CS 4033/5033 - Machine Learning: Homework 5

Spring 2022

Due: Monday, April 11, 2022

In this homework we will be practicing with techniques on regularization, model assessment, and model selection. In this direction we will work with two different datasets from the UCI repository.

Below I assume that you will be using scikit-learn.

Exercise 1 – Preprocessing for Regression (20 points). We will be using a UCI superconductivity data set that is available at:

https://archive.ics.uci.edu/ml/datasets/Superconductivty+Data.

This dataset has 21,263 examples using 81 real-valued attributes, with a real-valued target variable. For this assignment, we will use the file `train.csv` and we can ignore the file `unique_m.csv`. The target value that we are predicting is the critical temperature in Kelvins, which is the last column in the dataset.

(a) Remove 20% of the examples and keep them for testing. You may assume that all examples are independent, so it does not matter which 20% you remove. However, the testing data should not be used until after a model has been selected.

(b) Split the remaining examples into training (75%) and validation (25%). Thus, you will train with 60% of the full dataset (75% of 80%) and validate with 20% of the full dataset (25% of 80%).

Exercise 2 – Regression (30 points). We will be using the dataset from Exercise 1. For this problem please use the scikit-learn method,
`sklearn.linear_model.ElasticNet`.

(a) Fit an elastic net model to the training data with each possible combination of the following L_1 and L_2 regularization weights.

- $\lambda_1 = 0, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1$
- $\lambda_2 = 0, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1$

If you check the documentation, you'll see that the input arguments for `ElasticNet` do not include λ_1 and λ_2. Instead, they include `alpha`, and `l1_ratio`. λ_1 is `alpha * l1_ratio`, and λ_2 is `alpha * (1 - l1_ratio)`.

(b) For each model trained in step (a), make a prediction for each training example, using the `predict` method for `sklearn.linear_model.ElasticNet` and calculate the mean squared error (MSE) on the training examples. Report these values in your writeup.

As a reminder, the mean squared error (MSE) on a dataset of size m is:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - h(x_i))^2\,,$$

where, as usual, $h(x_i)$ is the prediction of our hypothesis and y_i is the correct (ground truth) prediction as this is defined in the dataset.

(c) This time, for each model trained in step (a), make a prediction for each validation example and calculate the mean squared error on the validation examples. Report these values in your writeup.

(d) Which model (i.e., pair of λ_1 and λ_2) performed best on the training data? Which model performed best on the validation data? Report this in your writeup.

(e) Find the best hyperparameter set (pair of λ_1 and λ_2) on the validation data. Train a model with the same λ_1 and λ_2 on both the training and validation data. Using this model, make predictions for each testing example and calculate the mean squared error on the test examples. Report this value in your writeup.

Exercise 3 – Preprocessing for Classification (20 points). Here we will do the same thing as we did for Exercise 1, but for a different dataset. In particular, we will be using a UCI simulated electrical grid stability data set that is available here:

https://archive.ics.uci.edu/ml/datasets/Electrical+Grid+Stability+Simulated+Data+.

This dataset has 10,000 examples using 11 real-valued attributes, with a binary target (stable vs. unstable). The target value that you are predicting is the last column in the dataset.

(a) Remove columns 5 and 13 (labeled p1 and stab); p1 is non-predictive and stab is target column that is exactly correlated with the binary target you are trying to predict (if this column is negative, the system is stable).

(b) Change the target variable to a number. If the value is stable, change it to 1, and if the value is unstable, change it to 0.

(c) Remove 20% of the examples and keep them for testing. You may assume that all examples are independent, so it does not matter which 20% you remove. However, the testing data should not be used until after a model has been selected.

(d) Split the remaining examples into training (75%) and validation (25%). Thus, you will train with 60% of the full dataset (75% of 80%) and validate with 20% of the full dataset (25% of 80%).

Exercise 4 – Logistic Regression (30 points). For this problem you can use the scikit-learn method `sklearn.linear_model.LogisticRegression`.

(a) Fit a logistic regression model with L_2 regularization (use the default value of λ) and another logistic regression model with no regularization. Note that, per the documentation, by default the function `sklearn.linear_model.LogisticRegression` performs L_2 regularization and in order not to use regularization we need to pass `penalty='none'` as parameter in the creation of the model.

(b) Using the two models created in part (a) make a prediction for each validation example. What is the empirical risk (using the 0-1 loss) of each model on the validation set and what is the confusion matrix of each model (again, on the validation set)?

(c) Which model performed better on the validation data? Report this in your writeup. We consider a model better than another one if the empirical risk of the first model is lower than the empirical risk of the second model. In case the empirical risk is a tie, compare the models using precision and determine which one is better.

Now, train a new logistic regression model on the training and validation data using whichever measure created the best model (i.e., is it beneficial to use L_2 regularization in this dataset, or not?) in (a)-(b). Make a prediction for each testing example. Report the empirical risk on the test set and the confusion matrix that corresponds to the predictions of this last model on the test data.