Optimization For Data Science Project Report

Anomaly Detection

Nora Nikoloska
nora.nikoloska@studenti.unipd.it

Sara Kartalovic
sara.kartalovic@studenti.unipd.it

June 21, 2021

1 Introduction

In this project two algorithms for the problem of minimum enclosing ball (MEB) on the task of anomaly detection are implemented and analyzed. Both algorithms are tested on a generated dataset with random values as well as two other publicly available datasets. The datasets analyzed are *Daphnet Freezing of Gait Data Set* and *EEG Eye State Data Set*. The first dataset contains the recordings collected from 3 different sensors that were used by people suffering from Parkinson's disease. On the other hand, the second dataset consists of chronologically ordered time-series data collected from *Emotiv EEG Neuroheadset* machine with purpose to distinguish the abnormalities in brain waves. The data is preprocessed and normalized before it is used for analysis. The report includes the description of the algorithms and datasets and shows convergence plots and accuracy results obtained from different experiments.

2 Algorithms description

In the referenced paper[1], two algorithms are proposed for the minimum enclosing ball (MEB) problem. The MEB problem in the Frank-Wolfe dual formulation can be seen as a greedy algorithm [2]. The optimization formulation of the problem is the minimization of the radius among the points in a given set A. The minimization problem as a Lagrangian dual is formulated as:

$$min_u \qquad \Phi(u) := \sum_{i=1}^{m} u_i (a^i)^T a^i - (\sum_{i=1}^{m} u_i a^i)^T (\sum_{i=1}^{m} u_i a^i)$$

such that:

$$\sum_{i=1}^{m} u_i = 1, \quad u_i >= 0, \quad i = 1, ..., m$$

The first algorithm starts with the selection of the point the furthest away from the first point in the set as alpha and the point the furthest away from

alpha as beta. These points are added in the core set. The vector of zeros u is generated with values of 0.5 for the alpha and beta points only. The center is calculated using the vector u and the objective function is calculated. The point that is the furthest away from the center is selected as k and the δ value is calculated. The algorithm loops until the δ value reaches the given threshold. In each iteration, the point k is added to the core set, the center is moved closer to the point k and the new furthest point is calculated. In the final iteration k, the ball radius is given as:

$$r = \sqrt{(1 + \delta_k)\gamma_k}$$

where

$$\gamma_k = \Phi(u^k)$$

The second algorithm is similar to the first, but other than the k point, it also calculates the closest point to the center as ξ. Two δ values: $\delta+$ and $\delta-$ are calculated using the k and ξ points respectively. The final δ value is the maximum between the two. When the δ value is larger than $\delta-$ the algorithm is identical to the first one. In the other case, it is possible for the ξ point to be removed from the core set and the center is moved away from the ξ making the so called *"away"* steps. Both algorithms achieve asymptotic complexity bound of $O(mn/\epsilon)$ where m is the number of samples, n the number of features and ϵ the threshold level.

3 Datasets description

Both algorithms are tested on two different datasets and the results are compared. The datasets description and preprocessing details are explained in this section.

3.1 Daphnet Freezing of Gait Data Set

The first dataset that is used is publicly available under name *Daphnet Freezing of Gait Data Set* [3]. It contains the annotated readings of 3 acceleration sensors. The sensors were located on the hip and legs of patients that suffer from Parkinson's disease. Specifically, the recordings were collected from patients that experience freezing of gait (FoG) during walking tasks. The sensors were prioritized to to recognize gait freeze while the emphasis of the data was generating many freeze events .

The dataset recordings were run at the Tel Aviv Sourasky Medical Center in 2008 as an outcome of collaboration between:

1. Laboratory for Gait and Neurodynamics, Tel Aviv
2. Sourasky Medical Center, Israel
3. Wearable Computing Laboratory, ETH Zurich, Switzerland

The dataset is composed of 237 instances and 10 features. Description of attributes is as follows:

1. Time of sample in millisecond
2. Ankle (shank) acceleration - horizontal forward acceleration [mg]
3. Ankle (shank) acceleration - vertical [mg]
4. Ankle (shank) acceleration - horizontal lateral [mg]
5. Upper leg (thigh) acceleration - horizontal forward acceleration [mg]
6. Upper leg (thigh) acceleration - vertical [mg]
7. Upper leg (thigh) acceleration - horizontal lateral [mg]
8. Trunk acceleration - horizontal forward acceleration [mg]
9. Trunk acceleration - vertical [mg]
10. Trunk acceleration - horizontal lateral [mg]
11. Annotation (Class)
    - 0: not part of the experiment. For instance the sensors are installed on the user or the user is performing activities unrelated to the experimental protocol, such as debriefing
    - 1: experiment, no freeze (can be any of stand, walk, turn)
    - 2: freeze

The first feature, 'Time of sample in millisecond' is not used because the class of the sample does not depend on it. All values annotated with zero are removed since they are not part of the experiment. Additionally, the set for all anomalies is set to '1', and '0' otherwise.

3.2 EEG Eye State

The second data used is publicly available as *EEG Eye State Data Set* [4]. It consists of 14980 instances where each instance provide information about 14 EEG values and a value indicating the eye state. The EEG test is performed in order to detect the abnormalities in brain waves, or in the electrical activity of brain. The mechanism used was 'Emotiv EEG Neuroheadset' which recorded one continuous measurement lasting 117 seconds. Also, camera is used to detect the state of the eye during the EEG examination. After further detailed analysis, the information provided by camera is manually added to the dataset. The eye state can take values '0' or '1' where '0' indicates the eye-open state and '1' the eye-closed state. At the top of the data is the first measure followed by other values in chronological order. The 14 features are time-series which means that features all together are a sequence that is collected at different points in time and ordered chronologically.

3.3 Datasets preprocessing

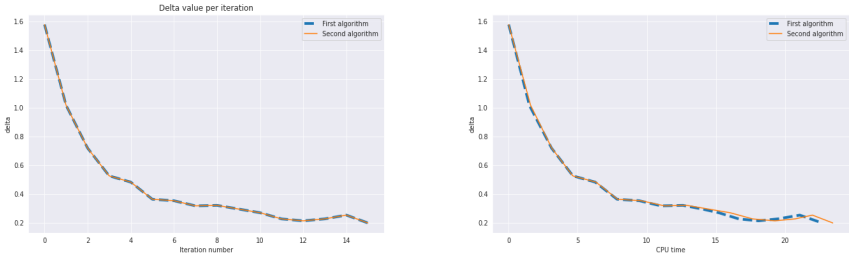
Firstly, the data is divided into training and test sets where 90% of the data is allocated for training purposes and the other 10% is designated for testing. The data is normalized by using min-max normalization where minimum value becomes 0 and maximum value becomes 1. This way the data is scaled between 0 and 1. The enclosing ball is trained on the points with class '0' which are not anomalies. Next, the enclosing ball is tested by using the data that has class '1' from the training set as well as the data that has class '0' from the testing set. The measured accuracy calculates the percentage of samples of class '1' outside of the enclosing ball and samples of class '0' inside the ball.

4 Convergence Results

Both algorithms are tested on the previously introduced datasets as well as on a generated dataset. The convergence results are depicted as plots for the delta value and core set size over the number of iterations or CPU time.

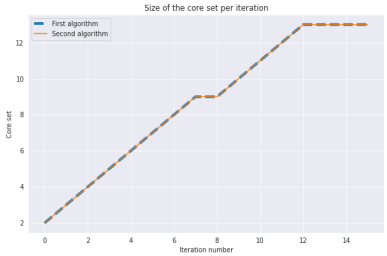
4.1 Generated Dataset

A random easily separable dataset with 7000 training and 3000 anomaly samples for testing was generated. Each subset consists of 10 features with values in the ranges from 0 to 0.7 for the training samples and 0.7 to 1 for the anomaly samples. The accuracy values obtained with both algorithms are 100% for both the samples inside the enclosing ball and the detected anomalies outside of the minimum enclosing ball. Figure 1 shows the convergence plots for the random dataset for both algorithms. It can be seen that the delta value convergence is identical for both algorithms with minimal differences in the CPU time. The size of the core set is also increasing identically, with the minor difference of faster convergence of the first algorithm in terms of CPU time.

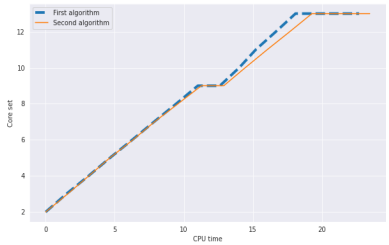


(a) Delta per iteration

(b) Delta per CPU time



(c) Core set size per iteration

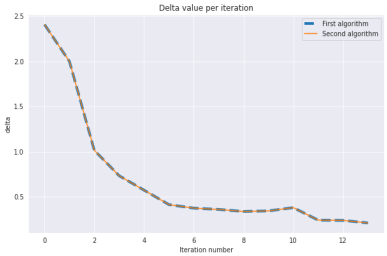


(d) Core set size per CPU time

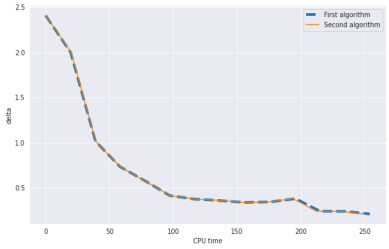
Figure 1: Convergence plots for the generated dataset

4.2 First Dataset

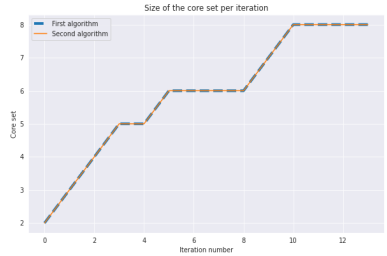
Figure 2 shows the convergence plots for the first dataset. Here the convergence times between the two algorithms are identical both in terms of number of iteration and CPU time. The threshold used for these plots is 0.1.



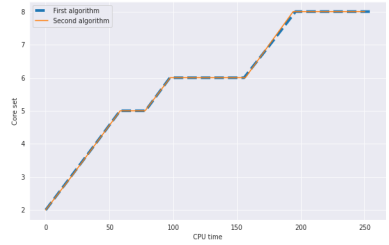
(a) Delta per iteration



(b) Delta per CPU time



(c) Core set size per iteration

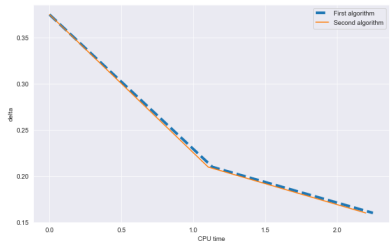


(d) Core set size per CPU time

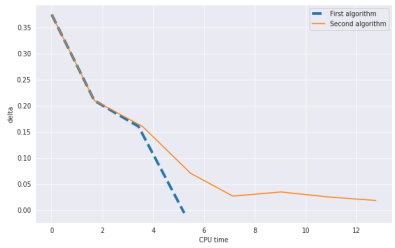
Figure 2: Convergence plots for the first dataset

4.3 Second Dataset

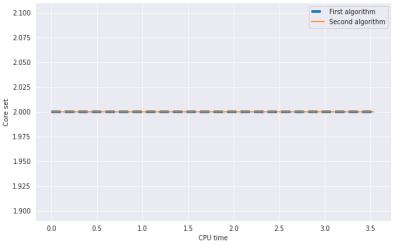
The convergence results for the second dataset are shown on Figure 3. The delta values and core set size are plotted against CPU time to detect the difference between the algorithms. It can be seen that for the threshold of 0.1 both algorithms behave identically and converge in few iterations without adding points to the core set apart from the starting points alpha and beta. However, for the threshold of 0.01 the first algorithm converges faster. With this threshold, another point is added to the core set and the size of the core set is 3.



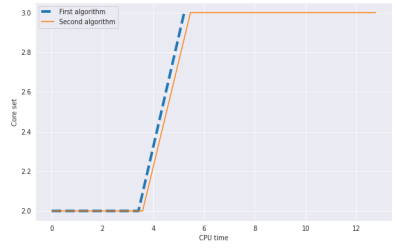
(a) Delta for e = 0.1



(b) Delta for e = 0.01



(c) Core set size for e = 0.1



(d) Core set size for e = 0.01

Figure 3: Convergence plots for the second dataset with different thresholds

5 Experiments

Several experiments with different values of e were conducted for both algorithms. The accuracy values are reported in accuracy tables for each dataset. The detection accuracy is the percentage of anomalies detected from the training set - points from the set outside of the ball. The in-ball accuracy is the percentage of training points inside the ball and calculates how well the algorithms encapsulates all training samples. The test in-ball accuracy shows if the algorithms correctly classifies new test examples that should be placed inside the encapsulating ball. In addition, the core set size is added for each of the experiments.

5.1 First Algorithm

The accuracy values for the first algorithm are depicted on Table 1 for the first and Table 2 for the second dataset respectively. On Table 1 it can be seen that the detection accuracy is very poor even for the threshold of 0.1. On the other hand, the in-ball accuracy and the test in-ball accuracy are very high which suggests that the algorithm manages to encapsulate all training samples. However, presumably because of the inseparability of classes, the anomalies are not detected outside of the enclosing ball. A slightly lower threshold of 0.08 does not improve the detection accuracy, but does result in additional point in the core set. Lower thresholds can not be reached and the algorithm does not converge, with the delta value infinitely looping in a plateau of several close values.

e	Detection Accuracy	In Ball Accuracy	Test In Ball	Core Set Size
0.1	0.10 %	100 %	99.94%	8
0.08	0.04 %	100 %	100 %	9
0.01	does not converge			

Table 1: Accuracy experiments for the first dataset

Table 2 shows the experiments with the second dataset. Here the anomaly detection accuracy is always 100% and it is achieved in few iterations. This is possibly attributed to the separability of the dataset. The number of points correctly enclosed within the ball is also high. The testing accuracy for the new points is also high and increases slightly with a lower threshold. The core set includes another point apart from the first two alpha and beta points in lower threshold values.

e	Detection Accuracy	In Ball Accuracy	Test In Ball	Core Set Size
0.1	100%	99.95%	93,93%	2
0.01	100%	99.95%	94.30%	3
0.001	100%	99.95%	94.30%	3

Table 2: Accuracy experiments for the second dataset

5.2 Second Algorithm

The same experiments are run on the second algorithm for comparison of the accuracy. The results are almost identical, reported on Table 3 for the first and Table 4 for the second dataset respectively. The difference between the algorithms on the first dataset is shown only in the detection accuracy where the

second algorithm performs slightly worse. Otherwise, they perform identically and the algorithm does not converge for low threshold values.

e	Detection Accuracy	In Ball Accuracy	Test In Ball	Core Set Size
0.1	0.09%	100%	99.94%	8
0.08	0.03%	100%	100%	9
0.01	does not converge			

Table 3: Accuracy experiments for the first dataset

For the second dataset the results are again identical, with slight difference in the testing accuracy of the new points to be added in the encapsulating ball. Here, the accuracy with the second dataset is slightly worse when using thresholds of 0.01 and 0.001.

e	Detection Accuracy	In Ball Accuracy	Test In Ball	Core Set Size
0.1	100%	99.95%	93.93%	2
0.01	100%	99.95%	93.93%	3
0.001	100%	99.95%	94.06%	3

Table 4: Accuracy experiments for the second dataset

6 Conclusion

From the results it can be seen that the algorithms performance is highly dependent of the structure of the dataset in use. For the randomly generated dataset with separable features, both algorithms reach perfect accuracy both for the points inside and outside of the minimum enclosing ball. However, for the *Freezing of Gait Dataset* the anomaly detection accuracy is very low while all of the training points are correctly enclosed in the ball. The failure of convergence at lower thresholds for this dataset further suggests that the radius of the ball can not be reduced enough in order for the anomalies to be outside of it. The structure of the second dataset, *EEG Eye State* allows for easy generation of a minimum enclosing ball in few iterations with high accuracy for both anomaly detection and training points encapsulation. In terms of CPU time, both algorithms converge almost in the same time, with the exception of slightly faster convergence of the first algorithm on the second dataset.

References

[1] E. Alper Yıldırım: Two Algorithms for the Minimum Enclosing Ball Problem

[2] L. Clarkson Kenneth: Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm

[3] Daphnet Freezing of Gait Data Set

[4] EEG Eye State Data Set