# *Enhanced Support Vector Machine with speed up and reduced sensitivity*

Nachiket Namjoshi
Department of Computer Science
Marathwada Mitramandal's College of Engineering
Pune, India

Harshal Chaudhari
Department of Computer Science
Marathwada Mitramandal's College of Engineering
Pune, India

Rahul Kolhatkar
Department of Computer Science
Marathwada Mitramandal's College of Engineering
Pune, India

Himanshu Londhe
Department of Computer Science
Marathwada Mitramandal'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 using CUDA platform 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***

## 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., eg: 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 1995 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 k-NN algorithm is implemented on GPU to increase the speed of execution and they found good results of upto 100x speedup on GPU as compared to CPU. Neetu Faujdar and Satya Prakash Ghrera [2] has implemented bubble sorting on GPU with 334 GPU cores which resulted into massive improvement of time complexity from O(n) on CPU to O(1) on GPU.

Sopyła K., Drozda P. and Górecki P [3] have suggested implementation of SVM using sequential minimal optimization (SMO) and compressed sparse row (CSR) on newsgroup20 and mnist dataset. 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, Q. Li, R. Salman and V. Kecman [6] have accelerated performance of SVM on various datasets viz. mnist*, adult, Web. 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 trade off to prediction accuracy loss, while the data chunking offers stable predicting accuracy for lower performance.

## III. PROPOSED WORK

### *A. Proposed Dataset*

A sample dataset (SUSY) taken from UC Irvine Machine Learning Repository will be used for testing and training of the classifier. The training data will consist of 9 million records and testing data will have 2 million records, in total 11 million records are present in the dataset. These records are pre classified into 28 attributes.

### *B. Proposed Method*

In the first step, 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. 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.

Training data is passed on to our spiral organiser, which organises the data set in order to make the data set compatible with linear SVMs. 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 Organiser operates in layer fashion to segregate the data by swapping places of each out of order elements

Performance of the algorithm can be improved i.e the computational complexity can be reduced if the algorithm is executed on the CUDA platform i.e with GPU support.

Using GPU along with the modified SVM algorithm will immensely increase the efficiency of the proposed algorithm.

### *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]

### *Pre-Classification*

#### a. *Spiral Organizer*

As discussed earlier, Linear SVM cannot classify scattered dataset efficiently. To overcome this drawback of Linear SVM, spiral organisation is to be performed beforehand so as to reduce the degree of scatteredness of the data set. The spiral organizer operates in a layered approach. The data is represented in a matrix format, with an arbitrary centre/pivot point around which the data will be reorganized. The attribute selected to pre-classify the data is such that it has 2 distinct values i.e binary attribute. On the basis of the pre-classification, a confidence value will be assigned, and the highest valued attribute is chosen for final reorganization.

#### b. *Processing on GPU*

A GPU is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the computations which take much longer to be performed On a CPU. A GPU has thousands of cores in it. This helps in computations of large scale data. Hence, we use GPU to speedup the process of both, Spiral Classification as well as SVM. [8]

- Each layer in spiral organizer is computed on the GPU, thus increasing the speed.
- Speed-up for SVM classifier is achieved by solving the function of finding the most accurate equation of the plane based on the x-y-z coordinates, parallely on the GPU.

To handle all GPU related operations, CUDA has been used. CUDA is a parallel computing platform and programming model developed by NVIDIA for general computing on GPU. The CUDA platform is designed to work with programming languages such as C, C++, and Fortran. This accessibility makes it easier for specialists in parallel programming to use GPU resources.

## IV. Performance evaluation

In classification, a false positive would always take place, which means to classify mistakenly an 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 analysing 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 will be evaluated with false positive rate and false negative rate. Along with it the overall classifier accuracy and precision will be used as metrics to evaluate system performance.

Furthermore, the time complexity obtained by classifying on CPU cores and on GPU core is compared to further discriminate the results obtained by the system and overall result graph will be shown.

## V. Expected outcome

Final data which is obtained from the system will yield an outcome which is accurate and will provide an efficient classification system. The outcome will solve the issue of data scattered-ness and sensitivity along with speeding up the classification process. In recent scenarios, data classification models are susceptible to outliers in the dataset. The proposed system will overcome such flaw and yield better overall performance.

## 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.

## 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 Minh 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.