University of Rome "La Sapienza"

Optimization Methods for Machine Learning

Project 2

Malick Alexandre Ngorovitch Sarr, Vigèr Durand Azimedem Tsafack

December 23, 2018

INTRODUCTION AND SET UP

The goal of this assignment was to implement optimization methods for training Support Vector Machines (supervised learning) applied to classication problems. 2 Data sets were provided and we were asked to choose between them based on some degree of difficulty. We have chosen the second data set as it was more challenging.

The dataset itself represents the Classication of normalized handwritten digits, automatically scanned from envelopes by the U.S. Postal Service. We were provided only the test and train data for the digit 2 and 8. We then concatenated those two data set into one and applied the following transformations:

- Assigning 1 as output for the digits provided as 2: $y = 1$,
- Assigning -1 as output for the digits provided as 8: $y = -1$

This dataset will be used to train three different types of Support Vector Machines Classification algorithm. We will first use the full QP method to solve the dual quadratic SVM problem, We will then use the decomposition method and we will finally close by implementing a Most Violating Pair (MVP) decomposition method solved using analytic solution instead of using a standard optimization routine. Additionally as a bonus, we will implement a Support Vector Machine classifier for multi-class classification introducing one more digit: 1.

This project was made with Python along few of its libraries: numpy, Scipy, Pandas, Cvxopt and Sklearn

1 Full QP

In order to implement the full QP, we consider an SVM with kernel $k(,)$ and implement the non linear decision function as

$$y(x) = sign(\sum_{p=0}^{P} \alpha_p y^p k(x^p, x) + b) \tag{1.1}$$

where α, b are derived from the optimal solution of the dual non linear SVM problem.

The right hyper parameter were choosen after running multiple simulations (Grid Search) using both polynomial and RBF kernels. We tested the polynomial kernel with a range of values with C and p from $0, ..50$ with a step size of 1, and the γ hyperparameter for our RBF kernel with the range of values 0...1 with a step size of 0.05. Our best results were obtained by the RBF kelnel and is defined as:

$$k(x, y) = e^{-\gamma||x-y||^2} \tag{1.2}$$

We will be using the RBF kernel for the remaining sections of this project. The best value of the test error were obtained with the hyper-parameters $C = 1$ and $\gamma = 0.1$.

With those values, we defined an error function

$$E(\hat{y}) = \frac{CorrectPredictions}{NumberInstances} \tag{1.3}$$

Using the above function and those parameters we are able to obtain a classification rate on training of 100% and a Classification rate on test of 97,53%. The output of the grid search procedure showed that as we increase the value of the hyperparameter γ we quickly manage to reach an optimum point at 0,1. This means that we should be able to perfectly fit a plane $w^T x + b$ that would correctly classify all the points of the testing data set.

The value C is instead used to accesses the training error within the primal problem. Indeed, an increase in C make the penalty term $C \sum \epsilon_i$ having more impact in the primal objective. This means that with low values of C, we increase the importance the function caused by wrongly classified points has. And with higher values of C, we increase the penalty given to misclassified points. The consequences of that relation means that theoretically, as we increase C, we will find an overfitting trends the boundary will be able to classified correctly as many points as possible, including outliers, and ergo decreasing its generalization capacity of unseen data.

Once we got the dual quadratic problem function and its constrains, we used the python package CVXOPT to solve it using the function

$$min\Gamma(\lambda) = -\frac{1}{2} Y * Y.TK(X, X.T) * \lambda * \lambda.T - 1.T * \lambda \tag{1.4}$$

subject to $Y.T * \lambda = 0$, $\lambda >= 0$ and $\lambda <= C$

Finding a solution to the above problem allowed us to compute the bias. A small threshold was chosen to find the support vectors (corresponding to non-zero Lagrange multipliers, by complementary slackness condition). From that we computed our prediction and obtained the following results.

- Optimization routine chosen: cvxopt
- Chosen kernel: RBF
- Classification rate on the training set: 100.0
- Classifcation rate on the test set: 97.52747252747253
- Time for finding the KKT point (in seconds): 16.19692301750183
- Number of optimization iterations: 5
- Value of the RBF kernel hyper parameter: 0.1
- Value of C: 1

2 Decomposed QP

In the decomposed QP SVM with the dual problem q=2, we solve a single quadratic sub-problems using the CVXOPT optimization routine similarly to the previous problem. First we selected $i \in R(\alpha_k)$ and $j \in S(\alpha_k)$ such that $\nabla(f(\alpha_k)).T * d_i, j < 0$; d_i, j is the descent direction and $\nabla(f(\alpha_k)).T$ is the transposition of the gradient vector. Then we computed with CVXOPT a solution to our QP $\alpha^* = (\alpha_i^*, \alpha_j^*).T$ with the help of the matrices Q, X, Y, α, W, N, n. We computed $\alpha_k + 1$, updated the gradient before checking the convergence. If the problem did not converge, we repeat the above steps until it converges. In here, it is interesting to note that almost all those variables had to be recomputed given the current working pair W_k, and as opposed of having the dimension given by the length of the training data, the dimensions were set at $q = 2$. The working pair W_k was found, as described before by computing the gradient for the overall dual problem in the current variable α_k.

We obtained the following results:

- Optimization routine chosen: cvxopt
- Chosen kernel: RBF
- Difference m(a)-M(a): 489
- Classification rate on the training set: 57.42340926944226
- Classification rate on the test set: 54.395604395604394
- Time for finding the KKT point (in seconds): 1,620114456176758
- Number of optimization iterations: 691
- Value of the RBF kernel hyper parameter: 0.1
- Value of C: 1

Using GridsSarch, we managed to get a classification rate up to 92% but because we were tasked to use the same param as the fullQP we had a low accuracy with our method. One think to note is the speed of convergence opposed to the first one. Indeed, it's much faster to reach optimality using the decomposed QP as opposed to the full one.

3 MVP Method

The idea behind the Most Violating Pair(MVP) is to find among indices $(i, j) \in R(\alpha^k)XS(\alpha^k)$ which define, among feasible direction with two non zero components, the steepest descent such that:

$$min_{st(\alpha)XS(\alpha^k)} \nabla f(\alpha^k)^T d^{st} \tag{3.1}$$

In order to find a analytic solution, we used the procedure defined in 5.A.1 of the teaching material [1, chapter 5.A.1, p. 116]with the rest of the set up being similar to that of section 2.

The following result were obtained

- Chosen kernel: RBF
- Difference m(a)-M(a): 748.0
- Classification rate on the training set: 57.580518460329934
- Classification rate on the test set: 54.395604395604394
- Time for finding the KKT point (in seconds): 1.7183501720428467

- Number of optimization iterations: 100
- Value of the RBF kernel hyper parameter: 0.1
- Value of C: 1

We observe here the same level of performance in with the decomposed QP. However, one think to note is the fact that we achieved optimality in much less iterations with the MVP.

4 Multi-class classication

The idea behind the multiclass SVM classificator is to select top ranked classes from a group of classes. This ranking can be thought of geometrically as the distances from the J linear separator. Thus the points that are near a class's separator are more likely to be misclassified, ergo the higher the distance from that separator, the higher there is a chance that a positive classification decision is correct. [2]

The training of the multiclass classifier was done with the following steps

1. Build a classifier for each class, where the training set consists of the set of documents in the class (positive labels) and its complement (negative labels).
2. Given the test document, apply each classifier separately.
3. Assign the document to the class with the maximum score,

The results of the multi class classificator are summarized below.

- Optimization routine chosen: cvxopt
- Chosen kernel: RBF
- Classification rate on the training set: 100.0
- Classification rate on the test set: 97.45222929936305
- Time for finding the KKT point (in seconds): 208.9875156879425
- Number of optimization iterations: 33
- Value of the RBF kernel hyper parameter: 0.1
- Value of C: 1

We have chosen the same values as in the previous section and observed a high value for both the training and testing data.

results

Question	Opt. Method	C	γ	Training Error (in %)	Test Error (in %)	Time
1	Full QP	1	0.1	100	97.53	16.59s
2	Decomposed QP	1	0.1	57.42	54.39	1.62s
3	MVP	1	0.1	54.39	54.39	1.62s
4	Mult. Var.	1	0.1	100	97.45	208.67s

Table 5.1: Results presented in the above section

5 Annexes

References

[1] Luigi Grippo, Laura Palagi, Marco Sciandrone *LaTeX:Optimization Methods for Machine Learning*, Teaching Notes 2016-17 2018.

[2] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *LaTeX:Introduction to Information Retrieval*, Cambridge University Press. 2008.