Optimization Methods for Machine Learning - Fall 2024

PROJECT # 2
Support Vector Machines

Laura Palagi & Edoardo Cesaroni

Posted on November 21, 2024 - due date December 22, 2024 (midnight 23:59)

Instructions

The project will be developed in team, the same as the first project unless severe exceptions: each team must hand in their own answers. We will be assuming that, as participants in a graduate course, every single student will be taking the responsibility to make sure his/her personal understanding of the solution to any work arising from such collaboration.

Submission

Projects must be uploaded using a google form

https://forms.gle/7b8JovPXxBKYkj8W6

Each team leader is in charge of uploading the files.

The team must upload the phyton files organized as requested in the instructions at the end of the project, in a compressed folder named "GroupNumber"_"GroupName"_code.zip.
A typed report in English and in pdf format must be uploaded too. **The report must be of at most 4 pages, excluded figures that must be put at the end. Do not include part of the python coding.** The report must be named "GroupNumber"_"GroupName"_report.pdf

Evaluation criteria

The grade is "Italian style" namely in the range [0,31], being 18 the minimum degree to pass the exam and 31=30 cum laude.
The project is due at the latest at midnight on the due date. For late submissions, the score will be decreased. It is worth 85% for the next 48 hours. It is worth 70% from 48 to 120 hours after the due date. It is worth 50% credit after a 120-hour delay.
The second project accounts for 35% of the total vote of the exam.
For the evaluation of the second project, the following criteria will be used:

1. 60% check of the implementation
2. 40% quality of the overall project as explained in the report.

Assignment Description

In this project, you will implement different optimization methods for training Support Vector Machines (supervised learning) in order to solve a classification problem.

Data sets.
You are asked to build a classifier that distinguishes of handwritten digits from The MNIST database. Each sample is a 28x28 grayscale image, associated with a label.
It is necessary to scale the data before the optimization.

You will be given a full dataset and you will extract the target set made up of images belonging to two (Question 1,2,3) or three (Question 4) classes. For Questions 1, 2 and 3, the target set will be made of images of the number 1 and 5 (identified by the labels 1 and 5, in the target set, respectively). In the 4 question, there will also be images of 7 (label 7). You can obtain test set by randomly splitting the target set into a training set and a test set with a percentage of 80%, 20 %. Use one of your student ID number as a seed for the random split. Instruction on how to import the data in Python will be provided in a separate file. A $k-$fold cross-validation can be used to set the hyperparameters. Please note that the teacher has defined a test set that might be used to check the accuracy of your models.

The SVM problem Consider a nonlinear SVM with kernel $k(\cdot,\cdot)$ and the corresponding nonlinear decision function

$$y(x) = sign\left(\sum_{p=1}^{P} \alpha_p y^p k(x^p, \bar{x}) + b\right)$$

where α, b are obtained as the optimal solution of the dual nonlinear SVM problem.

As kernel function you may choose either a RBF kernel

$$k(x,y) = e^{-\gamma\|x-y\|^2}$$

where $\gamma > 0$ is an hyper parameter, or a polynomial kernel

$$k(x,y) = (x^T y + 1)^{\gamma}$$

with hyper parameter $\gamma \geq 1$.

You are requested to train the SVM, namely to both set the values of the hyperparameters C and γ with a heuristic procedure, and to find the values of the parameters α, b with an optimization procedure. For this last task, you need to follow the different optimization procedures described in the following questions.

$$y(x) = sign\left(\sum_{p=1}^{P} \alpha_p y^p k(x^p, x) + b\right)$$

nuovo sample

$$K(x^p, x) = e^{-\gamma \|x^p - x\|}$$

oppure

$$K(x^p, x) = (x^{pT} x + 1)^{\gamma}$$

non dipende da w

α e b li ottengo dal problema SVM duale non lineare

Primal Linear SOFT SVM

$$\min \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{P} \xi_i$$

$$y^i \left[w^T x^i + b\right] - 1 + \xi_i \geq 0 \qquad i = 1, \ldots, P$$

$$\xi_i \geq 0 \qquad i = 1, \ldots, P$$

The dual problems is

$$\max_{w,b,\xi,\alpha,\mu} \quad L(w, b, \xi; \alpha, \mu)$$

$$\nabla_w L(w, b, \xi; \alpha, \mu) = 0$$
$$\nabla_b L(w, b, \xi; \alpha, \mu) = 0$$
$$\nabla_\xi L(w, b, \xi; \alpha, \mu) = 0$$
$$\alpha \geq 0, \mu \geq 0$$

$$\begin{aligned} L_{HARD}(w, b; \alpha) &= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{P} \alpha_i \left(y^i \left[w^T x^i + b\right] - 1\right) \\ &= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{P} \alpha_i y^i w^T x^i - \sum_{i=1}^{P} \alpha_i y^i b + \sum_{i=1}^{P} \alpha_i \\ &\quad \text{sub. in } w = \sum \alpha_i y^i x^i \qquad \text{sub. in } 0 = \sum \alpha_i y^i \\ &= \frac{1}{2}\|w\|^2 - w^T \sum_{i=1}^{P} \alpha_i y^i x^i - b\sum_{i=1}^{P} \alpha_i y^i + \sum_{i=1}^{P} \alpha_i \\ &= \frac{1}{2}\|w\|^2 - \underbrace{w^T w}_{\|w\|^2} + \sum_{i=1}^{P} \alpha_i = -\frac{1}{2}\|w\|^2 + \sum_{i=1}^{P} \alpha_i \end{aligned}$$

KKT

The Lagrangian is

$$L(w, b, \xi; \alpha, \mu) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{P} \xi_i - \sum_{i=1}^{P} \alpha_i \left(y^i \left[w^T x^i + b\right] - 1 + \xi_i\right) - \sum_{i=1}^{P} \mu_i \xi_i$$

$$L(w, b, \xi; \alpha\mu) = L_{HARD}(w, b; \alpha) + C\sum_{i=1}^{P} \xi_i - \sum_{i=1}^{P} \alpha_i \xi_i - \sum_{i=1}^{P} \mu_i \xi_i$$

$$\nabla_w L(w, b, \xi; \alpha, \mu) = w^* - \sum_{i=1}^{P} \alpha_i^* y^i x^i = 0$$

$$\nabla_b L(w, b, \xi; \alpha, \mu) = -\sum_{i=1}^{P} \alpha_i^* y^i = 0$$

$$\nabla_\xi L(w, b, \xi; \alpha, \mu) = Ce - \alpha - \mu = 0$$

N.B. $e = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$

The SOFT dual problem

$$\max_{w,b,\xi,\alpha,\mu} \quad L_{HARD}(w,b;\alpha) + \sum_{i=1}^{P}(C-\alpha_i-\mu_i)\xi_i$$
$$w = \sum_{i=1}^{P} \alpha_i^* y^i x^i$$
$$y^T\alpha = 0$$
$$Ce - \alpha - \mu = 0$$
$$\alpha \geq 0, \mu \geq 0$$

$$\max_{w,b,\alpha,\mu} \quad L_{HARD}(w,b;\alpha)$$
$$w = \sum_{i=1}^{P} \alpha_i^* y^i x^i$$
$$y^T\alpha = 0$$
$$Ce - \alpha - \mu = 0$$
$$\alpha \geq 0, \mu \geq 0$$

$$\max_{\alpha,\mu} \quad -\frac{1}{2}\left\|\sum_{i=1}^{P}\alpha_i y^i x^i\right\|^2 + e^T\alpha$$
$$y^T\alpha = 0$$
$$\mu = Ce - \alpha \geq 0$$
$$\alpha \geq 0,$$

$$\min_{\alpha} \quad \frac{1}{2}\left\|\sum_{i=1}^{P}\alpha_i y^i x^i\right\|^2 - e^T\alpha$$
$$y^T\alpha = 0$$
$$0 \leq \alpha \leq Ce,$$

Primal SOFT SVM

$$\min \quad \frac{1}{2}\|w\|^2 + C\ldots?$$
$$y^i\left[w^T x^i + b\right] \geq 1 - \xi_i \;\forall\, i$$
$$\xi_i \geq 0 \;\forall\, i$$

Convex Quadratic programming $n+P+1$ variables and P linear inequality constraints + P simple constraints

Dual SOFT SVM

$$\min_{\alpha\in\mathbb{R}^P} \quad \frac{1}{2}\alpha^T Q\alpha - e^T\alpha$$
$$\text{s.t.} \quad \alpha^T y = 0$$
$$0 \leq \alpha \leq C$$

Convex Quadratic programming P variables and one linear equality constraint and $2P$ simple constraints

Strong duality holds:

Let α^* be a solution of the dual problem, then there exists w^*, b^*, ξ^* for the primal and $w^* = \sum_{i=1}^{P}\alpha_i^* y^i x^i$, b^* is found by complementarity conditions on primal constraints.

The SOFT dual problem

The Wolfe dual problems of Linear Soft SVM

$$\min \quad \frac{1}{2}\left\|\sum_{i=1}^{P}\alpha_i y^i x^i\right\|^2 - e^T\alpha$$

$$\sum_{i=1}^{P}\alpha_i y^i = 0$$
$$0 \le \alpha_i \le C$$

where $e = (1,\ 1,\ 1,\ \ldots,\ 1)^T$

- It is a quadratic convex problem in $\alpha \in \mathbb{R}^P$
- It admits always a feasible solution $\alpha = 0$
- A vector α^* is a global solution iff it satisfies the KKT conditions

Primal-dual Linear SVM

Dual HARD SVM

$$\min_{\alpha\in\mathbb{R}^P} \quad \frac{1}{2}\alpha^T Q\alpha - e^T\alpha$$
$$\text{s.t.} \quad \alpha^T y = 0$$
$$0 \le \alpha$$

Dual SOFT SVM

$$\min_{\alpha\in\mathbb{R}^P} \quad \frac{1}{2}\alpha^T Q\alpha - e^T\alpha$$
$$\text{s.t.} \quad \alpha^T y = 0$$
$$0 \le \alpha \le C$$

with $Q = (y^j y^i {x^i}^T x^j)_{i,j=1,\ldots,P}$ positive semidefinite.

Only scalar products among samples appear.

$$Q = (y^j y^i {x^i}^T x^j)_{i,j=1,\ldots,P}$$

The classifier

The decision function of Linear SVM is

$$f(x; w^*, b^*) = sign(w^{*T} x + b^*)$$

with

$$w^* = \sum_{i=1}^{P} \alpha_i^* y^i x^i \qquad b^* = y^h - w^{*T} x^h \qquad \text{for some } h: \ 0 < \alpha_h^* < C$$

Hence we can write for some h s.t. $0 < \alpha_h^* < C$

$$w^{*T} x + b^* = \left(\sum_{i=1}^{P} \alpha_i^* y^i x^i \right) x + y^h - \left(\sum_{i=1}^{P} \alpha_i^* y^i x^i \right) x^h$$

$$f(x) = sign(w^{*T} x + b^*) = sign\left(\sum_{i=1}^{P} \alpha_i^* y^i x^{iT} x + y^h - \sum_{i=1}^{P} \alpha_i^* y^i x^{iT} x^h \right)$$

In the decision function only scalar products among samples appear.

The dual: kernel trick

N.B. NOI QUI USIAMO IL KERNEL NON LINEARE $K(x^h, x^k)$

$$\min \ \frac{1}{2} \sum_{h=1}^{P} \sum_{k=1}^{P} y^h y^k x^{hT} x^k \alpha_k \alpha_h - \sum_{h=1}^{P} \alpha_h$$

$$\sum_{i=1}^{P} \alpha_i y^i = 0$$

$$0 \le \alpha_i \le C$$

The Gram matrix: kernel trick

$Q = YX^T XY$

The $P \times P$ matrix $K = X^T X$ is the Gram or kernel matrix

where

$$X = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ x^1 & x^2 & \dots & x^P \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \qquad Y = \begin{pmatrix} y^1 & 0 & \dots & 0 \\ 0 & y^2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & y^P \end{pmatrix}$$

$K(x^h, x^k)$

$$\sum_{h=1}^{P} \sum_{k=1}^{P} y^h y^k x^{hT} x^k \alpha_k \alpha_h = \alpha^T Q \alpha$$

$K(x^1, x^1) \dots$

$$= \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_P \end{pmatrix}^T \begin{pmatrix} x^{1T} x^1 & y^1 y^2 x^{1T} x^2 & \dots & y^1 y^P x^{1T} x^P \\ y^2 y^1 x^{1T} x^2 & x^{2T} x^2 & \dots & y^2 y^P x^{2T} x^P \\ \vdots & \vdots & \dots & \vdots \\ y^P y^1 x^{1T} x^P & y^P y^2 x^{2T} x^P & \dots & x^{PT} x^P \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_P \end{pmatrix}$$

Question 1. (max score up to 23) Write a program to find the solution of the SVM dual quadratic problem. Perform a grid search with k-fold cross validation to set the values of the hyperparameters C and γ. For the solution to the SVM dual quadratic problem, you can use any routine that uses the gradient of the objective function (which is simply computable). Note that using a *specific* method for Quadratic Programming problems would be preferable and it may lead to higher grade.

Question 2. (up to "+4" points) Use the same kernel and hyperparameters of Question 1. Write a program which implements a decomposition method for the dual quadratic problem with any **even** value $\mathbf{q} \geq \mathbf{4}$. You must define the selection rule of the working set, construct the subproblem at each iteration and use a standard algorithm for its solution. You can use any routine that exploits the gradient of the objective function (which is simply computable). However, the use of the specific method for quadratic programming would be preferable and it may lead to higher grade. You must implement a stopping criterion in the outer optimization loop based on the optimality conditions.

Question 3. (up to "+3" points) Use the same kernel and hyperparameters of Question 1. Fix the dimension of the sub-problem to $q = 2$ and implement the most violating pair (MVP) decomposition method, which uses the analytic solution of the sub-problems. You must implement a stopping criterion in the outer optimization loop based on the optimality conditions.

Question 4. (up to "+1" points)
Consider a three-class problem with the classes of 1, 5, and 7. Implement an SVM strategy for multiclass classification (either one against one or one against all). You are free to use any method you find most suitable for solving the dual SVM problems.

Instructions for the report

In the report you must state:

- the chosen kernel;
- Only for Question 1: the final setting for the hyperparameter C (upper bound of the constraints of the dual problem) and of the hyperparameter of the kernel chosen; how you have chosen them and if you could identify values that highlight over/under fitting;
- Only for Question 1 and Question 2: which optimization routine you use for solving the quadratic minimization problem and the setting of its parameters, if any;
- For each Question:
  - machine learning performances: report the value of the accuracy on the training and test set; (for question 1, report also the value of the validation accuracy, if computed).

- optimization performance: report the initial and final value of the objective function of the dual problem, the number of iterations, the number of function evaluations, and the violation of the KKT condition, either as it is returned by the optimization routine used or evaluated by yourselves.

The comparison among all the implemented methods - in terms of accuracy in learning and computational effort in training - must be gathered into a final table (an example below)

	hyperparameters			ML performance		optimization performance			
question	C	γ	q	Training accuracy	Test accuracy	number its	number fun evals	KKT viol	cpu time
Q1									
Q2	As in Q1								
Q3	As in Q1								
Q4									

Instructions for python code

For each question $i = \{1,2,3,4\}$ you have to create a different folder named **Question_i_GroupName.py** where **you must provide the files according to the following instructions:**

- A file called **run_i_GroupName.py** (e.g. run_1_Example.py,). This file will be the only one executed in the phase of verification of the work done. It can include all the classes, and import all the functions and libraries you used for solving the specific question i, but it has to print ONLY:
  - Setting values of the hyperparameters
  - classification rate on the training set (% instances correctly classified)
  - classification rate on the test set (% instances correctly classified)
  - The confusion matrix
  - Time necessary for the optimization
  - number of optimization iterations
  - difference between $m(\alpha)$ and M(α)
  - the final value of the Dual SVM objective function $f(\alpha^*)$
  - solver status (e.g. optimal, infeasible etc..)

  Furthermore, in Question 2 you must print also:

  - value of q

  Furthermore, in Question 3 you must print also:

– Number of columns of the matrix in the quadratic objective function (i.e. Q) computed to solve the dual SVM.

Furthermore, in Question 4 you must print also:

– if you used OAA or OAO (i.e., the multiclass strategy that you implemented)

- A file called **functions_i_GroupName.py** with the complete code you have written that includes all the functions that are called from the **run_i_GroupName.py file** and all the procedures that have been used to select the hyperparameters.

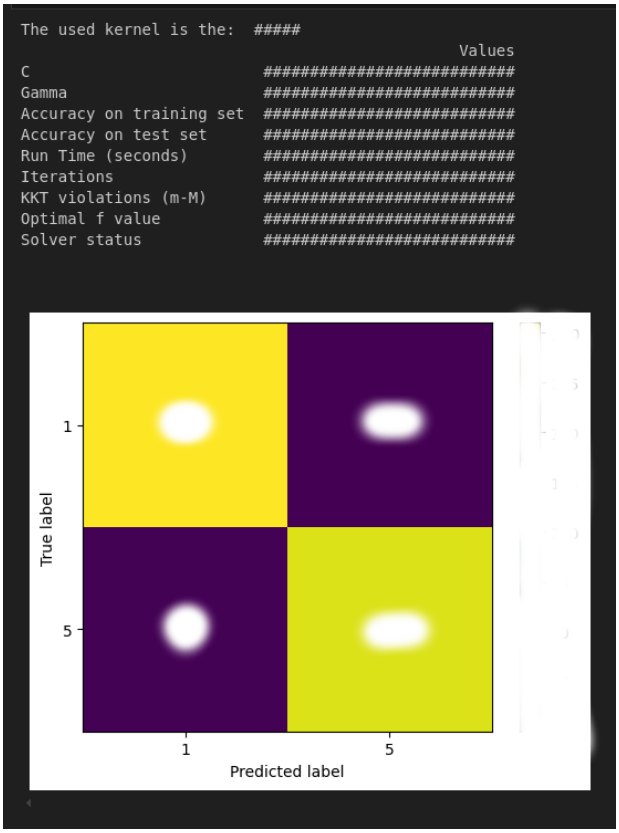

Figure 1: Example of the expected output of file run_i_GroupName.py

Note that:

1. Each question will be graded averaging the Performance score (25%), the Efficiency score (25%) and the Mathematical Correctness score (50%).

2. A test error on a blind test set might be evaluated to check the performance of the models.

3. There is no competition with other students. Your grade does not depend on other teams' performances.

4. You are warmly recommended to write clean and easily readable code. Not respecting the general assignment (for example, printing different values or not printing the requested outputs) will result in a penalization. Printing the progress of the algorithm or similar mistakes will slow down the code correction and result in a score decrease. Additionally, please make sure to comment your code sufficiently, so that an external reader can understand what each part of the code does.

5. Your code will not be debugged and will not be modified **in ANY case** after the submission. **If the code does not run, the resulting score will be zero**. You can schedule a meeting with teacher and/or teaching assistant to get some help **BEFORE the submission, not after**. You are warmly recommended to check multiple times and on different computers that your code is executable.

6. It is welcomed to take inspiration from other students, as well as from open-source contents online, but **"copy and paste" strategies will invalidate your project**. Be aware that **your code will be cross-checked with online materials, as well as with other students' code and with AI-generated code**.

7. The use of any library automatically building ML models (e.g, Pytorch, TensorFlow, sklearn.neuralnetwork etc.) is **FORBIDDEN** and automatically invalidates the project.