CSP 304: Machine Learning Lab (Spring 2023)

Indian Institute of Information Technology (IIIT), Kota
Instructor: Ankit Sharma

Posted in week: Feb 27- Mar 3, 2023
Due in week: Mar 6-10, 2023

Homework 1

Model selection and complexity control

Topic A — This part of homework illustrates the use of resampling methods for model selection (complexity control), and for comparing prediction accuracy of a learning method.

Data — Haberman's Survival Data Set, taken from UCI Machine Learning Repository at:

http://archive.ics.uci.edu/ml/datasets/Haberman%27s+Survival,

contains 306 cases from a study conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of female patients who had undergone surgery for breast cancer. Each of the 306 patient records has a class label (alive/dead) indicating whether the patient survived 5 years or longer after surgery, or died within 5 years. Each patient record has 3 inputs: Age of patient; Year of operation, and Number of positive auxiliary nodes detected. The goal is to estimate the decision boundary between the two classes, using $x_1 \sim$ Age of patient and $x_2 \sim$ Number of positive auxiliary nodes, to predict patient's survival. As a part of preprocessing, pre-scale each input to a $[0, 1]$ range, using min and max values of that input.

Learning Method — The decision boundary is estimated using k-nearest neighbors (KNN) classifier, where the class label y for a given input x is found by a majority vote of y-values of k training samples closest to x. Note that only odd k-values are used, in order to avoid ambiguity (in majority voting).

A1 — **Method Implementation (30 points):** Write your own code to implement the k-nearest neighbors classifier.

A2 — **Model Selection (30 points):** Estimate an optimal value of k using leave-one-out (LOO) cross-validation. Show this cross-validation error for different k-values ($k = 1, 3, 7, .., 99$) in a tabular or graphical form, and indicate the optimal k-value. Show the corresponding decision boundary along with the training data, in the two-dimensional input space (x_1, x_2).

A3 — **Prediction accuracy of a learning method (40 points):** The test error of a k-nearest neighbor classifier can be estimated using the double resampling procedure.

1. Use 5-fold cross-validation for estimating test error, by taking out every 5-th sample, ordered by Age, as a test set. This will result in 5 different partitions (folds) of the data into training + test sets
2. Within each fold, estimate an optimal value of k via LOO cross-validation.

Present the results of this double resampling in a table with 5 rows, where each row shows an optimal value of k, LOO validation error, and estimated test error (for that fold). Note that optimal k-values may be different for different folds. Calculate the true test error of a method is an average of test errors for 5 folds. Also calculate the mean value of LOO validation errors for 5 folds.

Does an optimal LOO cross-validation error used for model selection in A2 provide an accurate estimate of the true test error found in A3?

Notes — For A2 and A3 you can use a ready-made implementation of KNN as well i.e. it is not mandatory (although *strongly encouraged*) to use your own implementation (from A1).

Parametric Models

Topic B — This part of homework illustrates the estimation of parametric models using maximum likelihood estimation (MLE) methodology.

Data — Three pairs of training data and test data are given in the data.zip file attached.

In this problem, you will implement a program to fit two multivariate Gaussian distributions to the 2-class data and classify the test data by computing the log odds $log\frac{P(C_1|x)}{P(C_2|x)}$. The priors $P(C_1)$ and $P(C_2)$ should be estimated from the training data. Three pairs of training data and test data are given. The parameters μ_1, μ_2, $\mathbf{S_1}$ and $\mathbf{S_2}$, the mean and covariance for class 1 and class 2, are learned in the following three models for each training data and test data pair,

- **Model** 1**:** Assume independent $\mathbf{S_1}$ and $\mathbf{S_2}$ (the discriminant function is as equation (5.17) in the textbook).
- **Model** 2**:** Assume $\mathbf{S_1} = \mathbf{S_2}$. In other words, shared **S** between two classes (the discriminant function is as equation (5.22) in the textbook).
- **Model** 3**:** Assume $\mathbf{S_1}$ and $\mathbf{S_2}$ are diagonal (the *Naive Bayes* scenario as equation (5.24) in the textbook).

B1 — **Method Implementation (70 points):** Your program should return and print out the learned parameters $P(C_1)$,$P(C_2)$, μ_1 and μ_2 of each data pair. Your implementation of model 1 - 3 should return and print out the learned parameters $\mathbf{S_1}$ and $\mathbf{S_2}$.

B2 — **Prediction accuracy of the learning method (30 points):** For each test set, print out the error rates of each model (two models per each test set). Match each data pair to one of the models and justify your answer. Also, explain the difference in your results in the report.

Deliverables

Report — A brief report in PDF format describing the various experimental tasks mentioned above. Description should include the details of your experiments (process & setup), results and discussions/comments on the observations.

Code — Properly commented code file(s) or notebook(s) or any other setup and/or read-me files. All programming questions must be written in MATLAB/Python, no other programming languages will be accepted. And for Python only numpy, scipy and matplotlib can be relied on to implement the algorithm. Similarly, for MATLAB you have to provide your own implementation using basic functionalities and no ready made libraries.

For **B**1-**B**2 your main function should be named: MultiGaussian(*training data*: file name of the training data, *testing data*: file name of the testing data, *Model*: the model number). The function must output the learned parameters and error rates as required. You can submit additional

files/functions (as needed) which will be used by the main function.

If there are several of them, please bind all the code files (NOT the report file) into a single ZIP file and upload as a response to Google classroom. Although you have worked in groups, but each member of the group should return the Google classroom assignment by uploading same material.

Notes

Although not mandatory, the report is encouraged to be written in LaTeX preferably via www.overleaf.com using the NIPS format available here: https://neurips.cc/Conferences/2021/PaperInformation/StyleFiles. Also, an example overleaf NIPS template can be readily found here: https://www.overleaf.com/latex/templates/neurips-2021/bfjnthbqvhgs.
