A MULTICLASS SENTIMENT CLASSIFICATION USING SKIP-GRAM EMBEDDING WITH SUPPORT VECTOR MACHINE-STOCHASTIC GRADIENT DESCENT (SVM-SGD)

[1] Abdullah K.-K. A, [2] Sodimu S. M., [1] Odule T. J., [1] Solanke O. O.

[1] Department of Mathematical Sciences, Olabisi Onabanjo University, Ago Iwoye, Ogun State, Nigeria
[2] Ellcry Headquaters Public Benefits Company, Magodo, Lagos State, Nigeria

Corresponding Author: Abdullah K.-K. A., uwaizabdullah9@gmail.com

ABSTRACT: N-gram feature is used to represent documents in natural language processing but leads to curse of dimensionality. Sentiment classification based on word embedding to represent document with an incremental method reduce dimensionality with dense vector representation of words. This works focus on the skip-gram model with negative sampling for vector representation. The text class is predicted by maximising the probabilities of embedding vectors of words under the class. However, a Support Vector Machines with an extension of Stochastic Gradient Descent (SVM-SGD) is employed for effective classification of datasets into multiclass. This is achieved by maximising the margin between hyperplane of every two class pair using online learning as well as controlling the constraints and minimise the regularisation error. This reduces the effect of imbalanced classes in training the classifier parameters. Hence, solve a quadratic programming problem while running SGD for chosen iterations and returning the average point in the number of classes in terms of accuracy and computational cost.

KEYWORDS: SVM, Multiclass, Skip-gram embedding, Stochastic Gradient Descent, Negative sampling.

1.0 INTRODUCTION

Recent research studies have demonstrated that sentiment classifications are used to extract feelings, altitude and opinion towards an entity. This entity represents people, events or topics that are probable covered with the aid of evaluations. Therefore, sentiment classification provides significant commercial values and basis for academic research. Meanwhile, classification with machine learning helps to get vigorous knowledge and analysis for evaluation of sentiments such as clarifying consumer behaviour, recommending products and analysing feelings expressed in an automatic or semi-automatic way in order to understand public opinion [GME17]. Also, there has been a lot of research on sentiment classification such as sentiment classifiers [PLV02], opinion extraction [M+02], or recommender systems. The goal of the classifier knowledge is to build a concise version of the distribution of class labels in terms of prediction capabilities. Support Vector Machine (SVM) has been successful at addressing classification problem. It is expressed as the minimisation of an objective function involving empirical risk while keeping time as low as possible for the complexity of the classifier. Therefore, linear SVM classifier is a trade-off between training time and classification accuracy [DT05]. Literatures have shown that analysing polar sentiment in negative and positive is not always sufficient. Subsequently, SVM that involves binary classifiers performed poorly with large datasets [A+18], hence, poses a challenge for the computational complexity of a learning algorithm. This is due to imbalance classification of the prediction function $f: X \rightarrow Y$ implies $f(x) \neq y$, this brings the attention to methods using more classes for sentiment classification [YTA14]. Therefore, there is need to solve problem which has more than two classes by extending SVM to a multiclass classifier.

Approaches for multiclass SVM (MC-SVM) can be categorised into two; decomposition to binary classification methods while the other considers all data in single optimisation method. However, the two methods did not follow the motivation of maximising the margin between training points and hyperplane in support vector machine [RK14]. Although, the SVM with non-linear classification uses non-linear mapping to transform training data to higher dimension which searches for the linear separating hyperplane. The implementation is hindered by the quadratic dependence of memory requirements on the number of training examples, thus, processing large dataset is difficult. Therefore, there is need for SVM to build multiclass classifier that can handle large and sparse dataset using Stochastic Gradient Descent (SGD). The SGD has been proven to be an effective method for training machine learning algorithms with significant decrease in training time without sacrificing accuracy [Zan04]. Optimising SVM has been shown to be equivalent to quadratic programming problems, thus, minimised empirical loss as the sum of the pairwise errors [Gue02]. Parameter-dependent learning rate for stochastic gradient methods is sensitive to the choice of step sizes, which helps in the convergence rate of SGD and produced better results [DHS11]. Polyak & Anatoli [PA92] proved that if there is enough training samples for updating parameters, SGD can obtain the parameters as good as the empirical optimal in just one epoch at a time (t) through all data point, therefore, reduces time and computational cost. Most existing work used one-hot vector in representation, this waste time and memory space. Hence, not suitable for sentiment classification because it breaks the syntactic structures, disrupts the word order as well as discards some semantic information, hence, reduces classification accuracy. According to Gupta *et al.*, [GBW14], using feature vector for MC-SVM resulted in high-dimensional feature spaces, moreover, the number of k classes grows as well as the number of decision boundaries between classes at a worst-case rate of k^2. Tremendous progress has been made by distributed representations (word embeddings) which learn a transformation of each word from raw text to a dense, lower-dimensional vector space. Word embedding is a numerical representation of words $w \in \Re^m$ which is an unsupervised learned word representation. According to Mikolov *et al.*, [MSC13], the skip-gram formulation was introduced for neural word embeddings, wherein it predicts the context of a given word embeddings. Subsequently, the interdependence of the pairwise class decision boundary is too sensitive to the empirical error noise of imbalanced classes, therefore, minimise the total empirical error.

The study focuses on generating multiclass sentiment classification using online learning with SVM-SGD taking into consideration the number of misclassification labels. The study aims to optimise MC-SVM in order to find the optimal labels that can handle high dimensional data and get the best classification accuracy using word embedding as low-dimensional, dense vector representation of words. Although, mapping *d*-dimensional feature vector to *m*-dimensional embedded vector to separate classes in the embedding space takes longer time for large dataset. In this work, skip-gram word embedding is used for vector representation with negative-sampling algorithm to improve the computational feasibility of training the embeddings. This speeds up the training and rephrases the problem as a set of independent binary classification task by approximating training and setting randomly sample. Hence, adopt a softmax layer as the output to predict each word’s probability in the vocabulary. The proposed method is achieved by optimising single binary problem by maximising the margin between multiple *k*-classes as well as relationship between the classes using stochastic gradient descent algorithm with embedding matrix as vector representation. The SVM-SGD involves multiple binary class objective function between each two class pair to solve the constraints problem which forces the unlabelled examples to be far from the margin by creating hypothesis (parameter) and adjust it towards the gradient during the training with respect to the number of chosen iteration. The embedding matrix reduces the parameters and act as regulariser to reduce the memory and time usage, finally, convergence is overcome using a quadratic loss function problem.

The main contribution of this work is to maximise SVM training objective function with respect to the model parameters using optimisation process via Stochastic Gradient Descent (SGD) and context embedding by reducing the sparsity of the feature vector with negative samples towards the gradient. The rest of the paper is organised as follows: Section 2 presents background of the multiclass SVM on large-scale datasets with related works. Section 3 proposed multiclass using SVM-SGD with embedding as vector representation optimised with negative sampling, its improvement for multiclass and description of how to speed-up the training process. Section 4 gives experimental results, conclusion and future works are given in Section 5.

2.0 BACKGROUND AND RELATED WORK

Most approaches used for sentiment classification are machine learning, pattern based and natural language processing to find sentiments in words, sentences and/or sentiments in topics. Research conducted by Prabowo and Thelwall [PT09] shown that from all approaches, the best results were observed from machine learning approaches. Machine learning classification is the process of approximating the mapping function that maps the input sample to target class or label [CF09]. These labels are trained to produce reasonable outputs when encountered during decision making [GY14] which is capable of extracting, learning and classifying many real-world tasks, thus, performance depends on the scale of labelled data. SVM models were initially developed to perform binary classification, although, applications of binary classification are very limited. Therefore, SVM can be extends to a multiclass classifier of $k > 2$ classification by maximising hinge loss which can be derived from a margin bound. The SVM find the hyperplane that maximises the margin while, minimising a quantity proportional to the number of misclassification errors. This problem of maximising the margin can be solved using Quadratic Programming (QP) optimisation techniques, thus, multiclass problems determine n hyperplane.

A good work in text classification is the design of effective feature representations. Wang & Manning [WM12] demonstrated that bigram features are particularly useful for sentiment classification because it resolves the ambiguity of polysemous words. Meanwhile, semantic information plays an important role in sentiment, however, vector space representation lacks context and word sense information with high-dimensional. Thus, word embeddings represents word at low-dimensional, dense vector representation and trained by language model. Presently, the most popular models of word embedding are continuous bag-of-words model (CBOW) and continuous skip-gram model (Skip-gram) [MSC13] which is a simplification of neural language models for efficient training. CBOW deals with more syntactic information and obtain better result while embedding trained by skip-gram contains more semantic information and maximise the probabilities of words given its context windows.

With the learning process, multiclass can be classified into batch learning and online learning techniques. The data required for the training of the classifier is collected in prior which poses a constraint in the application of batch learning. While online learning, parameters are updated in an iterative way with sequential data [P+15]. Therefore, online learning techniques are preferred in this work for streaming datasets. A multistage SVM (MSVM) method is used for multiclass problem in [LXW03, B+01], this involved using Support Vector Clustering (SVC) to divide the training data into two parts to train binary SVM. In each partition, the procedure is recursively repeated until the binary SVM gives an exact label of class. There could be a repetition in the two clusters resulting in decreased predictive accuracy as well as the method cannot be used for large datasets.

The existing MC-SVM problem are categorised based on decomposition of binary classification methods and single optimization model with the concept of the margin to multiple classes. The former involve splitting the multiclass into multiple binary subproblems and employs the existing binary classifiers to solve MC-SVM [HL02] such as one versus-all (OvsA) and one-versus-one (OvsO). The OvsA split a k class classification problem into k binary subproblems and OvsO splits binary classification problem into $k(k-1)/2$ class. However, using SVM-SGD with these methods does not take into account the benefits of high-performance computing. Also, the method is too expensive for large dataset because it trains multiple binary classifiers. Approaches of parallelisation of MC-SVM training are based on OvsO or OvsR which can be done with SGD that are trained over multiple computers [GBW14, BMS16]. The second method is achieved by modifying the binary class objective function and adds constraints to every class which also called *All-in-one* MC-SVMs with examples such as Weston's multiclass SVM [WW99], Crammer multiclass SVM [CS02], and regression-like formulation [NWH17]). SGD used an iterative approximation method on the loss term function with step wise approach which is based on a subset of the data. Weston *et al.*, [WW99] applied weighting to stochastic gradients with automatic step-size adaptation called adagrad [DHS11] to MC-SVM, the result generated almost doubles the classification accuracy as well as shows substantial gains over OvsA SVM. This work is similar to Xu *et al.*, [X+17] that used the the single optimisation method to solve the multi class classification problem with stochastic gradient descent with variance reduction algorithm (SVRG) as optimisation model. But a word embedding with negative sampling is used as language representation to reduced the dimensionality and help in faster convergence rate as well as better local optimum.

3.0 THE PROPOSED APPROACH

The goal of classification is to classified a given document $d \in D$ into a fixed number of predefined categories k where D is the set of documents. The data are collected from University College Dublin at http://mig.ucd.ie/datasets/bbc.html, preprocessing is done on the extracted text collected such as stopwords, normalisation and tokenisation of text. The text documents are represented based on word embedding using skip-gram embedding to train the word documents. The document can be in one and/or multiple categories, in this work, document is most likely to belong to class $k \in K$, however, train the SVM to maximise the margin between each two classes pair. The multiclass SVM scores the correct class higher than all negatives scores by at least a margin of change. The SVM is analysed to solve dual problem subject to the constraints, but used SGD to optimise the constraints by updating or adjusting the weight w on t epochs with different learning rate until SGD converges. The main objective is to find weight that will satisfy the constraint for all examples in the training data and give a total loss as low as possible at optimal and constant level.

3.1 Preprocessing and Word Embedding

The text corpus C is prepared for learning the embedding by creating word tokens, removing punctuation and removing stop word as well as normalised the tokens. Words are represented as vector using word embedding to capture the semantic information between words. The *m*-dimensional vectors of all the words $w \in \Re^m$ in the vocabulary V form the embedding matrix. The skip-gram model trains word embeddings by maximising the probabilities of words given its context windows of size $'S'$, where the contexts are immediate neighbours of the target word. Skip-gram relies on the distributional similarity between each target word embedding (t) and the context embeddings (c) in a context window in the corpus with vocabulary set V. The objective of the skip-gram is to compute the probability $P(V_c|V_t)$ of V_c predicted as V_t content for all training pairs as follows:

$$\sum_{V_c, V \in D} \log P(V_c|V_t) \qquad (1)$$

In order to compute the closeness of the skip-gram, a dot product is used between input embedding of the target word and output embedding of the context word which is used to find the softmax objective function. Therefore, the closeness U_c is represented as follows:

$$U_c = I_i^E \cdot O_c^E \qquad (2)$$

To maximise the objective function of negative samples in skip-gram that is the sparse matrix. Each word-pair contains a correct word pair d and negative samples d' respectively, and then softmax function (output prediction) is in the form:

$$\sum_{V_t, V_{c \in D}} \log P(1/V_t, V_c) - \sum_{(V_t, V_c \in D)} \log(-P(C = 1/V_t, V_c) \qquad (3)$$

However, computing Eqn. 3 is computational expensive (calculating sum of the denominator of the softmax objectives), a negative sampling method is employed to speed up the training and rephrase the problem as a set of independent binary classification task according to [WM12]. This is done by approximate training and setting randomly sample word $S = S_1, S_2, \cdots S_n$ and set window size to be $S = 3$ with size 100 and negative sample of 10. The skip-gram negative sampling compares the correct word-context pairs with randomly-generated non-correct pairs and maximises the probability of the actual word-context pairs, while minimising the probability of the negative pairs. Therefore, the objective then predict for a given context pair U_c.

Using binary decision,

$$P(C = 1/V_t, V_c) = \delta(U_c) = \frac{1}{1+e^{=U_c}} \qquad (4)$$

where δ is the combination parameter which balances the contribution of each component in the training

Using Eqn. 4,

$$\sum_{V_t, V_{c \in D}} \log \delta(U_c) + \sum_{V_t, V_{c \in D}} \log \delta(-U_c) \qquad (5)$$

This can be implemented using Word2Vec module of the Gensim Library for the vector representation.

Algorithm for Word Embedding with Negative Sampling.

Input: Corpus C, labelled training set D, *m*-dimensional with vocabulary V and sample times T

Output: Embeddings $w \in \Re^m$ of all words in the vocabulary *V*.

Initialization: Randomly set $w \in \Re^m$ for all words in V; generate the token words set S;

Constructing T prediction using a window size $S = 3$.

```
1.  for t = 1, 2, . . . , T do
2.      optimise $U_c$ context using negative sample method
3.      if $w$ in S then
4.          for n do
5.              sampling the positive word $d$ and the negative word $d'$
6.              optimize $U_c$ function by updating $w, d, d'$
7.          end
8.      end
9.  end
10. return $X$ for all words in $V$
```

3.2 Maximise the Margin of SVM Hyperplane

Using SVM as a single optimisation problem for multiclass in large dataset involves high computational cost but it can be resolved by reducing dimensionality using embedding matrix in Eqn. (5) as well as optimising the gradient with Stochastic Gradient Descent (SGD). SVM is extended by simultaneously maximise the margin and minimise the error in every pair of binary classification subproblems between classes.

If there are n training data point of the form $\{x_i, y_i\}$ and k classes, such that $i = 1,2 \ldots m$; $x_i \in R^n$ in the *m*-dimensional input space and $y_i \in \{1 \ldots k\}$. SVM algorithms find the best separating plane for normal vector $w \in \Re^n$ and the scalar $b \in \Re$. In linearly separable support vector, the separating hyperplane with largest margin can be represented as:

$$x_i : (w + b) \geq +1 \; for \; y_i = +1 \qquad (6_a)$$
$$x_i : (w + b) \geq +1 \; for \; y_i = -1 \qquad (6_b)$$

The equations (6) are combined to form

$$y_i(w \cdot x_i + b) - 1 \geq 0 \; \forall i \qquad (7)$$

If the point for which equality in equation (6a & b) holds, then, it is equivalent to choose a scale for w and b which lies on the hyperplane at point zero (pt=0), Therefore, combining the two (2) hyperplane, the margin is $2/\|w\|$, where $\|w\|$ is the 2-norm of the vector) which gives maximum margin by minimising $\|w\|^2$ subject to constraints in Eq. (7). Therefore, any point x_i that falls on the other side of its supporting plane is considered to be an error (z_i) as in Figure 1. Thus, adding z_i to constraints in Eq. 7 $\Rightarrow y_i(w \cdot x_i + b) - 1 + z_i \geq 0$

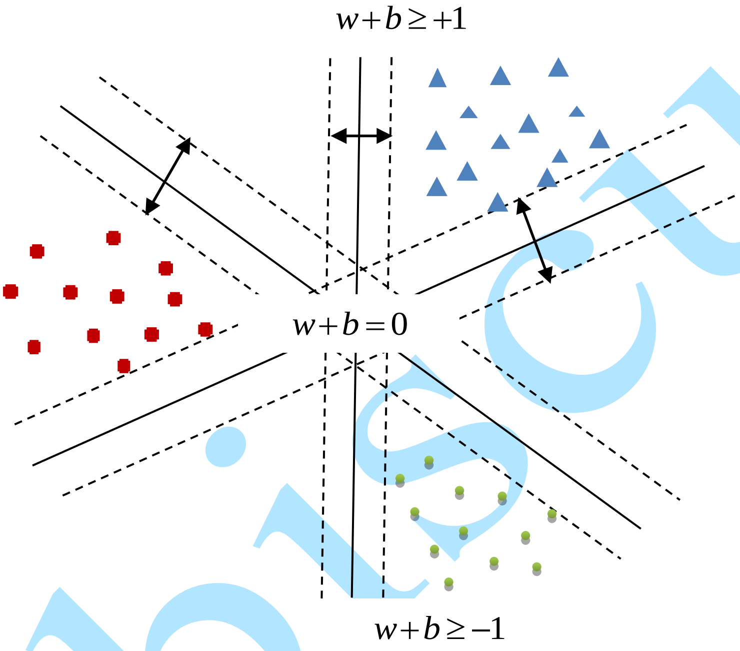


Figure 1: Separating datapoints into classes

When the data are not linearly separable, there is a penalty parameter $\eta_t > 0$ which implies $\eta_t \sum_{i=1}^{n} z_i$ where $\eta_t = \dfrac{1}{\lambda n}$. This reduces the number of training errors, in order to find SVM separating the training set under constraints which force the unlabelled examples to be far from the margin as in Figure 1. There must be small hinge loss, therefore, SVM search for a balance between the regularisation term $\frac{1}{2}\|w\|^2$ and the training errors. This is equivalent to quadratic programming problems as follows:

$$min \; \varphi(w, b, z_i) = \tfrac{1}{2}\|w\|^2 + \eta_t \textstyle\sum_{i=1}^{n} z_i \qquad (8)$$

subject to: $\quad y_i(w \cdot x_i + b) + z_i \geq 1 \qquad$ (i)
$\qquad\qquad\quad z_i = \geq 0 \qquad$ (ii)

The computational cost requires solving quadratic programming problems which depends on the number of training data point. This can be tackle with Stochastic Gradient Descent (SGD) for large datasets.

3.3 Optimise SVM with Stochastic Gradient Descent (SGD)

In this study, in order to solve the primal SVM objective function, SVM is optimised in a dual form (quadratic loss function) to gives exact solution which similar to Thanh-Nghi [Tha14]. There is need for online learning to prevent loading entirely training set examples (T) to the memory in order to adjust decision rule over time. Therefore, SGD is used to obtain a sequence of t predictors, whose average has a generalisation error that converges to an optimal value. SGD creates hypothesis (parameter) by adjusting the parameters such as epochs and learning rate on a variable dataset towards the gradient during the training time with respect to the number of chosen iteration.

SVM problem are formulated as quadratic programming in (8) in an unconstraint problem where bias can be ignored. Therefore, the constraints 8(i) and (ii) can be written as hinge loss function

$$z_i = max\{0,1 - y_i(w \cdot x_i)\} \qquad (9)$$

For a given dataset of size n , minimisation of the regularisation function with respect to w is equivalent to the minimisation of the objective function. Therefore, z_i is substitute in the objective function Ψ of Eqn. (8)

$$\varphi(w,[x,y] = \frac{1}{2}\|w\|^2 + \eta_t \textstyle\sum_{i=1}^{n} max\,\{0,1 - y_i(w \cdot x_i)\} \qquad (10)$$

The Iterative Method of SGD:

The stochastic (online) gradient descent is performed with respect to the objective function in Eqn. (10) and approximated by the instantaneous error in Eqn. (9) on a single example. The update rule is as follows:

For randomly class k, $1 \le k \le n$, training example $(t > 0)$, learning rate (η_t), subgradient w. r. t. w_t (∇w_t).

$$w_{t+1} = w_t - \eta_t \nabla w_t [\tfrac{1}{2}\|w^2\| + \tfrac{1}{\lambda} max\,\{0,1 - w_t \cdot x_k\} \qquad (11)$$

Hence, let learning rate $\eta_t = \frac{1}{\lambda t}$ for convergence analysis of stochastic approximations which will satisfy the conditions $\sum_{i=0}^{\infty} \eta_t < \infty$ and $\sum_{i=0}^{\infty} \eta < \infty$ as t increases

Using $\eta_t = \frac{1}{\lambda t}$ in Eqn (11) $\Rightarrow$

$$w_{t+1} = w_t - \frac{1}{\lambda t} w_t - \frac{1}{\lambda(t+1)} x_t \qquad (12_a)$$

For and $w_{t+1} = w_t - \frac{1}{\lambda t} w_t - \frac{1}{\lambda(t+1)} x \le 1$

$$w_{t+1} = w_t - \frac{1}{\lambda t} w_t \qquad (12_b)$$

Algorithm 2: SVM-SGD for Multiclass in Binary Optimisation Classification

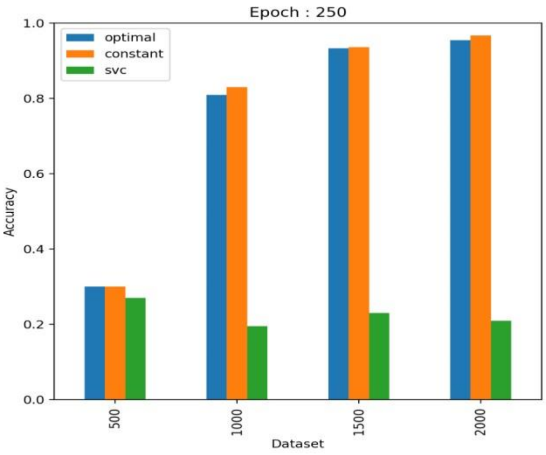
```
Input: Training data $k^+ \in D$ for positive class and $k^- \in D$ for negative class, epoch $T$, constant $\lambda > 0$
Output: SVM-SGD $w$
Initialisation: $w_1 = 0$
for $t = 1$ to $T$ do
    form reduced dataset $D$ from the set of positive class $k^+$ and sampling without replacement $D'$ from dataset $k^-$
    setting $\eta_t = \frac{1}{\lambda t}$
    for $i = 1$ to $k^+$ do
        randomly select a datapoint $[x_i, y_i]$ from reduced set of $k^-$
        if $y_i(w_i \cdot x_i) < 0$ then
            $w_{t+1} = w_t - \frac{1}{\lambda t} w_t - \frac{1}{\lambda(t+1)} x_t$
        else if $y_i(w_i \cdot x_i) < 1$ then
            $w_{t+1} = w_t - \frac{1}{\lambda t} w_t - \frac{1}{\lambda(t+1)} x_t - [1 - y_i(w_i \cdot x_i)$
        else
            $w_{t+1} = w_t - \frac{1}{\lambda t} w_t$
        end
    end
end
return $w_{t+1}$
```

4.0 EXPERIMENTAL ANALYSIS

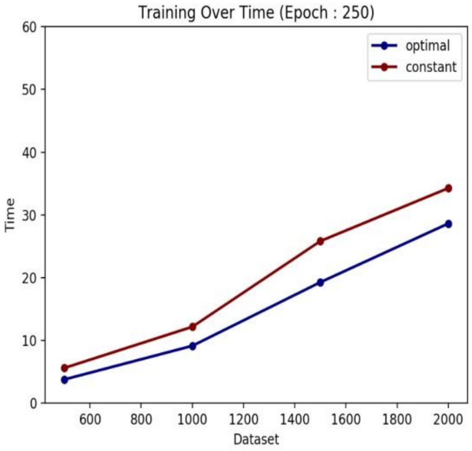
With a baseline support vector values of 0.270, 0.195, 0.230, 0.210 for epochs 250, 500, 1000 respectively at different Learning Rate (LR) such as optimal and constant are used with training time as described in table 1, 2 and 3 and also in figure 2, 3 and 4. The datasets are classified into five (5) different categories: Politics, Technology, Sports, Entertainment and Business.

Table 1: The epoch 250 on Optimal and Constant LR with Training Time

Dataset	Optimal	Time	Constant	Time
500	0.3	3.782	0.3	5.611
1000	0.81	9.142	0.83	12.177
1500	0.933	19.276	0.936	25.832
2000	0.955	28.635	0.967	34.272



(a)

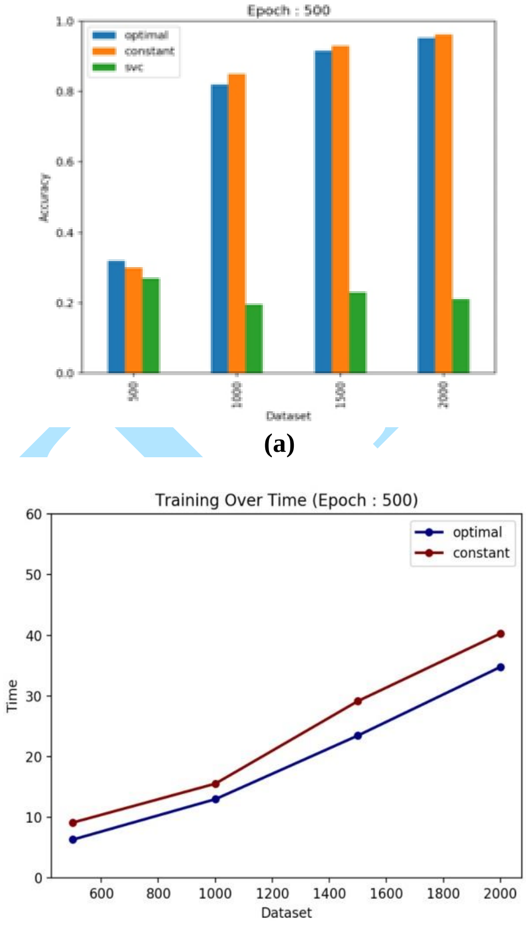


(b)

Figure 2a and 2b: The epoch 250 on Optimal and Constant LR with Training Time

Table 2: Showing the epoch 500 on Optimal and Constant LR with Training Time

Dataset	Optimal	Time	Constant	Time
500	0.32	6.325	0.3	9.134
1000	0.82	12.989	0.85	15.547
1500	0.916	23.48	0.93	29.162
2000	0.9525	34.789	0.962	40.304



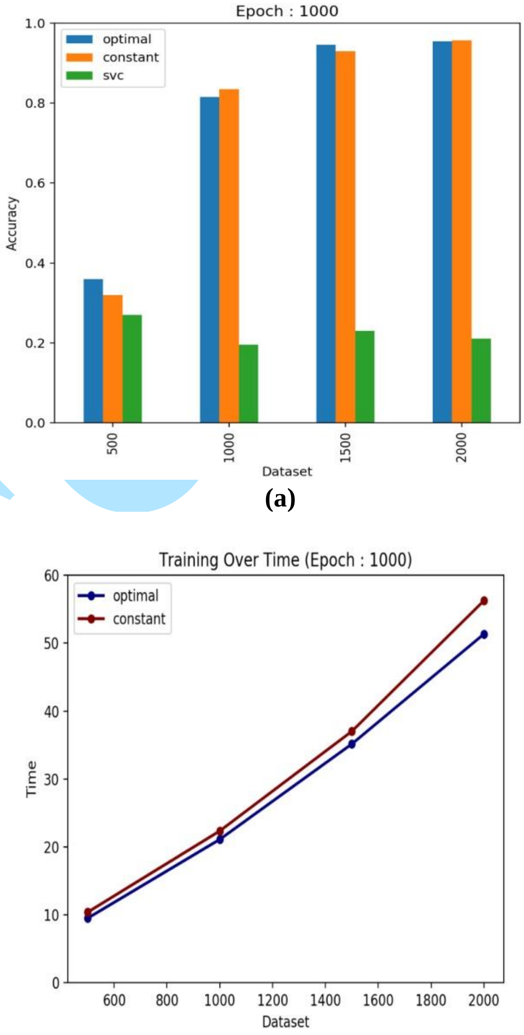
(a)

(b)

Figure 3a and 3b: The epoch 500 on Optimal and Constant LR with Training Time

Table 3: Showing the epoch 1000 on Optimal and Constant LR with Training Time

Dataset	Optimal	Time	Constant	Time
500	0.36	9.51	0.32	10.393
1000	0.815	21.11	0.835	22.354
1500	0.946	35.16	0.93	37.025
2000	0.955	51.35	0.9575	56.265



(a)

(b)

Figure 4a and 4b: The epoch 1000 on Optimal and Constant LR with Training Time

The effect of parameters: training set $(T > 0)$, learning rate (η_t) with time taken are investigated on the fraction of margin errors, as well as observed the behaviour on MC-SVMs. From the tables and figures, it shows that for epoch of 250, as the dataset increases, the time taken for constant learning rate is higher than at optimal level but with little changes in the learning rate as well as faster than SVM. However, for epoch of 500, the optimal has better accuracy with minimum learning rate. Subsequently, the accuracy increases as the number of datasets increase with the number of epoch increase to 1000. It shows that the greater the number of epoch, the better the learning rate.

5.0 CONCLUSION AND FUTURE WORK

The performance of the MC-SVM in term of classification accuracy increases using skip-gram with negative sampling and a window base embedding model. It maximises the probabilities of words being predicted by its context words. But reduce the computational cost as well as the time taken for optimal learning rate. Consequently, as datasets increases with an appropriate learning rate, SGD converges and speed-up the training process of the classifier. Stochastic gradient descent method with constant learning rate shows decreasing learning rate with low convergence rate Future research requires more comprehensive testing of the algorithm and application to real-world problems on very large datasets with sophisticated machine (GT FORCE) to solve big data.

REFERENCES

[A+18] **K.-K. A. Abdullah, S. O. Folorunso, O. O. Solanke, S. M. Sodimu** – *A Predictive Model for Tweet Sentiment Analysis and Classification*. Journal of Annals, Computer Science Series. 16:2, 35-44, 2018.

[BMS16] **R. Babbar, K. Maundet, B. SchoÈlkopf** – *TerseSVM: A Scalable Approach for Learning Compact Models in Large-scale Classification*. Proceedings of the 2016 SIAM International Conference on Data Mining. SIAM; 234-242, 2016.

[B+01] **A. Ben-Hur, D. Horn, H. Siegelmann, V. Vapnik** – *Support vector clustering*. Journal of Machine Learning Research, vol. 2:125-137, 2001.

[CF09] **A. C. de Carvalho, A. A. Freitas** – *A tutorial on multi-label classification techniques*, Foundations of Computational Intelligence Springer, 2009 5:177-195, 2009.

[CS02] **K. Crammer, Y. Singer** – *On the algorithmic implementation of multiclass kernel-based vector machines*. Journal of Machine Learning Research. 2:265-292, 2002.

[DT05] **N. Dalal, B. Triggs** – *Histograms of oriented gradients for human detection*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 886-893. 2005.

[DHS11] **J. Duchi, E. Hazan, Y. Singer** – *Adaptive subgradient methods for online learning and stochastic optimization*. Journal Machine Learning Research, 12:2121-2159, 2011.

[Gue02] **Y. Guermeur** – *Combining discriminant models with new multiclass SVMs*. Pattern Analysis and Applications, 5:168-179. 2002.

[GY14] **G. Geetika, D. Yadav** – *Sentiment analysis of twitter data using machine learning approaches and semantic analysis*. 2014 Seventh International Conference on Contemporary Computing (IC3), IEEE, 437-442, 2014.

[GBW14] **M. R. Gupta, S. Bengio, J. Weston** – *Training highly multiclass classifiers*. Journal of Machine Learning Research. 15:1, 1461-1492, 2014.

[GME17] **D. Gonzalez-Marron., D. Mejia-Guzman, A. Enciso-Gonzalez** – *Exploiting Data of the Twitter Social Network Using Sentiment Analysis*. In: Sucar E., Mayora O., Munoz de Cote E. (eds) Applications for Future Internet. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer, Cham 179, 2017.

[HL02] **C.-W. Hsu, C.-J. Lin** – *A comparison of methods for multiclass support vector machines*, IEEE Transactions on Neural Networks, 13:415-425, 2002.

[LXW03] **X. Liu, H. Xing, X. Wang** – *A multistage support vector machine*. 2nd International Conference on Machine Learning and Cybernetics, 1305-1308, 2003.

[MSC13] **T. Mikolov I. Sutskever, K. Chen** – *Distributed representations of words and phrases and their compositionality*. Advances neural information processing systems, pp. 3111-3119, 2013.

[M+02] **S. Morinaga, K. Yamanishi, K. Teteishi, T. Fukushima** – *Mining product reputations on the web*. Proc. of the 8th ACM SIGKDD International Conference 341-349. 2002.

[NWH17] **F. Nie, X. Wang, H. Huang** – *Multiclass capped lp-norm svm for robust classifications*. Thirty-First AAAI Conference on Artificial Intelligence, 2415-2421, 2017.

[PA92] **B. T. Polyak, B. J. Anatoli** – *Acceleration of stochastic approximation by averaging*, SIAM Journal on Control and Optimization 30(4) 838-855, 1992.

[PT09] **R. Prabowo, M. Thelwall** – *Sentiment analysis: A combined approach*. Journal of Informatics 143-157, 2009.

[PLV02] **B. Pang, L. Lee, S. Vaithyanathan** – *Thumbs up? Sentiment classification using machine learning techniques*. Proc. of the 2002 ACL EMNLP Conf., 79-86, 2002.

[P+15] **M. Pratama, S. G. Anavatti, J. Meng, E. D. Lughofer** – *pClass: An Effective Classifier for Streaming Examples*, IEEE Transactions on Fuzzy Systems, 23: 369-386, 2015.

[RK14] **R. Rifkin, A. Klautau** – *In defense of one-vs-all classification*. Journal of Machine Learning Research. 5:101-141. 2004.

[Tha14] **D. Thanh-Nghi** – *Parallel multiclass stochastic gradient descent algorithms for classifying million images with very-high-dimensional signatures into thousands classes*. Vietnam Journal of Computer Science. https://doi.org/10.1007/s40595-013-0013-2, 1(2):107-115, 2014.

[WM12] **S. Wang, C. D. Manning** – *Baselines and bigrams: Simple, good sentiment and text classification*. Proceedings of ACL, 90-94, 2012.

[WW99] **J. Weston, C. Watkins** – *Support vector machines for multiclass pattern recognition*. in: Verleysen M (ed.) Proceedings of the Seventh European Symposium on Artificial Neural Networks (ESANN). Evere, Belgium: d-side publications; 219-224, 1999.

[X+17] **J. Xu1, X. Liu, Z. Huo, C. Deng, F. Nie, H. Huang** – *Multiclass Support Vector Machine via Maximizing Multiclass Margins*. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17), pp 3154-3160, 2017.

[YTA14] **Y. Yuki, K. Tadahiko, N. Akiyo** – *Role of Emoticons for Multidimensional Sentiment Analysis of Twitter*. Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services. ACM., 107-115, 2014.

[Zan04] **T. Zhang** – *Solving large scale linear prediction problems using stochastic gradient descent algorithms*. Twenty-first International Conference on Machine learning; Omnipress. 2004.