Homework 2

COM SCI X 450.4 - Machine Learning

Thursday 2nd July, 2020

1 Introduction

This homework will be divided into three parts. First, you will implement, from scratch, three of the Machine Learning algorithms we discussed in class. The second part will ask you to use the algorithms you implemented to make predictions on two famous datasets. The last part will require comparing your results with the results from the models you implemented to the implementations from Scikit-Learn. Please time how long you took to finish this homework, as it will be a question at the end of the assignment.

2 Part I - Model Implementation (10 points + 14 bonus)

Important: For any part of the algorithms that you find too difficult to implement, feel free to explain, in you own words, how each step should work. This kind of Pseudo-algorithms will be accepted. This part does not apply to the bonus questions.

For this section, you will implement three famous supervised learning models. You will need to implement them in a Python file and you should follow good programming practices.

Coding Tips
To follow good coding practices, it is recommended that you create a separate class for each model. It is common that machine learning classes follow a default structure with the commands *fit* and *predict*, as shown in the code snippet below. It is recommended that you implement the models using vectorized approaches, but it is not mandatory.

```
class YourModelClass:
    def __init__(self, parameters):
        #initializes parameters

    def fit(self, X, y):
        # fits/trains your model

    def predict(self, X):
        # returns predictions
```

```
# when you want to use the model:
X, y = function_that_loads_data()

my_model = YourModelClass()

X_train, X_test, y_train, y_test = train_test_split(X, y, ..)

my_model.fit(X_train, y_train)
predictions = my_model.predict(X_test)
```

2.1 K-Nearest Neighbors

The first model you will be implementing is the K-Nearest Neighbors (KNN) model. As you have learned from the book and class discussion, KNN is an instance-based model that does not require training but mainly compares the new instance to all the existing instances. Your fit method might required you to think about this difference. Your model should be useful for both classification and regression models.

Your *__init__* method should accept the number of neighbors to be used, which distance function to use (you should implement at least two functions), and a parameter to distinguish if the model is a regression or classification model.

2.2 Linear Regression

You should implement linear regression using gradient descent.

Your *__init__* method should accept the learning rate. For bonus points (**extra 4 points if implemented for both linear and logistic regression**), you can add a parameter for *L1* and *L2* regularization and implement it accordingly.

2.3 Logistic Regression

You should implement logistic regression using gradient descent. You should only care for the binary case, where there is only a negative and a positive class. You can earn **5 bonus point** if you implement for multi-class problems (3 or more classes).

HINT: There is some overlap on how to implement linear and logistic regression. It is recommended that you try to reuse your code as much as possible.

Your *__init__* method should accept the learning rate. For bonus points (**extra 5 points if implemented for both linear and logistic regression**), you can add a parameter for *L1* and *L2* regularization and implement it accordingly.

3 Making Predictions and Evaluating Results (10 Points)

After you have implemented your models, we will be evaluating their results on two famous datasets, the Boston House-Prices and the Breast Cancer Wisconsin datasets. You can load

both using Scikit-learn through the provided links. You will need to split the results into train and test.

You will create a new Jupyter Notebook to run you analysis. You should be able to import your algorithms in the notebook. Write your answers to the questions in a separate document. You will submit it with your code and notebook.

Remember to preprocess the data with what you have learned in class and through the readings.

3.1 Boston House-Prices

Use your implementation of the Linear Regression and KNN algorithms. Train your models, make predictions and compare the results.

- Which one gets better results? If you were not able to implement one of them, which one you suspect should get better results and why?
- Which measures are you using to compare them?
- Which one is faster during training? And during inference time?

3.2 Breast Cancer Wisconsin

Use your implementation of the Logistic Regression and KNN algorithms. Train your models, make predictions and compare the results.

- Which one gets better results? If you were not able to implement one of them, which one you suspect should get better results and why?
- Which measures are you using to compare them?
- Which algorithm has more False Positives? And False Negatives? If you were not able to implement one of them, explain the concept of False Positives and False Negatives.
- If you were a doctor, which algorithm would you use for the prediction? If you were not able to implement one of them, explain how you would use a confusion matrix to answer this question.

4 Using Scikit-learn and comparing results (75 Points + 15 bonus)

Now, select the Linear Regression and]Logistic Regression models from sklearn. Train them using the same training set and make predictions using the same test set that you have used for your own models.

- Which one gets better results? Please specify per dataset.
- How could you modify the other methods to achieve similar results?

- Which one is faster during training? And during inference time?
- Using the methods from Scikit-Learn, play around with the parameters. What is the best result you can get? What parameter was the most important for each dataset?
- (Bonus) For Linear Regression, plot the coefficients after training models with L1 and L2 regularization. How do they differ?
- (Bonus) For Logistic Regression, plot the coefficients after training models with L1 and L2 regularization. How do they differ? Hint: You can get the coefficients of a trained model with the *model.coef_* method.

5 How long did it take? (5 Points)

Please add to your written document how many hours you took to finish this assignment.

6 Deliverable

You will submit your assignment before Sunday, July 19th, before 11:59pm. You should submit a zipped file with:

- The Python source files where you implemented your algorithms
- The Python Notebook with the code you wrote
- A separate PDF document with the answers to the questions.

Start early, especially if you are not familiar with Python! If you require help, please contact instructor. Office hours can be scheduled if necessary as long as the instructor is contacted with 48 hour notice.