DMW Assignment-7

Submitted By - [Akhil Shukla, IIT2018112] [Akhil Singh, IIT2018198][Javed Ali, IIT2018501][Manan Bajaj, IIT2018502][Lokesh, IIT2018503]

6 th Semester, B.Tech, Department of Information Technology, IIIT Allahabad

We have to understand the algorithm proposed in the paper "One-Class Slab Support Vector Machine" and implement it in python.

We have implemented the SMO algorithm as described in paper[2] to solve the quadratic programming problem in the given paper. The implementation is then tested on randomly created 2D binary classification data created using sklearn's dataset module.

The implementation is the same as discussed in paper[2] with little adjustment influenced by paper[3].

- The update on gammaB will be counted only when the update is above the *minChange* percent which by default is set to 0.1%. So for particular GammaB our code will keep on searching for GammaA in the order of heuristic mentioned in paper[2] till the MinChange doesn't occur in Gammas.
- There could be the case that none of the values for GammaB will result in proper change in Gammas; the case will be counted as pass. In such cases since the gammas are exactly the same as last iteration, searching the gammaB using it's heuristic as discussed in paper[2] will result in same iteration therefore, we will select the next gammaB randomly from all gammas and do this on the subsequent iteration as well till the gamma's does not effectively change.
- Even on selecting random gamma's it's not guaranteed that the gamma's will change; for this we adjusted the termination condition of code. The iteration will terminate if maximum allowed iterations are completed or the number of continuous passes exceed the *maxPass* limit
    - Here *maxPass* is defined as maxPass = max(maxPass2, (int)(0.1*iteration))
        - The MaxPass2 will be received from the user through an argument of class initiation which by default is set to 10.
- On algorithm mention paper[2] the values of *p1* & *p2* will be calculated from the average of the respective support vectors, but there could be the iteration (usually among the earlier iterations) where there is no support vector for either *p1* or *p2* in such case the algorithm can go heywire also in paper[1] they didn't face such conditions, In order to stop this we decided to ignore such changes and count it as pass and also adjust out initial parameter *v1, v2* and *epsilon* such that:
    - epsilon/(v2*m) and 1/(v1) are sufficiently greater than zero, where m is the number of samples in the dataset.

Observation

We have implemented both RBF and linear kernel for this assignment.
We have taken v1=0.0001, v2=0.09, epsilon=⅔ and iteration=10000 for the following result and for RBF kernel standard deviation was taken 1.

In the randomly generated dataset we have usually found an MCC score of about 0.41 in linear kernel and 0.35 in RBF kernel.

References

[1] Fragoso, Victor, et al. "One-class slab support vector machine." *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016

[2]Kumar, Bagesh, et al. "Sequential Minimal Optimization for One-Class Slab Support Vector Machine." *arXiv preprint arXiv:2011.03243* (2020).

[3]Platt, John. (1998). Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Advances in Kernel Methods-Support Vector Learning. 208.