SAPIENZA
UNIVERSITÀ DI ROMA

Report project2 - Group 4, MDB - OMML

Marco Casalbore, Daryna Hnidets, Bianca Ghiurca

Fall 2023

	Hyperparameters			**ML Performance**		**Optimization Performance**		
Question	**C**	γ	**q**	Training Acc.	Test Acc.	#Iterations	KKT Viols.	Comp. Time
Q_1	1	2		100%	97%	13	$2.5 \cdot 10^{-6}$	10.49 *sec*
Q_2	1	2	96	100%	97%	328	$9.8 \cdot 10^{-4}$	8.60 *sec*
Q_3	1	2	2	100%	97.25%	15169	$9.7 \cdot 10^{-6}$	8.49 *sec*
Q_{bonus}	1	2		100%	84.5%	41	x	47.18 *sec*

1 Q.1 - Soft SupportVectorMachine

For the first exercise was implemented, as requested, a Soft-SVM; a dataset featuring images of dresses and shirts was provided, that has been initially appropriately scaled as preliminary step to the entire optimization process.
The selected function as the Kernel for the non-linear Support Vector Machine is the polynomial function: $k(x, y) = (x^T y + 1)^\gamma$. This choice was guided by computational time; both kernels provided with good accuracy, but the polynomial kernel was faster.
As a first step, a combined process of grid-search and k-fold cross-validation is employed to determine the hyperparameters of the SVM: the upper bound of alphas C (for values of [1, 10, 100]), and the degree of the polynomial function γ (for values of [2, 3, 4, 5]).
The final choice of C and γ, as suggested by the grid search, setted on C = 1 and $\gamma = 2$.
In figure 2 is displayed the SVM's confusion matrix.
Considering that the problem involves a quadratic objective function and linear constraints, it was clear that the best decision was to implement an optimization routine specific for quadratic programming; for this reason the solvers.qp routine was adopted directly from the cvxopt python library. An additional parameter that has been carefully chosen is the tolerance (which serves as the lower bound for α): it was decided to set it to $1{\cdot}10^{-5}$. Decreasing this value ($1{\cdot}10^{-6}$, $1{\cdot}10^{-7}$, $1{\cdot}10^{-8}$, $1{\cdot}10^{-9}$) an increasing in the KKT violations was observed.

The optimization routine produces the following results:

- F.O.$_{initial}$ = 0
- F.O.$_{final}$ $\simeq$ -0.1074
- Number of iterations = 13
- KKT violations $\simeq$ 0.0000025 $\simeq$ 0
- Training accuracy = 100%
- Test accuracy = 97%

(As discussed with the teacher assistant, the selected routine solvers.qp does not provide the number of function evaluations as an output of the optimization routine, and for this reason, it is omitted from the optimization performances.)
Also the validation accuracy was computed and it turned out being 96%.
Regarding the behavior of the model in terms of overfitting and underfitting, several tests were conducted by varying the hyperparameters C and γ.
Varying the C values (1, 10, 100, 1000), it has been observed a slight increase in the number of iterations of the optimization routine (13, 14, 16, 18), while, in this particular case, all the other performance metrics remained almost unchanged.
On the other hand, as can be observed in figure 1, with the increase in γ, it was noted that the accuracy on the test set decreases while the accuracy on the training set remains constant (indicating overfitting). Several tests on γ have been done, considering the values (3, 4, 5, 6, 7, 8, 9, 10); as an example, $\gamma = 8$ resulted in a test accuracy of 66%. However, it was also observed that an increase in γ leads to a significant decrease in the final value of the objective function.

2 Q.2 - Soft SVM decomposition method q $\geq$ 4

For the second task, as requested, it's implemented a decomposition method for an SVM with the same kernel function and hyperparameters of Question 1 (polynomial kernel, C = 1, γ = 2).
As observed in Figure 3, the computational time is closely linked to the number of q; therefore, with the aim of minimizing it, q is set to 96 (all possible even values of q were tested in the range of q$\in [4, 100]$). This q value provided a validation accuracy of 96,75%. It was observed that for some q values the computational time increases up to a few minutes (figure 3).
As starting point it's picked $\alpha_0 = 0$ so that $\nabla f(\alpha_0) = -e$.
This allowed to update the gradient at each iteration without the need to construct the whole Q matrix, in fact only the columns related to the working set's indexes were used.
The structure of the decomposition method is based on a stopping criterion chosen to be, as requested, the optimality condition $m - M \leq 0$: a while loop was implemented for the evaluation of the condition, where, for computational reasons, instead of comparing it directly to 0, it is compared to an ϵ value, which, after several tests, was assigned to $1{\cdot}10^{-3}$. Using ϵ values equal to (10^{-4}, 10^{-5}), the computational time increased significantly.
As for the construction of the working-set, the indices q_1 and q_2 (both equal to $q/2$) related to W_k were determined by selecting them from R(α_k) and S(α_k), respectively, following the selection rule of W_k of the SVM$_{light}$ algorithm.
Also in this case a numerical tolerance as a lower bound for the α equal to $1{\cdot}10^{-5}$ was adopted.
Following the same reasoning employed in Question 1, to solve the quadratic programming problem it's implemented the solvers.qp routine directly from the cvxopt.

The optimization routine produces the following results:

- F.O.$_{initial}$ = 0
- F.O.$_{final}$ $\simeq$ -0.1073
- Number of iterations = 328
- KKT violations $\simeq$ 0.0009831 $\simeq$ 0
- Training accuracy = 100%
- Test accuracy = 97%

In figure 4 is displayed the SVM's confusion matrix obtained with the decomposition method.

3 Q.3 - Soft SVM decomposition method q = 2

For the final task, it was requested to implement an SMO algorithm, specifically employing the Most Violating Pair (MVP) algorithm. The basic structure of the program remains the same (polynomial kernel, C = 1, $\gamma = 2$).
Only in this case, as requested, an handmade analytical solution of the quadratic programming problem was performed using the exact lineasearch (consequently, no solver was implemented contrarily to Question 1 and 2 where solvers.qp from the cvxopt library was used). The direction has been updated, at each iteration, using the following rule:

$$d_h^{i,j} = \begin{cases} y_i & \text{if } h = i, \quad i \in I(\alpha_k) \\ -y_j & \text{if } h = j, \quad j \in J(\alpha_k) \\ 0 & \text{otherwise} \end{cases}$$

For the stepsize t_k the t_{max}^{feas} was computed. Again α_0= 0 was picked as starting point.
A numerical tolerance, as lower bound for the α, has been set to $1{\cdot}10^{-4}$; the choice was driven by considering both performances, in terms of computational time and accuracy; decreasing this value [10^{-5}, 10^{-6}, 10^{-7}], the computational time and the number of iterations increased significantly.
The structure of the SMO algorithm is based on a stopping criterion chosen to be the optimality condition $m - M \leq 0$: also in this case, in the while loop, instead of comparing this difference to 0, it was compared to an ϵ value equal to $1{\cdot}10^{-5}$ (for values grater then 10^{-4} the test accuracy worsened to 96.75%).
The optimization routine produces the following results:

- F.O.$_{initial}$ = 0
- F.O.$_{final}$ $\simeq$ -0.09927
- Number of iterations = 15169
- KKT violations $\simeq$ 0.0000097 $\simeq$ 0
- Training accuracy = 100%
- Test accuracy = 97.25%

In figure 5 is displayed the SVM's confusion matrix having implemented the MVP algorithm.

4 A brief comparison

First of all excellent accuracy values were achieved for all the three questions.
The MVP decomposition method has reached a slightly higher test accuracy.
Regarding computational time, as expected, there is a decrease in the decomposition methods compared to Question 1; in particular, the minimum value of computational time was obtained with the SMO with MVP method.
An important difference between the second question's SVM and the SVM obtained applaying the MVP method is in the use of the solver: in fact, an analytic solution was implemented in order to solve the sub problems of the MVP, while for the decomposition method of question 2 the solvers.qp routine of cvxopt was adopted.
Fot both decomposition methods it was never used the intire Q matrix for updating the α; only the strictly needed columns of the matrix were computed.
For all three questions the KKT violations always remained low.

5 Q.bonus - SVM for multiclass classification

As requested, an SVM model for multiclass classification has been implemented for the final point.
Once again, as with point 1, it was necessary to perform appropriate scaling of the dataset consisting in images of dresses, shirts and this time also pullovers.
For multiclass classification, the One Against All technique has been adopted. For the calculation of individual decision functions, the same parameters as those set for Question 1 have been setted, namely C = 1, $\gamma = 2$, polynomial kernel and tolerance = $1 \cdot 10^{-5}$. The solvers.qp routine was used.
To decide the belonging class of a new unseen sample x, it is taken the function that gives the maximum value of the argument of the sign in each of the sign functions previously calculated:

$$class(x) = arg \max_{1 \leq j \leq M} \left\{ \sum_{i=1}^{P} \alpha_i^{m*} y_m^i K(x^i, x) + b^{m*} \right\} \tag{1}$$

The performance measures of the model are:

- Training accuracy = 100%
- Test accuracy = 84.5%
- Computational Time = 47.18 *sec*
- Number of iterations = 41

In the following table are reported the KKT violations and the initial and final values of the objective function:

Istance	KKT Violations	Initial Obj Function	Final Obj Function
Class 2 against All	$1.17 \cdot 10^{-4}$	0	-0.89
Class 3 against All	$1.07 \cdot 10^{-5}$	0	-0.42
Class 6 against All	$1.94 \cdot 10^{-4}$	0	-1.34

It is interesting to note that in exercises 1, 2, and 3 where the SVM simply had to perform binary classification, the task was accomplished with excellent accuracy. However, in this case, involving classification into 3 classes, the SVM struggles significantly, exponentially increasing computational time and considerably diminishing accuracy performance on the test-set.
Nevertheless the same training accuracy was achieved as the one obtained in the previous questions.
In figure 6 is displayed the SVM's confusion matrix for the multiclass classification.

6 Images and Graphs

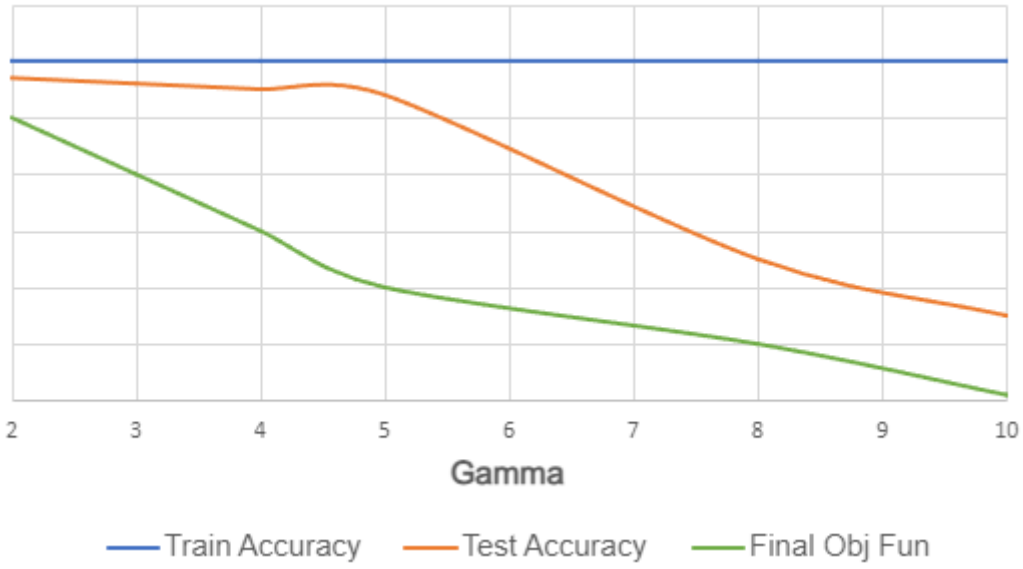

Figure 1: Performance measures varying γ, Q_1

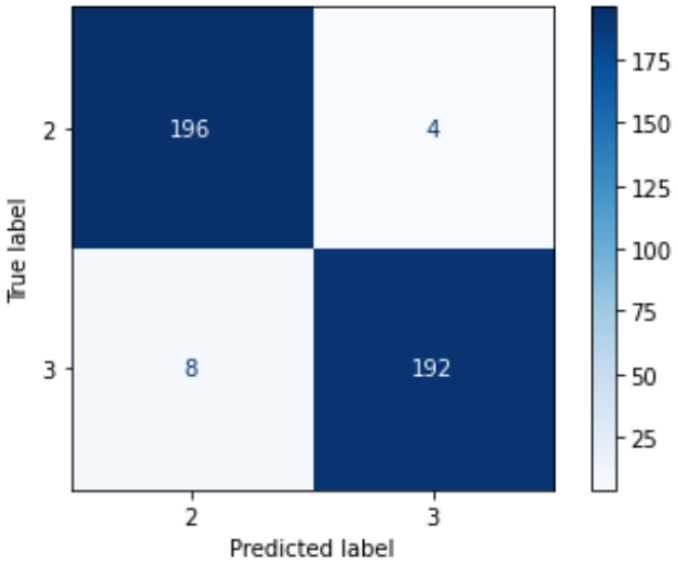

Figure 2: SVM's confusion matrix, Q_1

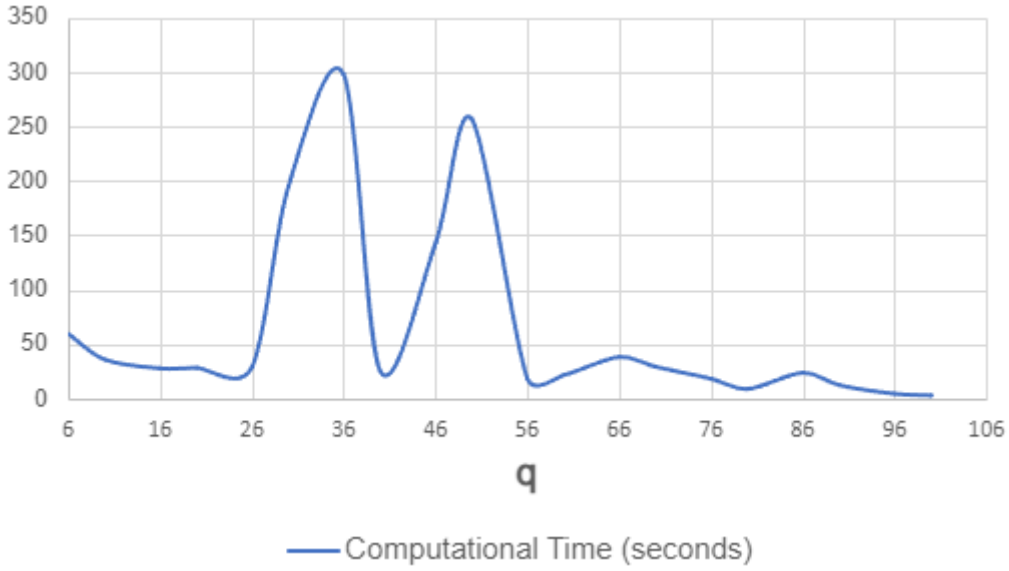

Figure 3: Computational time varying the number q, Q_2

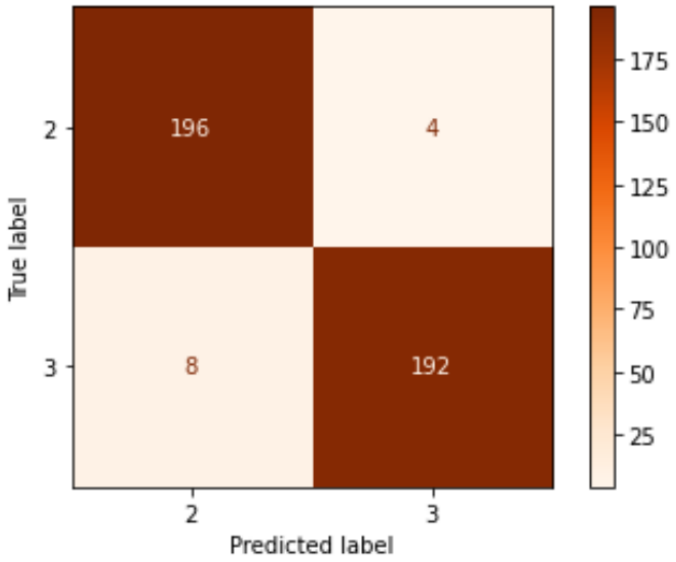

Figure 4: SVM's with decomposition method confusion matrix, Q_2

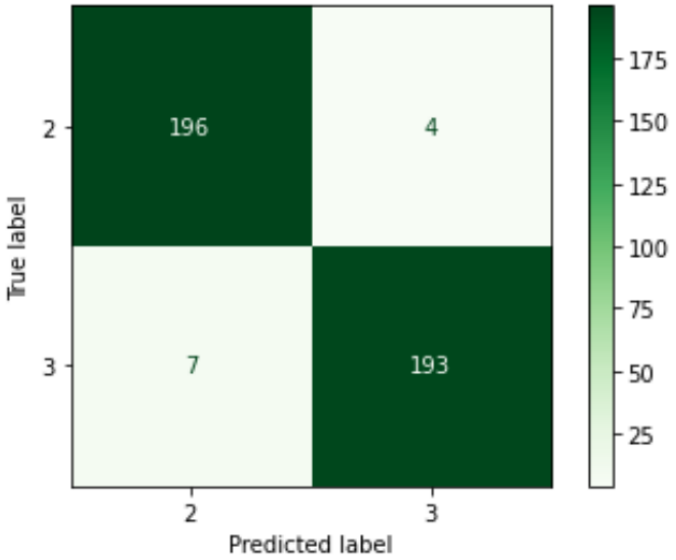

Figure 5: SVM's with MVP algorithm confusion matrix, Q_3

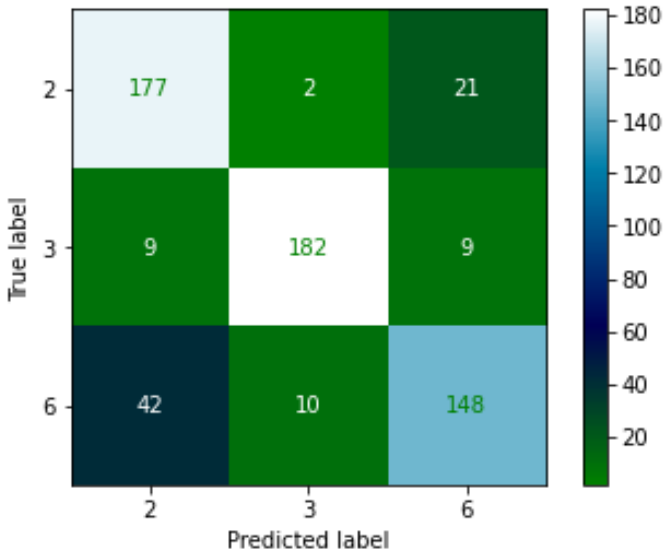

Figure 6: SVM's for multiclass classification confusion matrix, Q_{bonus}