Koç University
COMP341
Introduction to Artificial Intelligence
Assignment 5

Instructor: Barış Akgün
Due Date: December 21 2018, 23:59
Submission Through: Blackboard
Make sure you read and understand every part of this document

This programming assignment will test your knowledge and your implementation abilities of what you have learned in the machine learning part of the class. You are asked to complete a coding part and answer a few questions about the output. The coding part of this homework is custom made for the class. There are a lot of useful comments so be sure to investigate the files. The questions for the report part are given in this document.

This homework must be completed individually. Discussion about algorithms, algorithm properties, code structure, and Python is allowed but group work is not. Coming up with the same approach and talking about the ways of implementation leads to very similar code which is treated as plagiarism! Furthermore, do not discuss the answers directly as it will lead to similar sentences which is treated as plagiarism. If you are unsure, you should not discuss. Any academic dishonesty, will not be tolerated. **By submitting your assignment, you agree to abide by the Koç University codes of conduct.**

You may find yourself having trouble implementing the coding part. In this case, we are going to let you use someone else's code to answer the given questions, as long as you credit the person or the website you take the code from. If you chose this option, we are only going to grade your report.

Introduction

In this homework, you will implement a feature extraction method and two machine learning methods. The datasets are provided to you. You are asked to extract features from images, implement the kNN method for classification and implement linear regression. Data loading, regression data pre-processing, cross-validation to handle overfitting and performance visualization, all important parts of a machine learning pipeline, are handled for you. I recommend you go over the corresponding parts of the code.

As a result of the implementation part being relatively easy, since we are providing a lot of helper code, detailed explanations and help with installing extra modules will not be provided. The provided code has detailed comments and hints, it is up to you to go over them. Furthermore, learning to install needed modules for python will be a valuable experience.

In addition to implementing the aforementioned parts, you are asked to write a report about the output of your implementation. This homework will have a 50-50 balance of programming and report grade.

There will be no autograder provided to you for this HW. The autograder for this homework uses a reference implementation and compares the output of your ode with this implementation. Providing it would mean giving you the solutions. Instead, we are providing example outputs for you to match.

Grading

For this homework, the report and the programming part will have the same weight. You can still submit only the report. The report requires you to interpret the output of the coding part. We are providing example outputs, you can directly use them!

We are going to compare your code to each other and previous submissions of Koç students. If your code's similarity level is above a certain threshold, your code will be scrutinized. If we see any plagiarism, you will lose points in the best case and disciplinary action will be taken against you in the worst.

Part 1: Programming

Preliminaries

You are going to need the following python modules:

- Numpy (possible Numpy+Mkl)
- Matplotlib
- OpenCV
- Scikit-Learn

If you are using a Linux distribution, I would assume that you know what you are doing and that you probably do not need any help with installing these. (Hint: pip or apt-get)

If you are using windows, you can use pip. I suggest you use the libraries provided here with pip instead of their defaults: `https://www.lfd.uci.edu/~gohlke/pythonlibs/`.

If you are using Mac OS, you can also use pip with the default modules.

Hint for using pip: `python -m pip install <whl file OR package name>`

Datasets and the ML Problems

Classification

In the classification part of the homework, you are going to classify whether an image involves **wood** or **metal**. The images are provided under two folders. You are going to use the k-Nearest Neighbor and the Logistic Regression algorithms for this. You are going to only implement the kNN approach, logistic regression is provided to you.

WARNING: Using existing kNN implementations, for example the scikit-learn version, is prohibited and will not get any credit. You can however use these to verify your code.

In addition to implementing the kNN method, you are going to implement a feature extraction approach, involving histograms and image transformation. Both of these can be done readily with the aforementioned libraries. In addition, look up "image histogram" and "HSV color model" from the internet. Look at the code for specifics.

Regression

In the regression part of the homework, you are given three regression datasets. The inputs and targets are defined and extracted for you. You are going to use linear regression and ridge regression. You are going to only implement linear regression. Ridge regression is provided to you.

WARNING: Using existing linear regression implementations, for example the scikit-learn version, is prohibited and will not get any credit. You can however use these to verify your code. You are also allowed to use matrix operations provided by Numpy.

Implementation

There are 5 python files. You are going to complete the code in *data.py* and *learners.py* for this project. You would also want to play around with *main.py* for debugging. The entry point of the code is the *main.py*. You can directly call it as:

```
python main.py
```

If you are getting `ImportError: No module named ...` type errors, go back to Preliminaries. If not, you would get a `*** Method not implemented: ...` message.

At this point, what you need to do should be obvious. First read all the comments in *main.py*, then move on to the needed parts of the code. Specifically, you need to complete

- `extract` method of the `SaturationHistogramExtractor` class in file *data.py*
- `fit` and `predict` methods of the `knnClassifier` class in file *learners.py*
- `fit` and `predict` methods of the `LinearRegression` class in file *learners.py*

There are enough comments for you to go on. Do not forget that you have access to the slides as well. If there is something that you do not know, for example what a histogram is or how an image is represented, internet is your friend. Self-learning is part of this homework!

Outputs

The code that is provided to you prints to the standard output and saves performance figures as PNG files. We have provided example outputs for you under the *example-outputs* folder. If your machine is 64-bit, you should probably get the exact results (the plot colors and size might change but the numbers should be the same) since we are fixing the random seed. If you are not getting the exact same results, either your implementation is wrong or your version of the random number generator (RNG) is different than ours. Seek the instructor or the TAs if you think this is the problem.

In case you are sure your implementations are 100% correct (e.g. compared against a library implementation), take a look at your features! Finally, do not worry too much as there will be partial creadits.

Part 2: Report

This part includes answering the following questions based on your program's output on the given pacman tests. You are expected to answer the questions concisely. It is okay if you over-generalize, as long as your direction is clear and correct. Note that you do not need to write that much!
If you get different outputs then the provided ones, feel free to use them in your answers. Make sure you include them in your report!

Create a PDF file named *report.pdf* containing your answers for submission. **Write your name and your number on the report as well!**

Written Q1:
Why do you think that the comments in the code (in *data.py*, under `extract`) suggest you to normalize the histogram? Hint: Look at the input images.

Written Q2:
Look at the *knn.png*. What are the range of k values (horizontal axis) that correspond to overfitting and underfitting, if these are clearly observable? Which value is the best and why?

Written Q3:
Look at the *logreg.png*. What are the range of c values (horizontal axis) that correspond to overfitting and underfitting, if these are clearly observable? Which value is the best and why?

Written Q4:
Which method would you chose between kNN and logistic-regression based on the results? Why?

Written Q5:
Look at the *ridgereg_variation1.png*. What are the range of λ (horizontal axis) values that correspond to overfitting and underfitting, if these are clearly observable? Compare and contrast your linear regression and the ridge-regression, by looking at this graph and accuracy outputs. Make sure to comment on both the average error and its standard deviation. Do you have any other comments about applicability of these methods for this data set?

Written Q6:
Look at the *ridgereg_variation2.png*. What are the range of λ (horizontal axis) values that correspond to overfitting and underfitting, if these are clearly observable? Compare and contrast your linear regression and the ridge-regression, by looking at this graph and accuracy outputs. Make sure to comment on both the average error and its standard deviation. Do you have any other comments about applicability of these methods for this data set?

Written Q7:
Look at the *ridgereg_airfoil.png*. What are the range of λ (horizontal axis) values that correspond to overfitting, if these are clearly observable? Compare and contrast your linear regression and the ridge-regression, by looking at this graph and accuracy outputs. Make sure to comment on both the average error and its standard deviation. Do you have any other comments about applicability of these methods for this data set?

Submission

You are going to submit a compressed archive through the blackboard site. The file should extract to a folder with your student ID without the leading zeros. This folder should only contain *report.pdf*, *data.py* and *learners.py*. Other files will be deleted and/or overwritten. Do not submit any code if you only want us to grade your report.

Important: Download your submission to make sure it is not corrupted and it has your latest report/code. You are only going to be graded by your blackboard submission.

Submission Instructions

- You are going to submit a compressed archive through the blackboard site. The file can have *zip*, *rar*, *tar*, *rar*, *tar.gz* or *7z* format.
- This compressed file should extract to a folder with your student identification number with the two leading zeros removed which should have 5 digits. Multiple folders (apart from operating system ones such as MACOSX or DS Store) greatly slows us down and as such will result in penalties
- Code that does not run or that does not terminate will not receive any credits.
- Do not trust the way that your operating system extracts your file. They will mostly put the contents inside a folder with the same name as the compressed file. We are going to call a program (based on your file extension) from the command line. The safest way is to put everyting inside a folder with your ID, then compress the folder and give it your ID as its name.
- Once you are sure about your assignment and the compressed file, submit it through Blackboard.
- After you submit your code, download it and check if it is the one you intended to submit.
- **DO NOT SUBMIT CODE THAT DOES NOT TERMINATE OR THAT BLOWS UP THE MEMORY.**

Let us know if you need any help with setting up your compressed file. This is very important. We will put all of your compressed files into a folder and run multiple scripts to extract, clean up, grade and do plagiarism checking. If you do not follow the above instructions, then scripts might fail.