ISSN No. 0976-5697

Volume 3, No. 1, Jan-Feb 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Instance Subset Selection in SMS Classification using PSO-SVM

R.Parimala*
Research Scholar
National Institute of Technology
Tiruchirapalli, Tamil Nadu, India
rajamohanparimala@gmail.com

Dr. R. Nallaswamy
Professor, Dept. of Mathematics
National Institute of Technology
Tiruchirapalli, Tamil Nadu, India.
nalla@nitt.edu

Abstract: Text categorization is the task of classifying natural language documents into a set of predefined categories. It can provide conceptual views of document collections and has important applications in the real world. Short messages often consist of only a few words, and therefore present a traditional bag-of-words based spam filters using R package. In this paper we analyze the concept of a new classification model which will classify Mobile SMS into predefined classes**.** We have tested feasibility of applying Support Vector Machine (SVM) based machine learning techniques reported most effective in SMS spam filtering on NUS SMS dataset. We see that bag of-words filters improved substantially using different features. We conclude that classification for short messages is surprisingly effective.

Keywords: Text classification, pre-processing, sparse term, Support Vector Machine, Particle swarm optimization

I. INTRODUCTION

Text messaging has greatly increased in popularity in the past five years and the government is trying to keep up with rapidly changing technology. SMS spam (sometimes called cell phone spam) is any junk message delivered to a mobile phone as text messaging through the Short Message Service (SMS).

Although SMS spam is less prevalent than email spam, it still account for roughly 1% of texts sent in United States and 30% of text messages sent in parts of Asia. In the United States, SMS spam messages have been illegal under the Telephone Consumer Protection Act since 2004. Citizens who receive unsolicited SMS messages can now bring the solicitors to small claim court. In 2009, China's three main mobile phone operators (China Telecom, China Mobile Ltd and China Unicom) signed an agreement to combat mobile spam by setting limits on the number of text messages sent each hour.

In India SMS usage is increasing rapidly as more and more innovative uses are being found, such as contexts, shopping and location-based services. Apart from good volumes on normal days, SMS usage in India and abroad reaches abnormally high peaks on special days such as the New Year and other popular festivals. The Government has introduced anti-spam regulation, but has found that enforcing those regulations is almost impossible. As a result it is now claimed that almost 50% of all text messages sent in India are marketing spam.

Nowadays, SMS spam has become a major problem in India, which has become the world's largest and fastest-growing mobile market with over 700 million subscribers. In an effort to combat the problem, the Telecom Regulatory Authority of India (TRAI) has imposed a controversial limit of 100 SMS messages per day, per person. The new rule, which is in force, should end the dozens of unsolicited text messages received daily by Indians, according to the telecom regulator. Though regulatory agencies across the region have made broad effort to curb the amount of unsolicited messages users receive, there's still a huge problem with spam associated with things like credit cards and weight loss scams. Though limiting text messages to 100 per day shouldn't affect majority of mobile users, this kind of attack is higher on irritant value than on financial loss. Although at times the subscriber may end up paying for junk SMS message that the users does not want. Thus the user would suffer some extra financial burden. Most people expect only the most urgent of messages on one's cellular phone. Unsolicited messages on one's cellular phone are highly irritating. It is very difficult to filter out spam SMS messages just as it is difficult to intelligently filter out spam mail.

SMS spam is now prevalent in Singapore and Japan and it will undoubtedly spread throughout the world. The computational power of third generation cell phones and other devices as PDAs is increasing, making increasingly possible to do spam filtering at the devices, leading to better personalization and effectiveness. In some high-end mobile phones, Spam Managers installed which filter the messages and block SPAM to some extent. Several studies show that SMS is a spam or not, differs from person to person.

Instance subset selection plays an important role in classification. Random subset selection is actually a form of resampling. Each random sample extracted from training set with replacement. This will probably classify on different set of training instances. Optimal instance selection method proposed to choose the most suitable points in the data set to become instances for training data set used by the PSO-SVM algorithm.

In this paper, we focus on some well-known applications of text categorization and propose a new model for classification of SMS into predefined categories.

II. RELATED WORK

From our survey of user perceptions about SMS spam, we have found that perception of more than one user, the same SMS may differ. In recent times, there are many media reports published on SMS spam problem [1][2]. The most important fact here is that end users are helpless in controlling the number of SMS spam they are receiving. SMS text classifiers proposed in the literature using machine learning techniques such as support vector machines. Although many approaches proposed, text categorization is still a major area of research primarily because the effectiveness of current text classifiers is not faultless and still needs improvement. A classifier is built by applying a learning method to a training set of objects. This model is further used to predict the labels to new incoming objects. With all the effort in this domain there is still place for improvement and a great deal of attention is paid to developing highly accurate classifiers.

III. BACKGROUND

A. An overview of support Vector Machine Classifier:

Machine learning methods, including Support Vector Machines (SVMs), have tremendous potential for helping people more effectively to organize electronic resources. As a powerful statistical model with ability to handle a very large feature set, SVM is widely used in pattern recognition areas such as face detection, isolated handwriting digit recognition, and gene classification [3]. Recently SVM has been used for text categorization successfully. T. Joachims [4] classified documents into categories by using SVM and obtained better results than those obtained by using other machine learning techniques such as Bayes and K-NN. Similarly, J.T. Kwok [5] used SVM, to classify Reuters newswire stories into categories and obtained better results than using a k-NN classifier.

Support Vector Machines (SVMs) are a machine learning model proposed by V. N. Vapnik [6]. This introduction to Support Vector Machines (SVMs) is based on [3], [6], [7] and [8]. Support vector machines (SVMs) [6] are of great interest to theoretical and applied researchers and they have strong connections to computational learning theory. The basic idea of SVM is to find an optimal hyperplane to separate two classes with the largest margin from pre-classified data. After this hyperplane is determined, used for classifying data into two classes based on which side they are located. By applying appropriate transformations to the data space before computing the separating hyperplane, SVM can be extended to cases where the margin between two classes is non-linear.

Text classification is usually achieved by using Machine Learning techniques, which acquires labelled documents and no human intervention for coding rules or heuristics. Machine Learning algorithm has generated a model of the training data, used to classify new un-labelled documents automatically.

For a two-class classification problem, given a training set of instance-label pairs x_i, y_i , i =1, 2, 3 …ℓ where $x_i \in R^n$. The class label of the ith pattern is denoted by $y_i \in \{1,-1\}^t$, the SVM problem can be written as

$$\text{Minimize} \quad \frac{1}{2} w^T w + C\sum_{i=1}^{n} \xi_i$$

subject to the constraints

$$y_i (w^T \varphi(x_i) + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0, \quad i = 1,2,3..\ell,$$

where x_i are the input vector and y_i are the class labels. The problem is more commonly solved as the dual formulation, which can be written as

$$\text{Maximize} \quad -\frac{1}{2}\alpha^T Q\alpha + 1^T \alpha$$

subject to

$$y^t \alpha = 0$$

$$0 \leq \alpha_i \leq C, \quad i = 1,2.3..\ell .$$

where K is the Kernel matrix, $Q_{ij} = y_i y_j K(x_i, x_j)$ in some high dimensional space[9][10]. The output of the classifier for an unknown x is given by

$$y(x) = \text{sign}(\sum_{i=1}^{N} \alpha_i y_i k(x, x_i) + b).$$

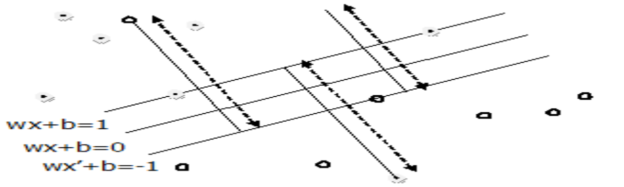

Figure: 1. Optimal separating hyperplane for binary classification problem

B. Particle Swarm Optimization:

Particle swarm optimization (PSO) is a population-based stochastic optimization technique, was first introduced by Kennedy and Eberhart in 1995 inspired by social behavior of bird flocking or fish schooling[11][12][13]. PSO has been successfully applied for training feed-forward [14][15][16] and recurrent [17][18]. Similar to Genetic Algorithms (GA), PSO is evolutionary computation technique. This technique initialized with population of random solutions and searches for optimum solution by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, are “flown” through the problem space by after the current optimum particles. The member of the entire population maintained through the search procedure so that information is socially shared among individuals to direct the search towards the best position in the search space. The applications of PSO on combinatorial optimization problems are still limited, PSO has certain advantages such as easy to carry out and computationally efficient. Hence PSO select the best training set using the cross-validation. PSO gives the global best object. Then predict the test set using global object.

PSO is a global optimization algorithm for dealing with problems in which a best solution represented as a point or surface search in n-dimensional space. Hypotheses plotted in this space and seeded with an initial velocity as well as a communication channel between the particles. Particles then move through the solution space and evaluated according to some fitness criterion following each time step. The particles accelerated give direction communication grouping which have better fitness values. Each particle keeps track of its coordinates in the problem space associated with the best solution it has achieved so far. This value called pbest. Particle swarm optimizer tracked the best value, obtained so far by any particle in the neighbors of the particle. This location called lbest. When a particle takes all the population as its topological neighbours, the best value is a global best and gbest. The particle swarm optimization concept consists of each time step, changing velocity of each particle toward its pbest and lbest locations.

C. Parameters of PSO:

Selecting good PSO parameters has been the subject of research. Pedersen et al (2000) presented a simple way of tuning the PSO parameters [19][20]. The technique for tuning PSO parameters called meta-optimization. The inertia weight employed to control the impact of the previous history of velocities on the current one. At time t update velocity from the previous velocity to the new velocity.

$$V_{ij}\left(+1\right) = w\, Vij(t) + c_1 r_1 (X_{ij}{}^{p}(t) - X_{ij}(t)) + C_2 r_2 (X_{ij}{}^{g}(t) - X_{ij}(t)) \quad (9)$$

The new position is then determined by the sum of the previous position and the new velocity by the following equation

$$Xij(t+1) = Xij(t) + V_{ij}(t+1) \qquad (10)$$

where w as the inertia factor, r_1 and r_2 are the random numbers which to keep up diversity of the population and are uniformly distributed in the interval [0, 1] for the jth dimension of ith particle. C_1 is a positive constant, called as coefficient of the self-recognition component. C_2 is also a positive constant called as coefficient of the social component from equation (1), a particle decides where to move next, considering its own experience the memory of the best past position and its most successful particle in the swarm. The parameter w regulates the trade-off between global and local exploration abilities of the swarm. A large inertia weight facilitates global exploration while a small one tends to facilitates global local exploration. A suitable value for the inertia weight w usually provides balance between global and local exploration abilities .This results in reduce the number of iterations required to find the optimum solution. The parameters $C_1=C_2=2$ can be set as default values [13]. Some experiment result show that $C_1=C_2=1.49$ might give even better results swarm size value might be 20. The basic structure of PSO as

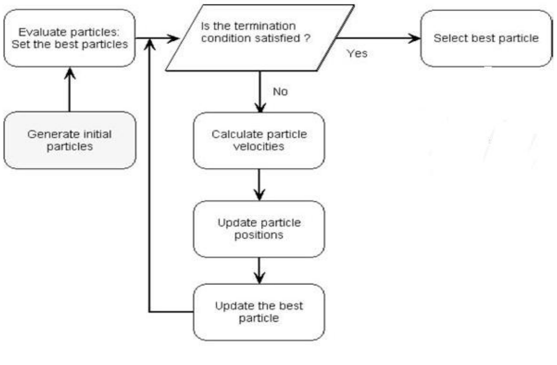
Evaluate particles: Set the best particles
Is the termination condition satisfied ?
Yes
Select best particle
No
Generate initial particles
Calculate particle velocities
Update particle positions
Update the best particle

Figure: 2

D. Review of text mining

Sebastiani gave an excellent review of text classification domain [21]. Thus, in this work apart from the brief description of the text classification. For this, text mining ™ framework implemented. Text mining is to mine the patterns from natural language than from structured database of facts. It is a process that employs a set of algorithms for converting unstructured text into structured data conveying the insightful information. Text mining process includes text preprocessing, feature generation and selection, pattern extraction, to analyzing results. For example, news stories are typically organized by subject categories (topics) or geographical codes; academic papers are often classified by technical domains and sub-domains; patient reports in health-care organizations are often indexed from multiple aspects, using taxonomies of disease categories, types of surgical procedures, insurance reimbursement codes and so on. Another widespread application of text categorization is spam filtering, where email messages classified into the two categories of spam and non-spam, respectively. The categorization task can divided into two sub-problems: (a) the text representation written in natural language as data suitable for machine learning algorithms and (b) categorizing the transformed data [21].

Text categorization [28][29] is the problem of automatically assigning predefined categories to free text documents. This problem is of great practical importance given the massive volume of online text available through the World Wide Web, Emails, and digital libraries. A major characteristic or difficulty of text categorization problems is the high dimensionality of the feature space. The original feature space consists of many unique terms (words or phrases) that occur in documents, and the number of terms is hundreds of thousands for even a moderate-sized text collection. This is prohibitively high for many mining algorithms. Therefore, it is highly desirable to reduce the original feature space without sacrificing categorization accuracy. The number of words is large even in relatively small documents such as short news articles or paper abstracts. Dimension of bag-of-words for a big collection can reach hundreds of thousands; moreover, the document representation vectors, although sparse, may still have hundreds and thousands of nonzero components. Most of those words are irrelevant to the categorization task and dropped with no harm to the classifier performance and may even result in improvement owing to noise reduction. Feature selection is the preprocessing step that removes the irrelevant words.

IV. PROPOSED WORK

This framework includes parsing plain documents, getting 'TermFrequencies' for terms in documents, mapping terms to featureIDs, computing weights and removing sparse terms. After documents transformed into a representation suitable for SVM an existing SVM library named, Kernlab called for training and testing. The graphical representation of the text preprocessing step is given in Fig1. For most bags of words representations, each feature corresponds to a single word found in the training corpus, usually with case information and punctuation removed. Tokenization takes a text (i.e. a string) and discovers sentences and `tokens'. The tokenizer performs this function. In English text, this process is fairly simple since white spaces and punctuation marks separate words. The most common way make independence is to remove suffixes from words using a *stemming* algorithm such as the one developed by Lovins [1968]. Stemming has the effect of mapping several morphological forms of words to a common feature [23]. A famous algorithm for stemming is the Porter stemmer [Por80] based on removing suffixes from words (e.g. removing s from plural words)[24][25][26].

The R package Rstem and Snowball (encapsulating stemmers provided by Weka) packages carry out such stemming capabilities and used in combine with our tm infrastructure. Stopwords are words that are so common in a language that information value is almost zero, in other words, entropy is very low. The next step is to create a term document matrix with the function Term Document Matrix. In this function, the weight of a term in a document is equal to the number of times the term appears in the document, i.e. to the raw frequency of the term in the document. The next step is to remove whitespaces and punctuations. The next set of tasks for text mining includes creating a Term Document Matrix (or TDM), identifying frequently occurring words, and removing sparse terms. In other words, we'll remove terms which have at least a sparse percentage of empty elements. Preprocessing aims to represent documents in a format that is understandable to the classifier. We choose the training and test set. Then we classify the test set based on the created SVM. Figure2 shows the flow diagram of text mining architecture and Figure 3 shows its space occupied in number of kilobytes. Figure 4 shows the sparsity vs number of features.

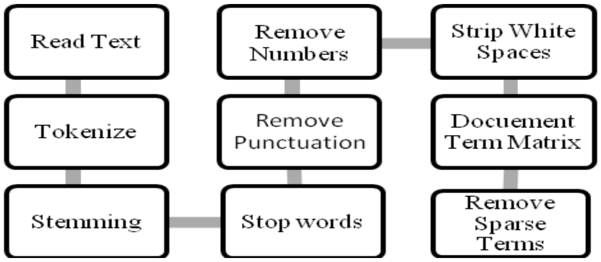

Figure2: Flow diagram of Textmining Architecture.

SVM is a practical algorithm that has been widely used in many areas. To guarantee its satisfying performance, it is important to set the proper parameters of SVM algorithm. As a simple and intelligent optimization algorithm, Particle Swarm Optimization (PSO) developed rapidly in recent years. The applications of PSO on combinatorial optimization problems are still limited, PSO has certain advantages such as easy to carry out and computationally efficient. The choice of SVM parameters has brought wide concern and many approaches during recent years; intelligent optimization has become more prevalent in recent years, involving genetic algorithm, PSO algorithm, and colony algorithm etc. During the optimization process, the key part is to select the fitness function.

A. *Direct Use of SVM:*

The penalty factor for training the SVM and PSO-SVM set to 5. RBF kernel function is a universal kernel function; after choice of the relevant parameters, applied to arbitrary distributive samples. In conclusion, RBF kernel function is generally applied in the Support Vector Machine [27]. The generalization ability of SVM algorithm depends on a set of parameters. We choose the training and test set. Then we classify the test set based on the created SVM. The table1 shows the performance measure of SVM classifier and Figure 3 shows the RBF function kernel. The parameters needs optimized are: RBF kernel parameter and the estimated accuracy. Use the 10-fold method to estimate the generalization ability. The original dataset was randomly divided into a two-third of a set (training set) and one-third of a set (testing set).The basic steps as follows:

a. Input the sample training set, and set a group of parameters {C, cross}.
b. Train SVM based on the parameters and calculates the cross validation error and obtains its object.
c. Test the SVM using object obtained from step 2.
d. Repeat the above step 25 times and find the average testing accuracy.

B. *The implementation of PSO-SVM:*

The procedure for describing proposed PSO-SVM is as follows:

a. Initialize PSO with population size, inertia weight.
b. Set cognitive and social learning rate as 2.
c. Set the number of particles and its dimension.
d. Set the training set as particle.
e. Take cross validation error of the SVM training set as fitness value. Evaluate fitness value of each particle.
f. Compare fitness value and calculates the local best and global best.
g. Update the inertia weight, velocity and position of each particle.
h. Repeat the step 4-7 till value of fitness function converges or the number of iteration reached.
i. After converging, the global best object fed in to SVM classifier for testing.

V. EXPERIMENTAL RESULTS

A. *Cross Validation:*

When we have finished the preprocessing, we use the SVM to do the classification. The cross validation will help to identify good parameters so that the classifier can accurately predict unknown data. In this paper, we used 10 fold cross validation to choose the penalty parameter C and γ in the SVM. When we get the nice arguments, we will use them to train model and do the final prediction [27].

B. *SMS Test Collections:*

A collection of about 10,000 legitimate messages collected for research at the Department of Computer Science at the National University of Singapore, called the *NUS SMS Corpus* (NSC).

A collection of English SMS messages, including 1002 legitimate messages randomly extracted from the NUS SMS Corpus and the Jon Stevenson Corpus, and 82 SMS spam messages collected from the Grumble text mobile spam site. This is a UK forum in which cell phone users make public claim about SMS spam messages, most of them without reporting the very spam message received. The identification of the text of spam messages in the claim is very hard and time-consuming task, and it involved carefully scanning 100 web pages.

We believe this collection resembles a realistic scenario, because both the legitimate and the spam messages are real messages; the proportion may be not accurate but we are not aware of the existence of real world statistics of spam received by cell phone users in the British/Singapore markets.

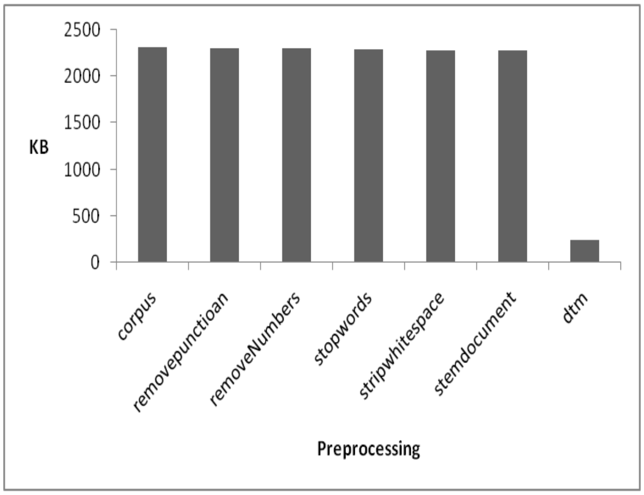

Figure3: Preprocessing method vs space.

C. *Used Environement and Libraries:*

Within the last years tm has gained interest from a variety of researchers and users of different background [28][29]. **R** is a programming language and software environment for statistical computing and graphicsR is more than a programming language. It is an interactive environment for doing statistics. We find it more helpful to think of R as having a programming language than being a programming language. The R language is the scripting language for the R environment. An R interface has been added to the popular data mining software Weka which allows for the use of the data mining capabilities in Weka and statistical analysis in R. kernlab for kernel learning provides ksvm and is more integrated into R so that different kernels can easily be explored [30][31]. The machine used was an Intel Core 2 Duo E7500 @ 2.93GHz with 2GB RAM.

D. *Performance Measure:*

PSO-SVM algorithm optimizes the best instance subset to classify the SMS as spam or non-spam. The research intends to compare the efficiency of SVM and PSO-SVM under different sparsity. Detection and identification of spam and non-spam SMS generalized as the following: True positive (TP): the number of spam SMS detected when it is actually spam SMS. True negative (TN): the number of non-spam SMS detected when it is actually non-spam. Classifiers have long been evaluated on their accuracy only. An often-used measure in the information retrieval and natural language processing communities is overall Accuracy, the other performance measures are kappa, rand and crand statistic. The Overall Accuracy (OA) is the most common and simplest measure to check a classifier. To measure performance of a classification model a random fraction of the labeled document set aside and not used for training. We may classify the documents of this test set with the classification model and compare the estimated labels with the true labels. Fraction of correctly classified documents to the total number of documents called overall accuracy and is a first performance measure. It is just defined as the degree of right predictions of a model. Figure 6 represent the performance of sparsity vs overall accuracy.Table.1 shows the performance measure of SVM and PSO-SVM under different sparse term document matrix. An often-used measure in the information retrieval and natural language processing communities is the F1-measure. According to Yang and Liu [1], this measure was first introduced by C. J. van Rijsbergen [32]. They state, the F1 measure combines recall (r) and precision (p) with an equal weight in the following form: $\frac{2RP}{R+P}$, where $R = \frac{TP}{TP+FN}x100\%$ and $P = \frac{TP}{FP+TP}x100\%$. Figure 6 shows the performance of F1-measure. TP is the number of true positives, i.e., the number of non-spam SMS cases predicted correctly. FP is the number of false positives, i.e., the number of cases incorrectly predicted as non-spam. FP is the number of false positives, i.e., the number of cases incorrectly predicted as non-spam. FN is the number of false negatives, i.e., the number of cases incorrectly predicted as spam. Figure 7 shows the F1 measure of SVM and PSO-SVM.

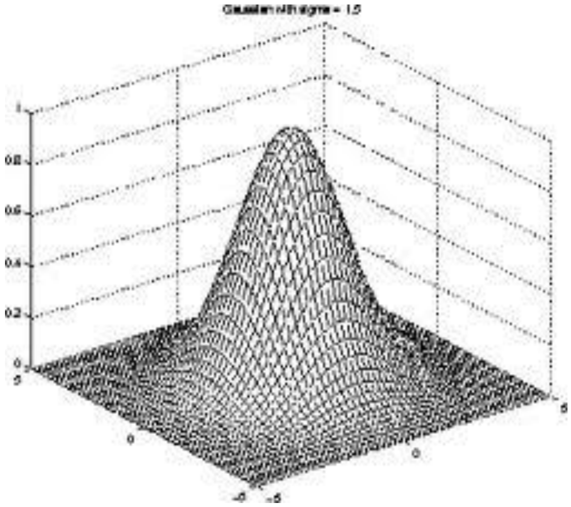

Figure8: RBF function kernel

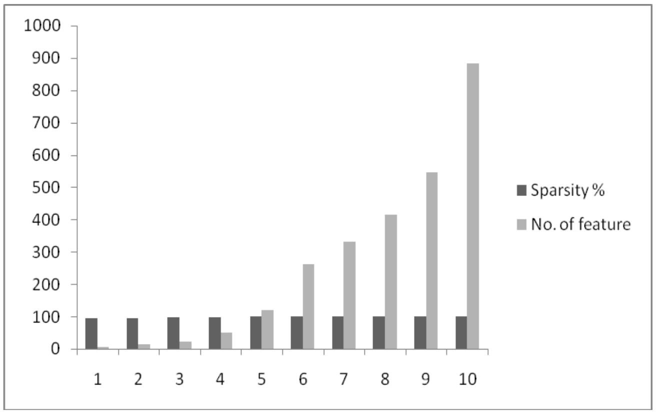

Figure 5: Sparsity Vs Number of features

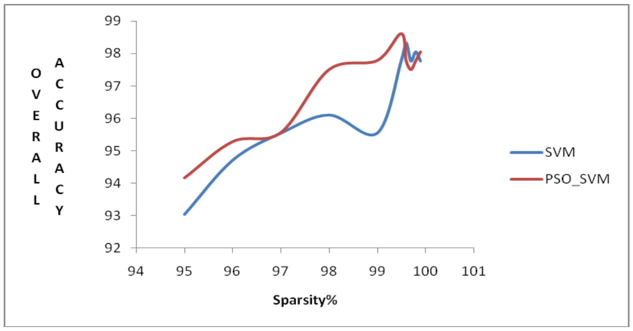

Figure 6: Performance of Overall Accuracy

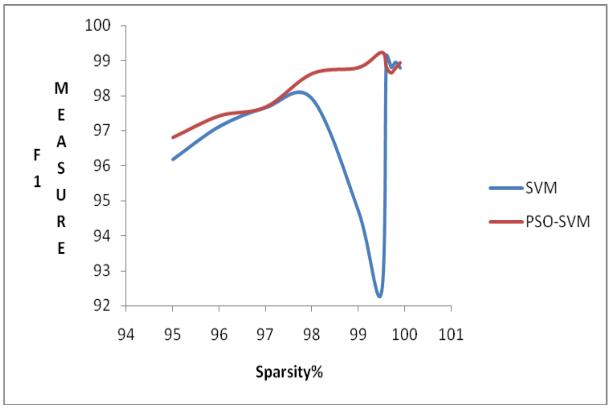

Figure 7:F1- Measure

VI. CONCLUSION

Efforts have been made for automatically discovering novel information from text. A goal of this paper is to make it easier for those interested in text mining to use open source software in their work. The required size of training set depends on the sparseness of the feature space. The size of the feature space is even high for small dataset. We expect that this model will be successful in efficiently classifying sms text documents. Implementation of this model will be further in future.

Table: 1Performance Measure

Sr.No.	Sparsity %	No. of features	sdtm1 (kb)	Accuracy		F1 Measure	
				SVM	PSO-SVM	SVM	PSO-SVM
1	95.0	5	55.1	93.05	94.16	96.18	96.81
2	96.0	13	60.8	94.72	95.28	97.12	97.43
3	97.0	22	66.5	95.55	95.56	97.66	97.68
4	98.0	49	78.0	96.11	97.50	97.91	98.64
5	99.0	119	97.0	95.55	97.78	94.70	98.81
6	99.5	262	118.9	97.77	98.61	92.42	99.25
7	99.6	332	126.9	98.33	97.78	99.12	98.83
8	99.7	415	135.1	97.77	97.50	98.81	98.66
9	99.8	545	145.9	98.05	97.77	98.95	98.80
10	99.9	884	168.9	97.77	98.05	98.79	98.95

VII. REFERENCES

[1]. http://timesofindiaindiatimes.com/tech/personaltech/computing/junk-sms-no-end-tomobile-pammes/articleshow/6247207.cms.

[2]. http://www.livemint.com/2010/07/27000020/Scourge-of-SMS-spam-swamps-mob.html.

[3]. C.J.C. Burges. „A tutorial on support vector machines for pattern recognition". Data Mining and Knowledge Discovery, 2(2): 955-974, 1998.

[4]. T. Joachims, "Learning to Classify Text Using Support Vector Machines" Dissertation, Kluwer, 2002.

[5]. J.T. Kwok. "Automated text categorization using support vector machine, In Proceedings of the International Conference on Neural Information Processing", Kitakyushu, Japan, Oct. 1998, pp. 347-351.

[6]. V. N. Vapnik. The nature of Statistical Learning Theory. Springer, Berlin, 1995.

[7]. N. Cristianini, and J. Shawe-Taylor, "Support Vector and Kernel Methods, Intelligent Data Analysis: An Introduction Springer – Verlag", 2003.

[8]. N.Cristianini, and J. Shawe-Taylor, "An introduction to support vector machines, Cambridge, UK: Cambridge University Press", 2004.

[9]. B. Schölkopf. C.J.C. Burges, and A.J. Smola,"Advances in Kernel Methods: Support Vector Learning", MIT Press, (Eds.), 1998.

[10]. A.J. Smola and B. Scholkopf, "Learning with kernels: Support Vector Machines, regularization, optimization, and beyond", Cambridge, MA: MIT press.

[11]. R.C. Eberhart, and J. Kennedy. "A new optimizer using particle swarm theory", Proceedings of the sixth international symposium on micro machine and human science pp. 39-43, IEEE service center, Piscataway,NJ, Nagoya, Japan, 1995.

[12]. R.C. Eberhart, and Y. Shi. "Particle swarm optimization: developments, applications and resources". Proc. congress on evolutionary computation 2001 IEEE service center, Piscataway, NJ., Seoul, Korea., 2001.

[13]. Y. Shi, and R.C. Eberhart, "Parameter selection in particle swarm optimization", Evolutionary Programming VII: Proc. EP 98 pp. 591-600. Springer-Verlag,, New York, 1998.

[14]. M.Carvalho, and T.B. Ludermir, "Particle swarm optimization of neural network architectures and weights", In Proc. of the 7th int. conf. on hybrid intelligent systems, (pp. 336_339), 2007.

[15]. M. Meissner, M. Schmuker, and G. Schneider, "Optimized particle swarm optimization (OPSO) and its application to artificial neural network training", BMC, Bioinformatics, 7, 125, 2006.

[16]. J. Yu, L. Xi, and S. Wang, "An improved particle swarm optimization for evolving feed forward artificial neural networks", Neural Processing Letters, 26(3), 217_231, 2007.

[17]. J. Salerno. "Using the particle swarm optimization technique to train a recurrent neural model", IEEE International Conference on Tools with Artificial Intelligence, 45_49, 1997.

[18]. M. Settles, B. Rodebaugh, and T. Soule,"Comparison of genetic algorithm and particle swarm optimizer when evolving a recurrent neural network", Lecture notes in computer science (LNCS): Vol. 2723, Proc. of the genetic and evolutionary computation conference, pp. 151_152, 2003.

[19]. M.E.H. Pedersen, "Tuning and Simplifying Heuristical Optimization",PhD Dissertation, University of Southampton, 2010.

[20]. M.E.H. Pedersen, and A.J. Chipperfield, Simplifying particle swarm optimization, Applied Soft Computing 10 (2) (2010) 618–628.

[21]. Fabrizio Sebastiani. Machine learning in automated text categorization, ACM Computing Surveys, 34(1):1–47, 2002.

[22]. Lovins, J. B, "Development of a stemming algorithm", Mechanical Translation and Computational Linguistics, 11, 22-31, 1968.

[23]. Porter M.F. "An algorithm for suffix stripping", 14, 130-137, 1980.

[24]. Porter M.F. "Snowball: A language for stemming algorithms",2001, http://snowball.tartarus.org/texts/introduction.html

[25]. Porter M.F, "The Lovins stemming algorithm",http://snowball.tartarus.org/algorithms/lovins/stemmer.html

[26]. SU Gao-li, Deng Fang-ping. "Introduction to Model selection of SVM Regression"[J]. Bulletin of Science and Technology, 2006.22(2):154-157

[27]. Ingo Feinerer. An introduction to text mining in R. R News, 8(2):19-22, October 2008

[28]. Ingo Feinerer, Kurt Hornik, and David Meyer. Text mining infrastructure in R. Journal of Statistical Software, 25(5):1-54, March 2008.

[29]. Karatzoglou, A., Smola, A., Hornik, K,, Zeileis, A., 2005, "kernlab – Kernel Methods.", R package, Version 0.6-2., Available from http://cran.R-project.org.

[30]. Alexandros Karatzoglou and Ingo Feinerer. Kernel-based machine learning for fast text mining in R. Computational Statistics & Data Analysis, 54(2):290-297, February 2010.

[31]. C. J. van Rijsbergen., 1979," Information Retireval". Butterworths, London.