CSCI1410 Fall 2020
Assignment 5: Supervised Learning

Code Due Monday, November 30 at 11:59pm ET

Writeup Due Tuesday, December 1 at 11:59pm ET

1 Goals

The topic of this assignment is supervised learning, *a.k.a.* function approximation. There are two kinds of function approximation, classification and regression. You will implement one algorithm, the perceptron, and will use it to solve both a classification and a regression problem. You will be able to gauge the performance of your solutions against TA solutions: k nearest neighbors for classification, and linear least squares (implemented using Python's `sklearn` library) for regression.

2 Introduction

In supervised learning, you are given a data set of pairs, where the first element of each pair is a list of features $\boldsymbol{x} \in \mathbb{R}^d$, and the second element is a label, $y \in R = \{0, 1\}$ in a (binary) classification problem, or $y \in R = \mathbb{R}$ in a regression problem. The underlying assumption is that those data represent a function $f : \mathbb{R}^d \to R$, and the goal is to recover f from the data.

A good predictor (classifier or regressor) is one which does not make too many mistakes, in the case of classification; or does not make predictions that are too far off from the true label, in the case of regression. In either case, it is not enough to tally errors on the training data alone: i.e., on all the points on which the predictor learns. It is necessary to hold out some data, as a test set, to see whether the predictor can generalize well. Analogously, it is not enough to test students on problems that were covered in class. An exam asks students slightly different questions to gauge the depth of their understanding of the material.

3 Algorithms

There are numerous classification and regression techniques. The TAs have implemented two baseline algorithms against which you can compare the performance of your perceptron. The first is k nearest neighbors (kNN), for classification, and the second is the method of least squares, for regression.

The kNN algorithm is not really much of a learning algorithm at all, as it does not have a training phase. When queried with a data point $\boldsymbol{x}$, it simply asks $\boldsymbol{x}$'s k closest neighbors to vote on whether $\boldsymbol{x}$ should be classified as 0 or 1. (The hyperparameter k is usually chosen to be odd, to avoid ties.) When $\boldsymbol{x} \in \mathbb{R}^d$, as is the case in this assignment, "closeness" can be measured using Euclidean distance.

Regression is also called *data fitting*, since the goal is to fit a curve to the data. The idea of linear least-squares regression is to fit a line to the data such that the sum of the squared residuals (i.e., errors) between the predictions and the labels (i.e., $\sum_{(\boldsymbol{x},y)}(\hat{f}(\boldsymbol{x}) - y)^2$, where $\hat{f}$ is the function learned by the regression algorithm) is minimal. There exists a closed form solution to linear least-squares regression, which involves inverting a (necessarily square) matrix, an $O(n^3)$ operation, where n is the dimension of the matrix. This method is usually preferable in cases where matrix inversion is tractable.

Your main task in this assignment is to implement the perceptron algorithm. Recall from lecture that perceptrons are feedforward neural networks. A *simple* perceptron is a feedforward neural network with only one layer: i.e., all the inputs (i.e., features) feed into a single output layer. More specifically, a weighted sum of all the inputs is fed into this output layer. Perceptrons can function as classifiers or as regressors, depending on the form of the output layer. When the outputs apply threshold functions (or smooth variants thereof: e.g., sigmoidals) to the weighted sum, they function as classifiers; alternatively, when the weighted sum itself is the output, they function as regressors. Although the perceptron learning rule was originally developed for classification (using a hard threshold function to compute the output), it turns out that the exact same perceptron learning rule also applies to regression. (See the lecture notes for details.)

4 Data Sets

4.1 Breast Cancer Diagnostic Dataset

This dataset contains features calculated from a digital image of a fine needle aspirate of a breast mass, and a label representing the diagnosis, i.e., benign or malignant. There are 569 data points and 31 features, which consist of the patient's ID together with the mean, standard error, and "worst" (i.e., the mean of the three largest values) of ten measurements, such as radius, perimeter, area, etc. Your goal is to use supervised learning to predict the classification label: i.e., benign or malignant. You can find out more about the data here. Please also refer to the file `breast_cancer_diagnostic_description.txt`.

4.2 Student Achievement Dataset

This dataset is a record of student achievement in 2008 at a secondary education at a Portuguese school. Each student is represented by a feature vector that summarizes their demographic, social, and school-related characteristics. Your goal is to use supervised learning to predict the regression label: i.e., a student's 3rd marking period grade. You can find out more about the data here. Please also refer to the file `student_dataset_description.txt`.

4.3 Data Files

These two datasets are available in `.csv` files, with accompanying descriptions in `.txt` files. Additionally, we are releasing `preprocess_student_data.py`, python code which "cleans" the student dataset (i.e., morphs it into a more readily usable form).

5 Stencil Code

The stencil code for this assignment consists eight files. Below we provide a high-level overview of each. Read the docstrings and comments for more detailed information.

- `classification.py` imports `breast_cancer_diagnostic.csv` and then classifies this dataset using a `KNNClassifier` or a `Perceptron`.
- `regression.py` imports `student_por.csv` and then regresses on this dataset using `SKLearnModel` or a `Perceptron`.
- `knn.py` contains the `KNNClassifier` class, which implements the kNN algorithm for classification. This implementation is provided for you.

  Note: In the stencil code, we use the phrase *anchor points* to describe the nearest neighbors.
- `sklearn_model.py` contains the `SKLearnModel` class, which implements regression using the `sklearn` library function. This implementation is also provided for you.

- `perceptron.py` contains the `Perceptron` class, for which you will implement `train`, `predict` and `evaluate`. **You must implement a perceptron for both classification and regression.**
- `supervisedlearner.py` contains `SupervisedLearner`, the abstract superclass of the other supervised learning classes for this assignment, namely `KNNClassifier`, `SKLearnModel`, and `Perceptron`.
- `cross_validate.py` splits the data into train and test sets for cross validation. You should be sure to cross validate both your classification and regression models.

For this assignment, you should edit only the following files:

- `perceptron.py` - implement the `Perceptron` class for both classification and regression by filling in `step_function`, as well as the abstract methods `train`, `predict`, and `evaluate`.
- `classification.py` - classify the Breast Cancer Diagnostic dataset using your `Perceptron` model. Adjust the `learning_rate` to optimize performance: i.e., the accuracy during cross validation, where accuracy is the number of *correct* predictions divided by the total number of predictions.

  Also implement the `optimal_k` method, which optimizes the value of k: i.e., the number of neighbors. *Be sure k is set to the best value you find when you submit.*
- `regression.py` - perform regression on the Student dataset using your `Perceptron` model. Adjust the `learning_rate` to optimize performance: i.e., the mean squared error—the residual sum of squares divided by the size of the dataset—during cross validation.

  Another way to gauge your performance in this regression task to measure your accuracy, which for the Student dataset can be defined as 1 less the mean squared error divided by 20 (20 is the normalizing factor because [0,20] is the range of the response variable). So, if your mean squared error is 1.5, your accuracy is $1 - {}^{1.5}/_{20} = 92.5\%$.)

Note: You will find that you achieve different degrees of accuracy depending on the `train_size` in `classification.py` and `regression.py`: i.e., the percentage of the dataset you use for training. Feel free to experiment with different values of `train_size` as well to optimize model performance.

6 Written Questions

Answer the following questions in clearly labeled sections.

1. Recall that a single-layer perceptrons can implement the Boolean functions AND and OR.
   (a) Design a single-layer, two-input perceptron that implements the Boolean function $X \wedge \neg Y$. By "design," we mean derive a set of weights that implements this function.
   (b) Design a two-layer, two-input perceptron that implements the XOR function.
   Hint: Make use of part (a) of this question.
2. Consider a two-dimensional binary classification problem, in which all points inside the circle of radius 1 are labelled blue, while all points outside the same circle are labelled red.
   (a) Are these classes linearly separable on the Cartesian plane?
   (b) Are these classes linearly separable in some derived feature space: i.e., after transforming the feature vectors into some higher-dimensional space by applying some basis functions? Explain.
3. List all the hyperparameters in your implementation of the perceptron algorithm, and briefly summarize how you optimized them. *Be sure all these values are set to the best values you find when you submit.*

7 Ethics Questions

One of the earliest commercial uses of AI was in healthcare. The MYCIN expert system was used during the 1970s to recommend antibiotics and dosages to patients, adjusted based on their body weights. Today, far more sophisticated supervised learning systems are available to assist healthcare workers in medical decision making. These systems take into consideration many more patient features than simply body weight. Read these current articles about AI in healthcare:

- On the ethics of algorithmic decision-making in healthcare
- Medical Algorithms Have a Race Problem
- Machine Learning for pain relief?

Now answer the following questions:

1. Suppose that the results of numerous trials indicate that when a doctor disagrees with the machine, the machine is more likely to be correct. If a doctor disagrees with the machine in a situation that is time-sensitive (she does not have enough time to consult additional medical experts), how should she decide what course of treatment to pursue?
2. In some cases, doctors use information regarding a patient's race when diagnosing and treating them. For example, Black Americans have a higher incidence of hypertension than others. What are some of the ethical implications of considering or not considering race information in supervised learning for healthcare? How is it similar or different to doctors using race information in their patient care?

8 Grading

The Breast Cancer Diagnostic dataset is fairly easy to classify correctly. You will earn full points by achieving 95% accuracy on this dataset.

The Student dataset is trickier. You will earn full points by achieving 90% accuracy on this dataset. You can earn extra credit by achieving 95% accuracy or above on the Student dataset. If you achieve 95% accuracy, please report your findings on Piazza, so that other students can try to match your results.

Note: You can can complete this assignment using only the "raw" feature values (normalized, most likely). However, you may also wish to experiment with using "derived" feature values, by which we mean feature values transformed by basis functions. You may be able to increase your accuracy in this way.

9 Install and Handin Instructions

To install, run `cs1410_install SL` in `~/course/cs1410/`.

To handin, run `cs1410_handin SL` in `~/course/cs1410/SL/`, which should contain your code and your five saved data files.

In addition, please submit the written portion of the assignment via Gradescope.