TU Delft
Faculty of Electrical Engineering, Mathematics, and Computer Science
Circuits and Systems Group

ET4350 Applied Convex Optimization

ASSIGNMENT

Linear Support Vector Machines

1 Context

In machine learning, support vector machines (SVMs) 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 deterministic binary linear classifier.

This exercise consists of two parts: (a) formulate the linear SVM problem as a suitable convex optimization problem; and (b) implement the binary classifier. In a group of 2 students, make a short report (4-5 pages; pdf file) containing the required Matlab scripts, plots, and answers. Also, prepare a short presentation to explain your results and defend your choices.

Dataset explanation

Given a training dataset of N points of the form

$$\{\mathbf{x}_1, y_1\}, \{\mathbf{x}_2, y_2\}, \cdots, \{\mathbf{x}_N, y_N\}$$

where the label y_n is either 1 or -1 indicating the class to which the data point $\mathbf{x}_n \in \mathbb{R}^2$ belongs to. We want to find a hyperplane (in this case, a line) that separates the group of points $\{\mathbf{x}_n\}_{n=1}^N$ into two classes so that this classifier can be used to label the test data. The dataset `linear_svm.mat` includes observations for 100 training samples that are labelled as well as 900 samples that can be used to test your classifier.

2 Assignment

You will have to answer the following questions:

1. (2 pts) Formulate the linear SVM problem as an optimization problem. Suggest a suitable convex approximation (i.e., derive a convex relaxed problem) if the true problem is not convex. Motivate the proposed formulation as well as the relaxation.

2. (2 pts) Implement the proposed convex optimization problem in your favorite off-the-shelf solver (e.g., `CVX`, `SeDuMi`, or `YALMIP`). How does this ready-made software solve your problem? Comment on the number of iterations, CPU time, and algorithm the ready-made solver uses.

3. (5 pts) Implement a low-complexity algorithm (e.g., projected (sub)gradient descent for the above problem, or provide a first-order algorithm to solve the primal and dual problems). Compare the obtained results with the solutions from the off-the-shelf solver. Comment on the number of iterations, CPU time, and convergence of your low-complexity algorithm.

4. (1 pt) Make a short presentation explaining your results clearly in 5 minutes.

3 Consultant

dr. S.P. (Sundeep Prabhakar) Chepuri
Tel: +31-15-27 81797
Email: s.p.chepuri@tudelft.nl
Room: HB 17.070
Office hours: By prior appointment only.