

Instructions:

In this lab, you will continue with model assessment and selection by

- computing the 2-fold cross-validation estimate of prediction error on paper for a trivial dataset
- implementing K -fold cross-validation for polynomial regression to determine the best model complexity, i.e., the p -th degree of the polynomial
- plotting learning curves to investigate the effects of the number of training samples and model complexity on the bias and variance of the model

Part 1: 2-fold Cross-Validation (on paper)

In this part of the lab, you will compute the 2-fold cross-validation estimate of prediction error for a trivial dataset.

Consider a training set consisting of the following four (4) training samples:

Sample	Value
(x_1, y_1)	(2, 4)
(x_2, y_2)	(3, 2)
(x_3, y_3)	(5, 3)
(x_4, y_4)	(6, 2)

Compute the average 2-fold cross-validation estimate of prediction error, using mean absolute error (MAE) as defined in Lab Exercise #2, for a first-degree polynomial regression model, i.e., $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$. Assume that the samples are sub-divided into two subsets (i.e., $K=2$) as follows: the first subset contains (x_1, y_1) and (x_2, y_2) , and the second subset contains (x_3, y_3) and (x_4, y_4) .

Part 2: K-fold Cross-Validation

[15 marks] In this part of the lab, you will implement K -fold cross-validation function for polynomial regression and output the MAE for both the training set and validation set in your script *CrossValidation_Lab5.py*. However, you shall implement K -fold cross-validation without using the scikit-learn package. Implement `poly_kfoldCV()` as defined below by replacing `???` with appropriate code:

```
# k-fold cross-validation for polynomial regression
# Inputs:
# x: training input
# y: training output
# p: degree of the fitting polynomial
# K: number of folds
# Outputs:
# train_error: average MAE of the training set across all  $K$  folds
# cv_error: average MAE of the validation set across all  $K$  folds
def poly_kfoldCV(x, y, p, K):
    ???

    return train_error, cv_error
```

You will verify the correctness of your `poly_kfoldCV()` implementation in Part 3.

Part 3: Model Assessment and Selection

[15 marks] In this part of the lab, you will perform polynomial regression on a provided dataset and use the `poly_kfoldCV()` function that you implemented in Part 2 to determine the best model complexity.

Steps:

- 1) Download the dataset, *data_lab5.csv*, which contains 100 rows and 2 columns, from BCIT Learning Hub (Content | Laboratory Material | Lab 5) and save it in your working directory. The header row denotes the x and y columns.
- 2) Add to your script, *CrossValidation_Lab5.py*, to read from *data_lab5.csv*. To verify the correctness of your `poly_kfoldCV()` implementation in Part 2, ensure that for $p = 1$, $K = 5$, `train_error`=1.0355 and `cv_error`=1.0848. In the event that your solution is incorrect, you will need to debug and correct your implementation before proceeding with this lab exercise.
- 3) Use `poly_kfoldCV()` to perform 5-fold cross-validation (i.e., $K = 5$) on the dataset for each $p = [1, 2, \dots, 15]$.
- 4) Plot the training error and cross-validation estimate of prediction error returned by `poly_kfoldCV()` as a function of p . Include in your plot, the title, x-axis label, y-axis label and a legend.
- 5) Based on the plot from Step (4), determine the best model complexity (i.e., p -th degree of the polynomial) for this dataset in the output.

Part 4: Learning Curves

[30 marks] In this part of the lab, you will plot learning curves as a function of the number of training samples, N , for different degrees of polynomial, p , and investigate the effects of N and p on the bias and variance of the model. You will implement the code in your script, *CrossValidation_Lab5.py*.

Steps:

- 1) For $p = 1$ and each training sample size, $N = [20, 25, 30, \dots, 100]$:
 - a) Use `poly_kfoldCV()` that you implemented in Part 2 to perform 5-fold cross-validation on the first N samples in the dataset.
 - b) Plot the training error and cross-validation error returned by `poly_kfoldCV()` as a function of N . Include in your plot, the title, x-axis label, y-axis label and a legend.
- 2) Repeat Step (1) for $p = [2, 7, 10, 16]$.
- 3) Based on the learning curves, which degree polynomial has the
 - a) highest bias? Substantiate your answer in the output.
 - b) highest variance? Substantiate your answer in the output.
- 4) Based on the learning curves, which degree polynomial would you use if only the first
 - a) 50 samples are provided? Substantiate your answer in the output.
 - b) 80 samples are provided? Substantiate your answer in the output.

Deliverable:

All work submitted is subject to the standards of conduct as specified in BCIT Policy 5104. No late assignments will be accepted.

[Oct 7, 2022 @1730] Ensure that your source code is adequately commented, along with output of the answers to Part 3 - Step (5), Part 4 - Step (3) and Part 4 - Step (4), and submit using the filename *CrossValidation_Lab5.py* to BCIT Learning Hub (Laboratory Submission | Lab 5).