ENGR-UH 4560
Selected Topics in Information and Computational Systems

Machine Learning

Project 02 - SVM for classification

Introduction

Support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a **non-probabilistic binary linear classifier** (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped **so that the examples of the separate categories are divided by a clear gap that is as wide as possible**. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. This gap is also called maximum margin and the SVM classifier is called **maximum margin classifier**.

Requirements

- Test your SVM module on example dataset (*ex6data1, ex6data2*).
    - Generate boundary line of different classes via your SVM module with a linear kernel (*ex6data1*).
    - Plot the result boundary line.
    - Generate boundary line of different classes via the SVM module with a Gaussian kernel (*ex6data2*).
    - Plot the result boundary line.
- Implement the cancer prediction model based on Scikit Learn Dataset.
- Set up the DataFrame.
    - Adding the target data to the DataFrame.
- Training and prediction.
    - Train Test Split.
    - Train the Support Vector Classifier.
    - Predictions and Evaluations: list the score of your model under evaluation matrix below
- Replace SVM module from the three-party library with your own code (**optional**).

Deliverables

A zip file containing the following:

1. a working project (source code, makefiles if needed, etc)
2. a report for the detailed description of the project
    a. explain the main aspects of your code
    b. how to run your project
    c. plots and diagrams

Before submitting your project, please make sure to test your program on the given dataset.

Notes

*Functions from Standard Python libraries (e.g. sklearn.svm) are **not recommended** in the task, you are encouraged to write your own module with similar performance.*
*You may discuss the general concepts in this project with other students, but you must implement the program on your own. **No sharing of code or report is allowed.** Violation of this policy can result in a grade penalty.*

Late submission is acceptable with the following penalty policy:

- ***10 points deduction for every day after the deadline***