[Project Code: HDSVM]
Heart Disease Prediction using Support Vector Machines

Project Duration : 26-Feb-2023 ~~ 18-Mar-2023
Submission Information : (via) CSE-Moodle

Objective:

A healthcare company wants to build a machine learning model to predict the heart disease of a person. The machine learning model will take the various information of the patients and predict whether he/she is suffering from a heart disease. In particular, they want to use a kernel method as it can be used for non-linear classification problems and has a theoretical guarantee. Your task is to help the company to build the classification model.

As per the problem requirement, the project manager has a particular solution in his mind. Thus, he has asked you to pose the training problem for the classifier as follows:

$$\min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=0}^{n-1} l(-y_i(\mathbf{w}^T\phi(\mathbf{x}_i) - b))$$

where $l(x) = \log(1 + e^x)$ and

$$\{(\mathbf{x}_0, y_0), (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n-1}, y_{n-1})\}$$

is the training set. Since it is a binary classification problem, the labels y_i s are either +1 or -1.

Note that you can check the sign of $\mathbf{w}^T\phi(\mathbf{x}_i) + b$ to determine the predicted class. If it is positive, then the predicted class of 1 and -1 otherwise. To help you to proceed further, the project manager asked you to re-write the problem as:

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}, \xi\in\mathbb{R}^n} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=0}^{n-1} l(\xi_i) \\ \text{subject to} \quad & \xi_i = -y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \end{aligned}$$

Your tasks are the followings:

1. First, you will find out the dual optimization problem of the above by the following steps:
    a. Formulate the Lagrangian of the above optimization problem. Also find out the corresponding KKT conditions.
    b. Then, formulate the dual optimization problem.
2. Secondly, you will write a class to implement the classifier with the following methods:
    a. Constructor: It will take the necessary hyper-parameters like C, the kernel type and the hyper-parameters related to the kernels as input and initialize the module. You have to only consider three types of kernels: linear, polynomial and RBF.
    b. Train: It will take the train data as input and learn the parameters. Note that the training requires solving the dual optimization problem. To solve the dual optimization problem you can use any library function like:

i. Scipy.optimize.minimize (python package)
ii. CVXOPT (python package)
iii. CVX (matlab package)

c. Predict: It will take the test data as input and return the predictions on the data.

3. Finally, you should generate results on the given data and compare its results with the sklearn module sklearn.svm.**LinearSVC.**

Note: *The program can be written in C++ / Java / Matlab / Python programming language from scratch. No machine learning /data science /statistics package / library should be used for model creation.*

Relevant information:

Dataset Filename: `heart.csv`
Number of Classes: `2`
Data Description:
Number of Instances: 1025
Number of Attributes: 13 (all numeric)
Detailed info: https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

Tasks to be done:

1. **Train-test split:** The dataset is not divided into train and test sets. First randomly split the data into train-test split with 80-20 ratio. You will be using only the train split for your training. The test split will be used only for the final evaluation. Even for the hyper-parameter tuning, you cannot use the test split.

2. **Data pre-processing**: Normalize each feature of the dataset to have zero mean and unit variance. Note that while normalizing the features, their mean and variance should be computed over the train split only. Once, the mean and variance is computed using only the train split, you normalize the test split using the mean and variance computed over the train split.

3. **Implementation of the model:**
   a. Implement the classifier as stated in the Objective Statement.
   b. Train the model using the train split of the dataset. Note that the training also involves the hyper-parameter tuning. Thus, for hyper-parameter tuning, you have to either split the train split into train and validation again or use the cross-validation on the train split. Whichever method you follow for the hyper-parameter tuning, clearly mention that in your report.
   c. Evaluate your trained model (with the best hyper-parameters) on the test split. And compare the results with the sklearn module sklearn.svm.**LinearSVC.**

4. **Outcomes and Reporting:** Prepare and submit a report with the following –
   a. The mathematical derivation of your solution.
   b. Results of the hyper-parameters tuning.
   c. Results on the test split of dataset.
   d. You need to calculate precision, recall, f1-score and accuracy for all the experiments.

Submission Details: (to be submitted under the specified entry in CSE-Moodle)

1. ZIPPED Code Distribution in CSE-Moodle
2. A brief (2-3 page) report/manual of your work
   (with your hyperparameter tuning results also presented in that report)

Submission Guidelines:

1. You may use one of the following languages: C/C++/Java/Python/Matlab.
2. Your Programs should run on a Linux Environment.
3. You are **not** allowed to use any library apart from these (Also explore all these libraries if doing in Python, or equivalent of these):

```
import numpy # linear algebra
import csv # data processing, CSV file I/O
import pandas # data processing, CSV file I/O
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
import sklearn.svm.LinearSVC
import operator
from math import log
from collections import Counter
```

   Your program should be standalone and should **not** use any *special purpose* library of Machine Learning for the classifier (apart from the library required to solve the dual optimization problem). Numpy and Pandas may be used. And, you can use libraries for other purposes, such as generation and formatting of data.
4. You should submit the program file and README file and **not** the output/input file.
5. You should name your file as <GroupNo_ProjectCode.extension>.
   (e.g., *Group99_HDSVM.zip* for code-distribution and *Group99_HDSVM.pdf* for report)
6. The submitted program file *should* have the following header comments:
   # Group Number
   # Roll Numbers : Names of members (listed line wise)
   # Project Number
   # Project Title
7. Submit through CSE-MOODLE only.
   Link to our Course page: https://moodlecse.iitkgp.ac.in/moodle/course/view.php?id=508

You should not use any code available on the Web. Submissions found to be plagiarized or having used ML libraries (except for parts where specifically allowed) will be awarded zero marks.

For any questions about the assignment, contact the following TA:
Suvadeep Hajra (Email: suvadeep.hajra@gmail.com)