Homework 3

Abderrahim AIT AZZI

abderrahim.ait_azzi@ens-paris-saclay.fr

1 Interior Points Method

The interior point algorithm uses the combination of self-concordant barrier and Newton method to efficiently solve many convex problems. In this homework, we focus on the general quadratic problem:

$$\begin{aligned} min_x \phi(x) &= \frac{1}{2} x^T Q x + p^T x \\ sc \quad Ax &\leq b \end{aligned}$$

where Q is a symmetric semi-definite matrix and $x \in R^d$. The goal of the interior point algorithm is to transform the constrained problem into

$$\phi_t(x) = t(\frac{1}{2} x^T Q x + p^T x) + \mathcal{B}(b - Ax)$$

where $\mathcal{B}(b - Ax)$ is a self-concordant barrier for the set $Ax \leq b$ and t the parameter of the barrier. In this homework, we will use the logarithmic barrier

$$\mathcal{B}(x) = -\sum_i log(x_i)$$

The expression of the constrained problem become then:

$$\phi_t(x) = t(\frac{1}{2} x^T Q x + p^T x) - \sum_i log(b_i - a_i^T x)$$

Where a_i represent the i^{th}column of the matrix A.
Question:

- Compute $\nabla \phi_t(x)$: $\nabla \phi_t(x) = t(Q^T x + p) - \sum_i^m \frac{a_i}{b_i - a_i^T x}$
- Compute $\nabla^2 \phi_t(x)$

$$\nabla^2 \phi_t(x) = tQ + \sum_i^m \frac{a_i a_i^T}{(b_i - a_i^T x)^2}$$

- We have implemented the functions phi(x,t,Q,p,A,b), grad(x,t,Q,p,A,b) and hess(x,t,Q,p,A,b) which return respectively the function value, gradient and hessian of $\phi_t(x)$ at point x. **See the enclosed code.**

2 Newton Method

For minimizing the function $\phi(x)$ we will use the Newton method. The Newton decrements:

$$\lambda^2(x) = (\nabla\phi_t(x))^T(\nabla^2\phi_t(x))^{-1}\nabla\phi_t(x)$$

- In this question, we have implemented the function **[xnew,gap]=dampedNewtonStep(x,f,g,h)** which compute the damped Newton step at point x. in each step the updated x has the form:

  $$x_{new} = x - \frac{1}{1+\lambda(x)}(\nabla^2\phi_t(x))^{-1}\nabla\phi_t(x)$$

  The gap is computed such that:

  $$gap = \frac{\lambda^2(x)}{2}$$

- In this question, we have implemented the function **[xstar,xhist]=dampedNewton(x0,f,g,h,tol)** which minimizes the function f starting at x0 using the damped Newton algorithm. the steps of the algorithm are:

  - initialize x=x0 and xhist=x0.
  - use the function [x,gap]=dampedNewtonStep(x0,f,g,h) to compute the initial gap
  - while $gap \leq tol$:
    - $[xnew, gap] = dampedNewtonStep(x0, f, g, h)$
    - updated x to xnew, and than set xhist to [xhist x]

  We can also add a maximum number of iteration to ensure the convergence of the algorithm.

- In this question we implemented the function $[xstar, xhist] = newtonLS(x0, f, g, h, tol)$, The one thing that change if we compare with the dampedNewton function is the size of he iteration step. We should in this case use backtracking line-search with parameters $0 < \alpha \leq 1/2, 0 < \beta < 1$. At each iteration, we start with s = 1 and while:

  $$f(x + sv) > f(x) + \alpha s\nabla f(x)^T v$$

  we shrink $s = \beta s$, else we perform the Newton update: here: $v = -(\nabla^2\phi_t(x))^{-1}\nabla\phi_t(x)$

- **Plot and Comparison of the results:**

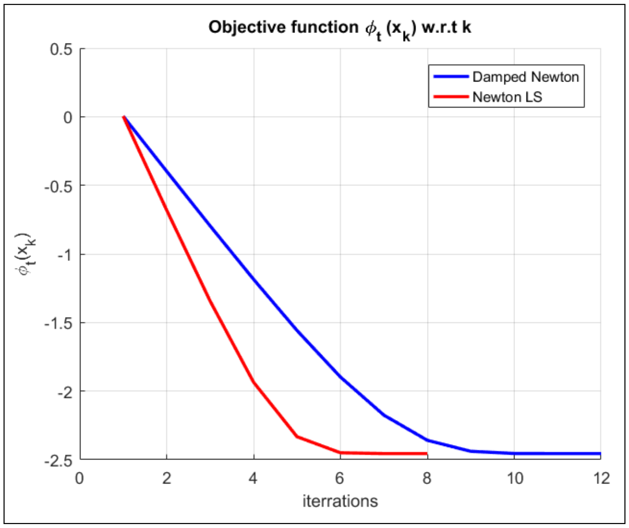


Figure 1: $\phi_t(x_k)$ w.r.t the number of iterations

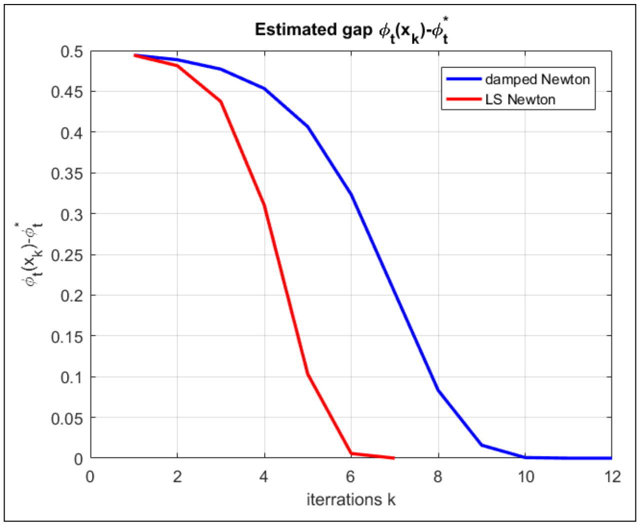


Figure 2: The gap to the optimal solution

We can notice that the use of the backtracking line-search allows to reach the optimal value in less number of iteration than by using the damped newton method defined above. Just for precision, the estimated gap that we have plotted: $\phi(x_k) - \phi(x^*) = \frac{\lambda^2(x_k)}{2}$

The value of x^* found for the two methods is the same:

$$\boxed{\mathbf{x^* = -10.3020}}$$

3 Support Vector Machine Problem

In the previous homework, we introduced the Data Separation problem (ex. 3), more commonly known as Support Vector Machine (SVM) problem. Its common formulation is the following. Given n data points $x_i \in R^d$ with labels $y_i \in \{-1, 1\}$ and a regularization parameter $\tau \succ 0$, the SVM reads:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{\tau n}\sum_{i=1}^{n} z_i + \frac{1}{2} \parallel \omega \parallel_2^2 & \text{(SVM-P)} \\ \text{subject to} \quad & y_i(\omega^T x_i) \geq 1 - z_i \qquad i = 1, ..., n & \\ & z \geq 0 & \end{aligned}$$

and its dual is written:

$$\begin{aligned} \text{maximize} \quad & \mathbf{1}^T\lambda - \frac{1}{2} \parallel \sum_{i=1}^{n} \lambda_i y_i x_i \parallel_2^2 & \text{(SVM-D)} \\ \text{subject to} \quad & 0 \leq \lambda \leq \frac{1}{n\tau} & \end{aligned}$$

1) Strictly feasible points for primal and dual:

- For the primal, we can choose z=2 × $\mathbf{1}_n$ and $\omega = \mathbf{0}_d$, this points satisfies strictly the constraints of (SVM-P) and therefore is strictly feasible.
- For the dual problem, we can choose $\lambda = \frac{1}{2n\tau}\mathbf{1}_n$

2) Implement the barrier method (using logarithmic barrier):

- First, we are asked to transform the primal problem (SVM-P) to a quadratic problem. The argument of the minimization in the problem (SVM-P) is $x = \begin{pmatrix} \omega \\ z \end{pmatrix}$. We want to find the form:

$$\begin{aligned} &min_x \quad \frac{1}{2}x^T Q x + p^T x \\ &sc \quad Ax \leq b \end{aligned}$$

  After some calculus, the expressions of Q, p,A,b found are:

$$Q = \left[\begin{array}{c|c} I_d & \mathbf{0}_{d\times n} \\ \hline \mathbf{0}_{n\times d} & \mathbf{0}_{n\times n} \end{array}\right] \qquad p = \left[\begin{array}{c} \mathbf{0}_d \\ \frac{1}{\tau n}\mathbf{1}_n \end{array}\right]$$

$$A = \left[\begin{array}{c|c} -diag(y)X & -I_n \\ \hline \mathbf{0}_{n\times d} & -I_n \end{array}\right] \qquad b = \left[\begin{array}{c} -\mathbf{1}_n \\ \mathbf{0}_n \end{array}\right]$$

- The dual problem is equivalent to:

$$\begin{aligned} -\text{minimize} \quad & -\mathbf{1}^T\lambda + \frac{1}{2} \parallel \sum_{i=1}^{n} \lambda_i y_i x_i \parallel_2^2 \qquad \text{(SVM-D)} \\ \text{subject to} \quad & 0 \leq \lambda \leq \frac{1}{n\tau} \end{aligned}$$

  We want then to find the form:

$$\begin{aligned} &-\, min_x \quad \frac{1}{2}x^T Q x + p^T x \\ &sc \quad Ax \leq b \end{aligned} \tag{1}$$

  The expressions of Q, p,A,b are:

$$Q = diag(y)XX^T diag(y) \qquad p = -\mathbf{1}_n$$

$$A = \left[\begin{array}{c} I_n \\ -I_n \end{array}\right] \qquad b = \left[\begin{array}{c} \frac{1}{n\tau}\mathbf{1}_n \\ \mathbf{0}_n \end{array}\right]$$

  That allows us to code the generic functions **[Q,p,A,b] = transform-svm-primal(tau,X,y)** and **[Q,p,A,b] = transform svm dual(tau,X,y)**

- Here we are asked to code a function **[x sol,xhist] = barr method(Q,p,A,b,x0,mu,tol)** which implements the barrier method to solve QP given the inputs (Q, p, A, b) and the initial point x0 (which should be strictly feasible), the steps of the implemented algorithm:

  given strictly feasible x0, $t := t^{(0)} > 0$, $\mu > 1$ and tolerance $tol > 0$ repeat

  - Damped Newton. Compute $x^*(t)$ by minimizing $\phi_t(x)$
  - Update. $x := x^*(t)$
  - Stopping criterion. quit if $gap = m/t < tol$.
  - Increase t. $t := \mu t$.

3) In this question we tested our code and try to separate the last two classes, i.e., Iris-versicolor versus Iris-virginica.. The vector of data has a dimension D=4 (4 attributes: 1. sepal length in cm / 2. sepal width in cm/ 3. petal length in cm / 4. petal width in cm) and N=100 (50 for each class). We have then **centred** the data and used 80% to train our algorithm and 20% for the test (out-of-sample)
In he figure below (cf fig 3) we plotted in 2D the petal width w.r.t to the petal length (centred) to see the distribution of the two classes:

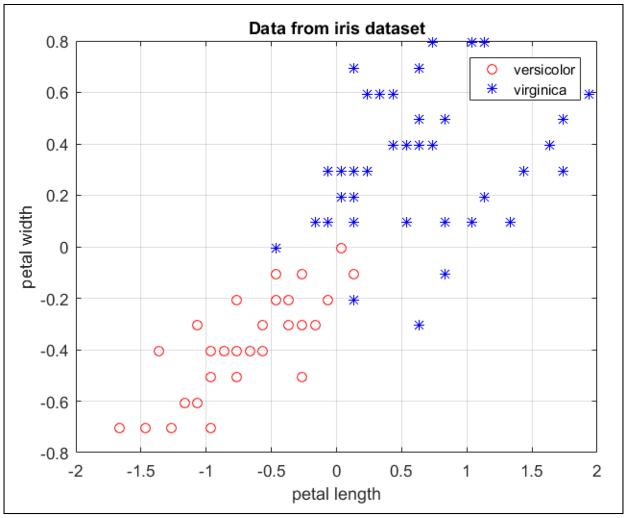


Figure 3: The petal width w.r.t the petal length

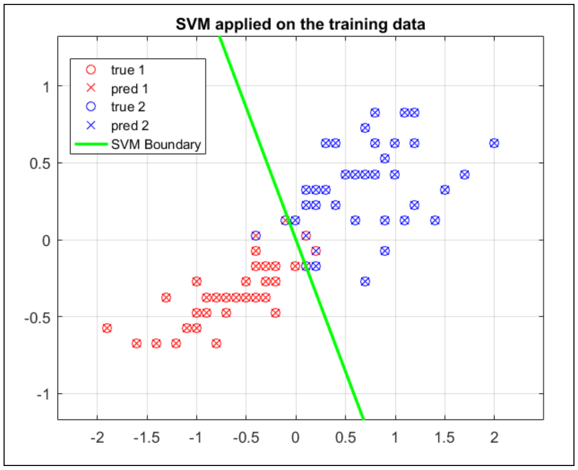
We applied the function transform **transform-svm-primal** in order to find the parameters of the QP problem :Q,p,A,b. Then we use our function barr_method in order to to compute the solution of our primal problem. Just to recall, the vector primal which is the argument of the objective function is $X = \begin{pmatrix} \omega \\ z \end{pmatrix}$. The solution gives us ω^* which is the optimal solution and the parameter of SVM. We chose as an initial vector a feasible vector verifying the condition of the question 1. The optimal solution found for $\tau = 10$ and $\mu = 10$ is:

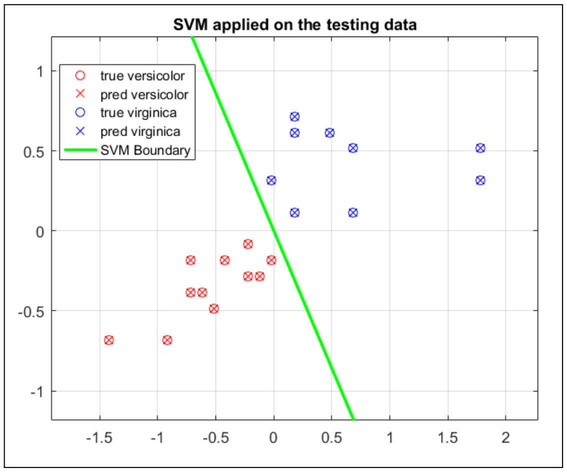
$$\omega^* = \begin{pmatrix} -0.4818 \\ -0.1809 \\ 0.6859 \\ 0.3345 \end{pmatrix}$$

Moreover, we were able to find the same optimal solution ω^* by solving the dual problem using the barrier method to find λ^* and then apply the formulat at the optimum:

$$\omega^* = \sum_{i=1}^{m} \lambda_i^* y_i x_i$$

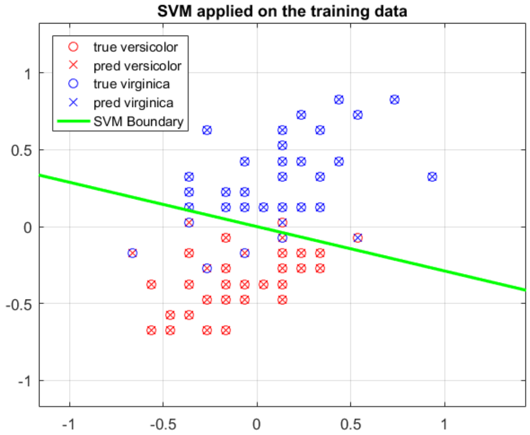
In the 2 figures below, we represented the result of classification with SVM (training (left) and testing (right)), using the parameter ω^*. In the same figure we plotted the true and the predicted labels in order to see the misclassifications points:

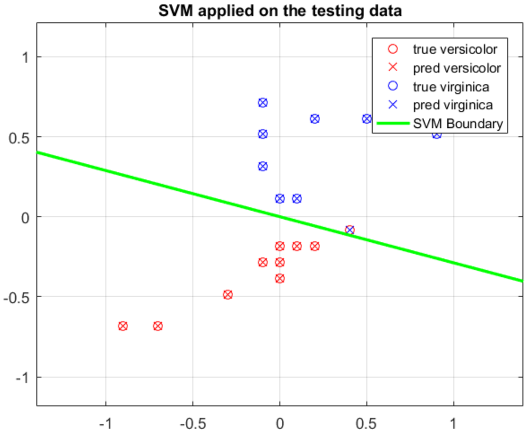




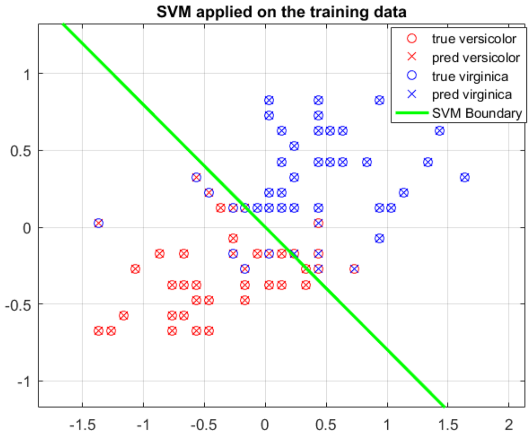
We also, represented some the plots of same other attributes to see the quality of classification:

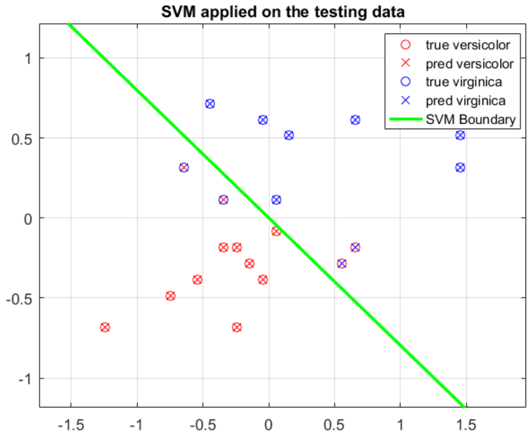
- petal width w.r.t sepal width





- petal width w.r.t sepal length:





In order to see how the performance of our SVM classifier changes w.r.t τ, we have applied the barrier function with different values of tau, and then classify according to the parameter ω found, The figure 4 shows the evolution of performance for $\tau = 0.001 : 0.01 : 5$

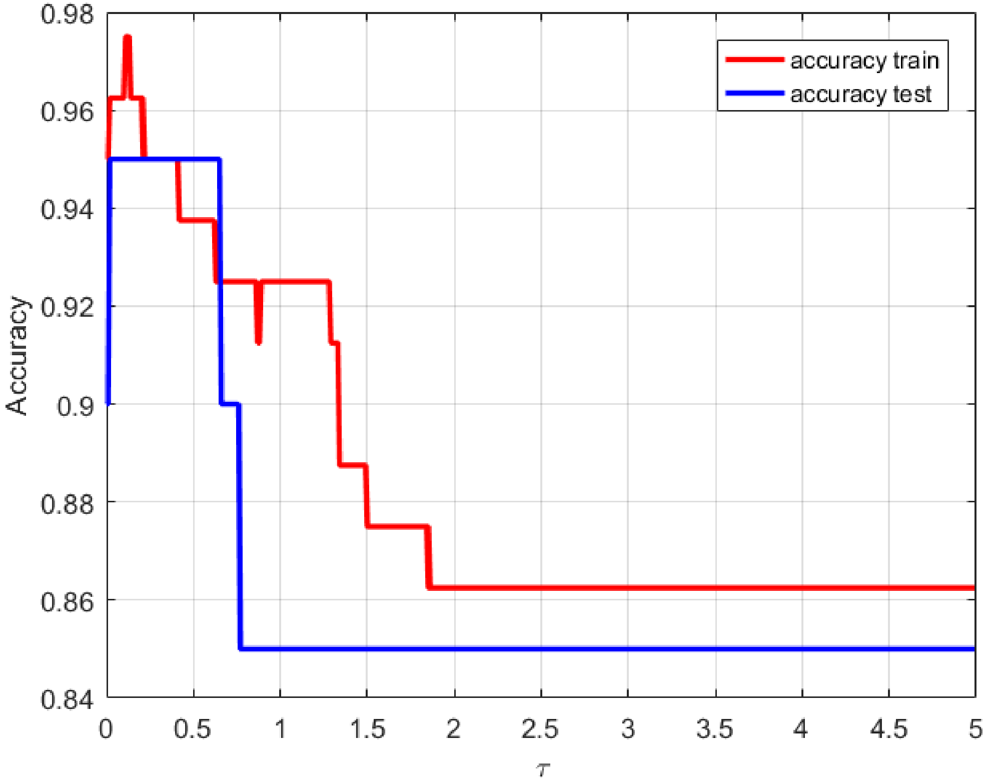


Figure 4: The accuracy of training and testing w.r.t τ

We can notice that in global, the performance of the SVM classifier decreases when τ increases. A best choice of the regularization parameter μ will be to choose it small enough to have best performances.

Remark: We will discuss the optional question in the last part of this homework, where we will change the data and apply the optimization method.

4)

- For the primal problem, we represent the semilog-scale for different values of the barrier method (Gap and the objective function), for $\mu = 2, 15, 50, 100$ with a fixed value of $\tau = 0.001$

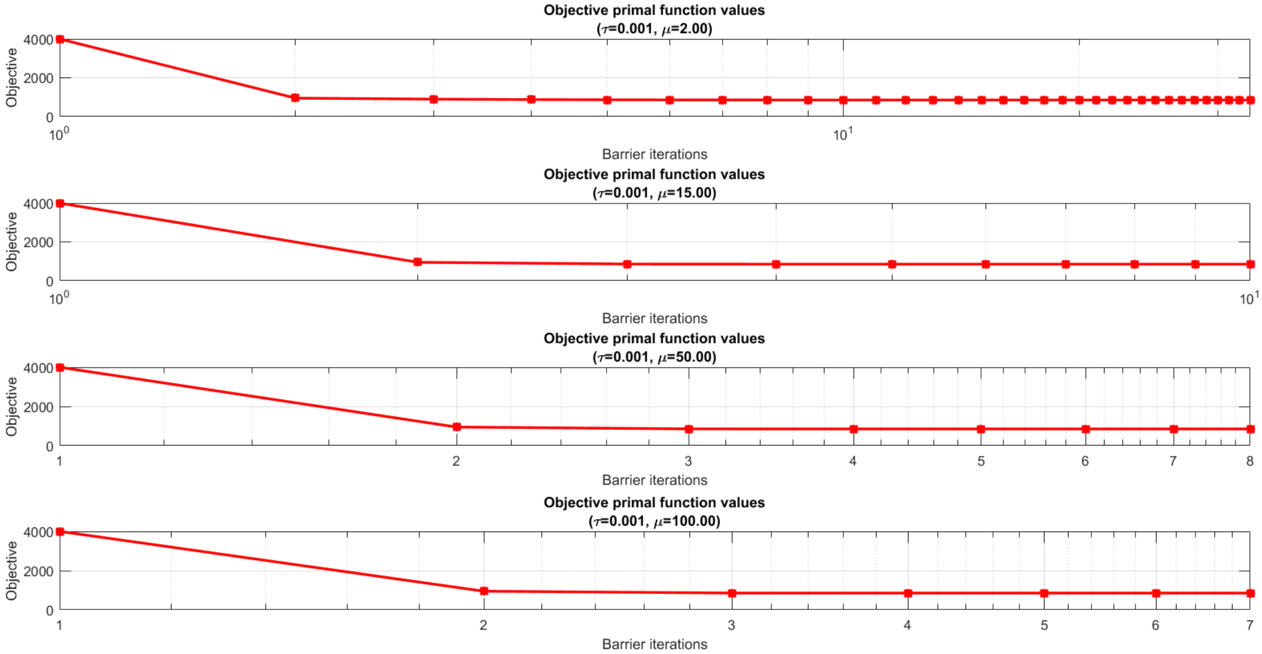


Figure 5: **The semilog-scale of the primal objective function w.r.t the barrier iterations**

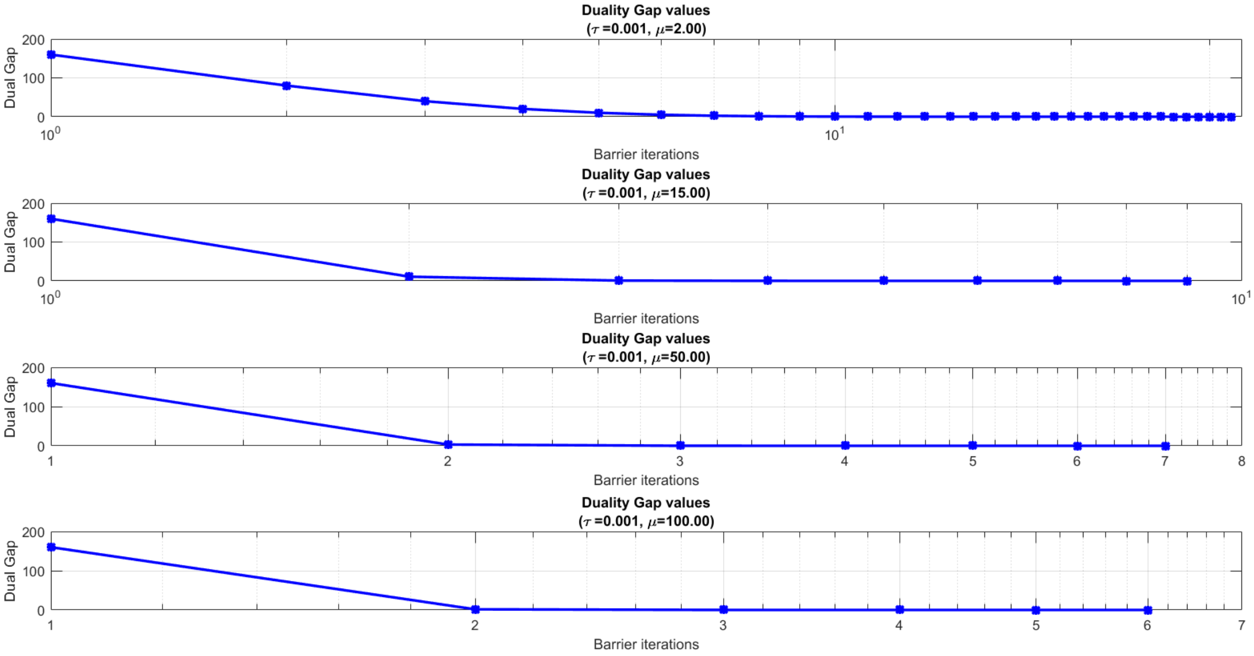


Figure 6: **The semilog-scale of the the duality Gap w.r.t the barrier iterations**

- For the dual problem, we represent the semilog-scale for different values of the barrier method (Gap and the objective function), for $\mu = 2, 15, 50, 100$

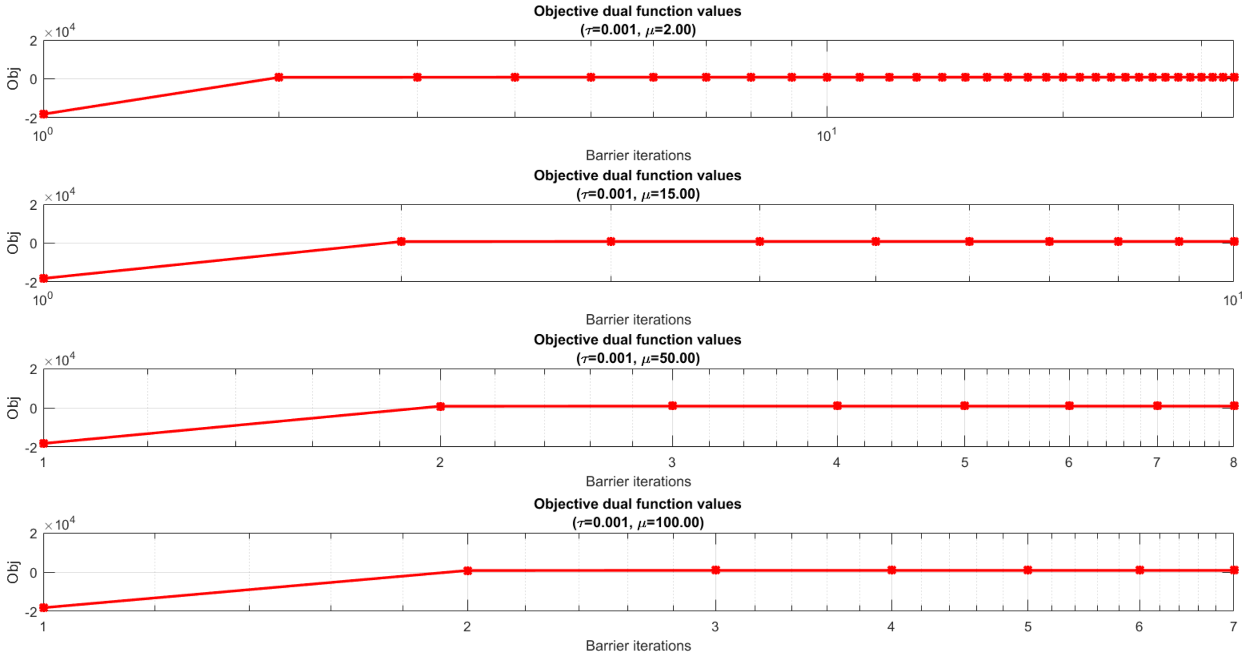


Figure 7: **The semilog-scale of the dual objective function w.r.t the barrier iterations**

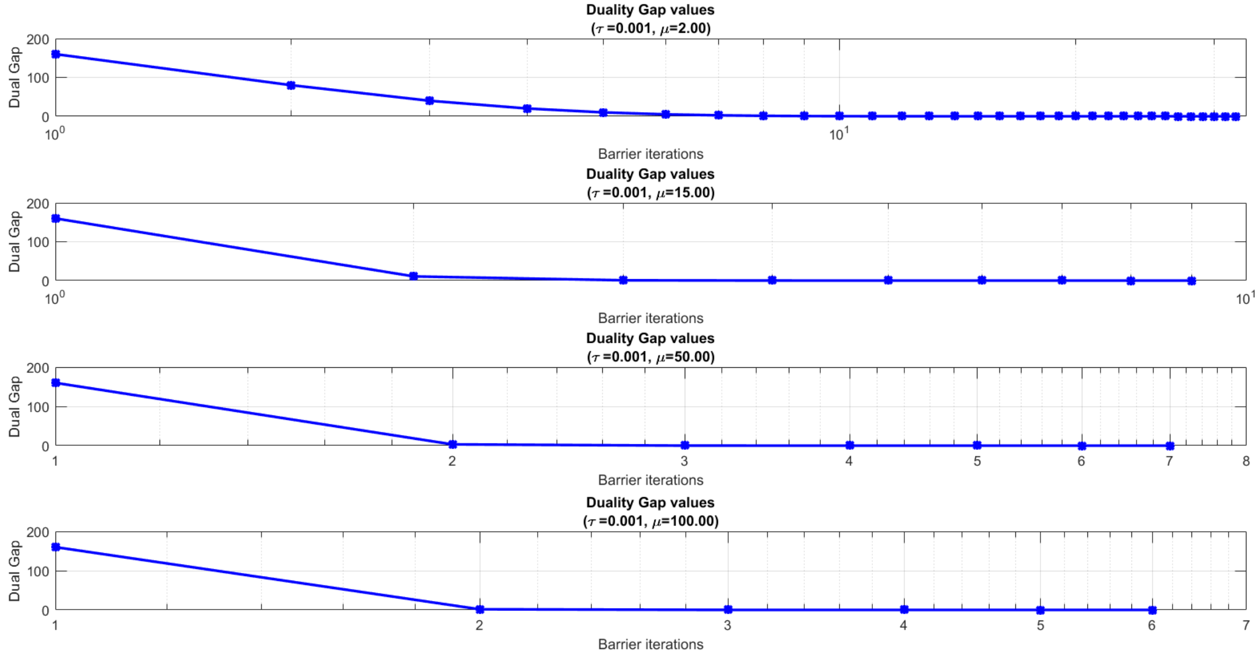


Figure 8: **The semilog-scale of the the duality Gap w.r.t the barrier iterations**

Remarks

- In the figure 5, we notice that the primal objective function ($\frac{1}{2}x^TQx+p^Tx$) started with a high value of the objective function (4000), and then started to decrease w.r.t the number of iteration until reaching its minimum. (Tis is true for all values of μ).
- We notice, that as μ increases, the number of iterations until convergence decreases. To see that, for $\mu = 2$, the iterations numbers was about 30, but it decreases until reaching 7 iterations for $\mu = 100$. The figure 9 below shows clearly the evolution of the iteration number w.r.t μ

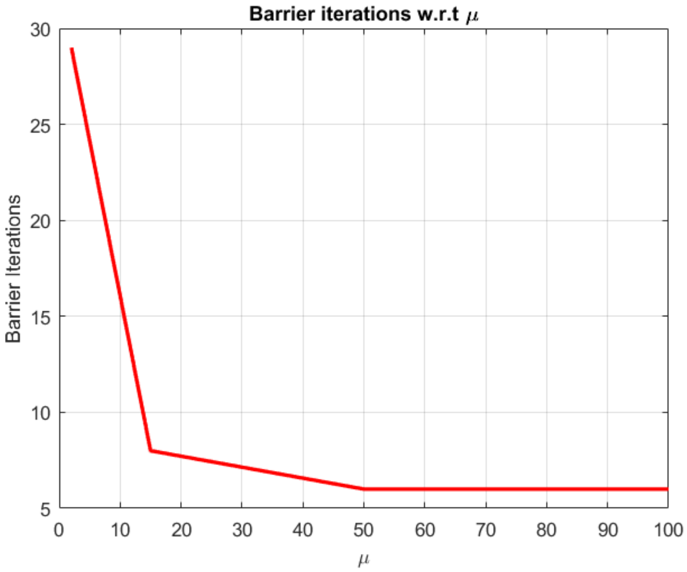


Figure 9: **Barrier iteration w.r.t μ**

- In the figure 6, we see that the duality gap converges faster with big values of μ.
- In the figure 7, we represented the dual objective function $\frac{1}{2}x^TQx + p^Tx$, with sign (-) (cf equation 1). We can notice the same remarks of the primal problem w.r.t μ (i.e less iteration as μ increases).
- The duality gap is the same between the primal and the dual problem (cf figures 6 and 8).
- The values of the optimal primal objective function is the same as the optimal dual objective function : strong duality for $\mu = 100$ for instance

$$\boxed{\textbf{p* =d* =855.0564}}$$

3.1 Optional question: New data

The data used in this question was provided in another course of MVA master (PGM), in order to make the classification. The training data is composed of 2 attributes and n=150 examples. As for the testing data, it is composed of 1500 examples. We should classify 2 classes.

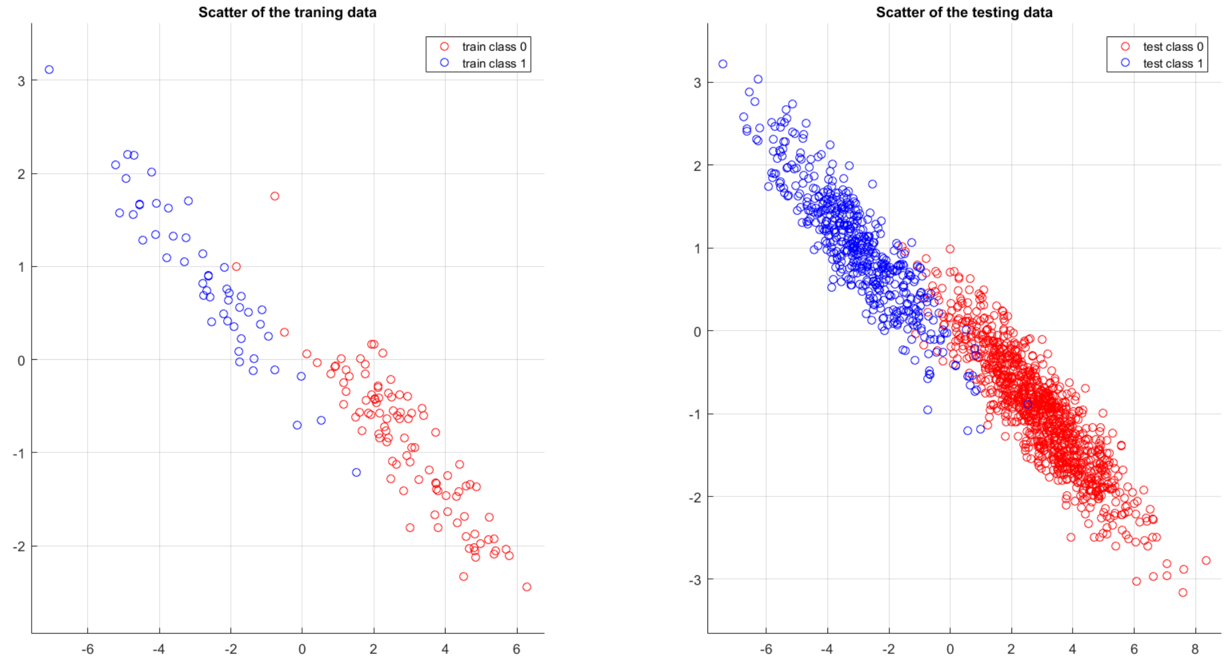


Figure 10: **Scatter plot of the training and testing data**

We applied the Barrier method to find the optimal parameter of SVM, the result of classification applied on:

- **The training data:**

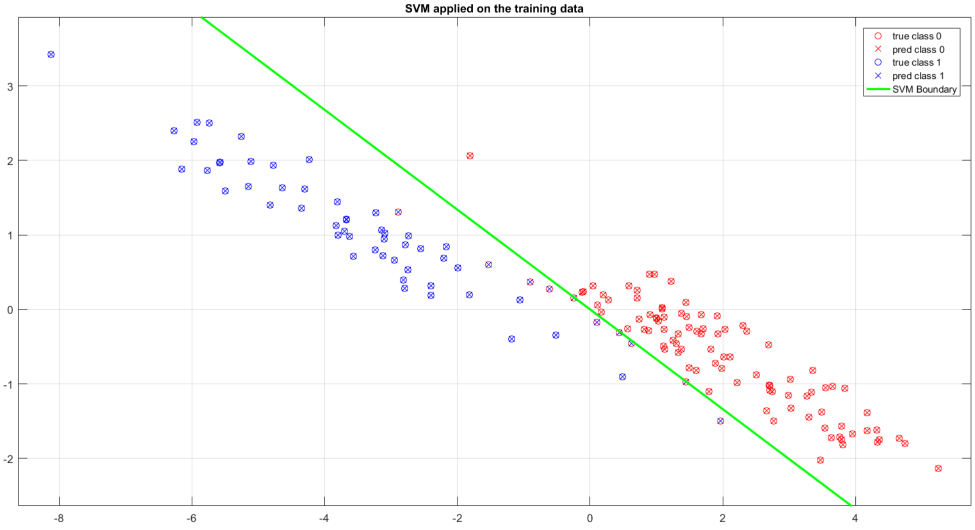


Figure 11: **SVM applied on the training data**

- **The testing data:**

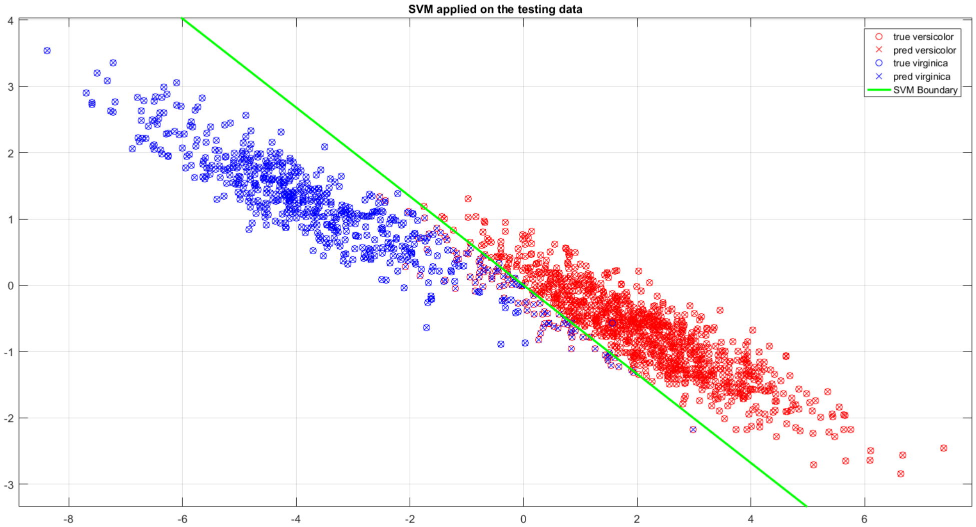


Figure 12: **SVM applied on the testing data**

Now we will see the influence of τ on the performance in order to confirm the remarks we made in the question 3 (cf fig. 4). This allows us to tune the parameter τ. The figure below (cf fig 13) shows the performance w.r.t τ:

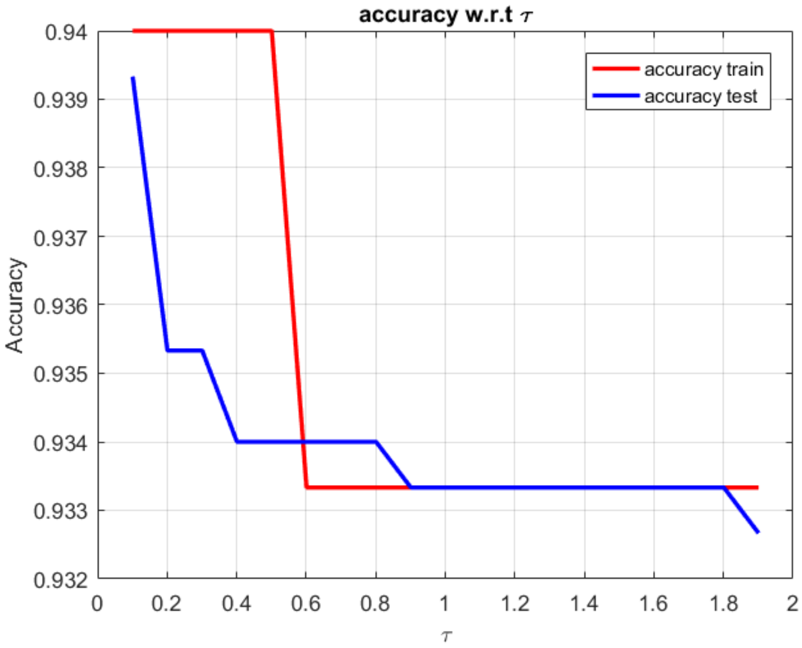


Figure 13: **Accuracy of training and testing w.r.t τ**

Same remarks: We should take τ relatively small!!