2021 Introduction to Machine Learning Program Assignment #3 - K-means Clustering & Support Vector Machine

TA: Yi-ann Chen viva23chen.am05@g2.nctu.edu.tw (mailto:viva23chen.am05@g2.nctu.edu.tw)

This programming assignment aims to help you understand *K-means Clustering* and *Support Vector Machine*.

Before we start

> Join the discord server for TA support (https://discord.com/invite/XJkvmNrcjp)
>
> - Ask questions on it, and we shall reply.
> - Try not to ask for obvious answers or bug fixes.
> - Memes and chit-chat welcome.

Objective

1. Data Input - 5%
    - K-means Clustering: Wheat Seeds Dataset (https://archive.ics.uci.edu/ml/datasets/seeds)
    - Support Vector Machine: Ionosphere Dataset (https://archive.ics.uci.edu/ml/datasets/Ionosphere)
2. Data Preprocessing - 10%
    - Transform data format and shape so your model can process them.
    - **Shuffle the data.**
    - Any data augmentation that can boost your final results.
3. K-means Clustering - 20%
    - You are asked to implement the K-means clustering algorithm

```
K-MEANS (P: a dataset of points, k: numbers of clusters)

    Randomly initial k centers C={c1, ..., ck}

    while stopping criterion has not been met

            1. assignment step:
                for i = 1, ..., n
                    find closest center cj to instance pi
                    assign instance pi to set Cj

            2. update step:
                for i = 1, ..., k
                    set ci to be the center of all points in set Ci
```

- Since the dataset has three types of target labels, execute your function with $k=3$
- In the end, calculate the labels of the instances in set Ci and assign Ci with the label that has the largest value count.
- **Do not pass the target label to your K-means function**
- In this part, you are **not allowed** to use any package-provided models

4. Support Vector Machine - 30%

- Train-Test-Split - 5%
  - Holdout validation (https://en.wikipedia.org/wiki/Cross-validation_(statistics)#Holdout_method) with the ratio of 7:3
- You need to use linear, polynomial, RBF kernels to construct three different models. - 15%
  - linear kernel: $\langle x,x' \rangle$
  - polynomial kernel: $(\gamma\langle x,x' \rangle+r)^d$
  - RBF kernel: $e^{-\gamma \| x-x' \|^2}$
- Parameter Search - 10%
  - For some kernels, there are several hyperparameters that need to be set.
    - polynomial kernel: degree d, gamma γ, coef0 r
    - RBF kernel: gamma γ
  - Use grid search to find the best hyperparameter pair for each model.
  - Make sure that you only use the training set for the parameter search.
  - Use K-fold cross-validation (https://en.wikipedia.org/wiki/Cross-validation_(statistics)#k-fold_cross-validation) with $K=5$ to check the performance of each hyperparameter pair.
- Construct the final models for each kernel with the best hyperparameter pair on the whole training set.

- In this part, you are **allowed** to use any package-provided models

5. Results - 10%
    - Obtain the performances of all experiment settings **in tables** by the following metrics:
        1. Confusion matrix
        2. Accuracy
        3. Sensitivity (Recall)
        4. Precision
6. Comparison & Conclusion - 5%
    - Also some feedback, anything you want to tell me.
7. Questions - 20%
    - K-means Clustering
        - Choose two features to execute your K-means function and draw a scatter plot with the computed centers and the predicted label for each instance. - 10%
    - Support Vector Machine
        - Show the average performance of K-fold cross-validation of parameter search **in tables** for each kernel. - 10%

Datasets

1. Wheat Seeds Dataset

- Data can be Downloaded here
    - https://archive.ics.uci.edu/ml/datasets/seeds (https://archive.ics.uci.edu/ml/datasets/seeds)
- Please **NOTE** that the last column is the label
- Data Set Information
    - The examined group comprised kernels belonging to three different varieties of wheat: Kama, Rosa, and Canadian, 70 elements each, randomly selected for the experiment. High-quality visualization of the internal kernel structure was detected using a soft X-ray technique. It is non-destructive and considerably cheaper than other more sophisticated imaging techniques like scanning microscopy or laser technology. The images were recorded on 13x18 cm X-ray KODAK plates. Studies were conducted using combine harvested wheat grain originating from experimental fields, explored at the Institute of Agrophysics of the Polish Academy of Sciences in Lublin.

- Attribute Information
    1. area A
    2. perimeter P
    3. compactness $C = 4\pi AP^2$
    4. length of kernel
    5. width of kernel
    6. asymmetry coefficient
    7. length of kernel groove

2. Ionosphere Dataset

- Data can be Downloaded here
    - https://archive.ics.uci.edu/ml/datasets/Ionosphere (https://archive.ics.uci.edu/ml/datasets/Ionosphere)
- Please **NOTE** that the last column is the label (good = g, bad = b)
- Data Set Information
    - This radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. See the paper for more details. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere.
    - Received signals were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Instances in this dataset are described by 2 attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal.
- Attribute Information
    1. All 34 are continuous
    2. The 35th attribute is either "good" or "bad" according to the definition summarized above. This is a binary classification task.

Submission & Scoring Policy

- Please submit a **zip** file, which contains the following, to the new E3 system.

    1. Report

        - Explanation of how your code works.
        - All the content mentioned above.
        - Your name and student ID at the very beginning - 10%

        - Accept formats: **HTML**
            - From markdowns (https://hackmd.io/?nav=overview) or jupyter notebooks.

    2. Source codes

        - Accept languages: **python3**
        - Accept formats: **.ipynb** (https://jupyter.org)

- Your score will be determined mainly by the submitted report.
    - If there's any problem with your code, TA might ask you (through email) to demo it. Otherwise, no demo is needed.
- Scores will be adjusted at the end of the semester for them to fit the school regulations.

- **Plagiarizing is not allowed.**
    - Plagiarizing is checked by MOSS (https://theory.stanford.edu/~aiken/moss/) and manually afterward.
    - You will get **ZERO** on that homework if you get caught the first time.
    - The second time, you'll **FAIL** this class.
    - 抄襲第一次作業零分，第二次當掉

Tools that might be useful

- Jupyter Lab (https://jupyter.org) - Better data science experience
- Numpy (https://numpy.org) - Math thingy
- matplotlib (https://matplotlib.org/stable/tutorials/introductory/pyplot.html) - Plot thingy
- pandas (https://pandas.pydata.org) - Data thingy
- scipy (https://scipy.org/) - Science thingy
- scikit-learn (https://scikit-learn.org/stable/) - Machine Learning and stuff