Optimization

4th Problem Set

Kalamarakis Theodoros: 2018030022

8/1/2023

Description of the problem

The purpose of this exercise is to solve a binary classification problem using the SVM classifier. The goal is to determine the boundary that makes the best possible separation of the two classes. The way SVM works is by trying to set a boundary that maximizes the margin between, the boundary itself, and its nearest data point from each class.

In this particular exercise we are studying the classification of linearly separable data, which means that the boundary can be a hyreplane ($\mathbf{a}^T\mathbf{x} - b = 0$ for $\mathbf{a}, \mathbf{x} \in \mathbb{R}^n$). Thus, the distance from the boundary to its nearest data point ($\mathbf{x}_i$) is :

$$\min_{\mathbf{x}=1,\dots n} \frac{|\mathbf{a}^T\mathbf{x}_i - b|}{\|\mathbf{a}\|_2}$$

and therefore the SVM problem can be interpreted as a (non-convex) optimization problem such as:

$$\begin{aligned} &\max_{\mathbf{a},b} \left\{ \min_{\mathbf{x}=1,\dots n} \frac{|\mathbf{a}^T\mathbf{x}_i - b|}{\|\mathbf{a}\|_2} \right\} \\ &\text{subject to} \quad y_i(\mathbf{a}^T\mathbf{x}_i - b) \geq 0 \end{aligned}$$

Where y_i is the label of the data point $\mathbf{x}_i$ and gets the values $+1$ and -1 corresponding to each one of the two classes.

By performing a slight reformulation and by setting $\mathbf{w} = (b \ , \ \mathbf{a})^T$ and $\bar{\mathbf{x}} = (1 \ , \ \mathbf{x})^T$, we can convert it into

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 \\ &\text{subject to} \quad 1 - y_i\mathbf{w}^T\bar{\mathbf{x}}_i \leq 0 \quad i = 1, 2, \dots n \end{aligned} \tag{1}$$

Which is now a convex optimization problem!

Solving the problem

Barrier function

We are already familiar with solving an unconstrained convex optimization problem using the Newton's algorithm. However in this case we need to include the inequality constraints into the solution. The idea is to adjust the cost function ($f_0(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2$) in a way, that the fuction value of the feasible points remains the same, whereas the function value of the infeasible points is increased as high as possible. In other words, we are trying to set a barrier so that the minimization algorithm (Newton) shall minimize only within the feasible region, and be prevented from considering an infeasible point (outside of the barrier) as minimum.

For this purpose, we develope the, so called, barrier function. A function whose value is close to infinite for infeasible points and close to zero for feasible points. It is also need to be convex and differentiable in order to maintain the problem's convexity. An example of a barrier function we are working with, in this exercise is:

$$\phi(\mathbf{w}) := -\sum_{i=1}^{N} \log(-f_i(\mathbf{w}))$$

where N is the number of inequality constraints (number of data points $\mathbf{x}_i$) and $f_i(\mathbf{w}) = 1 - y_i \mathbf{w}^T \mathbf{x}_i$

After having defined the barrier function, our new optimization problem is the following:

$$\underset{\mathbf{w}}{\text{minimize}} \left\{ f_0(\mathbf{w}) + \frac{1}{t}\phi(\mathbf{w}) \right\}$$

or equivalently

$$\underset{\mathbf{w}}{\text{minimize}} \ \{ g(\mathbf{w}) := t f_0(\mathbf{w}) + \phi(\mathbf{w}) \} \tag{2}$$

where $t > 0$

Suppose that $\mathbf{w}_*(t)$ is the optimal point calculated for the problem (2) and p_* is the actual optimal value for the initial problem (1). It is proven in the theory lectures that

$$f_0(\mathbf{w}_*(t)) - p_* \leq \frac{N}{t}$$

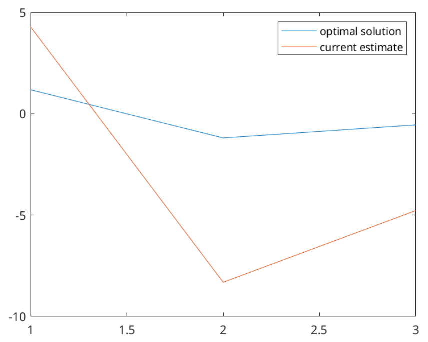
Barrier method

The last inequality shows that, using the barrier function we coclude to a solution which is, at most, $\frac{N}{t}$ suboptimal. This means that as $t \to \infty$, $\mathbf{w}_*(t)$ approaches the optimal solution. However setting an initial very large value for t and go on to solve the problem (2) does not work in all cases. Barrier method is solving a sequence of problems (2) for differents t and starting points (starting with a relatively small t). Each time, t is increased by a factor $\mu > 1$ ($t^{n+1} = \mu t^n$) and the starting point (for Newton's algorithm) is the last point found by the previous iteration. The algorithm terminates when the suboptimality of the calculated solution lies within a specific tolerance.

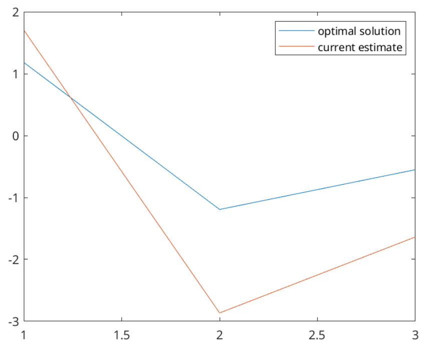
There is a trade-off between choosing a large or a small value of μ. In the large value case, we observe few barrier method iterations and many Newton's algorithm iteration, whereas in the small value case it happens the exact opposite.

Implementing in Matlab

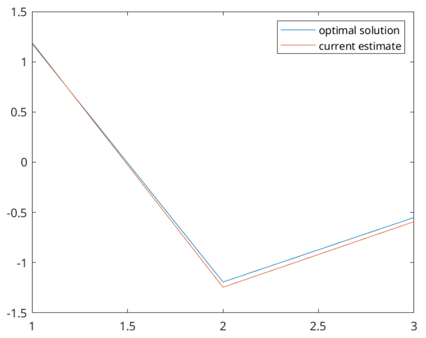
We are asked to implement and solve the SVM problem in matlab using a set of 1000 linearly separable two-dimensional data points. At first we start by selecting randomly $\mathbf{a}$ and b and generating supervised data points, separated by the hyreplane $\mathbf{a}^T\mathbf{x} = b$. Afterwards, we solve the SVM problem using cvx, the result of which, we consider as the optimal point. Then we proceed to the barrier method. The selected parameters are $t^{(0)} = 10$ and $\mu = 10$ and the starting point is the $\mathbf{w} = (b, \mathbf{a})^T$ by the initial (randomly chosen) $\mathbf{a}$ and b. In each barrier method iteration the optimal point is calculated, using Newton's algorithm with backtracking line search. The tolerance for the final result is $\epsilon = 10^{-5}$. The following plots are showing the progress of the estimated solution in each barrier method iteration, with respect to optimal solution computed via cvx



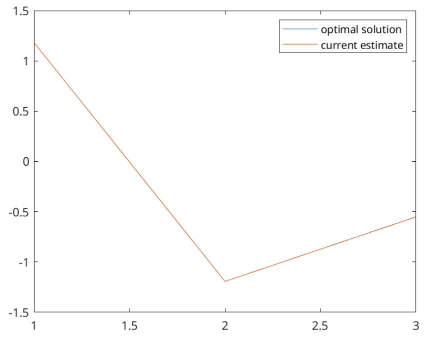
(a) Iteration 1



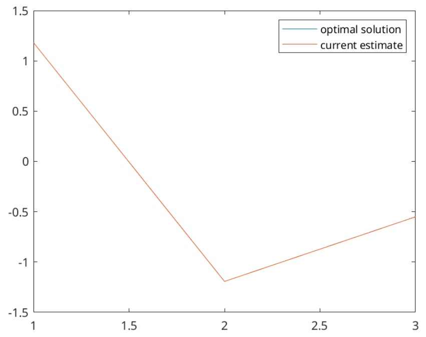
(b) Iteration 2



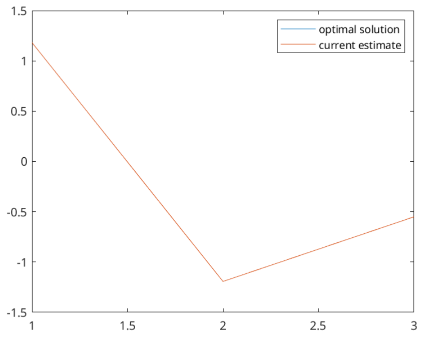
(c) Iteration 3



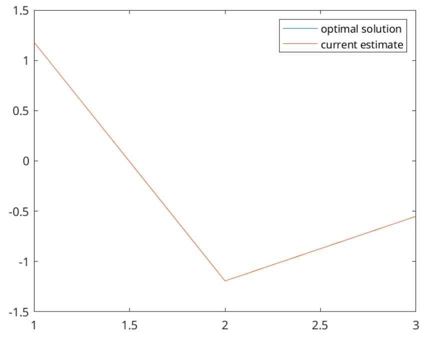
(d) Iteration 4



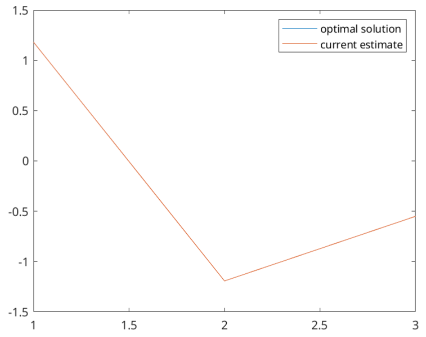
(e) Iteration 5



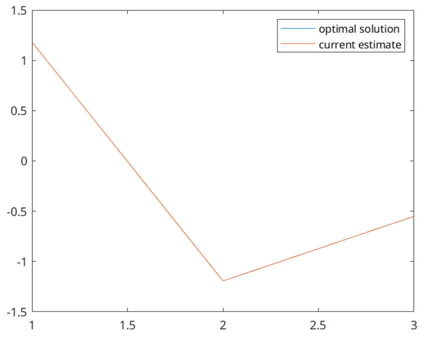
(f) Iteration 6



(g) Iteration 7



(h) Iteration 8



(i) Iteration 9

Figure 1: Eestimated solution (blue line) in each barrier method iteration, with respect to optimal solution computed via cvx (red line)

.

We observe how the estimated solution approaches the optimal one as the algorithm proceeds in each iteration, until they finally look like two coincident lines.

Finally the last plot shows the data points separated by the initial random hyperplane boundary, alogside with the optimal one that was computed by the SVM

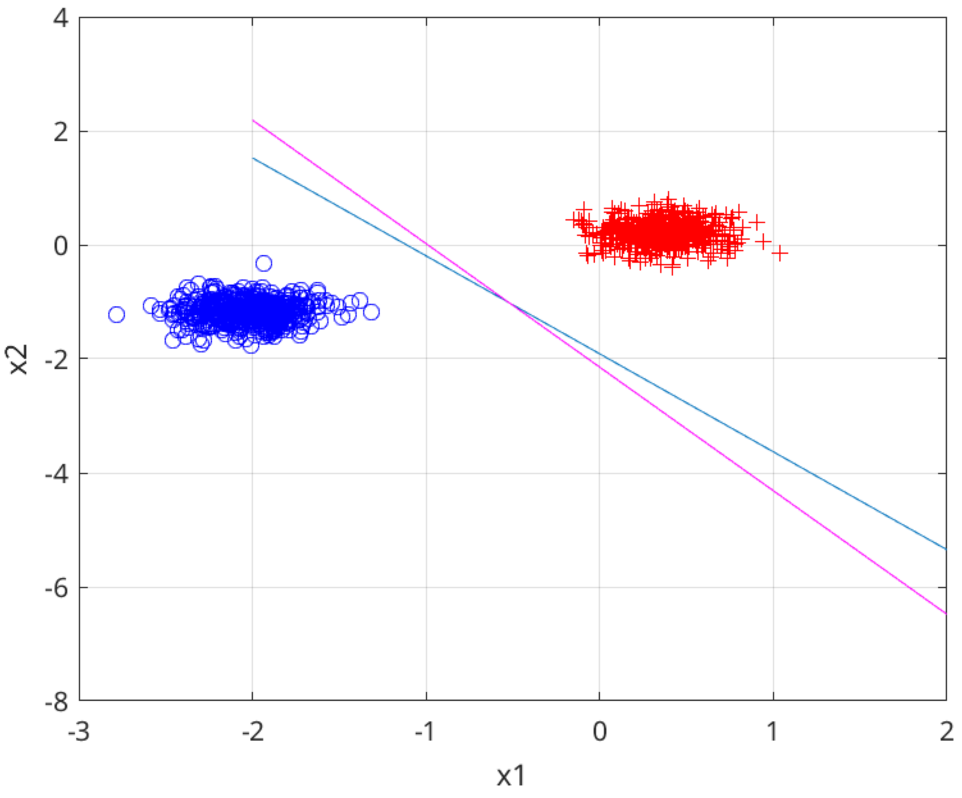


Figure 2: Separation of the data points by the two different boundaries. Initial random one in blue and the optimal one computed by the SVM in pink.

.