iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** XII **Month of publication:** December 2018

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue XII, Dec 2018- Available at www.ijraset.com

Enhanced Support Vector Machine with Speed Up and Reduced Sensitivity

Harshal Chaudhari[1], Himanshu Londhe[2], Nachiket Namjoshi[3], Rahul Kolhatkar[4], Sandeep Chaware[5]
[1, 2, 3, 4, 5]*Department of Computer Engineering, Marathwada Mitra Mandal's College of Engineering, Pune, India*

Abstract: ***Support Vector Machines (SVM) are the latest development in classification of data in machine learning. A new improved methodology is to be implemented in order to increase the performance. A novel approach to implement SVM with added modules and train the machine is proposed. Linear SVMs have a low time complexity. However, they cannot be used in a data set with several classes or in cases of scattered datasets, where Nonlinear SVMs are used, which have high time complexity. In order to achieve high-efficiency, a model which will implement a Linear SVM algorithm for complex data sets using spirals in the 2d plane to better classify the data in the hyperplane is proposed.***
Keywords: Data Mining, Support Vector Machines, Parallel Computing, GPU, Classification.

I. INTRODUCTION

In recent years, there has been a need for faster and more efficient algorithms, which has been achieved by parallelism. Parallelism can be achieved by using an extensive processor such as a Graphics Processing Units (GPU). Use of GPU provides us with the ability to work on ultra large data problems.

Data Classification is an important challenge in this modern world of information which is stored as data. Several institutions face with a challenge of handling and using their data. There are several data classification algorithms in existence, however, SVMs, currently are the latest development in data classification algorithms [1]. They have the best classification rate and can handle larger datasets. However, existing algorithms for classification of scattered datasets require greater time and computational complexity, which is not always desirable.

There are 2 types of SVM algorithms viz. Linear and Nonlinear. Linear SVMs can be used easily for simpler data sets, whereas for complex data sets, it becomes difficult to use them. So, we use Non-Linear SVMs. However, it must be considered that Non-Linear SVMs have a very high time complexity. Hence, an efficient approach must be developed.

Moreover, existing SVM algorithms, Linear as well as Non-Linear fail when the input dataset is mixed with various items and attributes i.e. scattered data set., e.g.: A student dataset in which, male and female data items are alternate.

II. BACKGROUND WORK

Few studies and research have been put forward over the recent years related to improving the efficiency of classification algorithms with the help of GPU. Support Vector Machines have been used since 1990s and is still used for classification of complex and large datasets. Implementing the correct pre-processing, SVMs performance can be increased, as shown in [1] Where KNN algorithm is implemented on GPU to increase the speed of execution and they found good results of up to 100x speedup on GPU as compared to CPU. A bubble sort has been implemented on GPU [2] with 334 GPU cores which resulted into massive improvement of time complexity from O(n) on CPU to O(1) on GPU.

An implementation of SVM using sequential minimal optimization (SMO) and compressed sparse row (CSR) on newsgroup20 and mnist dataset is suggested [3]. As compared to LibSVM on CPU they obtained a 6x to 36x speed up on GPU. The compressed sparse row (CSR) format represents a matrix M by three (one-dimensional) arrays, that respectively contain nonzero values, the extents of rows, and column indices. It is similar to COO, but compresses the row indices, hence the name. This format allows fast row access and matrix-vector multiplications (Mx) [4]. Sequential minimal optimization (SMO) is an algorithm for solving the quadratic programming (QP) problem that arises during the training of support vector machines. It was invented by John Platt in 1998 at Microsoft Research. SMO is widely used for training support vector machines and is implemented by the popular LibSVM tool. [5] By using data loading methods such as data chunking and data reduction, performance of SVM on various datasets viz. mnist*, adult, Web has been accelerated [6]. On GPU, 13x to 52x speed up was obtained compared to LibSVM on CPU. Although, the results obtained are not same for both the loading methods. The memory constraint issue brought by large datasets is addressed through either data reduction or data chunking techniques. The data reduction method achieved a significant speed improvement for a tradeoff to prediction accuracy loss, while the data chunking offers stable predicting accuracy for lower performance.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue XII, Dec 2018- Available at www.ijraset.com

III. PROPOSED WORK

A. Proposed Dataset

A sample dataset (SUSY) [15] taken from UC Irvine Machine Learning Repository will be used for testing and training of the classifier. The training data consists of 4.5 million records and testing data has 500 thousand records, in total 5 million records are present in the dataset. These records are pre-classified into 18 attributes.

B. Proposed Method

1) The data is split into approximate 2 halves. When envisioned as a matrix, the dataset classes will be the columns and the attributes will be the rows.
2) The attributes chosen in the first step are binary attributes i.e. they have only 2 discrete values. After the first step, the data obtained will be much easier to classify for the SVM algorithm.
3) Training data is passed on to our spiral organizer, which organizes the data set in order to make the data set compatible with linear SVMs.
4) Once the data is ready for linear SVMs, the SVM algorithm will be implemented, which, otherwise, would have failed if the data was unchanged. Spiral Organizer operates in layer fashion to segregate the data by swapping places of each out of order elements.

C. Classification

The support vector machine searches for the closest points which it calls the "support vectors". Once it has found the closest points, the SVM draws a line connecting them.

It draws this connecting line by doing vector subtraction (point A - point B). The support vector machine then declares the best separating line to be the line that bisects -- and is perpendicular to -- the connecting line.

We are given a training dataset of n points of the form $(\overrightarrow{x}_1, y_1)\ldots(\overrightarrow{x}_n, y_n)$

Where y_i are either 1 or −1, each indicating the class to which the point x_i belongs. We want to find the "maximum-margin hyperplane" that divides the group of points x_i for which $y_i = 1$ and also for 0. This is done by using Lagrange's Multiplier. The Equation comes out to be:

$$f(\overrightarrow{w}, b) = [\frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i(w.x_i - b))] + \lambda\,||w||^2$$

Where,

$f(\overrightarrow{w}, b)$ is the classifier function.

$\overrightarrow{w}$ is the width vector, which is to be minimized for the width to be maximum [7].

D. Pre-Classification

1) *The Spiral Organizer:* Spiral Organizer re-arranges the data in a spiral manner, where each layer of spiral is considered as a circle, where all the layers of the spiral are like concentric circles with each circle divided in 2 halves such that each semi-circle represent one class of data. The swapping of the data elements is done by the basis of comparison with the threshold value. This threshold is computed dynamically (it will change as per the attributes provided to it). The accuracy of the swapping is increased by using the same 2 attributes that are passed to the SVM classifier. This reduces the unwanted swaps and which results in increase in accuracy.

Algorithm of Spiral Organizer can be expressed as:

```
mat ←convertToMatrix(dataset)
meanbad ←getThreshold(thresholdParameter)
flatmat ←matflatten()
obj ←len(dataset)
        size lat ←len(flatmat)
        mid ←(int)(size flat/2 −−1)
        bot ←size flat –1
        for i:0 to mid do
          if    flatmat[i][3] > meanBad and
```

```
    flatmat[bot][3] > meanBad then
continue
else      if        flatmat[i][3] <= meanBad and
           flatmat[bot][3] > meanBad then
           swap(flatmat[i],flatmat[bot])
           bot ←bot –1
           continue
 else      if        flatmat[i][3] <= meanBad and
          flatmat[bot][3] > meanBad then
          bot ←bot –1
           i ←i –1
 if i == bot then
           break
       end if
    end if
end for
```

2) *Enhanced Spiral Organizer:* Spiral Organizer algorithm is designed in such a way that it can be parallelized. Spiral organizer takes dataset as well as feature list on which SVM is run on. However, in the real world, we may never know which feature set can result in maximum efficiency. As a result, we may need to use SVM on every probable permutation of the pairs of features and by extension, Spiral also. We can parallelize Spiral by passing every permutation to different thread. However, every thread must access to its own copy of the dataset which is not feasible for using GPU as there is a data- transfer latency. Hence, GPU is not usable for this algorithm.

IV. PERFORMANCE EVALUATION

In classification, a false positive would always take place, which means to classify mistakenly a class A as class B. A false negative, which classifies class B as class A, could also exist in class classification. A false positive could be more serious than a false negative because, in this case, the system would ignore certain elements due to element filtering without analyzing them. Under such circumstances, it is acceptable to have some false negatives. Thus, it is important to keep the false positives up to a minimum value.

The proposed algorithm performance is evaluated with false positive rate and false negative rate. Along with it the overall classifier accuracy and precision is used as metrics to evaluate system performance.

Furthermore, the time required by classifying on CPU with standard SVM and Spiral SVM is compared to discriminate the results obtained by the system and overall result graph is shown in Fig. 1.

V. RESULTS

TABLE I
Execution Timings

Data Size	Spiral SVM	SVM	Percentage Increase (Average Increase in time and accuracy)
10,000	0.226/87.8%	0.227/74.7%	8.96%
100,000	5.996/89.69%	6.108/75.63%	10.21%
500,000	47.339/89.7%	51.243/75.64%	13.09%

For testing the efficiency, accuracy of classification and total time of execution was calculated using the confusion matrix [12]. Results obtained showed 13-15% increase in accuracy and 2-3% decrease in execution time on CPU. This designed model works efficiently when executed parallelly as well. However, the internal of the spiral cannot be parallelized as it requires the entire dataset, to ensure maximum swaps and increased accuracy. The results are shown in Table 1.

For a better view, when a double bar graph is plotted (time against data size), we get the following plot as shown in Fig. 1.

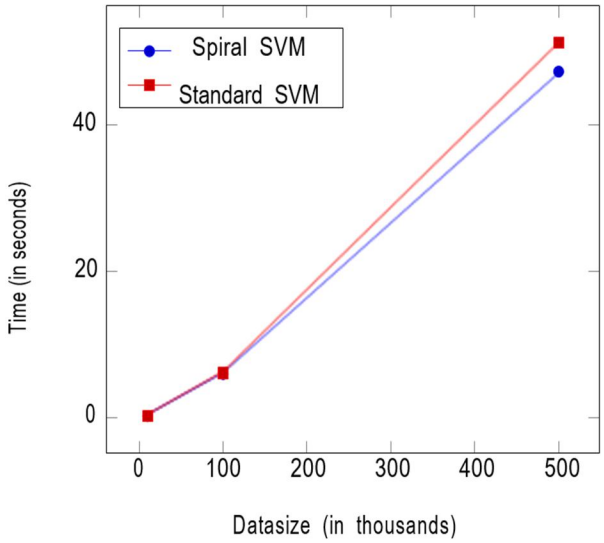


Fig. 1 Performance Plot of Spiral SVM vs Standard SVM

VI. CONCLUSION

Support vector machines have been used for a couple decades now by experts to classify large dataset. Classification is basically done by considering a real-life problem and then proposing solution for the same. Today, data is becoming more and more complex and growing rapidly in size, which brings us to the problem that, with the growth of data it is crucial to devise a system that can handle complex data.

Hence, our system emphasizes on various aspects related to processing of large data and resolves the issue of processing times and data compatibility.

VII. FUTURE WORK

Although, results obtained on CPU provide an increase in the efficiency of classification, more speed up can be achieved by running the SVM kernel on GPU. Similarly, same can be done in case of Spiral Organizer. During our experiment with the test data, we observed bottleneck in data latency and also GPU parallelization requires to load data onto GPU memory which is not possible with data of such size. One other way is to perform the classification operation on multiple clusters of GPUs [8].

REFERENCES

[1] Selvaluxmiy.S, Kumara.T. N, Keerthanan.P, Velmakivan.R, Ragel.R, Deegalla.S., "Accelerating k-NN classification algorithm using graphics processing units," 2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), Galle, 2016, pp. 1-6.

[2] N. Faujdar and S. P. Ghrera, "A practical approach of GPU bubble sort with CUDA hardware," 2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, Noida, 2017, pp. 7-12.

[3] Sopyła K., Drozda P., Górecki P. (2012) SVM with CUDA Accelerated Kernels for Big Sparse Problems. In: Rutkowski L., Korytkowski M., Scherer R., Tadeusiewicz R., Zadeh L.A., Zurada J.M. (eds) Artificial Intelligence and Soft Computing. ICAISC 2012. Lecture Notes in Computer Science, vol 7267. Springer, Berlin, Heidelberg.

[4] Compressed Sparse Row (CSR) URL: https://en.wikipedia.org/wiki/Sparse_matrix#Compressed_sparse_row_(CSR,_CRS_or_Yale_format)

[5] Sequential minimal optimization (SMO) URL: https://en.wikipedia.org/wiki/Sequential_minimal_optimization

[6] Q. Li, R. Salman and V. Kecman, "An intelligent system for accelerating parallel SVM classification problems on large datasets using GPU," 2010 10th International Conference on Intelligent Systems Design and Applications, Cairo, 2010, pp. 1131-1135.

[7] Support Vector Machines (SVM) URL: https://en.wikipedia.org/wiki/Support_vector_machine

[8] Graphics Processing Unit (GPU) URL: https://en.wikipedia.org/wiki/Graphics_processing_unit

[9] F. Padillo, J. M. Luna and S. Ventura, "An evolutionary algorithm for mining rare association rules: A Big Data approach," 2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastian, 2017, pp. 2007-2014.

[10] T. N. Do and V. H. Nguyen, "A novel speed-up SVM algorithm for massive classification tasks," 2008 IEEE International Conference on Research, Innovation and Vision for the Future in Computing and Communication Technologies, Ho Chi Mih City, 2008, pp. 215-220.

[11] Yunmei Lu, Yun Zhu, Meng Han, Jing (Selena) He, and Yanqing Zhang. 2014. A survey of GPU accelerated SVM. In Proceedings of the 2014 ACM Southeast Regional Conference (ACM SE '14). ACM, New York, NY, USA, Article 15, 7 pages.

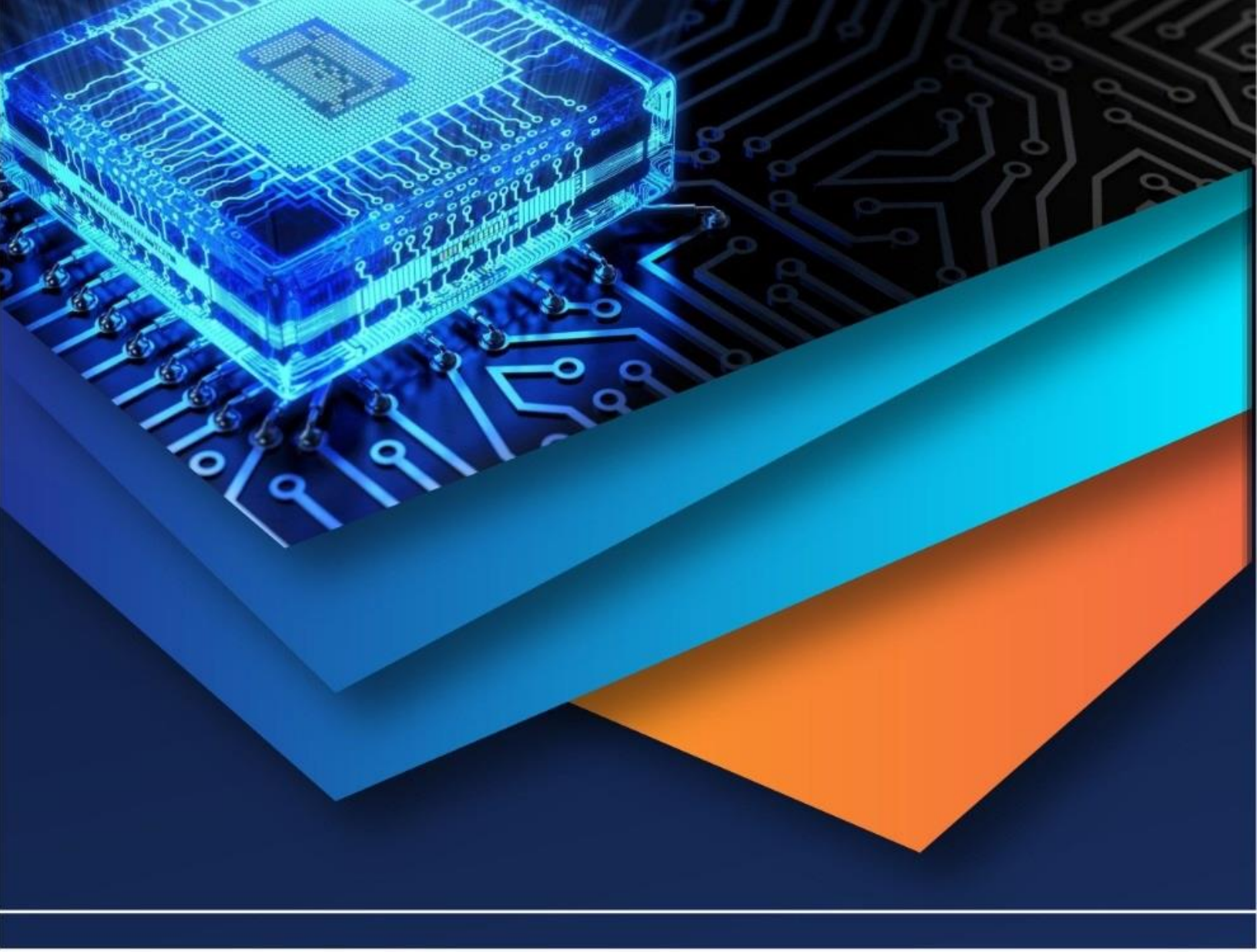
[12] Andreas Athanasopoulos, Anastasios Dimou, Vasileios Mezaris, Ioannis Kompatsiaris, "GPU acceleration for support vector machines", In Procs.

12th Inter. Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2011)

[13] K. Tan, J. Zhang, Q. Du and X. Wang, "GPU Parallel Implementation of Support Vector Machines for Hyperspectral Image Classification," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 8, no. 10, pp. 4647-4656, Oct. 2015.

[14] V. Vapnik, "The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995

[15] SUSY Data Set, URL: https://archive.ics.uci.edu/ml/datasets/SUSY



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY