Machine Learning HW1

曾以諾  
Institute of Data Science  
National Cheng Kung UniversityTainan, ROC  
re6111032@gs.ncku.edu.tw

*Abstract*—This is the homework 1 of Machine Learning. I apply linear classifier, voted perceptron, and SVM on two datasets, “data.csv” and “crx.csv”.

Keywords—linear classifier, voted perceptron, SVM

# Requirements

There are two datasets, crx.csv and data.csv:

1. The labels column in crx and data are "label" and " Diagnosis".
2. Please implement the Linear Classifier from scratch with the update rule in the slide. It means you cannot adopt any existing package like sklearn in this assignment.
3. When the "J=WX+b" could be represented as the matrix form for the linear classifier, please find the solution by solving this equation using least-squared manner. Also, please implement it and make a comparison between this method and the previous one implemented in 2.
4. Please implement the voted perception and make the performance comparison between methods.
5. With minimizing the ||w||^2, it should drive the marginal to be maximized as well. Please implement the linear classifier with the minimum ||w||^2 property and verify whether the margin of this version is larger than that of the conventional linear classifier or not.
6. Based on 5, please add the slack variable term in the linear classifier and find the most effective weighting value C.
7. Use any existing SVM package (libSVM, sklearn, or Matlab SVM), please make a performance comparison between the built-in SVM and your implementation.

All the source code should be uploaded to Github with public access. The performance comparison should be organized into a formal report with the standard template indicated in Moodle.

# Results of implementation on data.csv

## Linear Classifier

The accuracy is 92%.

## Linear Classifier with Matrix Form

The accuracy is 96%.

## Voted Perceptron

The accuracy is 96%.

## SVM without Slack Variables

The accuracy is 88%.

## SVM with Slack Variables

The accuracy is 38%. The weighting value C is 0.1.

## Comparison to sklearn.svm

The accuracy of sklearn.svm.SVC() is 94%

# Results of implementation on crx.csv

## Linear Classifier

The accuracy is 76%.

## Linear Classifier with Matrix Form

The accuracy is 53%.

## Voted Perceptron

The accuracy is 69%.

## SVM without Slack Variables

The accuracy is 67%.

## SVM with Slack Variables

The accuracy is 45%. The weighting value C is 0.1.

## Comparison to sklearn.svm

The accuracy of sklearn.svm.SVC() is 72%

Identify applicable funding agency here. If none, delete this text box.

# Tables

1. Comparison

| Method | Accuracy | |
| --- | --- | --- |
| data.csv | crx.csv |
| Linear Classifier | 0.92 | 0.76 |
| Linear Classifier with Matrix Form | 0.96 | 0.53 |
| Voted Perceptron | 0.96 | 0.69 |
| SVM without slack variables | 0.88 | 0.67 |
| SVM with slack variables | 0.38 | 0.45 |
| sklearn.svm.SVC | 0.94 | 0.72 |

# References

1. Ford Combs, “SVM from scratch: step by step in Python”,

article site: <https://fordcombs.medium.com/svm-from-scratch-step-by-step-in-python-f1e2d5b9c5be>,

Github: <https://github.com/frodoCombs/SVM_tutorials>.

1. Sidharth, “Implementing Support Vector Machine Algorithm from Scratch in Python”, <https://www.pycodemates.com/2022/10/implementing-SVM-from-scratch-in-python.html>.
2. vamc-stash, “Support-Vector-Machines”,

<https://github.com/vamc-stash/Support-Vector-Machines>.

1. vpcsilva, “voted-perceptron”,

<https://github.com/vpcsilva/voted-perceptron>.