Homework Optimization for Data Science

Capurso Graziana Di Labbio Daniela Garbin Agata Schiavo Leonardo

May 15, 2023

Abstract

In the contex of our supervised learning problem, we are faced with a situation where we posses a substantial amount of data, the majority of which is unlabeled. However, we do have a limited number of labeled examples. Our ultimate aim is to employ an optimization algorithm to effectively assign the appropriate labels to all the data points in the dataset.

1 Introduction

To investigate the classification of points between two sets, we conducted a simulation by generating two clouds of points on a plane. From these points, we selected a subset to create a labeled set. We then employed various optimization methods, including Gradient Descent, Randomized Block Coordinate Gradient Descent (BCGD), Gauss-Southwell BCGD method and Cyclic BCGD with blocks of dimension 1. The objective was to accurately classify all the points between the two sets and assess the accuracy of the implemented methods. Following the initial section, we proceeded to evaluate the performance of these methods on some real dataset.

2 Similarity Measure and Loss Function

Given $x, y \in \mathbb{R}^n$ we defined the similarity measure between x and y as

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

Here, γ is a hyperparameter that is proportional to the inverse of the square of the variance of our kernel. We set γ to one for both measuring the weight between unlabeled points. This similarity measure is an exponential decreasing function, where the similarity approaches 0 as the distance between x and y increases, and approaches 1 as the distance decreases. It is bounded everywhere, even between overlapping points, unlike the reciprocal of the Euclidean distance. To represent the similarity weights between points, we generated a symmetric weighted matrix W where each entry represent the weight between two points. The loss function $L(y)$ is defined as follows:

$$\sum_{i=1}^{l}\sum_{j=1}^{u} w_{ij}\,(y_i - \bar{y_i})^2 + \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{u} \bar{w_{ij}}\,(y_i - y_j)^2$$

Here, y is the vector of predicted labels, $\bar{y}$ contains the existing labels, w_{ij} represents the weights between unlabeled points. and $\bar{w_{ij}}$ represents the weights between labeled points. The goal is to minimize the loss function, i.e.,to find:

$$\min_{y \in \mathbb{R}^n} L(y)$$

To achieve this, we focused on first-order gradient methods, taking advantage of the smoothness of the objective function. We used a fixed step size method, empirically choosing the step as $\alpha = 0.0001$. Additionally, we implemented the Armijo Rule, Exact line search and the Lipschitz constant rule. Although these step size methods provided better accuracy in fewer epochs, they were computationally expensive. Sometimes we computed the loss function every several epochs to reduce computational overhead, as computing the loss function for

every single gradient update was too computationally heavy. Overall, our focus was on optimizing the loss function using different first-order methods, step size, and computational strategies to improve accuracy and efficiency.

3 Generated Dataset

The dataset we generated consists of n clouds of two-dimensional points in the same plane. Each could contains k points randomly generated from a bivariate normal distribution. The clouds are positioned at the centroids of points on the unit circle. We can generate the dataset with various options: n specifies the number of clouds; k specifies the number of points in each cloud; *cov* specifies the covariance matrix of the normal distribution. If needed the dataset is splitted into labeled and unlabeled data with a given percentage of labelled known data. In the provided example the dataset is created with 2 clouds, each containing 5000 points with only 150 known labels per cloud.

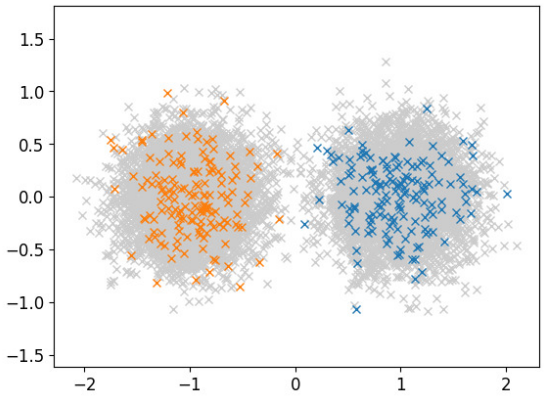

Figure 1: Generated Dataset
Size of labeled points: (300, 2)
Size of unlabeled points: (9700, 2)

We have used six approaches for the optimization process. For the Fixed and Gradient Descent; Fixed and Cyclic; Armijo and Gradient Descent; Exact and Gradient Descent; Gauss Southwell BCGD, we have provided results for different number of epochs,the goal is to minimize the loss. During each epoch, the value of the loss function and the accuracy of the model are reported. It can be observed that the value of the loss function decreases significantly from the early epochs until the tenth epoch, indicating an improvement in the model's performance in reducing the error during training. At the same time, the accuracy of the model seems to increase notably, approaching 100%. These trends suggest that the models are progressively improving and learning to make more accurate predictions as the training progresses. A decreasing loss function is generally a positive indication, as it signifies that the model is becoming better at minimizing the prediction errors. Furthermore, the increasing accuracy of the model indicates that it is making correct predictions for a large portion of the training data. However, it is important to note that the training accuracy is not always a reliable indicator of the model's performance on unseen data (such as test or validation data). Instead, for the approach that involves Fixed and Random methods, from the results, it can be observed that the loss does not significantly decrease over the course of the training epochs. The values remain relatively high, suggesting that the optimization process, utilizing the Fixed and Random methods, is not effectively reducing the loss.This could indicate that the model is struggling to learn and make accurate predictions on the training data.

4 Fraud dataset

The first dataset consists of data on fake credit cards. We had to considerably reduce the number of examples of non-fake cards to make the two classes balanced. Considering as usual a 15% labeled and the rest unlabeled data, we proceeded with a feature importance analysis via pca arriving at 11 different features that explain the 85% of the total veriance. On this preprocessed dataset we finally tested the methods exposed above.

4.1 Gradient Descent Fixed Stepsize

From the results, it can be observed that the loss slightly decreases over the course of the training epochs. The loss values remain relatively close, indicating that the optimization process is making incremental improvements in reducing the loss function. The reported accuracy remains relatively stable throughout the epochs, hovering around $91 - 92\%$. This suggests that the model is consistently making accurate predictions on the training data. Furthermore, the convergence or performance of the model might be influenced by various factors such as the complexity of the problem, quality and quantity of the training data, choice of optimization algorithm, and hyperparameter tuning.

4.2 Random Gradient Descent Fixed Stepsize

We have done the training for a large number of epochs. In the initial epochs, both optimization methods show similar performance with low accuracy values. However, as the training progresses, the accuracy improves for both methods. It's important to note that the loss value decreases over time, indicating that the model is learning and making progress in minimizing the error between predicted and actual values. However, the loss values remain high throughout the training process.

4.3 Armijo and Gradient Descent, Gradient Descent-Fixed stepsize, Gauss-Southwell

In the these last three methods, the accuracy tends to decrease after the first epoch.

5 Swarm dataset

The second dataset concerns the behavior of swarms. The class labels are binary, so the first refers to flocking, clustering, and alignment, while the other refers to non-flocking, non-clustering, and nonalignment. Also in this dataset we had to resort to PCA by reducing from a very high initial number of features down to 8 which explain a variance of 39%. When comparing the accuracy results, we found that the different optimization methods performed at a comparable level. They were able to achieve similar levels of accuracy in correctly classifying the data points within the dataset. This indicates that the methods were effective in capturing the underlying patterns and making accurate predictions. All plots for these dataset can be found in the notebook.

6 Conclusions

In this report, we have chosen to evaluate various optimization methods.
Initially, we applied these methods to a dataset generated by ourselves, and subsequently, they were tested on three different datasets. The results obtained were quite interesting, with some cases even achieving an accuracy of 80% or higher within a reasonable time frame. By conducting these evaluations, we aimed to assess the performance of different optimization methods and their suitability for the given datasets. We carefully analyzed the accuracy achieved by each method and compared the computational time required for convergence. This allowed us to gain insights into the effectiveness and efficiency of the optimization techniques employed.

7 Plots

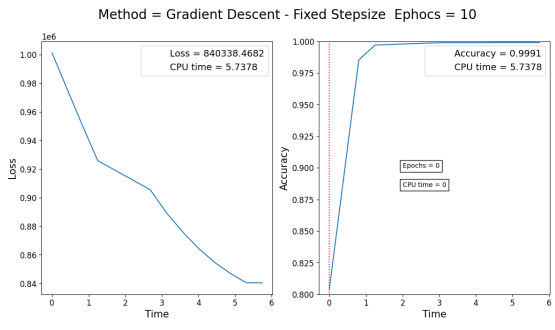

Figure 2: Gradient Descent

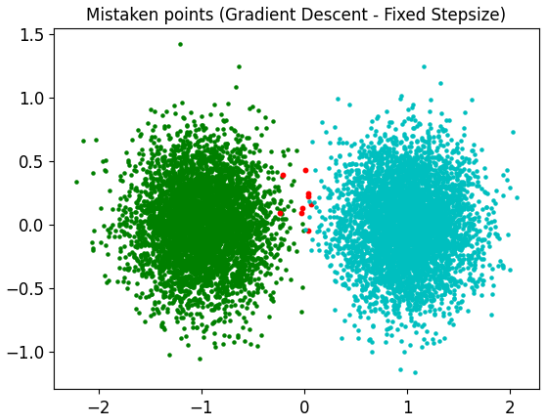

Figure 3: Mistaken points(Gradient Descent - Fixed Stepsize)

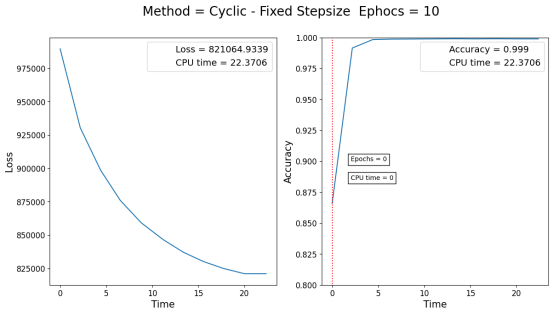

Figure 4: Cyclic Method

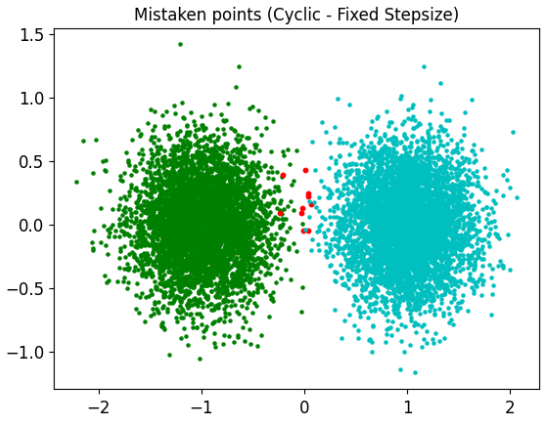

Figure 5: Cyclic Method mistaken points

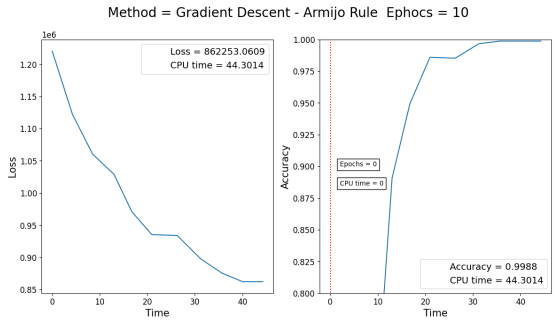

Figure 6: Gradient Descent - Armijo Rule

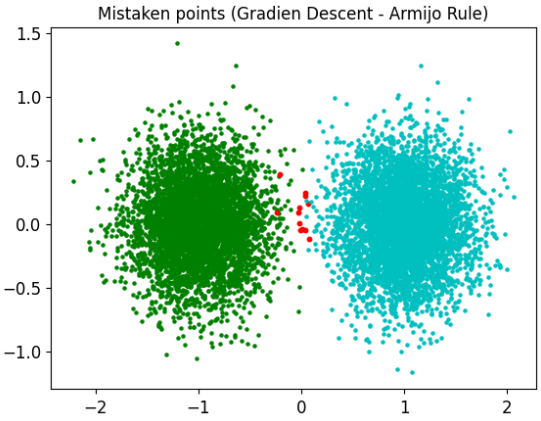

Figure 7: Mistaken points(Gradient Gescent - Armijo Rule)

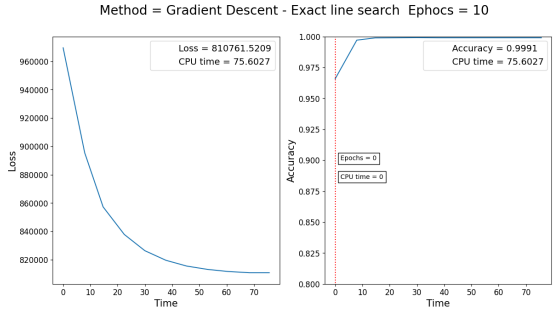

Figure 8: Gradient Descent - Exact line search

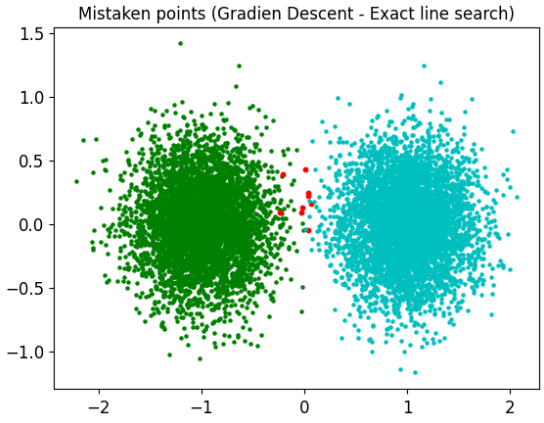

Figure 9: Mistaken points (Gradient Descent - Exact line search)

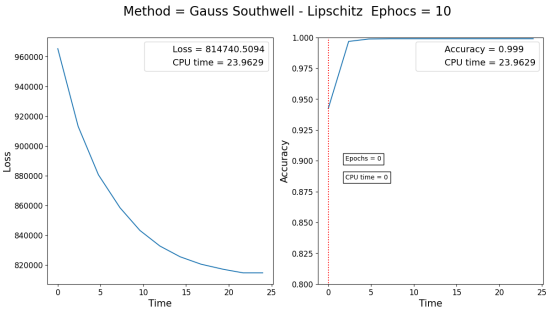

Figure 10: Gauss - Southwell

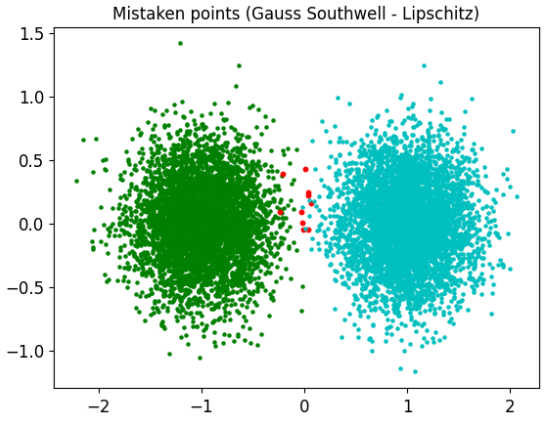

Figure 11: Mistaken point (Gauss Southwell - Exact line search)

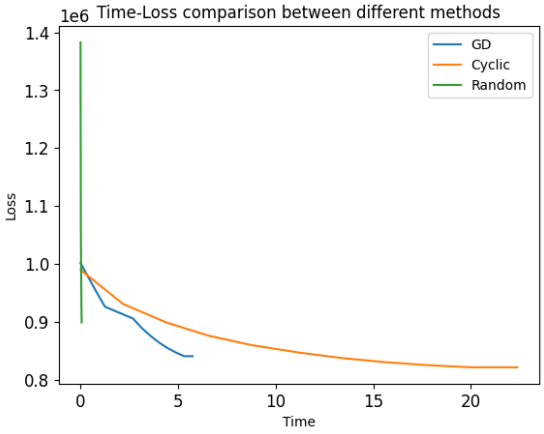

Figure 12: Time - Loss comparison between different methods

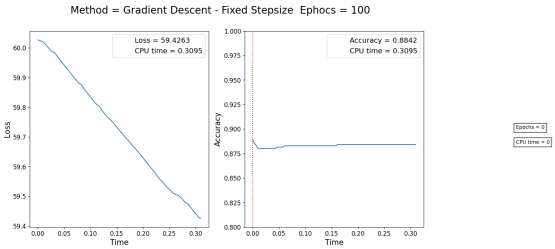

Figure 13: Gradient Descent (Fraud)

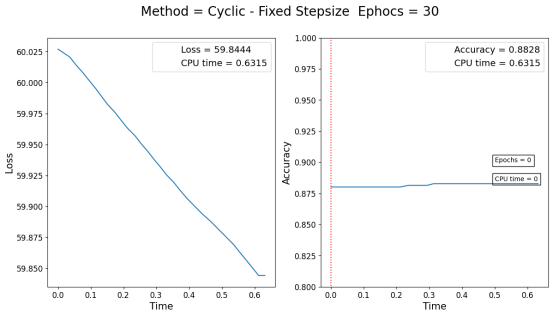

Figure 14: Cyclic Method (Fraud)

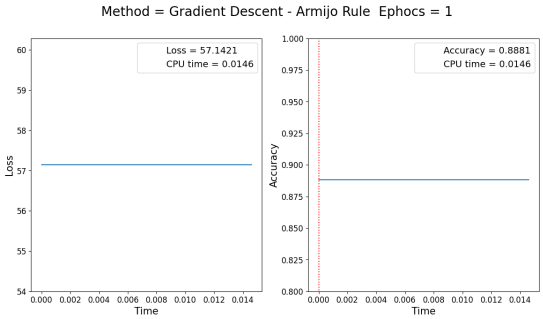

Figure 15: Gradient Descent - Armijo Rule (Fraud)

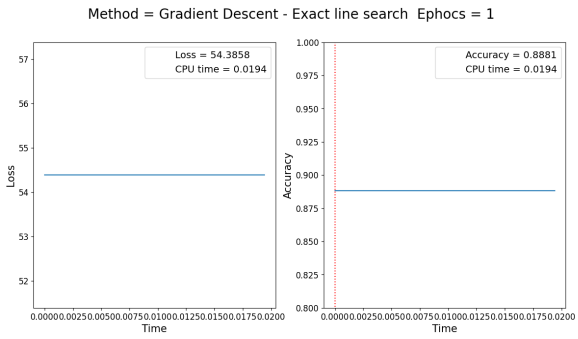

Figure 16: Gradient Descent - Exact line search (Fraud)

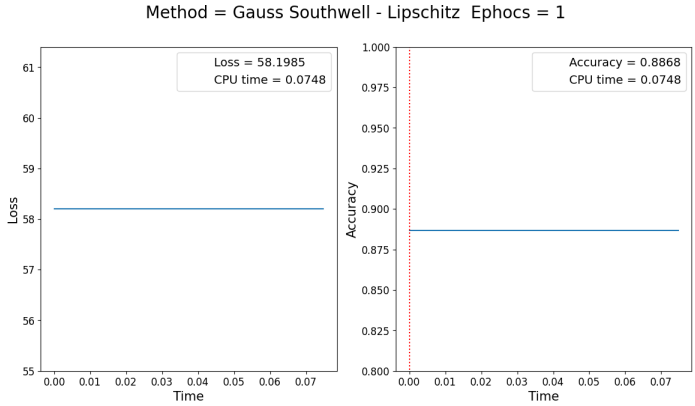

Figure 17: Gauss - Southwell (Fraud)

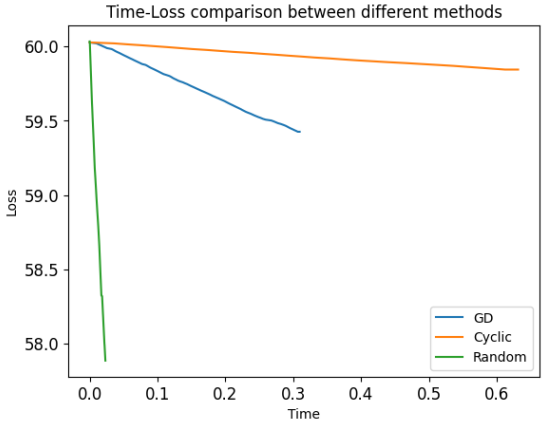

Figure 18: Time - Loss comparison between different methods (Fraud)