



Efficient Text Classification Model Based on Improved Hyper-sphere Support Vector Machine with MapReduce and Hadoop

Manisha Bhonde

Student, Padmashree Dr. D.Y.P.I.E.T.,
Pimpri, Pune University, India

Prof. Pramod Patil

Padmashree Dr. D.Y.P.I.E.T.,
Pimpri, Pune University, India

Abstract— the concept of text classification is process of sorting out the text documents automatically in predefined classes. There are many algorithms are presented for the automatic text categorization. Those algorithms are representing the bags or words and hence processing the large number of features. For the feature extraction semantic analysis is commonly used, removing the text representation errors caused by synonyms and polysemes, and hence removing the vector dimensions. Recently we have studied the Hyper-sphere-SVM (HS-SVM) as the recent machine learning method for text classification, this method later suppressed by Improved HS-SVM (IHS-SVM) by gaining more accuracy and efficiency. In this paper we are extending and investigating the IHS-SVM method by addition the parallel processing methods of text classification such as MapReduce and Hadoop. In case of IHS-SVM, additional to existing method of text classification new decision-making method based on concentration is presented for enhancing the classification of texts in overlapping regions. The practical evaluation of IHS-SVM with and without MapReduce as well as Hadoop is presented in this paper.

Keywords— Text Classification, machine learning methods, SVM, HS-SVM, IHS-SVM, MapReduce, Hadoop.

I. INTRODUCTION

Recently many researchers found that text classification accuracy is mainly affected by decision function used in machine learning methods. Thus existing HS-SVM approach is further improved by using the novel decision making approach over concentration, and hence this new method in turns called as Improved HS-SVM (IHS-SVM). Further LSA is used along with IHS-SVM in order to improve the performance. The method of LSA is used for features extraction as well as dimensionality reduction with good accuracy of text categorization and less computational overhead. In the text categorization, IHS-SVM is used for training as well as classification over different kinds of text datasets. This method in previous experimental studies found that better in terms of classification results with best accuracy as well as efficiency. In this paper our main aim is to investigate the algorithm of IHS-SVM and improve further its performance by using the Hadoop and MapReduce. Here mainly MapReduce parallel programming model is presented along with IHS-SVM; propose a MapReduce and the Hadoop distributed classification method, and presented its practical evaluation.

II. RESEARCH BACKGROUND

These are document indexing, classifier discovering and classifier evaluation. These are three stages in the life cycle of text classification.

A. Indexing Document

Document indexing denotes the undertaking of mapping a document d_j into a compact representation of its content that can be exactly interpreted (i) by a classifier construction algorithm and (ii) by a classifier, one time it has been constructed. An indexing procedure is characterized by (i) a delineation of what a term is, and (ii) a procedure to compute period weights. Concerning (i), the most common choice is to recognize periods either with the phrases happening in the article or with their arises. A popular choice is to add to the set of words or arises a set of sayings, i.e. longer (and semantically more significant) dialect units extracted from the text by superficial parsing and/or statistical methods. In relation to (ii), term weights may be binary valued or real-valued, counting on if the classifier building algorithm and the classifiers, once they have been built, need binary input or not. When weights are binary, these easily show presence/absence of the term in the document. When weights are non-binary, they are computed by either statistical or probabilistic methods, the previous being the most widespread option.

B. Learning Classifier

A text classifier for c_i is mechanically developed by a general inductive method (the learner) which, by discerning the characteristics of a set of documents reclassified under c_i or \bar{c}_i , gleans the characteristics that a new unseen document should have in order to belong to c_i . In alignment to build classifiers for C , one thus desires a set Ω of documents such that the worth of $\Phi(d_j, c_i)$ is renowned for every $(d_j, c_i) \in \Omega \times C$.

C. Evaluation Classifier

Teaching efficiency (i.e. average time required to construct a classifier $\hat{\Phi}_i$ from a given corpus Ω), as well as classification efficiency (i.e. mean time needed to classify a document by means of $\hat{\Phi}_i$), and effectiveness (i.e. mean correctness of $\hat{\Phi}_i$'s classification demeanor) are all legitimate measures of success for a learner.

In TC study, effectiveness is usually considered the most significant criterion, since it is the most reliable one when it arrives to experimentally matching distinct learners or distinct TC methodologies, granted that efficiency counts on too volatile parameters (e.g. distinct software / hardware platforms). In TC submissions, although, all three parameters are important, and one should carefully gaze for a tradeoff amidst them, counting on the application constraints. For example, in submissions engaging interaction with the client, a classifier with reduced classification effectiveness is unsuitable.

III. PROPOSED SYSTEM

Following figure 1 showing the proposed approach of text classification based on MapReduce and Hadoop.

The work will be expressed out as chases.

1. Investigation of available text classification designs.
2. Implementation of text pre-processor.
3. Characteristic extraction utilizing semantic investigation.
4. Vectorization of text.
5. Use of MapReduce and Hadoop during the training phase.
6. Finally classification of text utilizing SVM classifier.
7. Evaluation of design with actually accessible designs.
8. Production Evaluation and conclusion investigation.

To find out what methods are undertaking for discovering text classifiers, we should find out more about the properties of text.

High dimensional input space:

When discovering text Classifiers one has to deal with very many (more than 10000) features. Since SVMs use over fitting defense which does

Text pre-processing:

Unstructured texts or use to natural language of humans, which make its semantics tough for the computer to deal with. So they need essential pre-processing. Text pre-processing mostly segments texts into words.

LSA-based feature extraction, dimensionality Reduction.

LSA is utilized in this module for the characteristic ext reaction and the dimensionality decline of word-document matrix of educating set. K large-scale singular measures and corresponding singular vectors are extracted by the singular worth decomposition of word-document matrix, to constitute a new matrix for roughly representation of the primary word document matrix. Contrasted with VSM, it can contemplate the semantic attachment between phrases and the influence of contexts on saying meanings, eradicate the discrepancy of text representation caused by synonyms and polyesters, and decline the dimension of text vectors.

Vectorization of text:

In this form, each row vector of the word-document matrix represents a text that is the vectorization of text. All through a testing procedure, after each check experiment segmented into sayings, the prime text vectors are mapped to a latent semantic space in this module by LSA vector space form, to develop new text vectors.

IHS-SVM classifier and learning:

Eventually, the new text vectors are classified in IHS-SVM classification module. IHS-SVM is an improvement for HSSVM, both of which will use a least significant surrounding ball to define each kind of text. When employed out classes, HSSVM finds which hyper-sphere is the nearest one to the ascertain trial, and then the class it stands for is the one the ascertain trial pertains to. Where, the texts in overlapping localities will not be classified rightly by this way. IHS-SVM splits up trials into three types: those not in any hyper-sphere, those only contained in one, and those encompassed in multiples. The classification of the first two types is equal to HS-SVM. It compares the engrossment of the ascertain sample to each hyper sphere, and then classes the trial to the biggest one.

Feature Extraction and Dimensionality Reduction:

The method of characteristic extract ion is to make clear the boundary of each dialect structure and to eliminate as much as possible the dialect reliant factors, tokenization, and halt words exclusion, and stemming. Characteristic Ext reaction is fist step of pre processing which is utilized to presents the text articles into clear phrase format. Removing halts phrases and arising phrases is the pre-processing tasks. The documents in text classification are comprised by a large allowance of feature and most of them could be irrelevant or loud. Dimension decrease is the exclusion of a large number of keywords, groundwork preferably on a statistical criterision, to conceive a reduced dimension vector. Dimension Reduction methods have adhered much attention lately research productive dimension decrease make the learning task such as classification more effective and save more storage space. Commonly the steps taken delight for the characteristic extract ions are: Tokenization: An article is treated as a string and then partitioned into a list of tokens. Removing halt phrases: Stop words such as "the", "a", "and etc are frequently happening, so the insignificant words need to be taken. Stemming phrase: Applying the arising algorithm that converts distinct phrase form into alike canonical form. This step is the method of conflating tokens to their root pattern eg. Connection to attach, computing to compute etc.

VSM founded on text keywords quantizes document vector with the weights of the words, having high effectiveness and very simple to use. However, it only counts the frequency of the words, while disregarding the semantic link amidst

them and the influence of context on their meanings. Therefore texts similarity counts only on the number of the identical phrases they comprised, which decreases the classification correctness with the reality of polysemes and synonyms. In supplement, the text matrixes constructed by VSM are usually high-dimensional sparse matrices, inefficient training classification and not apt for management large-scale text groups. Although, LSA can effectively explain these limitations. It accepts as true that there is

a latent semantic structure between phrases of one text. And it hides in their context usage patterns. So, k biggest singular values and their corresponding singular vectors are extracted by the singular worth decomposition of word-document matrix, to constitute a new matrix for the about presentation of word-document matrix of the original documents set. Text offered by high-dimensional VSM is therefore mapped into a low-dimensional latent semantic space. You can extract latent semantic structure without the influence of the correlation between the words to get high text representation correctness. LSA is founded on singular worth decomposition. It charts texts and words form a high dimensional vector space to a reduced one, reducing text dimensions and advancing text representation accuracy.

Step1: Construct a word-document matrix A. In the LSA model, a text set can be conveyed as a word-document matrix of $m \times n$ (m is the number of applications comprised in a text, n is the number of texts).

Step2: Decompose singular value. A is decomposed into the multiply of three matrices: U, S, V. U' and V' are orthogonal matrices, S' is a diagonal matrix of singular worth. Retain the lines and the pillars of S' encompassing K biggest single-values to get a new diagonal matrix. Then keep the same part of U' and V' to get U and V. Thus, assemble a new word-document matrix $R = USV^T$. For a text d, phrases are screened by singular worth decomposition to form new vectors to restore the original text characteristic vectors. It disregards the components of smaller leverage and less importance. Key-words that don't emerge in the text will be comprised in the new word-document matrix if they are affiliated with the text semantics. Thus, the new matrix reflects the promise semantic relative among keywords from a numerical issue of outlook. It is nearest to the initial period frequency matrix with the least-squares. Meaning of each dimension vector space greatly altered in process. It reflects a strengthened semantic connection rather than of easy look frequency and circulation relationship of entries. And the dimension reduction of vector space can effectively advance the classification pace of text groups.

Hadoop Overview:

When data sets go beyond a single storage capacity, it is necessary to distribute them to multiple independent computers. Trans-computer network storage file management system is called distributed file system. A typical Hadoop distributed file system contains thousands of servers, each server stores partial data of file system.

MapReduce Overview:

In distributed data storage, when parallel processing the data, we need to consider much, such as synchronization, concurrency, load balancing and other details of the underlying system. It makes the simple calculation become very complex. MapReduce programming model [3] was proposed in 2004 by the Google, which is used in processing and generating large data sets implementation. This framework solves many problems, such as data distribution, job scheduling, fault tolerance, machine to machine communication, etc.

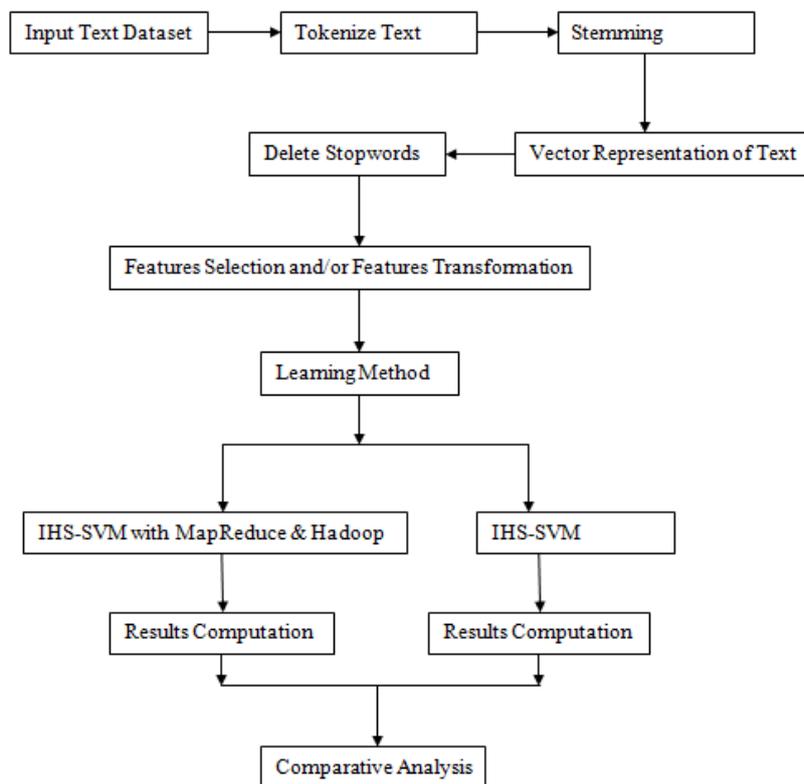


Figure 1: Proposed Method Architecture

MapReduce is applied in Google's Web search. Programmers need to write many programs for the specific purpose to deal with the massive data distributed and stored in the server cluster, such as crawled documents, web request logs, etc., in order to get the results of different data, such as inverted indices, web document, different views, worms collected the number of pages for each host a summary of a given date within the collection of the most common queries and so on.

MapReduce Programming Model:

MapReduce programming model, by map and reduce function realize the Mapper and Reducer interfaces. They form the core of task.

1. Mapper

Map function requires the user to handle the input of a pair of key value and produces a group of intermediate key and value pairs. <key,value> consists of two parts, value stands for the data related to the task, key stands for the "group number " of the value . MapReduce combine the intermediate values with same key and then send them to reduce function.

Map algorithm process is described as follows:

Step1: Hadoop and MapReduce framework produce a map task for each InputSplit, and each InputSplit is generated by the InputFormat of job. Each <Key,Value> corresponds to a map task.

Step2: Execute Map task, process the input <key,value> to form a new <key,value>. This process is called "divide into groups". That is, make the correlated values correspond to the same key words. Output key value pairs that do not required the same type of the input key value pairs. A given input value pair can be mapped into 0 or more output pairs.

Step3: Mapper's output is sorted to be allocated to each Reducer. The total number of blocks and the number of job reduce tasks is the same? Users can implement Partitioner interface to control which key is assigned to which Reducer.

2. Reducer

Reduce function is also provided by the user, which handles the intermediate key pairs and the value set relevant to the intermediate key value. Reduce function merges these values, to get a small set of values. The process is called "merge ". But this is not simple accumulation. There are complex operations in the process. Reducer makes a group of intermediate values set that associated with the same key smaller.

In MapReduce framework, the programmer does not need to care about the details of data communication, so <key,value> is the communication interface for the programmer in MapReduce model.

<key,value> can be seen as a "letter", key is the letter's posting address, value is the letter's content. With the same address letters will be delivered to the same place. Programmers only need to set up correctly <key,value>, MapReduce framework can automatically and accurately cluster the values with the same key together.

Reducer algorithm process is described as follows:

Step1: Shuffle. Input of Reducer is the output of sorted Mapper. In this stage, MapReduce will assign related block for each Reducer.

Step2: Sort. In this stage, the input of reducer is grouped according to the key (because the output of different mapper may have the same key). The two stages of Shuffle and Sort are synchronized;

Step3: Secondary Sort. If the key grouping rule in the intermediate process is different from its rule before reduce. We can define a Comparator. The comparator is used to group intermediate keys for the second time.

Map tasks and Reduce task is a whole, cannot be separated. They should be used together in the program. We call a MapReduce the process as an MR process. In an MR process, Map tasks run in parallel, Reduce tasks run in parallel, Map and Reduce tasks run serially. An MR process and the next MR process run in serial, synchronization between these operations is guaranteed by the MR system, without programmer's involvement. Following figure 2 is explaining the process of this method.

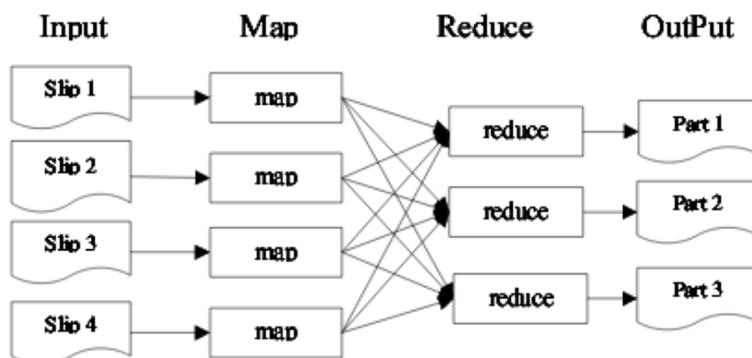


Figure 2: MapReduce Method Architecture

IV. PRACTICAL EVALUATION AND DISCUSSION

There are many metrics which we have studied for evaluating the effectiveness of machine learning methods. The most commonly used metrics are precision, recall and accuracy. In order to find out this performance metrics we have to do the understanding of if the classification of a document was a true positive (TP), false positive (FP), true negative (TN), or false negative (FN) as showing in below table 1:

TP	Determined as a document being classified correctly as relating to a category.
FP	Determined as a document that is said to be related to the category incorrectly.
FN	Determined as a document that is not marked as related to a category but should be.
TN	Documents that should not be marked as being in a particular category and are not.

Table 1: Text Classification

Precision (π_i) is determined as the conditional probability that a random document d is classified under c_i , or what would be deemed the correct category. It represents the classifiers ability to place a document as being under the correct category as opposed to all documents place in that category, both correct and incorrect:

$$\pi_i = \frac{TP_i}{TP_i + FP_i}$$

Recall (ρ_i) is defined as the probability that, if a random document d_x should be classified under category (c_i), this decision is taken:

$$\rho_i = \frac{TP_i}{TP_i + FN_i}$$

Accuracy is commonly used as a measure for categorization techniques. Accuracy values, however, are much less reluctant to variations in the number of correct decisions than precision and recall:

$$A_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}$$

Many times there are very few instances of the interesting category in text categorization. This overrepresentation of the negative class in information retrieval problems can cause problems in evaluating classifiers' performances using accuracy. Since accuracy is not a good metric for skewed datasets, the classification performance of algorithms in this case is measured by precision and recall [8].

Furthermore, precision and recall are often combined in order to get a better picture of the performance of the classifier. This is done by combining them in the following formula:

$$F_\beta = \frac{(\beta^2 + 1)\pi\rho}{\beta^2\pi + \rho}$$

where π and ρ denote precision and recall respectively. β is a positive parameter, which represents the goal of the evaluation task. If precision is considered to be more important than recall, then the value of β converges to zero. On the other hand, if recall is more important than precision then β converges to infinity. Usually β is set to 1, because in this way equal importance is given to each precision and recall.

In this experimental evaluation, we have implemented two methods IHS-SVM with MapReduce and Hadoop and another one is just IHS-SVM. We have used both English as well as Marathi text datasets for study. Following performance evaluation graphs are showing the performance of both datasets for both methods.

We have used following datasets (table 2) to compare the performance of both methods in terms of precision, recall and accuracy rates.

Dataset Number	Dataset Type	Dataset Name
1	Marathi	अर्थविश्व
2	Marathi	इतिहास
3	Marathi	पदार्थ
4	Marathi	क्रिडा
5	Marathi	डॉक्टर
6	Marathi	मनोरंजन
7	English	computer
8	English	environment
9	English	transportation
10	English	education
11	English	economy
12	English	military

Table 2: Experimental Datasets

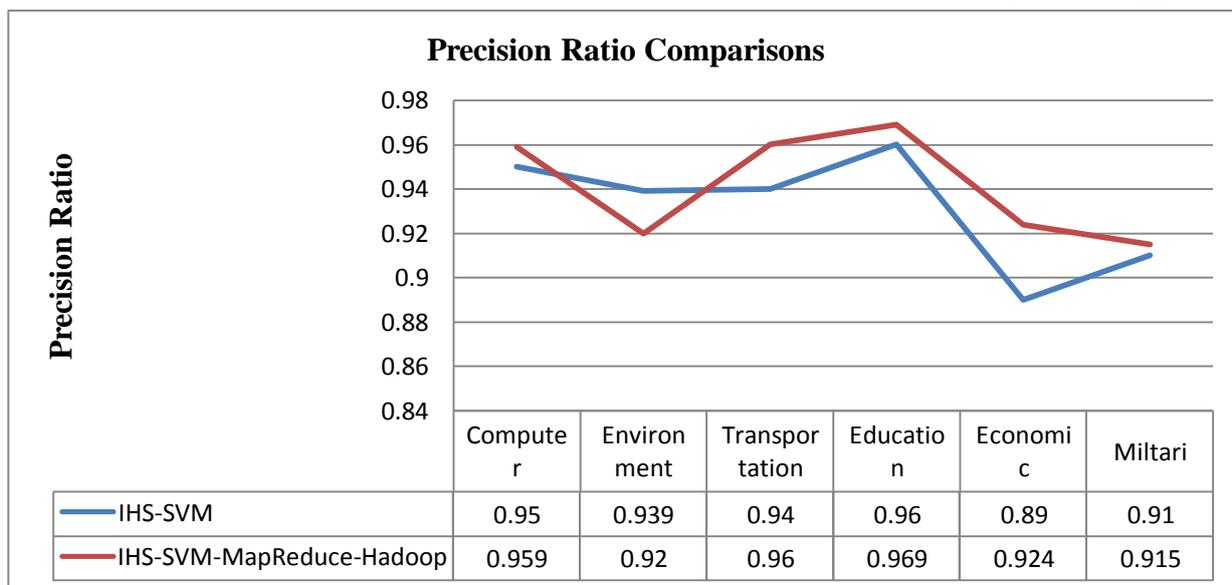


Figure 3: Precision Ratio Comparisons for English Dataset

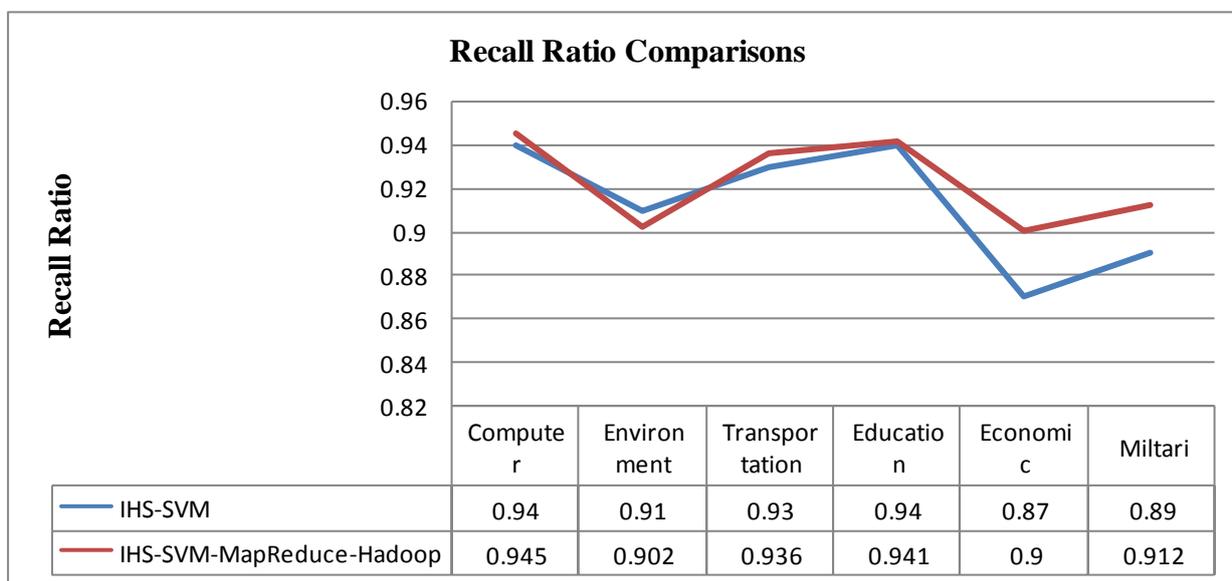


Figure 4: Recall Ratio Comparisons for English Dataset

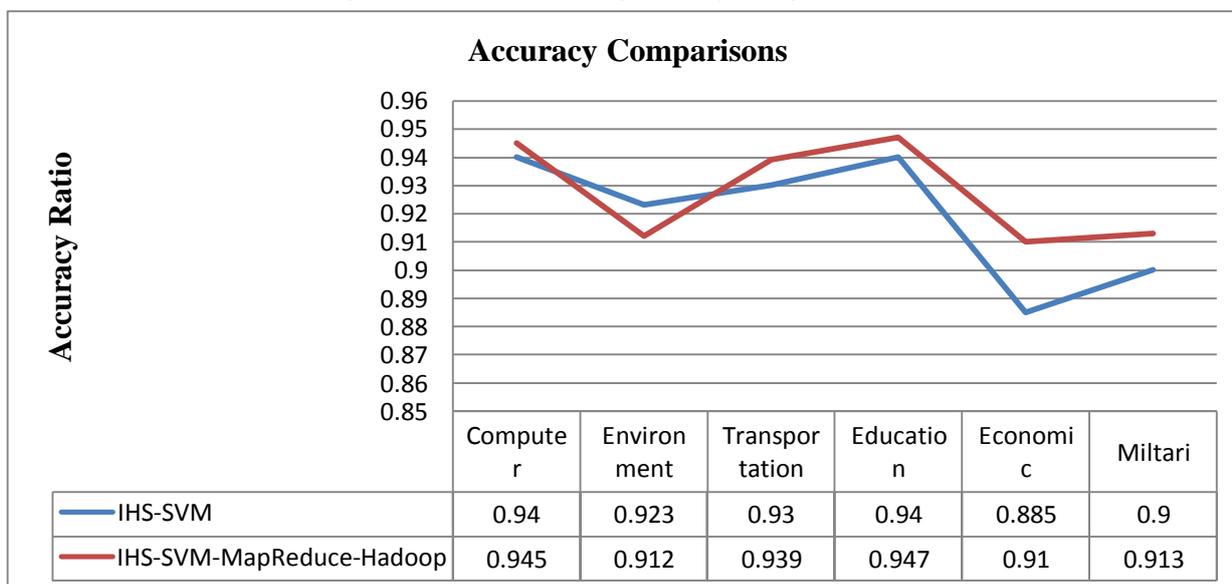


Figure 5: Accuracy Ratio Comparisons for English Dataset

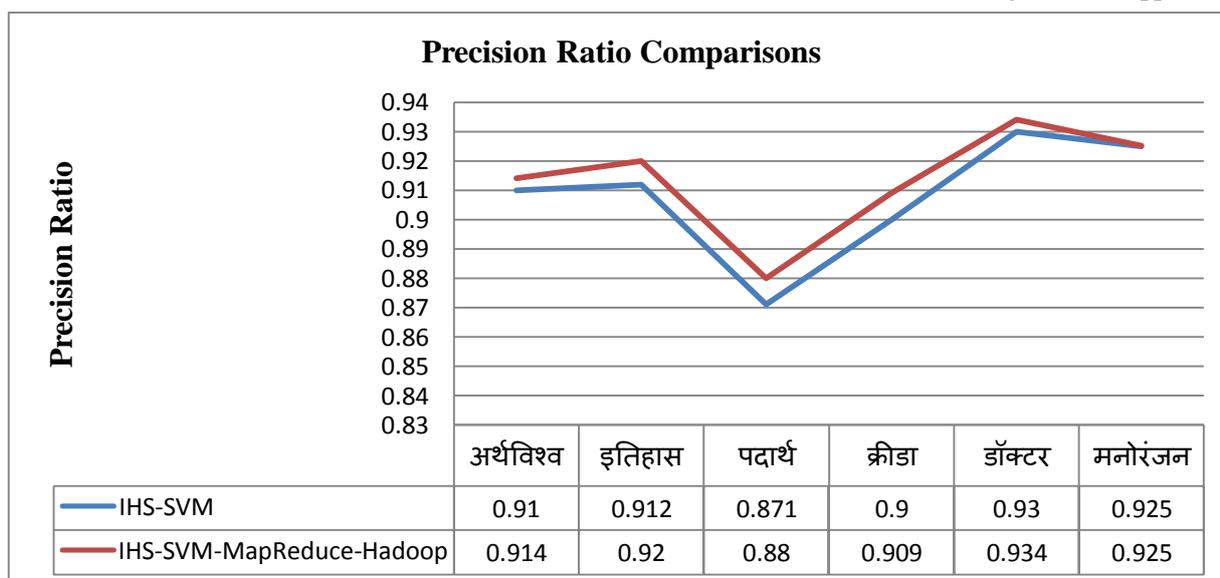


Figure 6: Precision Ratio Comparisons for Marathi Dataset

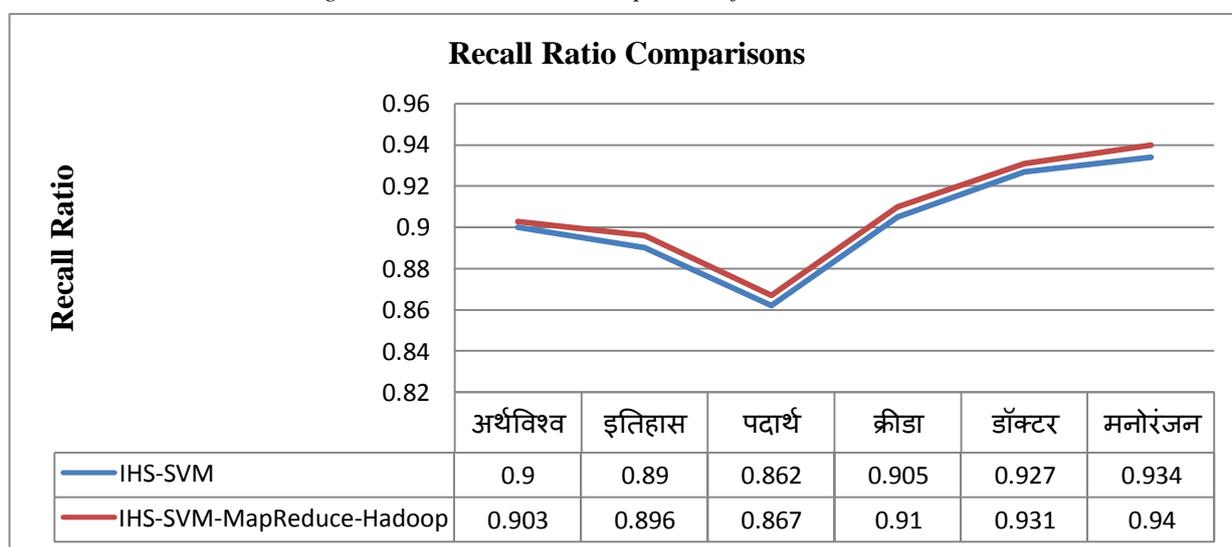


Figure 7: Recall Ratio Comparisons for Marathi Dataset

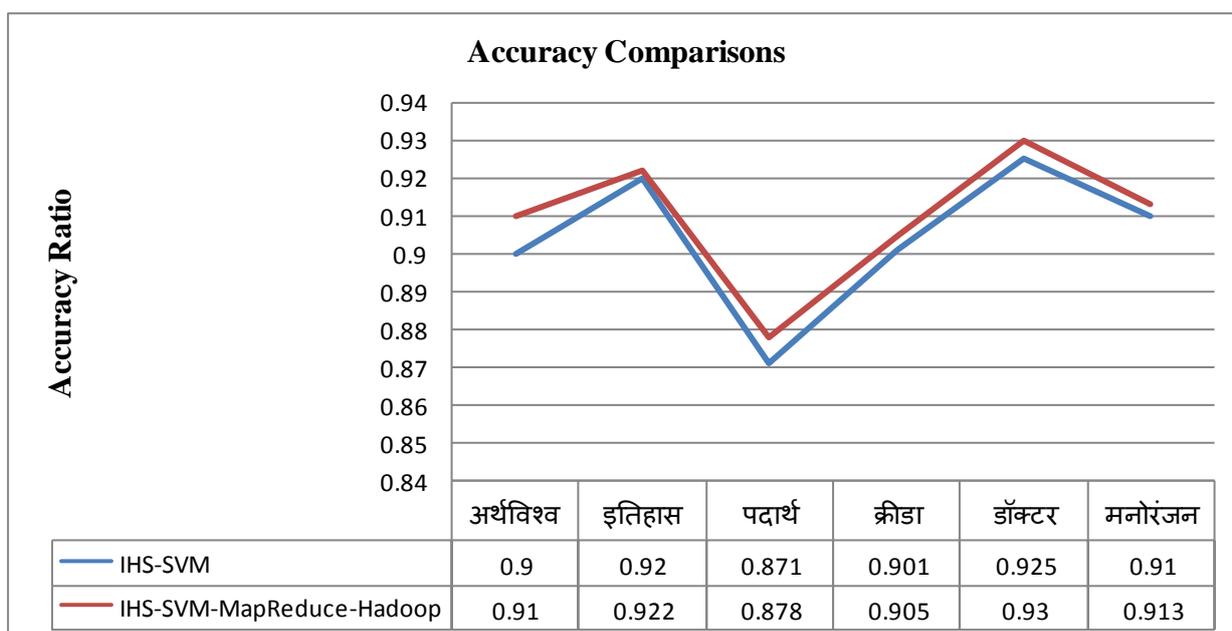


Figure 8: Recall Ratio Comparisons for Marathi Dataset

We have evaluated our proposed approach with two kinds of datasets like English and Marathi, we have observed all the expected results for precision, recall and accuracy rates for different types of datasets. We claim that from above results, our proposed or extended method of text classification is more accurate and efficient as compared to IHS-SVM method and hence we will further like to do analysis and investigation over the same.

V. CONCLUSION & FEATURE WORK

In this paper we simply discussed the literature review study over the all the recent methods of text classification those are based on machine learning techniques. We have discussed first most commonly used SVM, after that HS-SVM, and then most recent is IHS-SVM based approach for text classification. However we found that there is still place for improvement in terms of accuracy and efficiency of IHS-SVM method, and hence we have proposed to add the approach of MapReduce and Hadoop together with IHS-SVM to improve the accuracy and efficiency of text classification approach. We have presented the programming model for MapReduce as well as Hadoop and how it's included in IHS-SVM. The practical results showing that proposed method of text classification resulted into better as compared to existing one and hence we will further like do carry more investigation over the same.

REFERENCE AND BIBLIOGRAPHY

- [1] Yu-feng Zhang, Chao He, "Research of Text Classification Model Based on Latent Semantic Analysis and Improved HS-SVM", Sponsored by Key Base of Ministry of Education for Research of Humanities and Social Sciences(No.08JJD870225), IEEE 2010.
- [2] Evgeniy Gabrilovich, Shaul Markovitch. Text Categorization with Many Redundant Features: Using Aggressive Feature Selection to Make SVMs Competitive with C4.5. Proceedings of the 21st International Conference on Machine Learning, Banff, Canada, 2004.
- [3] C.H.Li, An efficient document categorization model based on LSA and BPNN, Sixth International Conference on ALPIT, pp.9-14, 2007
- [4] An Overview of E-Documents Classification Aurangzeb Khan, Baharum B. Bahardin, Hairullah Khan Department of Computer & Information Science Universiti Teknologi, PETRONAS 2009 International Conference on Machine Learning and Computing IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore
- [5] Hadoop. <http://hadoop.apache.org/>
- [6] S. Ghemawat, H. Gobioff, S.T. Leung, The Google file system, 19th Symposium on Operating Systems Principles, New York, pages 29-43 2003.
- [7] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In Proceedings of the 6th USENIX Symposium on Operating Systems Design and Implementation, pages 137-149, 2004.
- [8] S.Z. Selim, and M.A. Ismail. K-means-Type Algorithms: a Generalized Convergence Theorem and Characterization of Local Optimality. IEEE Trans on Pattern Analysis and Machine Intelligence ,Vol.6, No.1, pages 81-87, 1984.
- [9] Shuang Liu, Yongkui Liu and Bo Wang, "An Improved Hyper-sphere Support Vector Machine", Third International Conference on Natural Computation (ICNC 2007), IEEE 2007.
- [10] Peng Chen and Tao Wen, "Parameter Selection for Sub-hyper-sphere Support Vector Machine", Third International Conference on Natural Computation (ICNC 2007), IEEE 2007.
- [11] Q.Wang,C.Y.Jia,A.F.Zhang,S.Liu,Improved Algorithm Based on Sphere Structure SVMs and Simulation,Journal of System Simulation,Vol.20,No.2,pp.345-348,2008(in Chinese).
- [12] S.Liu,P.Chen,Improved hyper-sphere Support Vector Machine, Computer Engineering and Applications,Vol.45,No.16, pp.149-151,2009(in Chinese).
- [13] Kumar R, Mitchell J S B, Yildirim A, Approximate minimum enclosing balls in high dimension using core-sets, ACM, Journal of experimental algorithmics, Vol.8,pp.142-145,2003.