COMP.4220 - Machine Learning

Homework 1

student name:

---

1. Let $\mathbb{E}[f(x)]$ and $\mathbb{V}[f(x)]$ denote expectation and variance of function f, respectively. The variance of $f(x)$ is defined by $\mathbb{V}[f(x)] = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2]$. Expand out the square and show that the variance can be written in terms of the expectation of $f(x)$ and $f(x)^2$. Explain each step of the simplification. (4 Marks)

2. Suppose we have three colored boxes r (red), b (blue), and g (green). Box r contains 3 apples, 4 oranges, and 3 limes, box b contains 1 apple, 1 orange, and 0 limes, and box g contains 3 apples, 3 oranges, and 4 limes.
   (a) If a box is chosen at random with probabilities $p(r) = 0.2$, $p(b) = 0.2$, and $p(g) = 0.6$, and a piece of fruit is removed from the box (with equal probability of selecting any of the items in the box), then what is the probability of selecting an apple?
   (b) If we observe that the selected fruit is in fact an orange, what is the probability that it came from the green box? (6 Marks)

3. (Extra Credit) - Evaluate the Kullback-Leibler divergence between two Gaussians $\mathcal{N}(x|\mu, \sigma^2)$ and $\mathcal{N}(x|m, s^2)$. You should start from the KL divergence formula and use the properties of the Gaussian distribution to solve the integrals. (2 Marks)

$$KL(p\|q) = -\int p(x) \ln q(x) dx + \int p(x) \ln p(x) dx$$

4. In this question, you complete implementation of the linear regression algorithm for polynomial curve fitting. Given a data set including the training set $\mathbf{x} = (x_1, \ldots, x_n)^\top$ and the target set $\mathbf{t} = (t_1, \ldots, t_n)^\top$, the linear regression algorithm minimizes the following error function:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$

where $y(x, \mathbf{w}) = \sum_{j=0}^{M} w_j x^j$ is a polynomial function. Complete the following steps:

(a) The starter code uses the `generate_synthetic_dataset` function to generate a synthetic 1-D data set. Implement a $\sin(2\pi x)$ curve in the function `func` that is passed to this function.

(b) In the `main` function, call the `generate_synthetic_dataset` function to generate a 1-D training set and a target set $(\mathbf{x}_{\text{train}}, \mathbf{t}_{\text{train}})$. Use the function implemented in the previous step to generate 10 points with additional noise with standard deviation 0.25.

(c) In the `main` function, generate a test data set $(\mathbf{x}_{\text{test}}, \mathbf{t}_{\text{test}})$ using the `func` function. Let $\mathbf{x}_{\text{test}}$ be 100 points uniformly distributed in range $[0, 1]$. You will be testing your trained linear regression model using this test set.

(d) The generated dataset should be transformed with polynomial features. Use the given `PolynomialFeatures` class to transform the raw data with orders in $[0, 1, 3, 9]$. For each of the orders in the list, you should have a transformed training set X_{train} and a transformed test set X_{test}.

(e) In the `LinearRegression` class, complete the `fit` function by implementing the solution using the least square method. This line should solve $y(x, \mathbf{w}) = X\mathbf{w} = \mathbf{t}$ to find $\mathbf{w}$.

(f) In the `LinearRegression` class, complete the `predict` function. Given a new input, this line should use the trained model (i.e., $\mathbf{w}^*$ from the previous step) to predict the output values.

(g) Now in the `main` function, instantiate and use the `LinearRegression` class to train four models for each given order.

(h) Test each model using the `predict` function and the test data set.

(i) Write a function that produces a plot similar to the Figure 1.4 in the textbook. This function can be part of the `main` function or a separate function that you call in `main`.

(j) In this step, you investigate the fitting error for the training and testing data sets. To do this firs implement the root mean squared error function, `rmse`. This should calculate the error between two vectors a and b as $\sqrt{(\sum_{n=1}^{N}(a-b)^2)/N}$.

$\mathbf{w}/M$	0	1	2	3	4	5	6	7	8	9
w_0^*										
w_1^*										
w_2^*										
w_3^*										
w_4^*										
w_5^*										
w_6^*										
w_7^*										
w_8^*										
w_9^*										

Table 1: Trained (optimal) model weights.

(k) Include a new part in the `main` function that loops over all polynomial orders in $[0, 9]$, fits a Linear Regression model using the training set and tests using the test set. Then plot the RMSE for each case in one figure with x-axis being the orders and the y-axis the RMSE. Your plot should look like Figure 1.5 in the textbook.

(l) Prepare a table with columns including the trained $\mathbf{w}^*$ values for orders in $[0, 9]$. Your table should look like the template Table 1. You can write code to format the table or use Word/Excel/Latex (do not return handwritten results).

(m) Based on the results in part (j) and in Table 1, which order would be best for this data set? Explain why?

(n) Explain at least two issues with the Linear Regression using least squares. What approaches can be used to mitigate each problem.

(15 Marks)