ISSN No. 0976-5697

Volume 4, No. 2, Jan-Feb 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Noval Approach for Text Classification using Self Clustering Algorithm

S.Shanmugapriya*
Department of Computer Science & Engineering
Veltech Hightech Dr.Rangarajan Dr.Sakunthala Engineering College Avadi, Chennai-62
spriyavasan@yahoo.co.in

M.Monisha Devi
Department of computer Science & Engineering
Veltech Hightech Dr.Rangarajan Dr.Sakunthala Engineering College Avadi, Chennai-62
mmonishadevi.cse@gmail.com

Abstract: In a document, the high dimensionality of text has not been fruitful for the task of categorization, for which, feature clustering has been proven for text categorization problems and to reduce the dimensionality of feature vectors. In this paper, we propose a Self Clustering Algorithm (SC) for feature clustering in which the number of extracted Features is obtained automatically. In this method words are represented as distributions and processed one by one sequentially. Words with specific similar feature are clustered together. A new cluster is created for a word which is not similar to any existing cluster. Each and Every Cluster is characterized by a membership function with statistical mean and deviation. Once all the words have been fed in, a desired number of clusters are formed, having an extracted feature. Besides the user need not specify the number of extracted features in advance and trial -and - error for determining the appropriate number of extracted features can be avoided. Evaluation results for these tasks show that the proposed methodology obtains reliable performance for text classification tasks.

Keywords: Feature Clustering, Text Classification, Self Clustering

I. INTRODUCTION

Recently, text data processing approaches have attracted more and more attention. For example, two real-world data sets, 20 Newsgroups [1] and Reuters 21578[2] top-10, both have more than 15,000 features. Such high dimensionality is a severe obstacle for classification algorithms [3 & 4]. To alleviate this difficulty, feature reduction approaches are applied before document classification tasks are performed [5]. The main purpose of Feature Reduction is to reduce the classifiers computation load and to Increase data consistency. Two ways of doing Feature Reduction are Feature Selection and Feature Extraction [6], [7]. The feature selection methods select a subset of the original features and the classifier uses subset instead of all the original features to perform the text classification task. The feature extraction methods convert the representation of the original documents to a new representation based on a smaller set of synthesized features. The feature extraction methods convert the representation of the original documents to a new representation based on a smaller set of synthesized features. Feature clustering is one of effective techniques for feature extraction. The idea of feature clustering is to group the words with a high degree of pair wise semantic relatedness into clusters and each word cluster is then treated as a single feature. In this way, the dimensionality of the features can be drastically reduced. The current problems of the existing feature clustering methods are

a. The desired number of extracted features has to be specified in advance
b. When calculating similarities, the variance of the underlying cluster is not considered

In this paper, Self Clustering (SC) approach is to reduce the number of features for document classification. By this approach, the number of extracted features is obtained automatically. Words are represented as distributions and processed one by one sequentially. If a word is not similar to any existing cluster, a new cluster is created for this word. When all words have been fed in, desired clusters are formed. We then have one extracted feature for each cluster [8]. The extracted feature corresponding to a cluster is a weighted combination of the words contained in the cluster

II.LITERATURE SURVEY

In this section, we briefly discuss the related research in text categorization Most algorithms are based on the bag-of-words model for text (Salton and McGill, 1983). A simple but effective algorithm is the Naive Bayes method (Mitchell, 1997). For text classification, different variants of Naive Bayes have been used, but McCallum [9] and Nigam (1998) showed that the variant based on the multinomial model leads to better results. Support Vector Machines have also been successfully used for text classification (Joachims, 1998, Dumais et al., 1998). For hierarchical text data, such as the topic hierarchies of Yahoo! (www.yahoo.com) and the Open Directory Project (www.dmoz.org), hierarchical classification has been studied by Koller and Sahami (1997) [, Chakrabarti et al. (1997), Dumais and Chen (2000) [11].

To counter high-dimensionality various methods of feature selection have been proposed by Yang and Pedersen (1997), Koller and Sahami (1997) [12], [13] and Chakrabarti et al. (1997). Distributional clustering of words has proven to be more effective than feature selection in text classification and was first proposed by Pereira, Tishby, and Lee (1993) where "soft" distributional clustering was used to cluster nouns according to their conditional verb distributions. Note that since our main goal is to reduce the number of features and the model size, we are only interested in "hard clustering" where each word can be represented by its unique word cluster. For

text classification, Baker and McCallum (1998) [9] used such hard clustering, while more recently- Slonim and Tishby (2001) have sed the Information Bottleneck method for clustering words. Both Baker and McCallum (1998) [7], [8], [9] and Slonim and Tishby (2001) use similar agglomerative clustering strategies that make a greedy move at every agglomeration, and show that feature size can be aggressively reduced by such clustering without any noticeable loss in classification accuracy using Naive Bayes. Similar results have been reported for Support Vector Machines (Bekkerman et al., 2001).

To select the number of word clusters to be used for the classification task, Verbeek (2000) has applied the Minimum Description Length (MDL) principle (Rissanen, 1989) to the agglomerative algorithm of Slonim and Tishby(2001)[10].Two other dimensionality/feature reduction schemes are used in latent semantic indexing (LSI)(Deerwester et al., 1990) and its probabilistic version (Hofmann, 1999) . Typically these methods have been applied in the unsupervised setting and as shown by Baker and McCallum (1998) [7],[8], LS results in lower classification accuracies than feature clustering. We now list the main contributions of this paper and contrast them with earlier work. As our first contribution, we use an information-theoretic framework to derive a global objective function that explicitly captures the optimality of word clusters in terms of the generalized Jensen-Shannon divergence between multiple probability distributions To process documents; the bag-of-words model [8] is usually used. Let D be the matrix consisting of all the original documents with n features D = $\{d_1, d_2, \ldots\ldots d_n\}$ represented n documents. Let the word set W= $\{w_1, w_2, \ldots\ldots w_m\}$ be the feature set of the documents. Each document d_i, 1< i< n, can be represented as $d_i=\{d_{i1},d_{i2},d_{i3}, \ldots\ldots d_{im}\}$ where each d_{ij} denotes the number of occurrence of w_j in document d_i. The feature reduction task is to find a new word set W'= $\{w_1', w_2' \ldots\ldots w_k'\}$, k<m such that W and W′ work equally well for all the desired properties with D [8]. After feature reduction, each document d_i is converted to a new representation $d_i'= \{d_1', d_2' \ldots\ldots d_n'\}$ and the converted document set is D′= $\{d_{i1}', d_{i2}' \ldots\ldots d_{ik}'\}$ If k is very much smaller than m, computation cost can be drastically reduced.

III. FEATURE REDUCTION TECHNIQUES

In pattern recognition area, methods for dimensionality reduction are divided into two categories

A. Feature Selection:

The dimensionality is reduced [9] by selecting a subset of original features. The removed features are not used in the computation anymore. The aim of feature selection methods is to determine a subset of d features from the set of m, for which a criterion J will be maximized.

B. Feature Extraction:

The original vector space is transformed into a new one with some special characteristics and the reduction is made in a new vector space. Comparing to feature selection, all data features are used. In this case, they are just transformed (using a linear or non-linear transformation) to a reduced dimension space with the aim of replacing the original features by a smaller set of underlying features

C. Feature Clustering:

The aim is to find groups of similar features (or in other words, features that have the same or similar function in the vector space) and group them together [11]. A group (or cluster) is forming a new feature, which is also sometimes called concept. Feature clustering is an efficient approach for feature reduction [10] which groups all features into some clusters where features in a cluster are similar to each other, in which each word contributes to the synthesis of only one new feature. Each new feature is obtained by summing up the words belonging to one cluster [12]. Let D be the matrix consisting of all the original documents with m features and D′ be the matrix consisting of the converted documents with new k features. The new feature set W′= $\{w_1', w_2', \ldots\ldots w_k'\}$, corresponds to a partition $\{W_1, W_2 \ldots\ldots W_k\}$ of the original feature set W.

A document set D of n documents $d_1,d_2,\ldots\ldots\ldots,d_n$ and the Feature vector W of *m* words $w_1, w_2,\ldots .,w_m$ and the p classes of $c_1,c_2,...,c_p$ Then, the j^{th} feature value of converted document d_i' is calculated as follows

$$d_{ij}' = \sum w_t \in w_j \, d_{it}$$

IV. PROPOSED METHOD

There are some difficulties with the clustering-based feature extraction methods described in the previous section. First they have to be given the parameter k indicating the desired number of clusters to which all the patterns have to be assigned. Second the computation time depends on the number of iterations [13], which may be expensively high. We propose an approach to overcome these difficulties We develop an incremental word clustering procedure which uses a pre-specified threshold to determine the number of clusters automatically. Each word contains a similarity degree, between 0 and 1, to each cluster. Based on these degrees, a word with a larger degree will contribute a bigger weight than another one with a smaller degree to form a new feature corresponding to the cluster

A. Self Clustering (Sc) Algorithm:

In the self clustering algorithm, initially there will be no cluster and it will be created with increments from the trained data set. One feature pattern is considered in each time. If the input feature is similar enough to none of the existing clusters, a new cluster for the feature is created and the corresponding membership functions should be initialized [14], [15]. Otherwise, the input feature is combined to the existing cluster to which it is most similar and the corresponding membership functions of that cluster should be updated. Let *k* be the number of existing fuzzy clusters and S_j be the size of cluster G_j. Apparently, *k* is 0 initially. For a word w_i, where $\vec{w}_i = <w_{i1}, w_{i2}, \ldots\ldots\ldots w_{ip}>$. We calculate $\mu_{Gj}(\vec{w}_i) \geq þ$ where þ, $0 \leq þ \leq 1$, is a predefined threshold. The order in which the word patterns are fed in influences the clusters obtained and Sort all the patterns, in decreasing order, by their largest components.

Two cases may occur. First, there are no existing fuzzy clusters on which word $\vec{w}_i$ has passed the fuzzy similarity test. For this case, we assume that word $\vec{w}_i$ is not similar enough to any existing word cluster and a new word cluster G_h, h = k + 1 is created with

$$\vec{m}_h = \vec{w}_i, \quad \vec{\sigma}_h = \vec{\sigma}_0 \text{ ----------> (1)}$$

Where $\vec{\sigma}_0 = < \sigma_1\ \sigma_2\ \ldots\ldots\ldots\sigma_n >$ is a user –defined constant vector. Note that the new Cluster h contains only one member, word $\vec{w}_i$ at this time. The Number of Clusters is increased by 1 and the size of cluster h should be initialized

ie k=k+1, S_h=1----------> (**2**)

On the other hand, if there are existing fuzzy clusters on which word $\vec{w}$i has passed the similarity test, let the cluster with the largest membership degree be cluster *t*, i.e.,

$$t = \arg\max_{1 \le j \le k} (\mu_{Gj}(\vec{w}_i)) \text{ --------> (3)}$$

In this case, we assume that word is closer to cluster and cluster should be modified to include word as its member. The modification to cluster is as follows:

$$A = \frac{(S_j - 1)(\sigma_1 - \sigma_0)^2 + S_j m_{j1}^2 + w_{i1}^2}{S_j}$$

$$B = \frac{S_j + 1}{S_j}\left(\frac{S_j m_{j1} + w_{i1}}{S_j + 1}\right)^2$$

Where

$$\sigma_1 = \sqrt{A - B} + \sigma_0$$

$$m_j = \frac{S_j m_{j1} + w_{i1}}{S_j + 1} \text{ and } S_j = S_j + 1$$

The above process is iterated until all the input words have been processed. At the end, we have k fuzzy clusters. Note that with this approach, the data contained in a cluster have a high degree of similarity[16] among them. Besides, when new training data are considered, the existing clusters can be adjusted or new clusters can be created, without the necessity of generating the whole set of clusters from the scratch.

B. Procedure:

We give a more detailed Procedure of this process below:

a. Initialization:

a) Initialization of the Original Features: m and Classes as p
b) Initialization of the Initial Deviation as σ and Cluster as K=0
c) Input as W_1=[$P(C_1|W_1)$,.....,$P(C_p|W_i)$] where $1 \le i \le m$

b. Procedure:

For each pattern W_1, $1 \le i \le m$
Temp = $\{W_j | G_j(W_1) \ge p, 1 \le j \le k\}$
If temp == $\emptyset$

A new Cluster W_h h=k+1, is created by Eq (1) & (2)
Update Cluster K

Else let $W_a \in$ Temp be the cluster to which W_j is closed according to Eq (3)

Incorporate W_1 into W_a by A & B
End if
End for
Return with the Created K Clusters

C. Weighting Approach:

The feature reduction task can be written in the following form

$$D' = DT$$

Where

$$D = \begin{bmatrix} d1 \\ d2 \\ | \\ dn \end{bmatrix},\ D' = \begin{bmatrix} d1' \\ d2' \\ | \\ dn' \end{bmatrix},\ T = \begin{bmatrix} t11 & \cdots & t1k \\ | & & | \\ tm1 & \cdots & tmk \end{bmatrix} \text{ ---->4}$$

The goal is to find a transformation matrix *T* to convert *D* to *D'* in a desirable way. There are two weighting approaches to transform data sets with new features and then these transformed data sets are fed to a classifier to show the performances. One is "hard-clustering" approach [10], each feature exactly contributing to a new feature. In this case, the elements of transformation matrix *T [11]* in Eq-4 are defined as follows:

$t_{ij} = 1$, if $j = \arg\max_{1<\alpha<K}(\mu_{G\alpha}(wi))$;
0, Otherwise

The other one is "soft-clustering" [10] approach, each feature contributing to several features. To avoid the effect of large low-degree features [17], we only consider features which have degrees higher than the threshold ρ.

$t_{ij} = \mu_{Gj}(w_i)$, if $\mu_{Gj}(w_i) \ge \rho$
0, Otherwise

V. EXPERIMENTAL RESULTS

To show the effectiveness of our proposed method, experiments on well-known data sets for text classification research are performed Experiment works on the 20 Newsgroups[1] corpus which contains about 20000 articles taken from the Usenet newsgroups. These articles are evenly distributed over 20 categories; each category of 20 Newsgroups has about 1000 articles. We use two-thirds of the documents for training and the rest for testing. We compare our method with the Divisive Clustering (DC) method on classification accuracy and running speed. For convenience, our method with hard-clustering is abbreviated as H-SC and our method with soft-clustering is abbreviated as S-SC.

Table 1 and Table 2 show the classification accuracy (%) and execution time (sec) of the 20 Newsgroup[1] data set obtained by DC, H-SC, and S-SC, respectively. Note that the 20 Newsgroups data set contains 25718 features. As shown in Table 1, both H-SC and S-SC achieve higher accuracy[8] than DC with the number of extracted features less than 508 and almost the same accuracy when the number of extracted features is more than 508. S-SC achieves higher accuracy than H-SC, especially when the number of extracted features is small. Because the execution time of H-SC is equal to S-SC, we label "SC" in Table 2 to show the execution time of these two methods. In this case, our methods obviously perform

better than DC in execution time, especially when the number of extracted features is small. As the number of extracted features increases, patterns in each cluster have higher similarities to each other for both DC and our methods. For our methods, since the patterns are fed only once through the algorithm, the accuracy maybe lower than iterative learning approaches when average cluster size is small. DC obviously spends much more time than others such as H-SC & S-SC

Table: 1 Accuracy % Of Three Approaches On 20 Newsgroups Data With 1/3-2/3 Test-Training Split

METHOD	NUMBER OF FEATURES						
	21 (0.75)	**41 (0.85)**	**82 (0.87)**	**214 0.9)**	**508 (0.93)**	**1122 (0.945)**	**1452 (0.95)**
DC	80.54	82.28	84.43	85.82	87.08	87.76	88.08
H-SC	83.35	85.02	85.34	86.75	87.12	87.62	88.06
S-SC	84.75	85.93	85.99	86.92	87.49	87.86	88.24

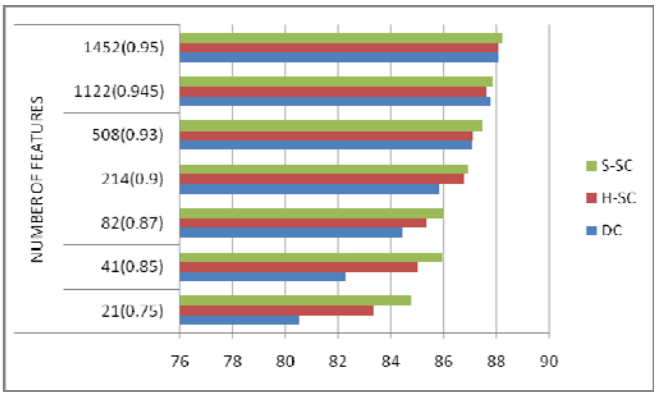

Figure1. Accuracy % Of Three Approaches On 20 Newsgroups Data With 1/3-2/3 Test-Training Split

Table 2. Accuracy % Of Three Approaches On 20 Newsgroups Data With Distributed Clustering & Self Clustering

METHOD	NUMBER OF FEATURES						
	21 (0.75)	41 (0.85)	82 (0.87)	214 (0.9)	508 (0.93)	1122 (0.94)	1452 (0.95)
DC	80.54	82.28	84.43	85.82	87.08	87.76	88.08
SC	8.87	14.9	24.18	57.59	313.6	635.5	994.9

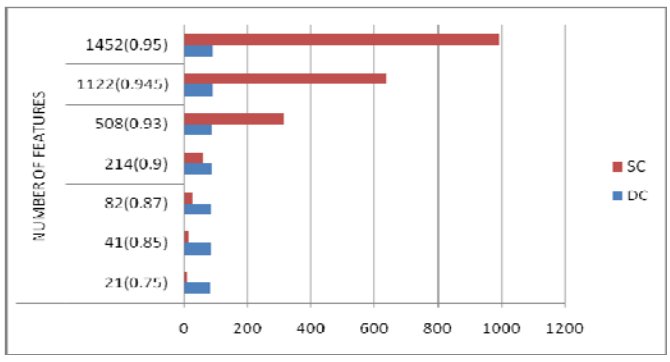

Figure 2 Accuracy % of Distributed Clustering with Self Clustering

VI. CONCLUSION AND FUTURE WORK

In this Self Constructing feature Clustering Algorithm for text Classification is described. Original features contribute weights to form new features according to a fuzzy similarity measure between original features and new features. The number of extracted features is obtained automatically according to a specified threshold. The Proposed approach has two advantages. Trial-and-error for determining the appropriate number of extracted features can be avoided. Computation demand is small and the method runs fast. In our approach it will give better performance than other methods.

VII. REFERENCES

[1] Http://people.csail.mit.edu/jrennie/20Newsgroups/, 2010.

[2] Http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html. 2010.

[3] H. Kim, P. Howland, and H. Park, "Dimension Reduction in Text Classification with Support Vector Machines," J. Machine Learning Research, vol. 6, pp. 37-53, 2005.

[4] F. Sebastiani, "Machine Learning in Automated Text Categorization,"ACM Computing Surveys, vol. 34, no. 1, pp. 1-47, 2002.

[5] B.Y. Ricardo and R.N. Berthier, Modern Information Retrieval. Addison Wesley Longman, 1999

[6] E.F. Combarro, E. Montan, I. Dıaz, J. Ranilla, and R. Mones, "Introducing a Family of Linear Measures for Feature Selection in Text Categorization," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 9, pp. 1223-1232, Sept. 2005.

[7] R. Kohavi and G. John, "Wrappers for Feature Subset Selection,"Aritficial Intelligence, vol. 97, no. 1-2, pp. 273-324, 1997.

[8] Jung-Yi Jiang, Ren-Jia Liou, and Shie-Jue Lee, "A Fuzzy Self Constructing Feature Clustering Algorithm for Text Classification", IEEE transactions on knowledge and data engineering, Volume. 23, NO:. 3, March 2011, pp. 335-349.

[9] L. D. Baker and A. McCallum. "Distributional clustering of words for text classification," 21st Annual International ACM SIGIR, pages 96–103, 1998

[10] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y.Winter. "Distributional word clusters vs. words for text categorization," Journal of Machine Learning Research, 3:1183–1208, March 2003.

[11] M.C. Dalmau and O.W. M. Florez. "Experimental results of the signal processing approach to distributional clustering of terms on reuters-21578 collection," 29th European Conference on IR Research, pages 678–681, 2007.

[12] I. S. Dhillon, S. Mallela, and R. Kumar. "A divisive information-theoretic feature clustering algorithm for text classification," Journal of Machine Learning Research, 3:1265–1287, March 2003.

[13] D. Ienco and R. Meo. "Exploration and reduction of the feature space by hierarchical clustering,"2008 SIAM Conference on Data Mining, pages 577–587, 2008.

[14] S. J. Lee and C. S. Ouyang. "A neuro-fuzzy system modeling with self-constructing rule generation and hybrid svd-based learning," IEEE Transactions on Fuzzy Systems, 11(3):341–353, June 2003.

[15] G. Salton and M. J. McGill. Introduction to Modern Retrieva,. McGraw-Hill Book Company, 1983.

[16] F. Pereira, N. Tishby, and L. Lee. "Distributional clustering of English words," 31st Annual Meeting of ACL, pages 183–190, 1993.

[17] F. Sebastian. "Machine learning in automated text categorization," ACM Computing Surveys, 34(1):1–47, March 2002.