ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
TECHNICAL UNIVERSITY OF CRETE

Course "Optimization" Report

Exercise 4

Georgios Frangias 2018030086
gfrangias@tuc.gr

Course Professor:
Athanasios Liavas

Technical University of Crete
School of Electrical & Computer Engineering

January 2023

7 Exercise

1. `flag = point_is_feasible(w_init, X_augm, y)`

This function should return true when `w_init` $\in \mathbf{dom}\phi$. This means that it will return true only if

$$-f_i(\mathbf{w}) > 0 \Rightarrow y_i \mathbf{w}^T \overline{\mathbf{x}}_i > 1$$

2. `grad = gradient_SVM_barrier(w(:,inner_iter), X_augm, y, t)`

The gradient of the cost function g_k method at point $\mathbf{w}$ is

$$\nabla g_k(\mathbf{w}) = t_k \nabla f_0(\mathbf{w}) + \nabla \phi(\mathbf{w}) = t_k \mathbf{w} + \sum_{i=1}^{n} \frac{y_i}{1 - y_i \overline{\mathbf{x}}_i^T \mathbf{w}} \overline{\mathbf{x}}_i$$

3. `Hess = Hess_SVM_barrier(w(:,inner_iter), X_augm, y, t)`

The Hessian of the cost function g_k method at point $\mathbf{w}$ is

$$\nabla^2 g_k(\mathbf{w}) = t_k \nabla^2 f_0(\mathbf{w}) + \nabla^2 \phi(\mathbf{w}) = t_k \mathbf{I} + \sum_{i=1}^{n} \frac{1}{\left(y_i \overline{\mathbf{x}}_i^T \mathbf{w} - 1\right)^2} \overline{\mathbf{x}}_i \overline{\mathbf{x}}_i^T$$

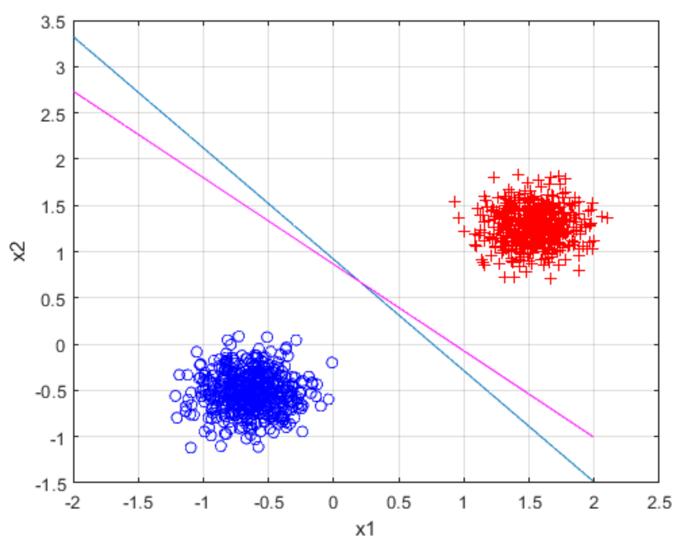
4. `barrier_SVM_cost_function(w_new, X_augm, y, t)`

The value of the cost function g_k method at point $\mathbf{w}$ is

$$g_k(\mathbf{w}) := t_k f_0(\mathbf{w}) + \phi(\mathbf{w}) = \frac{1}{2} t_k \|\mathbf{w}\|_2^2 - \sum_{i=1}^{n} \log\left(y_i \mathbf{w}^T \overline{\mathbf{x}}_i - 1\right)$$

Comments

After implementing the aforementioned functions, the main MATLAB program produced estimations of the SVM using the barrier method. At first, the estimation has a large error comparing it tot the `CVX` solution, but gradually after some iterations it collides with it. Below you can see an example set that the program produced



You can watch the video of the multiple iterations HERE