Fourth Problem Set for the Optimization Course

Leonidas Bakopoulos AM 2018030036

January 19, 2023

Introduction

The purpose of the final problem set was for the student to familiarize with the concept of Support Vector Machines (SVMs), in order to solve simple binary classification problems. In this pset, separable data was assumed, in order to avoid using the kernel trick procedure. It is worth mentioning, that in this code, an interior-point algorithm was used.

Code

Inside the given code,the basic structure of Newton's algorithm was implemented in order to solve the classification problem. But first, 2-class data were created and checked to see whether they were separable or not. In the first figure, we can see with the blue line, the first random SVM prediction/initialization and with the magenta line, the prediction for the maximized margin(distance between two support vectors). After making sure that the data was indeed linearly separable, Newton's algorithm was applied in order to calculate the value of $f_0 = \frac{1}{2}||w||^2$. In order for the code to be completed we were first required to create four functions, whose incorporation inside the given code will be explained after the the creation of the functions . The first one being : flag = point is feasible(w init, X augm, y), which returns 1 if w init exists in dom ϕ and 0 otherwise.

0.1 point is feasible function

For this function we had to figure out a way to satisfy the given constraint of w init existing in the domain of ϕ. After carefully reading the given theoretical notes, we designed the required function by exploiting the information that $f_i(w) = 1 - y_i w^T x_i, i > 0$. If the returning value of f_i was positive then the flag was set to zero, otherwise set to one.

0.2 gradient SVM barrier function

For this,second, function we had to compute the gradient of the cost function g_k at the given point $w(:, inner_iter)$. The gradient of $g_k(w)$ was calculated in chapter 6 of the notes. Its use was to minimize the cost functions by using the Newton Method for an increasing sequence of coefficients. With the given gradient of g we created the required function.

0.3 Hess SVM barrier function

The purpose of this function is to compute the Hessian (second gradient) of the cost function gk at the previously mentioned point w. Using the already calculated Hessian of g, we created the required function.

0.4 barrier SVM cost function

Last but not least, we computed the value of g_k at the new w point (w_new) . In order to do that, we used the equation containing the cost functions we wish to minimize : $g_k(w) := t_k f_0(w) + \phi(w)$. In order to find g_k value for the new w, we calculated $f_0 = \frac{1}{2}||w||^2$. and used the given equation for calculating $\phi(w)$.

Results

With the given code, and the previously mentioned additions, we were able to solve a binaryclassification problem using SVM,as mentioned before. As it is shown in the figures above (figure one and two), using Newton's step, the SVM algorithm approximates the optimal solution that derived from the CVX, and represents the border that maximizes the minimum distance between the two classes.

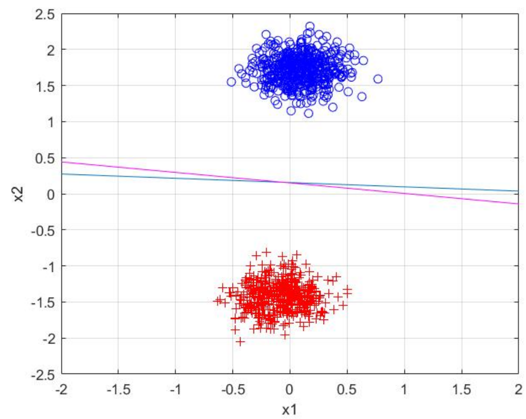


Figure 1:

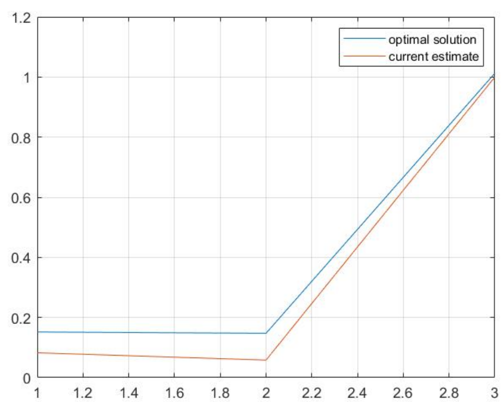
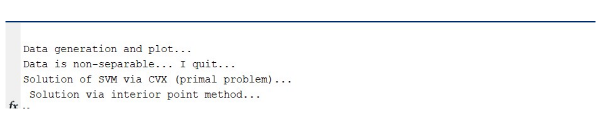


Figure 2:

But this SVM is not the best. As it was mentioned earlier, in this SVM is not implemented the kernel trick in order to "project" the data in higher dimension hyperplanes in order to be able to classify them even when they are not linear separable. As a conclusion to this problem set, I would



```
Data generation and plot...
Data is non-separable... I quit...
Solution of SVM via CVX (primal problem)...
 Solution via interior point method...
```

Figure 3: Matlab console where data are not linearly separable

like to highlight the value of that exercise, because SVMs were handled as an optimization problem and not as an oracle as they were in other classes (e.g. pattern recognition)