CS 412 Machine Problem 2

1 Question Answering

Q1. What are overfitting and underfitting? Hint: Page 11-12 of L3_Regression.pdf (10 points)

Q2. Please describe at least two strategies that can help avoid overfitting. (10 points)

Q3. Linear regression can only handle scenarios where the target value is a linear function of the input features. Is this statement correct? Please justify. (10 points)

Q4. Suppose you want to build a linear regression model to predict the price of a house based on its size, built year, and score of school district. No basis function or feature extraction will be applied to the input features. How many learnable parameters are there in the model? What are they? (10 points)

Q5. Please describe in your own words what cross-validation is, and what it is used for? Hint: Page 22-27 of L6_KNN.pdf (10 points)

2 Programming

The goal of this programming assignment is to give you a chance to play with linear regression and cross-validation. It is mandatory to use Python 3. Only the numpy package can be used to implement model learning and prediction, and cross-validation.

P1. Please follow the steps below to implement a linear regression model to learn a 3rd-order polynomial function. (20 points)

1. Generate 20 pairs of (x, y) values following the steps in P4 of MP #1. They are your training data.
2. Learn the parameters of a 3rd-order polynomial function on the training data via linear regression. You can use the linear regression code in our code tutorial as a template (available in Blackboard) or build your own code from scratch. Print the learned parameters. Also type or write down the formulation of the learned 3rd-order polynomial function in your submitted PDF file. Hint: think about how to construct the matrix corresponding to the input.
3. Use the learned model to make predictions on input values $x_1 = 0, x_2 = 0.25, x_3 = 0.5, x_4 = 0.75, x_5 = 1$, respectively. Print the predicted output values.
4. Use four different colors to respectively draw in a single figure (1) the noiseless sine function[1] used for data generation, (2) training data in Step 1, (3) (x, y) values in Step 3, and (4) the learned 3rd-order polynomial function.

P2. Perform 5-fold cross-validation to calculate the prediction error of a 3rd-order polynomial function on each validation fold. Use the root mean square error (Page 12 of L3_Regression.pdf) as the

[1] This tutorial (https://www.codesansar.com/python-programming-examples/sine-wave-using-numpy-and-matplotlib.htm) illustrates how to draw a sine function via numpy and matplotlib. Polynomial functions can be drawn similarly.

performance metric. For each validation fold, print the learned parameters and prediction error. Also print the average prediction error on all validation folds. (15 points)

Note a prediction error is calculated for a validation fold, and an average prediction error is calculated for a cross validation, averaging the prediction errors on all validation folds.

P3. Follow the steps below to find the optimal hyperparameter. (15 points)

1. Set the order of the polynomial function to 1, 3, 5, 7, 9, respectively, and for each order, perform 5-fold cross-validation as in P2 to calculate the average prediction error. Print (1) the average cross-validation error of each order, and (2) the optimal order.
2. For each order, draw the polynomial function learned in the first iteration of the cross-validation. Also draw all training (x, y) values in this figure to see how well different polynomial functions fit the data.

3 Optional Programming

The two questions below are optional. But completing them will give you a few bonus points.

O1. Use ridge regression (Page 15 of L3_Regression.pdf) with $\lambda = 0.001$ to fit a 9th order polynomial function to the training data in P1. Draw in a single figure (1) the training data, (2) the polynomial function learned via ridge regression, and (3) the polynomial function learned via linear regression as in P3. (3 points)

O2. Perform 5-fold cross validation to find the best hyperparameter λ of a 9th order polynomial function. At least 5 different values of λ should be chosen and tested. Print (1) the average prediction error of each λ value and (2) the optimal λ value. Draw in a single figure (1) the training data and (2) the 9th order polynomial functions corresponding to different λ values learned in the first iteration of the cross-validation. (3 points)

4 Submission

Please follow the instructions below for submission.

- You need to upload two files to Blackboard: a PDF file and a .py file[2]. Do not compress them into a single ZIP file.
- The PDF file contains all your solutions to this homework. For Question Answering, you can either type answers or handwrite them and take a photo. For Programming, you need to print the results or draw figures, and take screenshots.
- The .py file contains all your code for the programming problems.

[2] Using Jupyter Notebook and submitting a .ipynb file instead of a .py file are fine.