Labs
Machine Learning Course
Fall 2021

EPFL
School of Computer and Communication Sciences
Nicolas Flammarion & Martin Jaggi
www.epfl.ch/labs/mlo/machine-learning-cs-433

Problem Set 4, Oct 14, 2021 (Cross-Validation and Bias-Variance Decomposition)

Goals. The goal of this exercise is to

- Implement 4-fold cross-validation.
- Understand bias-variance decomposition.

Setup, data and sample code. Obtain the folder `labs/ex04` of the course github repository

github.com/epfml/ML_course

This exercise is partly based on the materials from last week (`labs/ex03`). If you don't have it already, please finish the ex03 first. You might directly reuse/copy some of the functions you implemented yourself during previous exercises, e.g., `ridge_regression()`, `least_squares()` and `split_data()`.

1 Cross-Validation

Exercise 1:

Implement 4-fold cross-validation.

- Copy your code from last week, and fill in the corresponding templates, i.e., `ridge_regression() to ridge_regression.py`, `build_poly()` to `build_polynomial.py`, and `least_squares()` to `least_squares.py`.

  In this exercise, please fill in the notebook function `cross_validation_demo()`, and perform 4-fold cross-validation for polynomial degree 7. Plot the train and test RMSE as a function of λ. The resulting figure should look like Figure 1.

  Note that we can use `numpy.random.seed(seed)` to generate two independent train and test data splits.
- How will you use 4-fold cross-validation to select the best model among various degrees, say from 2 to 10? Write code to do it.
- **BONUS:** Using cross-validation, one can also compute the variance of the RMSE, over the folds. This tells us how confident we could be of our *model selection*. You can use box-plots to do this.

2 Visualizing Bias-Variance Decomposition

Last lecture we have introduced model selection, and we have seen that the model complexity is crucial to the performance. In this problem, we will further investigate the effect of *model complexity* with the concept of *bias-variance decomposition* (you will learn later in the lectures this afternoon).

Consider linear regression with a one-dimensional input and using polynomial feature expansion. The maximum degree d regulates the complexity of the class. Assume that D is the data distribution, and f_S denotes the model trained on the dataset S.

We will show by experiment that the following is true:

- Assume that we only allow simple models, i.e., we restrain the degree to be small:
  - We then typically will get a large bias, i.e., a bad fit.
  - On the other hand the variance of $L_D(f_S)$ as a function of the *random* training set S is typically small.

  We say that we have high bias but low variance.
- Assume that we allow complex models, i.e., we allow large degrees:
  - We then typically will find a model that fits the data very well. We will say that we have small bias.
  - But we likely observe that the variance of $L_D(f_S)$ as a function of the *random* training set S is large.

  We say that we have low bias but high variance.

Often it is hard to achieve both small bias and variance. This is called *bias-variance trade-off.*

Exercise 2:

Visualizing the bias-variance trade-off.

- Complete the notebook function `bias_variance_demo()` and experiment with different seeds, number of training and testing examples with least square, to get a plot that looks similar to Figure 2, i.e. on average the train error should go down and test error should go up for higher degrees. NOTE that you should COPY the function `split_data()` implemented in `ex03` to the template file `split_data.py`.
- Look at the variance of test errors. Does it increase with the degree of polynomial?
- **BONUS:** Another good visualization is to use the box-plot to get an appropriate visualization. You can clearly see the distribution of test error.
- **BONUS:** Produce the same plot with Ridge regression. You will have to automatically find the best λ for each degree. You do this using K-fold cross-validation (CV) only on the training set. So you need to insert the CV code inside. You also have to make sure that you chose an appropriate range of λ, so that you achieve the minimum test error. What would you expect to happen if you replace least-squares with Ridge regression? Does the experiment result meets your expectation?

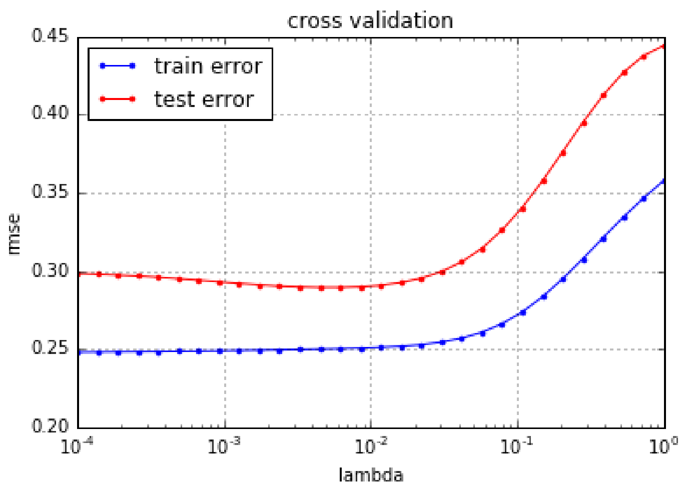

Figure 1: Effect of λ on training and test errors, calculated using 4-fold cross-validation

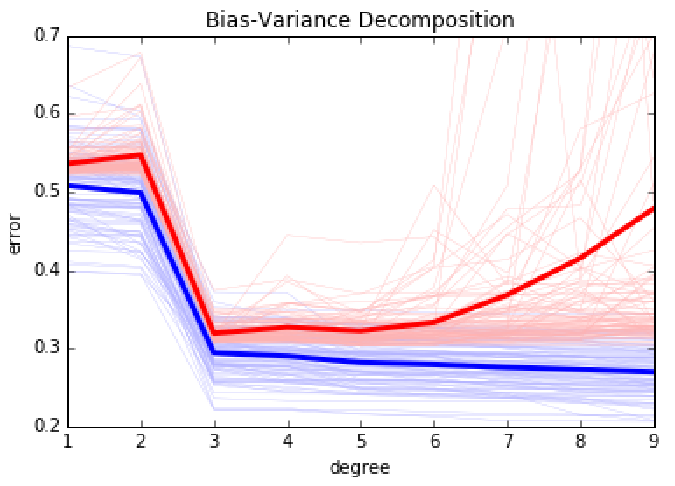

Figure 2: Visualizing Bias-Variance Trade-off: Error vs. Model Complexity