Kernel Optimized-Support Vector Machine and Mapreduce framework for sentiment classification of train reviews

RASHMI K THAKUR^{1,*} and MANOJKUMAR V DESHPANDE²

¹Mukesh Patel School of Technology Management & Engineering, Narsee Monjee Institute of Management Studies, Mumbai 400 056, India

²Prestige Institute of Engineering Management and Research, Indore, India
e-mail: thakurrashmik@gmail.com; manojvilasrao@gmail.com

MS received 26 May 2017; accepted 7 July 2018; published online 17 December 2018

Abstract. Sentiment analysis is one of the popular techniques gaining attention in recent times. Nowadays, people gain information on reviews of users regarding public transportation, movies, hotel reservation, etc., by utilizing the resources available, as they meet their needs. Hence, sentiment classification is an essential process employed to determine the positive and negative responses. This paper presents an approach for sentiment classification of train reviews using MapReduce model with the proposed Kernel Optimized-Support Vector Machine (KO-SVM) classifier. The MapReduce framework handles big data using a mapper, which performs feature extraction and reducer that classifies the review based on KO-SVM classification. The feature extraction process utilizes features that are classification-specific and SentiWordNet-based. KO-SVM adopts SVM for the classification, where the exponential kernel is replaced by an optimized kernel, finding the weights using a novel optimizer, self-adaptive lion algorithm. In a comparative analysis, the performance of KO-SVM classifier is compared with SentiWordNet, Naive Bayes, neural network, and LSVM, using the evaluation metrics, specificity, sensitivity, and accuracy, with train review and movie review database. The proposed KO-SVM classifier could attain maximum sensitivity of 93.46% and 91.249%, specificity of 74.485% and 70.018%; and accuracy of 84.341% and 79.611%, respectively, for train review and movie review databases.

Keywords. Sentiment classification; SVM; MapReduce; LA; sentiwordnet.

1. Introduction

Public transportation plays a major role in creating supportive communities, where a group of people frequently book transportation tickets to reach the same destination, as they may have the same purpose of visit, like sightseeing, doing business, or visiting relatives [1]. Therefore, inferring the purpose of travel of the passenger groups may help the transporters in providing particular and personalized recommendations or services to the passengers. To evaluate the passenger service, few measurements like waiting time, number of transfers, in-vehicle time, and so on, are analyzed from the experienced journey of the passengers [2]. Presently, people share their opinions on various problems related to transportation, such as traffic jams, traffic collisions, landslides, rockslides, etc., through the social media sites. Based on reviews made by the people regarding the same subject, namely the jamming of roads or city streets, street-side organizations, and so on, new clients gather to react. On the contrary, immediate and safe routes can be determined through the web surfers due to the tweets or the reviews made [3]. Sentiment analysis plays a

crucial role in reacting to the reviews or the tweets such that the user satisfaction is met [4].

Sentiment analysis, which is also named as opinion mining [5], is an emerging research space in natural language processing, where opinions, appraisals, attitudes, sentiments or emotions of the people toward entities are analyzed and are expressed in the texts [6, 7]. Due to the recent advancements made in this field, the topic has drawn attention from research and industrial communities [8]. The advantages of sentiment analysis are many in different areas, like education, e-commerce, and opinion polls [9, 10]. Sentiment classification, which is a kind of text classification approach, has become a growing research field in sentiment analysis due to the large development of subjective texts in the forums social media, and blogs [11]. Sentiment analysis process utilizes a sentiment lexicon comprised of few words and phrases that can convey positive or negative sentiments. However, for sentiment classification, a sentiment lexicon alone is not adequate due to the opposite orientations of the words and the phrases in various domains [10].

There are various terms given by the researchers for sentiment classification, namely subjectivity analysis, opinion mining, review mining, sentiment analysis, and

*For correspondence

opinion extraction [12]. Sentiment classification has considerable importance in day-to-day life, commercial activities, activities associated with commodity production, and political activities. Identifying a solution for the classification of emotion effectively and appropriately is a challenge [13]. Traditional text classification approaches utilize several classical techniques, like Chi-square test, Document Frequency (DF), and so on, for the feature extraction [14]. However, these approaches are not directly applicable to sentiment classification [15]. In sentiment analysis, one of the fundamental and widely studied research areas is sentence-level sentiment classification. Normally, a lexicon of sentiment words is utilized in Lexicon-based approaches [16], where each of the words is interpreted with its sentiment strength. Intensifications and negations are some of the linguistic rules that are integrated to collect the sentiment polarity of the sentences. Sentiment classification is considered as a special kind of problem in text classification based on corpus techniques [7]. In such techniques, the sentiment classifier is built from the sentences having sentiment polarity annotation. The sentiment supervision can be done either by manual annotation or can be collected automatically using sentiment signals, including emoticons in tweets [17] or review ratings [8, 18].

This paper presents a technique for sentiment classification of train reviews using novel KO-SVM and MapReduce approach. The MapReduce technique performs sentiment analysis on big data utilizing mapper and reducer, where the reducer performs classification using the proposed KO-SVM classifier. In the mapper, feature extraction is done over the pre-processed data that transforms the reviews into a feature vector. Various features including All-caps, Emoticon, Hashtag, Elongated words and punctuation marks and that developed with the utilization of SentiWordNet are considered for the feature extraction. For the classification, a classifier KO-SVM is developed by modifying SVM with an optimized kernel, where the selection of weighted coefficients is based on SLA with a newly formulated update rule.

The main contributions of the proposed scheme of sentimental classification are as follows,

- Development of a feature extraction model with six new features contributed by SentiWordNet in addition to five classification-specific features developed for the effective classification in sentiment analysis.
- Introduction of KO-SVM to classify the positive and negative reviews by modifying SVM with an optimized kernel function, which has weighted coefficients that are selected using a new optimization algorithm, SLA.

The paper is structured as follows: Section 2 summarizes the literature study, where the existing techniques of sentiment classification are described. In section 3, the proposed KO-SVM classifier used for the classification is

explained. Section 4 illustrates how the proposed classifier is adapted to the MapReduce model with architecture. The results discussing the performance of the proposed approach is given in section 5, and in section 6 the conclusions are provided.

2. Motivation

2.1 Literature review

Different methods used for sentiment classification in the existing systems are reviewed in this section, stating their limitations and the challenges faced.

Duyu Tang *et al* [8] had presented a technique for sentence-level sentiment classification using a framework of joint segmentation and classification. This technique could generate the segmentation by predicting sentence-level polarity simultaneously depending on the segmentation results. Three models: candidate generation model, segmentation ranking model, and classification model, were developed to produce segmentation candidates of a sentence; to assign a score to the segmentation candidate; and to predict the sentiment polarity, respectively. The major limitation is that all the expressions cannot be composed by its constituent meaning.

Çagatay Catal and Mehmet Nangir [10] developed a classification approach by investigating the advantage of multiple classifier systems on classification problem focussing on the reviews written in Turkish. Vote algorithm was used with three classifiers, such as Support Vector Machine (SVM), Naive Bayes (NB), and Bagging. The comparative assessment was made with various meta-classifiers, ensemble and parameter optimization methods for the sentiment classification. Even though the approach had better performance, it was expensive and consumes more time for larger datasets.

Chihli Hung and Hao-Kai Lin [19] had re-evaluated the words in SentiWordNet by measuring the sentimental relevance of objective words and their related sentiment sentences. The authors had developed two sampling approaches—random sampling without replacement and position-based document segmentation, which were integrated with the SVMs for the sentiment classification. The decision-making process is better with good classification accuracy, but it fails to consider word sense disambiguation during extraction.

Vo Ngoc Phu *et al* [13] developed a method for sentiment classification of big data in the parallel network environment. This model used the Fuzzy C-Means (FCM) method for English sentiment classification with Hadoop MAP (M) /REDUCE (R) in Cloudera, which was a parallel network environment. This approach could perform the classification of sentiments of millions of English documents in the Cloudera. It could process big data comprised of millions of English documents in a short execution time.

However, it requires more time for the implementation and is expensive.

Mohammad Salehan and Dan J Kim [20] had presented a sentiment mining method by examining the predictors of readership and helpfulness of Online Consumer Reviews (OCR) for big data analytics. The length and longevity of a review could influence the readership and helpfulness positively. Since other methods of OCR sorting may bias their readership and helpfulness, this approach could be used by online vendors to build automated systems for the classification and sorting of big OCR data. The drawback here is that it did not consider the differences in expressing emotions between the actual and the virtual space.

Farman Ali *et al* [4] designed a method for sentiment analysis and decision-making based on fuzzy ontology and Semantic Web Rule Language (SWRL), to monitor the activities of transportation, such as traffic volume, accidents, street conditions, and vehicles, so that city-feature polarity map could be made for the travelers. This method could retrieve reviews and tweets associated with transportation activities and city features. From the data retrieved, the feature opinions were extracted, and the transportation and city-feature polarity were determined using fuzzy ontology. The categorization was done successfully, but irrelevant reviews were executed reducing the precision rate of analysis.

Tao Chen *et al* [21] had presented a divide-and-conquer technique that classified the sentences into various types, and then, sentiment analysis was carried out individually on sentences from each type. The authors designed a sequence model using a neural network to classify opinionated sentences into three types based on the number of targets in a sentence. Each sentence group was applied into a one-dimensional Convolutional Neural Network (CNN) for the sentiment classification. Even though performance enhancement is possible, it did not discover other learning methods for the detection of target expressions.

Jinyan Li *et al* [15] evaluated various classification techniques to develop an approach, called hierarchical classification, using three filtering methods that could reduce the original data set according to the contextual polarity and repeated terms of a document. The performance of this approach in various combinations of classification and filtering techniques was compared with that of three sets of online news articles that performed binary and multi-class classifications. The limitations found in this technique are difficulty in achieving higher performance and lack of sensitivity towards the missing data.

2.2 Challenges

From the study made on the existing techniques of sentimental classification, the challenges that various techniques possess are as follows,

- One of the main issues in sentimental classification is due to the online reviews comprised of user opinions in the form of abbreviations, conjoined words, and shortenings, which are commonly used by the people to express their feelings about a particular point. The existing systems could not cover the entire online reviews and had polarity inconsistency [15].
- Another challenge observed in classification techniques deals with the selection of the best feature that provides a better result. The features to be selected include n-grams, words, syntactic n-grams of different types or a combination of these features. Feature size is also a problem that most of the existing schemes face [13].
- The third challenge in classification is regarding the precision factor, as it gets affected due to the usage of nouns, adjectives, verbs, etc., by the classifiers rather than considering only the sentimental words in the reviews [4]. Also, the existing techniques had concentrated mainly only on the sentence polarity instead of the sentence type [21].
- Certain classification techniques had utilized only the numerical rating and the word count of the review to evaluate the performance of the approach. Those techniques failed to consider the positive review, the negative review, and the number of users [20].
- In [11], Linear SVM (LSVM) was used for sentiment classification after the extraction of the segment level features. Since LSVM is a traditional approach, it has several issues regarding overfitting and convergence within the bound of optimization problems, which limit the approach.
- FCM clustering with MapReduce approach was used in [13] for sentiment classification. This classification approach did not consider any supervised learning methods for the classification. The facts that limit the approach are that the manual classification is very challenging and it is expensive, taking more time for the big data.

3. The proposed method of sentiment classification of train reviews using KO-SVM and MapReduce algorithm

This section presents the framework developed for the sentiment classification using MapReduce mechanism with the proposed KO-SVM classifier. The sentiment analysis using MapReduce framework is comprised of mapper and reducer that performs feature extraction and classification. The mapper pre-processes the reviews in the input database and creates a feature vector considering eleven important features using SentiWordNet for a semi-supervised classification, in the feature extraction process. The utmost intention of this paper is to categorize the reviews into two

classes: positive and negative class. The proposed KO-SVM classifier employed in the reducer performs the classification to find positive and negative reviews. KO-SVM is designed by adopting SLA to determine the weighted coefficients in the newly designed kernel function. For the feasible selection of weights in the kernel function, a multi-objective fitness function, with sensitivity, specificity, and accuracy, as three objectives, is utilized. The block diagram of the proposed mechanism of sentiment classification is depicted in figure 1.

3.1 Pre-processing

The classification process is initiated by pre-processing, where the input database is pre-processed. Let D be the database that consists of a number of train reviews, as represented below,

$$D = \{S_1, S_2, \dots, S_n\}; 1 \leq i \leq n \quad (1)$$

where, n is the total number of reviews in D . Here, the reviews are considered as sentences, which is to be identified positive or negative. S_i represents the i^{th} review comprised of a number of words, given as

$$S_i = \{w_1, w_2, \dots, w_m\}; 1 \leq j \leq m \quad (2)$$

where, m is the total number of words available in the i^{th} review.

The input database D undergoes stop word removal [22], where the stop words, which are the meaningless words (also known as “functional words”), are eliminated. The words that do not provide any significant meaning in one context may provide some information in another. Hence, it is important to choose a method for the stop word removal wisely. The database with the stop words removed is subjected for the feature extraction process.

3.2 Feature extraction based on SentiWordNet

In feature extraction, the input database is converted into a feature vector such that the size of the features is reduced, which finds a major application in handling big data. In the proposed approach, five features called classification-specific features [8] that influence the classification, and six statistical features developed using SentiWordNet are extracted that aid the classification algorithm to provide an effective classification of reviews. SentiWordNet [23] is a resource that contains a dictionary of opinion information obtained from the WordNet database. The five features adopted for the classification are All-Caps, Emoticon, Hashtag, Elongated units and Punctuation. A feature vector

Figure 1. Block diagram of the proposed technique of sentiment classification using MapReduce and KO-SVM classifier.

is thus constructed with the eleven considered features. Let F_i be the feature vector of i^{th} review, as represented below,

$$F_i = \{f_i^1, f_i^2, \dots, f_i^l, \dots, f_i^{11}\} \quad (3)$$

where, f_i^l represents the l^{th} feature in the feature vector.

The feature f^1 represents the All-Caps feature, which denotes the number of capitalized words in a review, given as,

$$f^1 = \sum_{i=1}^m w_C^i \quad (4)$$

where, w_C^i represents the number of words with all the letters in upper case. It assumes a value 1 or 0 based on a condition that depends on the presence or the absence of capitalized words, as given below,

$$w_C^i = \begin{cases} 1; & \text{if caps word} \\ 0; & \text{otherwise} \end{cases} \quad (5)$$

f^2 is the second feature that represents either the positive or the negative emoticons in a review, given by,

$$f^2 = \sum_{i=1}^m s_E^i \quad (6)$$

where, s_E^i is the number of emoticons present in the i^{th} review and considers a value either 1 or 0 based on its presence in the review, as represented,

$$s_E^i = \begin{cases} 1; & \text{for the presence of emoticon} \\ 0; & \text{otherwise} \end{cases} \quad (7)$$

The third feature considered indicates the presence of hashtags in a review and is given by,

$$f^3 = \sum_{i=1}^m s_H^i \quad (8)$$

where, s_H^i is the number of hashtags in the i^{th} review.

f^4 is the fourth feature that depends on the elongated words, which have a character repeated more than twice, in a review and is defined as,

$$f^4 = \sum_{i=1}^m w_E^i \quad (9)$$

where, w_E^i is the total number of elongated words in S_i . The term is assigned with a value 1, for each of the elongated words present in the review and 0, in the absence of an elongated word.

The fifth feature is the punctuation, which may be an exclamation mark, apostrophe, dot and so on, in a review.

$$f^5 = \sum_{i=1}^m s_p^i \quad (10)$$

where, s_p^i is the number of punctuations occurred in the i^{th} review. The value of s_p is 1, if there occurs punctuation in the review and 0, for the other case.

SentiWordNet [24] groups the words into several sets of synonyms, named synsets. All the synsets are related with a score of polarity, such as positive, negative and objective. The scores take a value between 0 and 1, and their sum gives a value 1 for each synset. Based on the scores provided, it is possible to decide whether the opinion is positive, negative or objective. The words in the SentiWordNet database are based on the parts of speech obtained from WordNet, and hence, it utilizes a program to apply the scores to each word. The weight adjustment of score values can be represented as,

$$[\phi^i(p), \phi^i(n), \phi^i(o)] = h(w_i) \quad (11)$$

where, $\phi^i(p)$ is the positive score, $\phi^i(n)$ is the negative score, $\phi^i(o)$ is the objective score and h is the SentiWordNet function.

A word in the review is said to be positive, denoted as $\{w^P\}$, such that $\{w^P\} < w_j$ if $\phi^i(p) > \phi^i(n)$. Similarly, the term is negative, represented as $\{w^N\} < w_j$, if $\phi^i(p) < \phi^i(n)$. When the score is zero, the word is neither positive nor negative and thus, is said to be an objective word. Depending on the categorization made using SentiWordNet, six additional features can be extracted by utilizing the statistical information, as defined below,

$$f^6 = \frac{1}{|w^P|} \sum_{j=1}^m \phi_j(p) \quad \forall w_j \in w^P \quad (12)$$

where, $\phi_j(p)$ is the positive score of the j^{th} word and w^P represents the positive word. The seventh feature is determined based on the negative score as,

$$f^7 = \frac{1}{|w^N|} \sum_{j=1}^m \phi_j(n) \quad \forall w_j \in w^N \quad (13)$$

where, $\phi_j(n)$ is the negative score of the j^{th} word and w^N represents the negative word. The next two features, i.e., f^8 and f^9 are based on the frequency of positive and negative words that makes the identification of positive and negative review easy.

$$f^8 = \frac{1}{|w^P|} \sum_{j=1}^m \phi_j(p) * f_j \quad \forall w_j \in w^P \quad (14)$$

where, f_j is the frequency of positive words used in a review, which indicates the word belongs to a positive word.

$$f^9 = \frac{1}{|w^N|} \sum_{j=1}^m \phi_j(n) * f_j \quad \forall w_j \in w^N \quad (15)$$

where, f_j in Eq. (15) is the frequency of negative words used in a review, representing that the j^{th} word is a negative word. The last two features are developed in association with the variance to find whether the word belongs to a positive word or not. f^{10} is defined in terms of variance computation by taking the difference between one of the features belonging to the positive word and the positive score as

$$f^{10} = \frac{1}{|w^P|} \sqrt{\sum_{j=1}^m (f^6 - \phi_j(p))^2} \quad \forall w_j \in w^P \quad (16)$$

where, f^6 represents the sixth feature and $\phi_j(p)$ is the positive score of j^{th} the frequency of positive words used word.

The last feature takes the variance measured between the feature pointing negative word and the negative score, stating that the word is negative, as defined below,

$$f^{11} = \frac{1}{|w^N|} \sqrt{\sum_{j=1}^m (f^7 - \phi_j(n))^2} \quad \forall w_j \in w^N \quad (17)$$

where, f^7 represents the feature seven and $\phi_j(n)$ is the positive score of the j^{th} word.

Thus, the review database is converted into a feature vector of dimension 1×11 , denoted as $[F_i]_{1 \times 11}$. The feature vector is subjected to the classification performed by KO-SVM classifier, as explained in the following section.

3.3 Proposed KO-SVM classifier for the sentiment classification

The proposed classifier, KO-SVM used for the classification of reviews is summarized in this section. KO-SVM is designed by modifying the exponential kernel in the SVM classifier by an optimized kernel that determines the hyperplane based on weighted coefficients selected by SLA. SVM [25] can solve unconstrained optimization problems using a large-margin classifier that maximizes the margin to find the optimal hyperplane. This training algorithm can adjust the capacity providing better classification accuracy. Let the input vector F_i comprised of n reviews belongs to either of the two classes, Class 1 or Class 2, represented as given in Eq. (18),

$$(F_1, x_1), (F_2, x_2), \dots, (F_n, x_n) \\ \forall \begin{cases} x_v = 1; & \text{if } F_v \in \text{Class1}(w^P) \\ x_v = -1; & \text{if } F_v \in \text{Class2}(w^N) \end{cases} \quad (18)$$

where, x_v represents the class labels. Based on the rule provided below, the classification is performed,

$$\begin{aligned} F &\in \text{Class1} ; \text{ if } N(F) > 0 \\ F &\in \text{Class2} ; \text{ otherwise} \end{aligned} \quad (19)$$

where, $N(F)$ is the decision function, designed based on certain parameters that are determined by the classification algorithm and is given by,

$$N(F) = \sum_{v=1}^n \alpha_v k(F_v, F) + d \quad (20)$$

where, α_v is a parameter, d is the bias and $k(F_v, F)$ is the kernel function. SVM classifier usually adopts an exponential or Gaussian kernel function, which is defined as follows,

$$k(F_v, F) = \exp(-\gamma \|F_v - F\|^2) \quad (21)$$

where, γ is a constant. This kernel function is replaced in KO-SVM by an optimized function based on a regression coefficient, as the weights it considers are varying rather than constants.

$$k(F_v, F) = \frac{1}{K} \sum_{i=1}^K \exp(-\gamma_i \|F_v - F\|^2 * \alpha_i) \quad (22)$$

where, K is the regression coefficient. γ_i and α_i are the weighted coefficients, which are selected using the proposed SLA model that will be discussed in section 3.3a.

To maximize the margin, the norm of a weight vector W is to be minimized. This is a difficult task when the space dimensionality is large. Hence, it adopts the Lagrangian model, given by,

$$L(W, d, \alpha) = \frac{1}{2} \|W\|^2 - \sum_{v=1}^n \alpha_v [x_v N(F_v) - 1] \quad \forall \alpha_v \geq 0 \quad (23)$$

where, $N(F_v)$ is the decision function, and α_v is termed as Lagrange multiplier that obeys the following condition,

$$\alpha_v [x_v N(F_v) - 1] = 0 ; 1 \leq v \leq n \quad (24)$$

The solution to this problem is finding a saddle point, which is the minimum of $L(W, d, \alpha)$ in terms of W and the maximum with respect to α_v . Finally, by solving a partial derivative based condition, the following vector is obtained,

$$W^* = \sum_{v=1}^n \alpha_v^* x_v \delta_v \quad (25)$$

where, δ_v represents the function that is predefined in the dual space. According to Eq. (25), the hyperplane having maximum margin is equivalent to a linear combination of patterns for which α_v^* is nonzero. Following this equation, the decision function becomes,

$$N(F) = \sum_v x_v \alpha_v^* k(F_v, F) + d \quad (26)$$

where, $k(F_v, F)$ is the kernel function and d is the bias.

3a SLA-based generation of weighted coefficients In this section, the proposed SLA model used for the selection of

kernel weights utilized in the KO-SVM classifier is described. The proposed SLA model modifies the Lion Algorithm (LA) [26] by the introduction of a new function in the fertility evaluation to make it self-adaptive. LA is an optimization algorithm inspired by the unique behavior of lion. The selection of a solution is based on two behaviors, such as territorial defense and territorial takeover, of a lion. The former is carried out between the resident males and the nomadic males, whereas, the later is between the old and new territorial males. Adopting LA, optimal selection of weighted coefficients in large search space is possible. The following divisions explain the proposed approach in detail.

A. Solution encoding

The simplest form of explaining the procedure of an algorithm is by solution encoding. The proposed SLA model intends to find the weighted coefficients in the kernel function using an appropriate fitness function, for the sentiment classification. The aim is to select the weights γ and α optimally, in the kernel function of KO-SVM. Let P be the solution set, represented as $P = \{1, 2, \dots, r\}$, where r is the total number of solutions. Each solution P_i ; $1 \leq i \leq r$ in the solution set P is of dimension u that is represented using binary elements. The algorithm selects the solution optimally based on a multi-objective fitness function. Once the kernel weights are selected, the classification will be performed using KO-SVM, in the reducer employed in the MapReduce mechanism.

B. Proposed SLA model

The SLA model designed by utilizing LA with an effectual modification is illustrated in this section. In SLA, the update equation is modified by the inclusion of a function so that the algorithm becomes self-adaptive and can solve the optimization problems. The reason to make the algorithm self-adaptive is to cope with the issues that occur at various stages of the optimization process depending on the quality of the solutions generated by the algorithm. LA model follows the social behavior of lion, where the strongest lion owns the territory. The steps involved in SLA is described as follows,

I. Initialization

The algorithm begins with the initialization of lion population, where the pride, Y^m and Y^f , which denotes the male and the female, are initialized, as represented below,

$$Y^m = \{Y_1^m, Y_2^m, \dots, Y_L^m\}; 1 \leq i \leq L \quad (27)$$

$$Y^f = \{Y_1^f, Y_2^f, \dots, Y_L^f\}; 1 \leq i \leq L \quad (28)$$

where, L is the length of the population such that $L > 1$.

Moreover, the nomadic lion, denoted as Y^n , is also initialized, in addition to the initialization of the pride. Initially, one of the nomadic lions Y_1^n will be initialized,

whereas, the other will be initialized during the territorial defense.

II. Objective function evaluation

The objective function that decides the strength of the solution is formulated in SLA considering three objectives, such as sensitivity, specificity, and accuracy. The multi-objective fitness function is defined as,

$$F^{SLA} = \frac{1}{3} [S^V + S^C + A^R] \quad (29)$$

where, S^V , S^C and A^R represent the sensitivity, specificity, and accuracy.

Sensitivity: It refers to the proportion of positives that are identified correctly, as given below,

$$S^V = \frac{T_P}{T_P + F_N} \quad (30)$$

where, T_P is true positive and F_N is a false negative.

Specificity: It is defined as the proportion of true negatives that are obtained correctly.

$$S^C = \frac{T_N}{T_N + F_P} \quad (31)$$

where, T_N is true negative and F_P is false positive.

Accuracy: Accuracy measures the degree of closeness specifying the proportion of true results, represented as follows,

$$A^R = \frac{T_N + T_P}{T_N + T_P + F_N + F_P} \quad (32)$$

Each solution in the current population will be evaluated using Eq. (29) and the solution having maximum fitness will be considered as the best solution.

III. Fertility evaluation by updating

In this step, the fertility of the pride is evaluated to avoid convergence at local minima. Let g_r and s_r be the laggardness rate and the sterility rate, which are initialized as zero. The female update function is then derived as follows,

$$Y_i^f(t+1) = \begin{cases} Y_j^f(t+1); & \text{if } i = j \\ Y_i^f(t+1); & \text{Otherwise} \end{cases} \quad (33)$$

$$Y_j^f(t+1) = \min \left[Y_j^{\max}, \max \left(Y_j^{\min}, Y_j^f(t+1) \right) \right] \quad (34)$$

where, $Y_i^f(t+1)$ and $Y_j^f(t+1)$ are the i^{th} and j^{th} vector elements of $Y^f(t+1)$, which is the female update of LA, given by,

$$Y^f(t+1) = \lfloor Y^f(t) + (0.1b_2 - 0.05)(Y^m(t) - b_1 Y^f(t)) \rfloor \quad (35)$$

where, $Y^f(t)$ represents the female lion and $Y^m(t)$ is the male lion generated at the time t . b_1 and b_2 are the random numbers in the interval $[0, 1]$.

In SLA, the female update equation in (35) is modified by the function β that makes the algorithm adaptive to various strategies. This function is computed based on the iteration of the process by a replacement in LA, such that $\beta = 0.05$ and hence, $2\beta = 0.1$. This can improve the searching process to find the optimal solution by making the algorithm adaptive to various fitness landscapes. The function provides a value that determines the updating of the current solution. Therefore, at iteration $t + 1$, the female update is done regarding the female and male solution at previous iteration with the newly designed function, as formulated below,

$$Y^f(t+1) = \lfloor Y^f(t) + (2\beta b_2 - \beta)(Y^m(t) - b_1 Y^f(t)) \rfloor \quad (36)$$

where, the function replaces the rate as $\beta = 0.05$ and is defined as,

$$\beta = \left\lfloor \frac{I_{\max} - I_C + 1}{I_{\max}} \right\rfloor \quad (37)$$

where, $I_{\max}$ is the maximum iteration and I_C is the current iteration. The update equation of SLA based on this function gives preference to strategies necessary on various problems.

IV. Mating

The mating process involves two steps, such as crossover and mutation, by which four cubs will be generated with a probability p . The generation of y^{th} cub can be represented mathematically as,

$$Y^c(y) = C_y \circ Y^m + \bar{C}_y \circ Y^f; y = 1, 2, 3, 4 \quad (38)$$

where, C is the crossover mask whose value depends on p and the operator $\circ$ denotes the Hadamard product. The Cubs obtained undergo mutation to form a number of cubs equivalent to that in the crossover. From a number of cubs generated, two solutions having the first and the second best fitness will be considered as the male cub Y^{m-c} and the female cub Y^{f-c} . These will get replaced by other solutions for which the fitness is in maximum, in the further iterations.

V. Territorial defense The search space is improved in this step by following nomad coalition, survival fight and by updating coalition. The second nomadic lion Y_2^n is initialized in this step based on g_r by mutation. In the survival fight, the nomadic lion that wins in the coalition Y^{e-n} is engaged such that its fitness is lesser than that of Y^m , Y^{m-c} and Y^{f-c} , i.e. $f(Y^{e-n}) < f(Y^m)$, $f(Y^{e-n}) < f(Y^{m-c})$ and $f(Y^{e-n}) < f(Y^{f-c})$. Then Y^m is updated by replacing it with Y^{e-n} , whereas nomad coalition is updated based on the selection of Y^n , for which an exponential function, $E(n)$, computed is greater than the exponential of unity z , as given in Eq. (39),

$$E(n) = \exp \left[\frac{a_1}{\max(a_1, a_2)} \right] \frac{\max(f(Y_1^n), f(Y_2^n))}{f(Y_1^n)} \quad (39)$$

where, a_1 is the Euclidean distance between Y_1^n and Y^m ; and a_2 is the Euclidean distance between Y_2^n and Y^m , respectively.

VI. Territorial takeover

In this stage, the territory will be given to the cubs, Y^{m-c} and Y^{f-c} , once they become stronger than Y^m and Y^f . If Y^{m-c} is better than Y^m , Y^{m-c} replaces the position of Y^m . Y^m remains unchanged for the other case. Once the process of territorial takeover is completed, the iteration t will be incremented by one.

VII. Termination

The procedure is repeated from step II to VI, until a termination criterion is met. When the maximum number of iterations is reached, the algorithmic procedure is halted.

The pseudocode of the proposed SLA employed for the sentiment classification is described in table 1.

4. Adapting KO-SVM- based classification in MapReduce for sentiment classification

This section illustrates the process of adapting KO-SVM classifier in the MapReduce framework for the sentiment classification. It is a programming model that allows parallel processing by splitting a large volume of data into chunks using a map and reduce functions. The input to the MapReduce approach employed here is the train review database. The MapReduce model developed for the proposed technique of sentiment classification is shown in figure 2. It consists of a master node that distributes data to worker nodes, which is comprised of mappers and reducers. As shown in figure 2, let the train review database be denoted as R_W , which is partitioned into n number of reviews, represented as $R_W = \{R_{W1}, R_{W2}, \dots, R_{Wn}\}$. Let U be the mapper, such that $U = \{U_1, U_2, \dots, U_s\}$, where s is the total number of mappers, whereas, $R = \{R_1, R_2, \dots, R_p\}$ represents the p number of reducers, in the MapReduce model.

The map and the reduce function include two concepts, 'key' and 'value'. The mapper includes pre-processing and feature computation, explained in subsections 3.1 and 3.2, whereas the reducer consists of the training algorithm, KO-SVM, for the classification, discussed in subsection 3.3. In the mapper, the review database is pre-processed and converted into feature data by the feature extraction process. The feature data is classified into positive and negative classes by KO-SVM classifier employed in the reducer. The map function processes the input train review database that is partitioned by the master node and forms an intermediate data by generating $\langle \text{key}, \text{value} \rangle$, in which the key is the data, and the value is the feature vector obtained using the SentiWordNet-based feature extraction. The intermediate data will be sent to the reducer and

Table 1. Proposed SLA model.

	<i>Proposed SLA</i>
1	Input: Y^m , Y^f and Y^n
2	Output: Y^m , Y^f and Y^n
3	Begin
4	Initialize the population
5	Initialize g_r and s_r
6	for each solution in Y^m , Y^f and Y^n
7	Compute the fitness using equation (29)
8	Assign the solutions having maximum fitness in the corresponding population as Y^m , Y^f and Y^n
9	end for
10	for ($t < I_{\max}$)
11	Female update using equation (36)
12	Generate cubs through mating using equation (38)
13	if $E_1(n) > z$
14	$Y^n = Y_1^n$
15	else
16	$Y^n = Y_2^n$
17	endif
18	if $f(Y^{m-c}) > f(Y^m)$
19	$Y^m = Y^{m-c}$
20	else
21	$Y^m = Y^m$
22	endif
23	if $f(Y^{f-c}) > f(Y^f)$
24	$Y^f = Y^{f-c}$
25	else
26	$Y^f = Y^f$
27	endif
28	$t = t + 1$
29	Return Y^m , Y^f and Y^n
30	Terminate

creates $\langle \text{key, listed value} \rangle$, where the listed value is the class. The key and the listed value will be processed by evaluating the sentiments, such as positive, negative or objective and thus, the MapReduce model classifies the reviews.

5. Results and discussion

In this section, the results of the proposed KO-SVM classifier used for sentiment classification of train reviews are illustrated. The performance of the proposed technique is

compared with existing techniques used for classification, in a comparative analysis.

5.1 Experimental set-up

The experimentation of the proposed technique using KO-SVM classifier is performed in a system with Windows 8 Operating System, 2 GB installed memory, and Intel core i-3 processor. The software tool used for the implementation of the MapReduce technique is JAVA based on the datasets described below.

5.1a Dataset description: The datasets used for the experimentation are two: a publicly available database of train reviews [27] and benchmark database on movie reviews [28]. The train review database considered is taken from a popular northern-side train in India, Rajdhani Express, which connects New Delhi with most of the capital cities. The review is available in MouthShut.com, which is one of the best sites for getting online reviews. This publicly available database consists of 61 reviews, with 33 positive and 28 negative reviews. The benchmark data on movie review used for sentiment analysis is comprised of collections of movie-reviews based on their sentiment polarity. The sentiment polarity dataset utilized is polarity dataset v1.0, which has 700 positive and 700 negative reviews. The dataset is released in July 2002.

Parametric values assigned: The parameter considered in the proposed KO-SVM classifier is the size of the population to be initialized in the proposed SLA model. The population size is assigned a value: $L = 10$.

5.2 Comparative techniques

The techniques employed for the comparison to evaluate the performance are (i) SentiWordNet [19], (ii) NB [10], (iii) NN [21] and (iv) LSVM [11]. In [19], the sentimental relevance of objective words and their related sentiment sentences are measured for the classification using SentiWordNet, whereas, in [10], NB was adopted in Vote algorithm for the classification of reviews in Turkish. NN utilized in [21] performs classification for each group of sentences applied. In [11], LSVM is used for sentiment classification after feature extraction based on a segment level. The performance of these four existing techniques is compared with that of the proposed KO-SVM classifier in the performance analysis.

5.3 Evaluation metrics

The metrics used for the evaluation of the performance of the techniques utilized for the comparison are sensitivity, specificity, and accuracy. These three performance

Figure 2. MapReduce architecture for sentiment classification.

evaluation metrics are considered in the fitness evaluation SLA, as defined in section 3.3a B.

5.4 Comparative analysis

This section demonstrates the performance analysis of the comparative techniques based on the three evaluation metrics using the train review dataset by varying the percentage of data for the number of iterations 50 and 100.

5.4a Analysis based on sensitivity: The comparative analysis based on the metric, sensitivity, for two fixed number of iterations is explained using figure 3. The sensitivity analysis chart obtained for 50 iterations in the techniques, SentiWordNet, NB, NN, LSVM, and KO-SVM, is shown in figure 3(a). As the percentage of data is increased, the sensitivity attained by the four techniques is increased. When the data percentage is 50, SentiWordNet, NB, NN, LSVM and KO-SVM, seems to have a sensitivity of 87.731%, 88.863%, 89.896%, 89.989%, and 92.804%,

Figure 3. Comparative analysis based on sensitivity. (a) Number of iterations = 50, (b) Number of iterations = 25.

Figure 4. Comparative analysis based on specificity. (a) Number of iterations = 50, (b) Number of iterations = 25.

Figure 5. Comparative analysis based on accuracy. (a) Number of iterations = 50, (b) Number of iterations = 25.

respectively. For the maximum percentage of data, the sensitivity attained by these techniques is 90.091%, 91.11%, 90.998%, 91.872%, and 93.46%. In figure 3(b), the sensitivity analysis chart for 25 iterations is presented by varying the percentage of data from 50 to 80. The maximum sensitivity attained by the existing SentiWordNet, NB, NN, and LSVM, is 84.773%, 83.992%, 85.231% and 85.758, while in KO-SVM, it is 87.126% when the percentage of data is 80. This shows that the proposed KO-SVM classifier has maximum sensitivity than the techniques compared.

5.4b Analysis based on specificity: The specificity analysis plot made against varying percentages of data in the five comparative techniques for the number of iterations, 50 and 100, is explained using figure 4. Figure 4(a) shows the analysis based on specificity for the iteration number fixed as 50. Increasing the percentage of data, the specificity can be increased. The value increases from 70.728% to 71.261%, in NB, when the data percentage is fixed to 80 from 50. Meanwhile, in KO-SVM, the specificity is raised to 74.485% from 72.389%. Thus, for the maximum

percentage of data, the specificity of KO-SVM is 4.328% more than in the existing NB approach. The specificity analysis for the number of iterations 25 is given in figure 4(b). The maximum specificity achieved by SentiWordNet, NB, NN, LSVM, and KO-SVM is 67.339%, 68.051%, 68.744%, 67.983% and 70.649%, respectively.

5.4c Analysis based on accuracy: The comparative analysis based on accuracy using the train review database for varying number of iterations is demonstrated in figure 5. In figure 5(a), the accuracy analysis chart for the number of iterations 50, is given. Higher the accuracy, higher is the performance of the classifier. When 83.224% is the maximum accuracy attained by the existing LSVM technique, the proposed KO-SVM classifier has 1.324% accuracy more with 84.341% as its maximum accuracy. The result of analysis based on accuracy, for 25 iterations, is depicted in figure 5(b). For 70% data, the accuracy achieved by SentiWordNet, NB, NN, LSVM and KO-SVM, is 76.089%, 76.793%, 77.104%, 77.4696% and 79.542%, which increases to, 76.391%, 76.813%, 77.23%, 77.641% and 80.326%, respectively, when the percentage of data is maximum.

Figure 6. Comparative analysis based on sensitivity. (a) Number of iterations = 50, (b) Number of iterations = 25.

Figure 7. Comparative analysis based on specificity. (a) Number of iterations = 50, (b) Number of iterations = 25.

5.5 Comparative analysis on benchmark data

Apart from the train review database, analysis based on benchmark data using movie review is made. This is to ensure the effectiveness of the proposed technique, rather than considering only the publicly available train review data. The analysis is performed similar to the previous database by varying the data percentage.

5.5a Analysis based on sensitivity: The performance analysis based on sensitivity, for two fixed number of iterations, is shown in figure 6. In figure 6(a), the sensitivity analysis for 50 iterations in the proposed as well as existing classifiers are given, where the maximum sensitivity observed is for the percentage of data 80. At that instant, the sensitivity attained using SentiWordNet, NB, NN, and LSVM, is 87.116%, 87.835%, 88.863%, and 90.265%, whereas, in KO-SVM, it is 91.249%. In this case, a percentage increment of 1.078% is found in the proposed KO-SVM than the existing LSVM classifier. The analysis based on sensitivity with 25 as the number of iterations, is shown in figure 6(b). Initially, for 50% data, the sensitivity obtained by

SentiWordNet, NB, NN, LSVM, and KO-SVM is 81.025%, 82.101%, 82.031%, 84.002%, and 86.273%, respectively. For 80% data, these techniques have a sensitivity of 82.391%, 83.113%, 84.445%, 85.904% and 87.454%. Hence, from the sensitivity analysis, the proposed KO-SVM classifier seems to have better sensitivity than the existing techniques.

5.5b Analysis based on specificity: Figure 7 demonstrates the specificity analysis chart, with the number of iterations 50 and 100, in the comparative techniques using the benchmark database. The performance analysis based on specificity for 50 iterations is depicted in figure 7(a). When the percentage of data is 80, SentiWordNet, NB, NN, LSVM and KO-SVM, has a specificity of 66.849%, 67.816%, 68.617%, 69.508%, and 70.018%, respectively. In figure 7(b), the specificity analysis for 25 iterations is shown, where the maximum specificity attained by the proposed KO-SVM classifier is 65.603%, whereas, in LSVM, it is 65.17%. From the analysis, it is seen that the proposed classifier is having the maximum specificity than the existing classifiers used for sentiment analysis.

Figure 8. Comparative analysis based on accuracy. (a) Number of iterations = 50, (b) Number of iterations = 25.

Table 2. Performance comparison using train review database.

Comparative techniques	Evaluation metrics		
	Sensitivity	Specificity	Accuracy
<i>SentiWordNet</i>	90.091	70.355	80.457
<i>NB</i>	91.11	71.261	80.922
<i>NN</i>	90.998	71.114	83.189
<i>LSVM</i>	91.872	72.257	83.224
<i>KO-SVM</i>	93.46	74.485	84.341

Table 3. Performance comparison using movie review database.

Comparative techniques	Evaluation metrics		
	Sensitivity	Specificity	Accuracy
<i>SentiWordNet</i>	87.116	66.849	77.899
<i>NB</i>	87.835	67.816	78.074
<i>NN</i>	88.863	68.617	78.747
<i>LSVM</i>	90.265	69.508	79.252
<i>KO-SVM</i>	91.249	70.018	79.611

5.5c Analysis based on accuracy: The result of accuracy analysis in all the five comparative techniques for two different iteration numbers is presented in figure 8. In the accuracy analysis chart for 50 iterations, shown in figure 8(a), the accuracy is found to be in maximum when the percentage of data is increased. When 77.899%, 78.074%, 78.747%, and 79.252%, is the maximum accuracy attained by SentiWordNet, NB, NN, and LSVM, the proposed KO-SVM has an accuracy of 79.611%. The accuracy analysis for 25 iterations is given in figure 8(b). The maximum accuracy attained among the existing techniques is 74.099% by LSVM, while in the proposed KO-SVM, it is possible to achieve an accuracy of 75.391%.

5.6 Discussion

From the overall comparative analysis made using the publicly available train review and benchmark movie database, the discussion is made separately, as shown in tables 2 and 3.

As shown in table 2, the maximum results regarding the three evaluation metrics in the five comparative techniques are shown, using train review data. The maximum performance attained among the existing techniques is by LSVM, which has 91.872%, 72.257%, and 83.224%, as the maximum sensitivity, specificity, and accuracy. Meanwhile, the proposed KO-SVM classifier has a sensitivity of 93.46%, specificity of 74.485% and accuracy of 84.341%.

Similarly, the performance comparison using movie review database is made and discussed, as presented in table 3. When LSVM had 90.265%, 69.508%, and 79.252% as the maximum sensitivity, specificity, and accuracy, the proposed KO-SVM classifier could attain the same as 91.249%, 70.018%, and 79.611%, respectively. Hence, it can be concluded that the proposed KO-SVM classifier could achieve better results than the existing techniques employed for sentiment classification.

6. Conclusion

This paper presents a technique for sentiment classification of train reviews using the MapReduce approach based on the proposed KO-SVM classifier. The MapReduce framework handles big data using map and reducer functions, which performs feature extraction and classification. The mapper converts the pre-processed train review data into a feature vector by extracting classification-specific and SentiWordNet analysis-based features. In the reducer, the classification is done using the proposed KO-SVM classifier that adopts SVM by substituting the exponential kernel by an optimized kernel, where the selection of weights is carried out by the proposed SLA model. The SLA

optimizer is developed by modifying the update process with a parameter that makes the algorithm self-adaptive. The performance of the proposed KO-SVM classifier is compared with SentiWordNet, NB, NN, and LSVM, using specificity, sensitivity, and accuracy, as the evaluation metrics, with train review and movie review database. From the performance analysis, the proposed KO-SVM classifier could achieve maximum performance with a sensitivity of 93.46%, specificity of 74.485% and accuracy of 84.341% for train review and the same of 91.249%, 70.018%, and 79.611%, respectively, for movie review database. Thus, it can be stated that KO-SVM classifier could perform better classification than the existing comparative techniques.

References

- [1] Lin Y, Wan H, Jiang R, Wu Z and Jia X 2015 Inferring the travel purposes of passenger groups for better understanding of passengers. *IEEE Trans. Intell. Transp. Syst.* 16(1): 235–243
- [2] Hurk E, Kroon L, Maróti G and Vervest P 2015 Deduction of passengers' route choices from smart card data. *IEEE Trans. Intell. Transp. Syst.* 16(1): 430–440
- [3] Ali F, Kim EK and Kim Y G 2015 Fuzzy ontology-based opinion mining and information extraction: a proposal to automate the hotel reservation system. *Appl. Intell.* 42(3): 481–500
- [4] Ali D, Kwak P, Khan S M R, Islam K H, Kim and Kwak K S 2017 Fuzzy ontology-based sentiment analysis of transportation and city feature reviews for safe traveling. *Transp. Res. Part C: Emerg. Technol.* 77: 33–48
- [5] Havasi C, Cambria E, Schuller B, Liu B and Wang H 2013 Knowledge- based approaches to concept-level sentiment analysis. *IEEE Intell. Syst.* 28(2): 0012–14
- [6] Manning C D and Schütze H 1999 *Foundations of statistical natural language processing*, Cambridge, MA: MIT Press
- [7] Pang B, Lee L and Vaithyanathan S 2002 Thumbs up? Sentiment classification using machine learning techniques. In: *Proceedings of the EMNLP*, pp. 79–86
- [8] Tang D, Qin B, Wei F, Dong L, Liu T and Zhou M 2015 A joint segmentation and classification framework for sentence level sentiment classification. *IEEE/ACM Trans. Audio, Speech Lang. Process.* 23(11): 1750–1761
- [9] Quan C and Ren F 2014 Unsupervised product feature extraction for feature-oriented opinion determination. *Inf. Sci.* 272: 16–28
- [10] Catal C and Nangir M 2017 A sentiment classification model based on multiple classifiers. *Appl. Soft Comput.* 50: 135–141
- [11] Xia R, Zong C and Li S 2011 Ensemble of feature sets and classification algorithms for sentiment classification. *Inf. Sci.* 181(6): 1138–1152
- [12] Liu B 2012 *Sentiment analysis and opinion mining*. San Rafael: Morgan & Claypool
- [13] Phu V N, Dat N D, Tran V T N and Chau T A, Nguyen 2016 fuzzy C-means for English sentiment classification in a distributed system. *Appl. Intell.* pp. 1–22
- [14] Yang Y and Pedersen J O 1997 A comparative study on feature selection in text categorization. In: *Proceedings of the ICML'97*, pp. 412–420
- [15] Li J, Fong S, Zhuang Y and Khoury R 2016 Hierarchical classification in text mining for sentiment analysis of online news. *Soft Comput.* 20(9): 3411–3420
- [16] Taboada M, Brooke J, Tofiloski M, Voll K and Stede M 2011 Lexicon-based methods for sentiment analysis. *Comput. Linguist.* 37(2): 267–307
- [17] Zhao J, Dong L, Wu J and Xu K 2012 Moodlens: An emoticon-based sentiment analysis system for chinese tweets. In: *Proceedings of the SIGKDD*
- [18] Maas A L, Daly R E, Pham P T, Huang D, Ng AY and Potts C 2011 Learning word vectors for sentiment analysis. In: *Proceedings of the ACL*
- [19] Hung C and Lin H 2013 Using objective words in SentiWordNet to improve word-of-mouth sentiment classification. *IEEE Intell. Syst.* 28(2): 47–54
- [20] Salehan M and Kim D J 2016 Predicting the performance of online consumer reviews: A sentiment mining approach to big data analytics. *Decis. Support Syst.* 81: 30–40
- [21] Chen T, Xu R, He Y and Wang X 2017 Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Syst. Appl.* 72: 221–230
- [22] Saif, Hassan, Fernández, Miriam, He, Yulan and Alani, Harith 2014 On stopwords, filtering and data sparsity for sentiment analysis of Twitter. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pp. 810–817
- [23] Esuli A and Sebastiani F 2006 SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In: *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06)*, 417–422
- [24] Ohana B and Tierney B 2009 Sentiment classification of reviews using SentiWordNet. In: *9th IT&T Conference, Dublin Institute of Technology, Dublin, Ireland*
- [25] Boser B E, Guyon I M and Vladimir N Vapnik 1992 A training algorithm for optimal margin classifiers. In: *Proceedings of the Fifth Annual Workshop on COMPUTATIONAL Learning Theory COLT '92*, pp. 144–152
- [26] Rajakumar B R 2014 Lion algorithm for standard and large scale bilinear system identification: A global optimization based on Lion's social behavior. In: *2014 IEEE Congress on Evolutionary Computation (CEC), Beijing*, pp. 2116–2123
- [27] <http://www.mouthshut.com/product-reviews/Rajdhani-Express-reviews-925004322-page-3>
- [28] <http://www.cs.cornell.edu/people/pabo/movie-review-data>