) = \frac{P(X=x|Y=y)P(Y=y)}{P(X=x)}$ with respect to y, where in this case x is an observed list of words present in the text and y is one of three labels. As $P(X=x)$ does not depend on Y, we can simply maximize $P(X=x|Y=y)P(Y=y)$, and, assuming that features are independent given the label, we have $P(X=x|Y=y)P(Y=y) = P(Y=y)\prod_{j=1}^{p} P(x_j|Y=y)$.

Beginning with the first text, from www.wsj.com/, ignoring all other words whose word frequencies are not given in the table, we have the following words: ‘Dow’, ‘GM’, ‘forecast’, ‘$71.92’. We then calculate $P(Y=y|X=$[‘Dow’, ‘GM’, ‘forecast’, ‘$71.92’]) for all values y.

Given the word frequencies from the table, and the prior distribution for labels given by: 50% for ‘technical’, 40% for ‘financial’, and 10% for ‘no interest’ we calculate the following:

$P(Y=technical)*P(Dow|Y=technical)*P(GM|Y=technical)*P(forecast|Y=technical)*$
$P(\$<number>|Y=technical) = 0.50*0.00*0.00*0.00*0.01 < 5*10^{-9}$

$P(Y=financial)*P(Dow|Y=financial)*P(GM|Y=financial)*P(forecast|Y=financial)*$
$P(\$<number>|Y=financial) = 0.40*0.08*0.03*0.03*0.07 = 2.016*10^{-6}$

$P(Y=no\,interest)*P(Dow|Y=no\,interest)*P(GM|Y=no\,interest)*P(forecast|Y=no\,interest)*$
$P(\$<number>|Y=no\,interest) = 0.10*0.00*0.00*0.01*0.05 < 5*10^{-9}$

*Note that some frequencies show up as 0.00... in the table, meaning they must be less than 0.01.

As $Y=financial$ maximizes $P(Y=y|X=$[‘Dow’, ‘GM’, ‘forecast’, ‘$71.92’]), we predict the label of this text to be ‘financial’.

For the second text, from slashdot.org/, we have $X =$[‘network’, ‘capacity’, ‘hub’, ‘deal’, ‘IP’, ‘gigabits’]). Note that some words such as ‘gigabits’ require stemming to obtain ‘gigabit’.

Performing similar calculations:

$P(Y = technical|X = x) = 0.50 * 0.04 * 0.01 * 0.06 * 0.01 * 0.03 * 0.03 = 1.08 * 10^{-10}$
$P(Y = financial|X = x) = 0.40 * 0.01 * 0.00 * 0.00 * 0.02 * 0.00 * 0.00 < 8 * 10^{-13}$
$P(Y = no\, interest|X = x) = 0.10 * 0.00 * 0.00 * 0.01 * 0.00 * 0.00 * 0.00 < 1 * 10^{-13}$

As $Y = technical$ maximizes $P(Y = y|X = x)$, we predict the label of this text to be ‘technical’.

For the third text from www.linuxdevices.com/, we have $X =$[‘Intel’, ‘processor’, ‘processor’, ‘businesses’, ‘\$600’, ‘deal’, ‘smartphone’, ‘processors’, ‘Intel’, ‘wireless’, ‘chipset’, ‘businesses’, companies]). Again, some words require stemming.

Performing similar calculations:

$P(Y = technical|X = x) = 0.50*0.02*0.07*0.07*0.01*0.01*0.01*0.04*0.07*0.02*0.02*0.04*0.01*0.01 = 2.1952 * 10^{-22}$
$P(Y = financial|X = x) = 0.40*0.02*0.01*0.01*0.07*0.07*0.02*0.04*0.01*0.02*0.01*0.01*0.07*0.04 = 1.75616 * 10^{-22}$
$P(Y = nointerest|X = x) = 0.10*0.00*0.00*0.00*0.04*0.05*0.00*0.01*0.00*0.00*0.00*0.00*0.04*0.05 < 4 * 10^{-25}$

As $Y = technical$ maximizes $P(Y = y|X = x)$, we predict the label of this text to be ‘technical’.

2.
a. A Naive Bayes classifier can either handle it by treating it as one word, or as two separate words. Taking “network capacity” as an example, if handled as one word, then “network” and “capacity” must appear consecutively and in that order to be treated as one word, otherwise they will be counted as two separate words. In addition, “network capacity” will have its own word frequency, independent of the word frequencies of “network” and capacity”. In other words, an expression should not contribute to the word frequencies of its constituents. If expressions are to be treated as separate words then we can apply Naive Bayes as in (1), without having to change anything.

b. After text-preprocessing such as removing punctuation, removing stop-words, and stemming, the module CountVectorizer can be imported from the package sklearn.feature_extraction.text to build these expressions. The CountVectorizer object should be initialized with ngram_range=(min_n, max_n), where min_n and max_n are the lower and upper bounds for the range of n-values to be considered for different word n-grams. For example, ngram_range=(1, 2) means unigrams and bigrams should be considered, which will include words, as well as expressions such as “network capacity”. After, we can apply the CountVectorizer on the corpus using .fit_transform(corpus) and easily extract the feature names and frequencies using .get_feature_names_out() on the CountVectorizer and transforming the return value of .fit_transform(corpus) into an array.

Exercise 3

1. In this 3-fold cross-validation, during each split, two data points will be assigned to the training set and the remaining one data point will be assigned to the validation set. We have the following three (x,y) data points: (0,2), (2,2), and (3,1).

In the first split we have (2,2) and (3,1) in the training set, and (0,2) in the validation set. The algorithm will predict the average of y in the training set, which is equal to $\frac{2+1}{2} = 1.5$. This lead to a training MSE of $\frac{(2-1.5)^2+(1-1.5)^2}{2} = 0.25$ and a validation MSE of $(2 - 1.5)^2 = 0.25$.
Repeating the same process with (0,2) and (3,1) in the training set and (2,2) in the validation set results in a prediction of $\hat{y} = \frac{2+1}{2} = 1.5$, training MSE of $\frac{(2-1.5)^2+(1-1.5)^2}{2} = 0.25$ and validation MSE of $(2-1.5)^2 = 0.25$.
Once again repeating the same process, this time with (0,2) and (2,2) in the training set, and (3,1) in the validation set, we obtain a prediction of $\hat{y} = \frac{2+2}{2} = 2$, a training MSE of $\frac{(2-2)^2+(2-2)^2}{2} = 0$, and a validation MSE of $(1 - 2)^2 = 1$.

Thus, the overall average training and validation mean square errors from the 3-fold cross-validation are $\frac{0.25+0.25+0}{3} = 0.167$ and $\frac{0.25+0.25+1}{3} = 0.5$, respectively.

2. Using the same training and validation sets in each split as in (1), we fit a linear regression $Y = \beta_0 + \beta_1 X + \epsilon$ on the training data that minimizes MSE as in OLS regression. In split 1, we can fit $Y = 4 - X + \epsilon$, where $B_0 = 4$ and $B_1 = -1$, which gives predictions $Y = 4 - (2) = 2$ for the data point (2,2) and $Y = 4 - (3) = 1$ for (3,1), resulting in a training MSE of $\frac{(2-2)^2+(1-1)^2}{2} = 0$ and validation MSE of $(2 - (4 - (0)))^2 = 4$.
Similarly, in split 2 we estimate $B_0 = 2$ and $B_1 = -\frac{1}{3}$, predicting $Y = 2 - \frac{1}{3}(0) = 2$ for the data point (0,2), $Y = 2 - \frac{1}{3}(2) = \frac{4}{3}$ for the data point (2,2), and $Y = 2 - \frac{1}{3}(3) = 1$ for (3,1). This results in a training MSE of $\frac{(2-2)^2+(1-1)^2}{2} = 0$ and validation MSE of $(2 - \frac{4}{3})^2 = \frac{4}{9}$.
In split 3 we estimate $B_0 = 2$ and $B_1 = 0$, predicting $Y = 2 - 0(0) = 2$ for the data point (0,2), $Y = 2 - 0(2) = 2$ for the data point (2,2), and $Y = 2 - 0(3) = 2$ for (3,1). This results in a training MSE of $\frac{(2-2)^2+(2-2)^2}{2} = 0$ and validation MSE of $(1 - 2)^2 = 1$.
Thus, the overall average training and validation mean square errors from the 3-fold cross-validation are $\frac{0+0+0}{3} = 0$ and $\frac{4+\frac{4}{9}+1}{3} = 1.815$, respectively.

Exercise 4

1. Refer to submitted code (ECO481_Assignment2.ipynb).

2. Refer to submitted code (ECO481_Assignment2.ipynb).

3. Refer to submitted code (ECO481_Assignment2.ipynb).

4. Most topics seem to revolve around events in the US. Topics include US politics, COVID-19, the attack on the US capitol, weather, schools, and deadly events such as shootings and car accidents.

5. One approach is to take the headlines of each day and see what proportion of them contain at least n words from the vocabulary, where n is a positive integer to be decided, perhaps 2.

6. Refer to submitted code (ECO481_Assignment2.ipynb).

7. This is not a very good way of assessing economic policy uncertainty because it will create many false positives. It is very probable that many of these words, such as “uncertain”, “deficit”, and “legislation” will be used in contexts that do not relate to economic policy uncertainty. Thus, the coarse economic policy uncertainty index ends up overestimating by a significant amount.

8. Refer to submitted code (ECO481_Assignment2.ipynb).

9. Refer to submitted code (ECO481_Assignment2.ipynb).

10. Refer to submitted code (ECO481_Assignment2.ipynb).

11. This paragraph will highlight how Covid has stagnated growth in the stock market, given the supporting evidence, which includes the correlation between Covid and economic policy uncertainty, and the low returns on SP500. The general feeling is not optimistic, so people are hesitant to invest.