CSC2515: Homework #1

Due on September 26

Roger Grosse

Thomas Hollis

Problem 1

First, consider two independent univariate random variables X and Y sampled uniformly from the unit interval [0,1]. Determine the expectation and variance of the random variable Z, defined as the squared distance $Z = (X-Y)^2$. You are allowed to evaluate integrals numerically (e.g. using `scipy.integrate.quad` or `scipy.integrate.dblquad`), but you should explain what integral(s) you are evaluating, and why.

Solution

From Tutorial 1 we saw that:

$$\mu = \mathbb{E}[X] = \int_{-\infty}^{\infty} Xp(X)dX$$

$$\sigma^2 = \text{Var}[X] = \int_{-\infty}^{\infty} (X-\mu)^2 p(X)dX = \mathbb{E}[(X-\mu)^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

And we know that:

$$\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$$

$$\text{Var}[X+Y] = \text{Var}[X] + \text{Var}[Y] + 2\text{Cov}[X,Y]$$

In our case, since X and Y are independent (i.e. $\text{Cov}[X,Y] = 0$), thus:

$$\text{Var}[X+Y] = \text{Var}[X] + \text{Var}[Y]$$

We also know that since X and Y are random variables sampled from the interval [0,1] thus:

$$\mathbb{E}[X] = \frac{1}{2} = 0.5$$

$$\mathbb{E}[Y] = \frac{1}{2} = 0.5$$

$$\text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \frac{1}{3} - \frac{1}{4} = \frac{1}{12}$$

$$\text{Var}[Y] = \mathbb{E}[Y^2] - \mathbb{E}[Y]^2 = \frac{1}{3} - \frac{1}{4} = \frac{1}{12}$$

Since:

$$\mathbb{E}[X^2] = \int_0^1 X^2 p(X)dX = \frac{1}{3}\left[X^3\right]_0^1 = \frac{1}{3}$$

Hence, since we are given that $Z = (X-Y)^2$, we can say:

$$\mathbb{E}[Z] = \mathbb{E}[(X-Y)^2] = \mathbb{E}[X^2 - 2XT + Y^2] = \mathbb{E}[X^2] - 2\mathbb{E}[XY] + \mathbb{E}[Y^2]$$

$$\therefore \mathbb{E}[Z] = \int_0^1 X^2 p(X)dX - 2\int_0^1 \left(\int_0^1 (XY)p(X)dX\right)p(Y)dY + \int_0^1 Y^2 p(Y)dY$$

$$\therefore \mathbb{E}[Z] = \frac{1}{3} - \frac{2}{4} + \frac{1}{3} = \frac{1}{6}$$

$$\text{Var}[Z] = \mathbb{E}[Z^2] - \mathbb{E}[Z]^2 = \int_0^1 \left(\int_0^1 (X-Y)^4 p(X)dX\right)p(Y)dY - \left(\frac{1}{6}\right)^2 = \frac{1}{15} - \left(\frac{1}{6}\right)^2 = \frac{7}{180}$$

Thus, the answer is $\mathbb{E}[Z] = \frac{1}{6}$ and $\text{Var}[Z] = \frac{7}{180}$

Problem 2

Now suppose we sample two points independently from a unit cube in d dimensions. Observe that each coordinate is sampled independently from [0, 1], i.e. we can view this as sampling random variables $X_1, \cdots, X_d, Y_1, \cdots, Y_d$ independently from [0, 1]. The squared Euclidean distance can be written as $R = Z_1 + \cdots + Z_d$, where $Z_i = (X_i - Y_i)^2$. Using the properties of expectation and variance, determine $\mathbb{E}[R]$ and $\text{Var}[R]$. You may give your answer in terms of the dimension d, and $\mathbb{E}[Z]$ and $\text{Var}[Z]$ (the answers from part (a)).

Solution

From the question given we know that:

$$\mathbb{E}[R] = \mathbb{E}[Z_1 + \cdots + Z_d] = \mathbb{E}[(X_1 - Y_1)^2 + \cdots + (X_d - Y_d)^2]$$

Hence, we can state (from properties in Problem 1 and since all $\mathbb{E}[Z]$ are equal):

$$\mathbb{E}[R] = \mathbb{E}\left[\sum_{i=1}^{d} Z_i\right] = \sum_{i=1}^{d} \mathbb{E}[Z] = d \cdot \mathbb{E}[Z]$$

Similarly, we can state:

$$\text{Var}[R] = \text{Var}[Z_1 + \cdots + Z_d] = \text{Var}[(X_1 - Y_1)^2 + \cdots + (X_d - Y_d)^2] = \mathbb{E}[R^2] - \mathbb{E}[R]^2$$

$$\text{Var}\left[\sum_{i=1}^{d} Z_i\right] = \sum_{i=1}^{d} \text{Var}[Z] = d \cdot \text{Var}[Z]$$

Thus, the answer is $\mathbb{E}[R] = d \cdot \mathbb{E}[Z]$ and $\text{Var}[R] = d \cdot \text{Var}[Z]$

Problem 3

Write a function `load_data` which loads the data, preprocesses it using a vectorizer (`http://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_extraction.text`), and splits the entire dataset randomly into 70% training, 15% validation, and 15% test examples.

Solution

See code written in sections 1.1 to 1.6.

Problem 4

Write a function `select_model` which trains the decision tree classifier using at least 5 different values of `max_depth`, as well as two different split criteria (information gain and Gini coefficient), evaluates the performance of each one on the validation set, and prints the resulting accuracies of each model. You should use `DecisionTreeClassifier`, but you should write the validation code yourself. Include the output of this function in your solution PDF (`hw1_writeup.pdf`).

Problem 4 continued on next page...

Solution

See code written in sections 2.0 to 2.2. The performance outputs of the 10 different models are as follows:
Gini, depth = 2: 0.6530612244897959
Gini, depth = 10: 0.6979591836734694
Gini, depth = 50: 0.7122448979591837
Gini, depth = 200: 0.7142857142857143
Gini, depth = 1000: 0.7142857142857143
Entropy, depth = 2: 0.6224489795918368
Entropy, depth = 10: 0.7061224489795919
Entropy, depth = 50: 0.7224489795918367
Entropy, depth = 200: 0.726530612244898
Entropy, depth = 1000: 0.726530612244898
Maximum is: 0.726530612244898 (#9, i.e. Entropy, depth = 200)

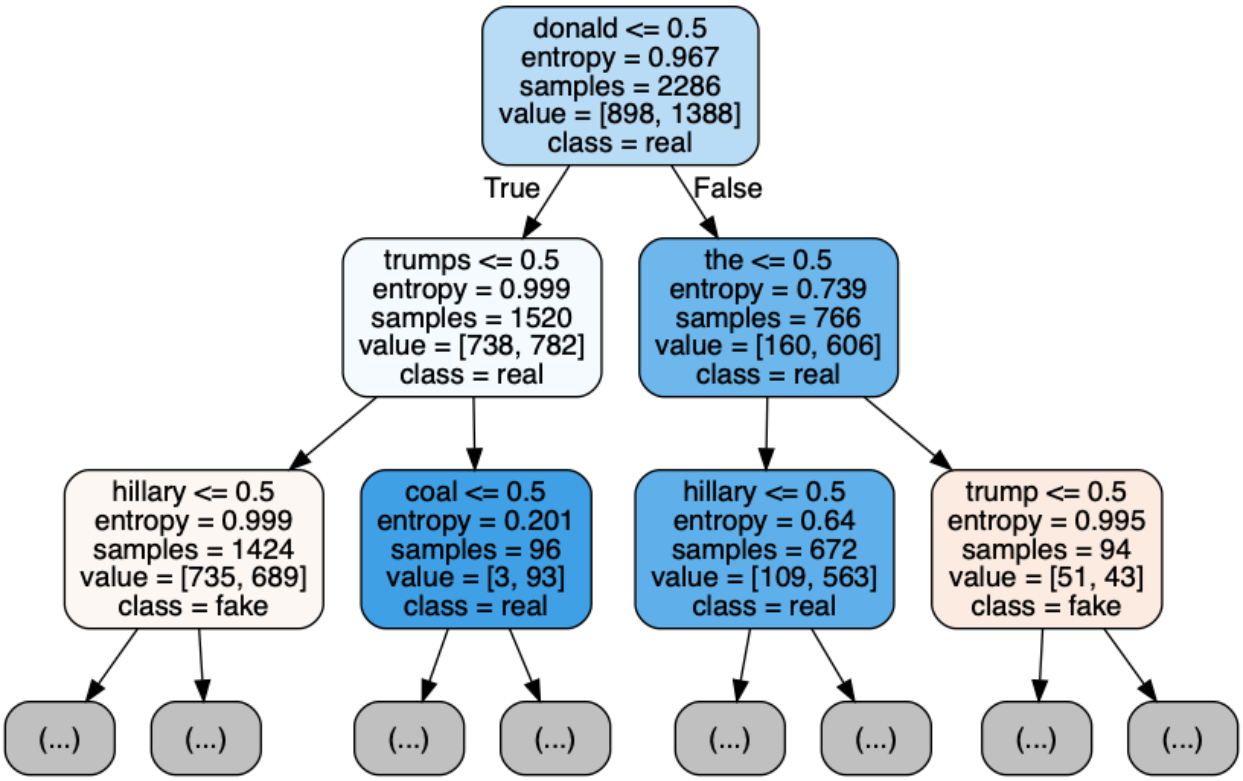
It is worth noting that this changes upon each iteration of the code (since the randomiser uses a new seed each time). Further investigation would have to be done for better hyperparameter modelling but this is not the purpose of the exercise at this time. Thus, the criterion chosen was entropy at a depth of 200.

Problem 5

Now lets stick with the hyperparameters which achieved the highest validation accuracy. Extract and visualize the first two layers of the tree. Your visualization may look something like what is shown below, but it does not have to be an image: it is perfectly fine to display text. It may be hand-drawn. Include your visualization in your solution PDF (`hw1_writeup.pdf`).

Solution

See code written in section 2.3. The first two layers of the tree above can be visualised as follows:



Problem 6

Write a function `compute_information_gain` which computes the information gain of a split on the training data. That is, compute $I(Y, x_i)$, where Y is the random variable signifying whether the headline is real or fake, and x_i is the keyword chosen for the split. Report the outputs of this function for the topmost split from the previous part, and for several other keywords.

Solution

See code written in section 3.

Topmost split ("donald") information gain: 0.05434667917675895
"the" information gain: 0.03521446940171813
"trumps" information gain: 0.042800696562309004
"hillary" information gain: 0.0342878764951946
"coal" information gain: 0.00027073544903877256
"trump" information gain: 0.011072155135465