Int. j. inf. tecnol. (December 2020) 12:1147–1157
https://doi.org/10.1007/s41870-018-0137-9

ORIGINAL RESEARCH

A novel semi supervised approach for text classification

Debaditya Barman[1] · **Nirmalya Chowdhury**[2]

Received: 25 December 2017 / Accepted: 2 April 2018 / Published online: 20 April 2018

Abstract Text categorization, also known as text classification is a supervised classification problem. It aims to assign a predefined class label or group to a new or unknown text document. Most of the time we need a collection of large data from each class to train the classifier. It may be noted that, it is very hard or expensive to collect labelled text data. In most cases we assign the label manually which is neither cost effective nor efficient. In this paper, we have introduced a semi-supervised classification approach where the learner needs very small amount of labelled data with a large amount of unlabeled data to assign a class label to a new or unknown text document. The proposed method uses Kohonen self organizing map (SOM) for labelling the unlabeled data and three classifiers namely support vector machine (SVM), Naïve Bayes (NB), and decision tree (DT): classification and regression tree (CART) for observing the accuracy of classification. The experimental results obtained show the effectiveness of our proposed method.

Keywords Text categorization · Semi supervised learning · Kohonen self organizing map · Naïve Bayes · Decision tree · Support vector machine

✉ Nirmalya Chowdhury
nirmalya63@gmail.com

Debaditya Barman
debbarmanhere@gmail.com

[1] Department of Computer and System Sciences, Visva-Bharati, Santiniketan 731235, India

[2] Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, India

1 Introduction

Advancement of social networking, blogging, micro-blogging has provided the opportunity to develop various applications in natural language processing (NLP). Text classification is a very popular research area of NLP. Over the last decade it grew exponentially [1–6] due to easy accessibility of the digital text documents. Text classification implies automated assignment of textual data to predefined classes. Sometimes either the number of such classes is not known or the class labels are not known. In this case initially some clustering technique is employed to obtain the appropriate grouping of a given set of text documents, then such groups are labelled based on some criteria or heuristic.

Several machine learning algorithms had been applied successfully to categorize text documents based on their content. Perhaps Naive Bayes (NB) algorithm is the frequently used classifier to solve text categorization problem. Researchers used two different generative models: *multi-variate Bernoulli* [7–11] and *multinomial* [12–16] event model while designing the NB classifier. Algorithms based on artificial neural network (ANN) [17–19], decision tree (DT) [20–23], k-nearest neighbor (KNN) [24–27], and SVM [28, 29] had been employed frequently to solve the text categorization problem. Ruiz and Srinivasan [5] proposed a method based on principal component analysis (PCA) and class profile based features (CPBF). It requires the number of classes present in the set of text documents to be known a priori. Text classification methods using linear vector quantizer (LVQ) network also require the number of classes to be supplied. A web news classification technique using neural networks proposed by Selamat et al. [6], which is based on PCA and requires the class profile-based features to be known a priori. Supervised text classification algorithms need a collection of sufficient number text data from each of the classes to predict. Classifiers performance varies with the nature of the dataset. So, we need a dataset that is not only large but also well distributed over the range of the classes. But labelled text data is hard to find/expensive. Most of the time we do the class labelling task manually which is neither efficient nor cost effective. Using clustering algorithms, we can extract some groups from a dataset when the class label is absent or not available. Text clustering is another popular research area in Text domain. Details of the several text clustering algorithms can be found in [30–38]. Clustering may find the correct groups presented in a collection of text, but it cannot provide the context or class label of the groups.

In this paper we have proposed a semi-supervised approach to predict a class label for new text document. In this approach we need only few labelled text data and large number of unlabeled data to do the prediction. Our proposed method uses Kohonen SOM and three classifiers SVM, NB, and DT (CART). We have extracted the groups present in the text data by applying Kohonen SOM. Then we took votes using the labelled data to assign class label to our extracted groups. In other words, we find the class label for which the number of labelled data present in that group is larger than that for all the remaining class label. This method is used to label all the groups of the text documents generated by SOM. In this way we are able to generate a complete labelled dataset from a dataset which is mostly unlabeled as stated earlier. The newly labeled data are used to train three classifiers namely SVM, NB, and DT (CART) and previously labelled data are used for the testing purpose. It may be noted that the results obtained with the aforesaid three most widely used classifiers are satisfactory.

The basic sketch of this paper is as follows. Next section presents the statement of the problem. Section 3 thoroughly describes the proposed method. Experimental results are presented in Sect. 4. Concluding remarks can be found in Sect. 5.

2 Statement of the Problem

Given a set of text documents $D = \{d_1, d_2, \ldots d_i, \ldots d_n\}$, where i_{th} document d_i is represented by its pattern vector x_i and n is the number of documents. Let the set of pattern vectors $S = \{x_1, x_2, \ldots x_i, \ldots x_n\} \subseteq \mathbb{R}^m$, where m is the dimensionality of the feature space. The value of m is decided in the Step 5 of feature selection steps described below. The problem is to assign each text document d_i into one of the predefined categories or labels (l_i). Let the set of categories is $L = \{l_1, l_2, \ldots l_i, \ldots, l_p\}$, where p is the number of class labels such that $p \ll n$. A typical text or document categorization task assigns a boolean value to each pair $\langle x_i, l_j \rangle \in D \times L$. A *True* value is assigned to $\langle x_i, l_j \rangle$ when a decision has been made to assign a label l_j to a document x_i otherwise a *False* value is assigned. Let a subset $D_l \subseteq D$, consists of labelled text documents and $D_{ul} \subseteq D$ consists of unlabelled text documents, where $|D_l| \ll |D_{ul}|, D_l \cup D_{ul} \equiv D$ and $D_l \cap D_{ul} \equiv \emptyset$. So our problem is to assign boolean values to the pairs $\langle x_i, l_j \rangle \in D_{ul} \times L$.

According to *Bag-of-words* technique, we can represent a text document (x_i) as a vector of frequency count of the words present in the text document. Essentially, the dimension of such a vector will be very high if we consider all the words. We have applied the following steps to remove some insignificant words that leads to dimensionality reduction of the feature vectors.

Step 1: Remove stop words from the documents: Remove stop words like determiners (e.g. a, an, the etc.), conjunctions (e.g. and, or, but etc.), prepositions (e.g. across, after, behind etc.), some adverbs (e.g. here, there, out etc.) from all the text documents. Complete list of the stop words can be found in [39].

Step 2: Extraction of the root words: Use 'Porter stemmer' to remove morphological affixes from the words.

Step 3: Select only the unique words from the documents. Suppose we have two documents d_1 and d_2, where $d_1 = \{w_1, w_2, w_1, w_3, w_4\}$, $d_2 = \{w_1, w_2, w_3, w_4, w_3, w_5, w_6\}$, so the set of unique words is W_p, $W_p = \{w_1, w_2, w_3, w_4, w_5, w_6\}$.

Step 4: Define EF_{wl_k} to be the elimination factor for the word w in context l_k as follows: $EF_{wl_k} = \frac{n_{l_k}}{N}$, where n_{l_k} is the frequency count of the word in the context l_k and N is the total frequency count. Remove all the words that have EF_{wl_k} value less than a predefined threshold (0.9) from W_p [41]. Let W_{ef} denotes a set of words that are having EF_{wl_k} value less than the predefined threshold. So $W_q = W_p - W_{ef} = \{w_i : w_i \in W_p \; and \; w_i \notin W_{ef}\}$. We have observed that $|W_q| \ll |W_p|$.

Step 5: All the remaining words in W_q are selected as the features of the text documents. Note that the feature values are finite positive integers (i.e. n_{l_k}) in nature.

Step 6: Construct pattern vectors for each of the document using the features selected in Step 4 and the length of the pattern vector is $|W_q|$.

Please note that using Step 4, we can remove those words that are used frequently irrespective of the class labels or context. For example, the word "win" can occur frequently in all the news article category (i.e. *Sports, Travel, Business,* and *Entertainment*) considered for the experimentation. Thus, the word *"win"* will have lower value of w_{efl_j}. It may also be noted that although the parameter '$tf - idf$' value is useful for estimating the relative importance of a specific word in terms of its total number of occurrence in the whole corpus but it does not at all reflect its importance for detection of a specific context/class.

We need to design a system which can assign a class level to a new or unlabelled text document.

3 Proposed method

In this work we have proposed a semi supervised approach to predict the class label of new or unlabelled text documents. Our proposed method uses Kohonen SOM and SVM algorithm. Kohonen SOM is used to discover the groups present in a given set of text documents. SVM is used to assign class labels to unlabelled text documents. Following algorithm presents the method for grouping the text documents.

We have manually labelled a small portion $(D_{labelled})$ of a given dataset (D). By taking about 50% of the labelled data, a test dataset D_{test} has been constructed. Rest of the labelled data, $D_{koho-labelled}$, $(D_{koho-labelled} = D_{labelled} - D_{test})$ will be used to label unknown data. It may be noted that, the test dataset (D_{test}) is used only for the testing purpose. We have applied Kohonen SOM algorithm on the Dataset D_{input},$(D_{input} = D - D_{test})$ to extract the groups. In other words, the unlabeled data and 50% of the manually labelled data are used in this process. Extracted groups are labeled by majority voting with the help of labeled data $(D_{koho-labelled})$. We have used only the newly labelled data $(D_{train} = D_{input} - D_{koho-labelled})$ to train our classifiers. The part of the manually labelled data (D_{test}) initially kept aside to act as a test dataset are now being used to test the performance of the chosen classifiers.

Following algorithm describes the proposed method for labelling the unknown data.

3.1 Algorithm: to assign label to a set of unlabeled text documents

For each text document d_i of a set of text documents $D_{input} = \{d_1, d_2, \ldots d_i, \ldots d_n\}$, apply its pattern vector x_i to the Kohonen's network. Here the Kohonen network will have m number of input nodes (since m is the dimensionality of the feature space) and $k(k = 16)$ output nodes. Output nodes are arranged in a 2-dimensional 4×4 grid. After convergence of the Kohonen's Algorithm, each high-density region of the feature space will be represented by one or more output nodes of Kohonen networks.

Input:
Set of pattern vectors $S, S = \{x_1, x_2, ..x_i, ..x_n\} \subseteq \mathbb{R}^m$, where $n = |D_{input}|$ and m is the dimensionality of the feature space.

Output:
A labelled set $(D_{input} - D_{koho-labelled})$ of text documents which were previously unlabeled.

Steps:

1. Initialize the weight $W_j, W_j \in \mathbb{R}^m$ from m inputs to the k output nodes to small random values.
2. Present a new input.
3. Compute distance to all output nodes

   $$dist_j = \sum_{i=1}^{n} (x_i(t) - w_{ij}(t))^2.$$

4. Select the kohonen-output node (j^*) with minimum distance.
5. Update weight to node (j^*).

   $$W_{ij}(t+1) = W_{ij}(t) + \beta(t)\big(x_i(t) - w_{ij}(t)\big), where\ 0 < \beta(t) < 1.$$

6. Repeat by going step 2. (Until all the inputs are presented).
7. Repeat by going step 3 (Until $t = max\_itn$).
8. Construct a Minimum Spanning Tree $S_{op}(V, E_s)$, where V is the set of output nodes $(V \equiv J = \{j_1, j_2, \ldots j_i, \ldots j_k\})$ and $E_s \subset E, E = \| j_u, j_v \|, 1 \le u \le k\ and\ 1 \le v \le k$.
9. Compute $e_{mean} = \frac{\sum_{i=1}^{n} e_i}{n}$, $where\ e_i \in E_s$.
10. Compute the number of clusters (q) present from S_{op}: $q = \left|E_s^{'}\left(where\, e \in E_s^{'}\, and\, e_i > c * e_{mean}\right)\right| + 1$, where $E_s^{'} \in E_s$ and c is a real constant.
11. Divide the $k(k = |V|)$ number of nodes into q number of clusters $(C_1, C_2, \ldots C_i, \ldots C_q)$ by disconnecting the edges $e_i > c * e_{mean}$, where c is a real constant. We need to have $p = q$. If $p < q$ increase the value of c in steps of 0.1 until we get $p = q$ else if $p > q$ decrease the value of c in steps of 0.1 until we get $p = q$.
12. Assign each d_i of D_{input} to the cluster C_y if the kohonen node which is nearest to d_i belongs to cluster C_y.

$$d_i \in C_y \quad \text{if} \quad \| d_i - j_y \|^2 < \| d_i - j_z \|^2 \; \forall y,$$
$$z \in \{1, 2, \ldots, k\}, y \neq z \text{ and } j_y \in C_y.$$

13. Assign class label l to i_{th} cluster C_i based on voting of class labels of manually labelled members present in C_i. $C_i = \{l : \#n_l > \#n_r\} \forall l, r \in \{1, 2, .., p\}$ and $l \neq r$, where $\#n_l$ and $\#n_r$ denotes the number of instances with the class label l and r respectively in cluster C_i and p is the number of class labels.
14. Return a labelled set $(D_{input} - D_{koho-labelled})$ of text documents which were previously unlabeled.

To validate the class label assignment, we can use any exiting classifier. We have chosen three very popular and widely used classification algorithms—Naïve Bayes (NB), decision tree (DT): classification and regression tree (CART), and support vector machine (SVM). We have trained these classifiers with the dataset D_{train} (training dataset) and tested the classifiers with the dataset D_{test} (test dataset). In this experiment we have used 10-fold cross validation approach.

3.2 Naïve Bayes

Assume we have a training dataset D, given as $\{(X_1, y_1), (X_2, y_2), \ldots, (X_i, y_p), \ldots, (X_{|D|}, y_q)\}$ where X_i is the set of training tuples or feature vector with class label y_p and $X_i = \vec{X} = \{x_1, x_2, \ldots, x_n\}$. Now, NB predicts the class label y_q for any $\vec{X}$ according to the following probability.

$$p(y_q|\vec{X}) = p(y_q|x_1, x_2, \ldots, x_n) \text{ for } q = 1, \ldots, r \tag{1}$$

According to the *Bayes's* theorem, we can write,

$$p(y_q|\vec{X}) = \frac{p(\vec{X}|y_q) \cdot p(y_q)}{p(\vec{X})} = \frac{p(x_1, x_2, \ldots, x_n|y_q) \cdot p(y_q)}{p(x_1, x_2, \ldots, x_n)}. \tag{2}$$

The denominator is not dependent on y_q and the value of $\vec{X}$ is given so effectively it is a constant. So, we are interested only in the numerator. The factor $p(x_1, x_2, \ldots, x_n|y_q)$ in numerator can be written as following using *chain* rule.

$$p(x_1, x_2, \ldots, x_n|y_q) = p(x_1|x_2, \ldots, x_n, y_q)p(x_2|x_3, \ldots, x_n, y_q) \cdots p(x_{n-1}|x_n, y_q)p(x_n|y_q) \tag{3}$$

Now, if we assume that any feature x_i is independent of any other feature x_j given the class label y_q then using Eq. (3) we can write

$$p(x_i|x_{i+1}, \ldots, x_n, y_q) = p(x_i|y_q) \Rightarrow p(x_1, \ldots, x_n|y_q) = \prod_{i=1}^{n} p(x_i|y_q). \tag{4}$$

So,

$$\begin{aligned} p(y_q|x_1, \ldots, x_n) &\propto p(y_q, x_1, \ldots, x_n) \\ &\propto p(y_q) \cdot p(x_1, \ldots, x_n|y_q) \\ &\propto p(y_q) \cdot \prod_{i=1}^{n} p(x_i|y_q) \end{aligned} \tag{5}$$

In practice, Binomial distribution or Gaussian distribution is used to model the class conditional feature probabilities $p(x_i|y_q)$. So we can assign a class label y_q to a data $\vec{X}$ by computing $p(y_q) \cdot \prod_{i=1}^{n} p(x_i|y_q)$ for $q = 1, \ldots, r$ and assigning $\vec{X}$ the class y_q for which the value is largest. So we can write-

$$\bar{y} = \underset{q \in \{1, \ldots, r\}}{\operatorname{argmax}} \, p(y_q) \cdot \prod_{i=1}^{n} p(x_i|y_q),$$

where $\bar{y}$ is the predicted class label for $\vec{X}$.

3.3 Decision tree (CART)

CART algorithm recursively partitions a given training dataset (D) to obtain pure subsets with respect to a given class. A certain set of tuples $F(F \subseteq D)$ associated with each node of a tree can be splitted based on a specific test on an attribute. Suppose, we have a continuous attribute C, a threshold t, and the test is $C \leq t$. We can do the partition on F and generate two subsets (one for right branch and another one for left) based on the following criteria.

$$F_r = \{f \in F : f(C) \leq t\} \text{ and}$$
$$F_l = \{f \in F : f(C) > t\}$$

Similarly, the split can be done based on any categorical feature H. If, $H = \{h_1, h_2, \ldots, h_k\}$, then each branch i can be constructed using the test $H = h_i$.

This recursive algorithm has a *divide* step to construct the decision tree by selecting the best split according to a predefined quality measure from all possible splits. Usually the algorithm evaluate each splits based on the *impurity* measurement. CART uses Gini index to measure the *impurity*. It may be noted that, the split should decrease the *impurity* of the parent node. Let the dataset has following scheme $(C_1, C_2, \ldots C_i, \ldots, C_m, Y)$, where C_i is the attribute and Y is the target class label. Let $(F_1, F_2, \ldots, F_k)$ be a split generated from a set of tuples F and $IM()$ is the *impurity* measure function then the splitting criterion is

$$\Delta = IM(F) - \sum_{i=1}^{k} \frac{|F_i|}{|F|} IM(F_i) \tag{6}$$

For CART, $IM()$ is the Gini index.

$$IM(F) = Gini(F) = 1 - \sum_{j=1}^{n} q_j^2 \tag{7}$$

where n is the number of class and q_j is the fraction of tuples in F that belong to a particular class y_j.

$$q_j = \frac{|\{f \in F : f[Y] = y_j\}|}{|F|} \tag{8}$$

3.4 Support vector machine

Assume we have a training dataset D, given as $\{(X_1, y_1), (X_2, y_2), \ldots, (X_i, y_p) \ldots, (X_{|D|}, y_q)\}$ where X_i is the set of training tuples or feature vector with class label y_p. It may be noted that $X_i \in \mathbb{R}^n$ and $y_p \in (+1, -1)$. Given above stated situation, SVM [44, 45] requires solving the following minimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i, \tag{9}$$

$$\text{subject to } y_p\left(w^T \emptyset(X_i) + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0$$

The function $\emptyset$ maps the feature vector X_i to higher dimension space. In higher dimension SVM finds the hyperplane which separates the classes by maximum margin. C is a penalty parameter and $C > 0$. We can define a kernel function $K(X_i, X_j) = \emptyset(X_i)^T \emptyset(X_j)$. Frequently used kernel functions are [46]:

1. Linear: $K(X_i, X_j) = X_i^T X_j$.
2. Polynomial: $K(X_i, X_j) = \left(\gamma X_i^T X_j + r\right)^d, \gamma > 0$.
3. Radial basis function:

   $K(X_i, X_j) = \exp(-\gamma \parallel X_i - X_j \parallel^2), \gamma > 0.$
4. Sigmoid:

   $K(X_i, X_j) = \tanh(\gamma X_i^T X_j + r).$

Here $\gamma, r,$ and d are kernel parameters.

Linear kernel is best suited for linearly separable dataset. Polynomial, radial basis function, and sigmoid kernel functions are generally used on non-linearly separable dataset. These kernels are also known as non-liner kernels.

In our experiment, we have used linear kernel instead of non-linear kernels for following reasons:

(i) Since text dataset is very high dimensional in nature, we can use linear kernel instead of nonlinear kernel without sacrificing any significant performance [46, 48].

(ii) For linear kernel, there is no need for implicit mapping of data to high dimension ($\emptyset(X)$). So, training time for SVM with linear kernel is less than the SVM with nonlinear kernels.

(iii) Unlike the non-liner kernels, we need to estimate only the penalty parameter (C). Please note that, *"grid search"* have been used to find the optimum value of C. There exists several advanced methods (e.g. Randomized parameter optimization, Gradient based optimization, Bayesian optimization etc.) which can save computation cost. However, we prefer the simple *"grid search"* for following reasons.

   (a) Since it is an exhaustive process it will return a good value.

   (b) The computational time required to find a good value for only one parameter by *"grid search"* is not much than the time required for the above mentioned advanced techniques.

It may be noted that, in SVM *'one vs. one'* approach is used over *'one vs. all'* approach for two reasons. One is that, our problem deals with multiple classes. The other reason is that, training time required for *'one vs. one'* approach is much less than *'one vs. all'* approach [47].

4 Experimental results

We have tested our proposed technique on five different datasets from two different domains (i.e. News Articles and Product reviews).

Following are the news article datasets.

1. BBC Dataset [42]: This dataset contains 2225 news articles collected from BBC news website. Class labels are business (B), entertainment (E), politics (P), sport (S), and technology (T).
2. BBCSport Dataset [42]: This dataset contains 737 news articles collected from BBC Sport news website. Class labels are athletics (A), cricket (C), football (F), rugby (R), and tennis (T).

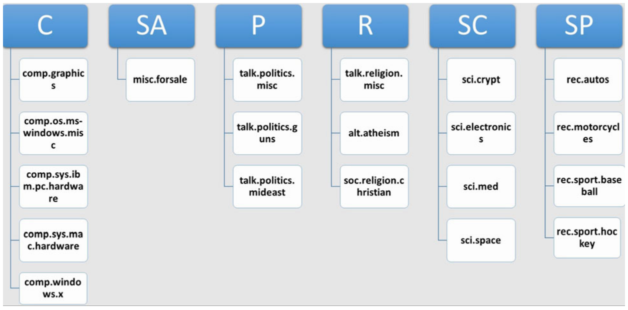

Fig. 1 Class description of the 20_newsgroup_subset dataset

Table 1 Description of experimental dataset

Name of the dataset	Labelled ($D_{labelled}$)	Test (D_{test})	Input data (D_{input})	Total ($D = D_{test} + D_{input}$)
BBC News	223	111	2114	2225
BBC Sport News	74	37	700	737
20 News Groups_subset	600	300	5700	6000
Product Review 1	490	245	4655	4900
Product Review 2	330	165	3135	3300

Table 2 Description of test data (D_{test}) for ‘BBC News’

Name of the dataset	Categories				
	B	E	P	S	T
BBC News	25	20	21	25	20

Table 3 Description of test data (D_{test}) for ‘BBC Sport News’

Name of the dataset	Categories				
	A	C	F	R	T
BBC Sport News	5	6	13	8	5

Table 4 Description of test data (D_{test}) for ‘20 Newsgroups_subset’

Name of the dataset	Categories					
	C	SA	P	R	SC	SP
20 Newsgroups_subset	75	15	45	45	60	60

3. 20 Newsgroups_subset Dataset [43]: Original dataset contains 20,000 messages taken from 20 different newsgroups. 6000 messages are randomly chosen to form the subset. Multiple newsgroups are grouped together to form the class labels. Class labels are computer (C), sales (SA), politics (P), religion (R), science (SC), and sports (SP). Following Fig. 1 describes how class labels are formed.

To construct product review datasets, we have developed a web crawler and a page parser program. These programs will automatically collect consumer’s reviews on various products from a popular Indian e-Commerce site.

Following are the product review datasets.

1. Product Review 1: This dataset contains 4900 product reviews collected from Flipkart.[1] Class labels are camera (C), television (TV), mobile (M), and laptop (L).

Table 5 Description of test data (D_{test}) for ‘Product Review 1’

Name of the dataset	Categories			
	C	TV	M	L
Product Review 1	52	52	79	62

Table 6 Description of test data (D_{test}) for ‘Product Review 2’

Name of the dataset	Categories			
	AC	R	T	WM
Product Review 2	45	25	50	45

Table 7 Confusion matrix for the classification results using NB classifier on BBC News Dataset

	Truth data					Precision (%)
	B	E	P	S	T	
Classifier result						
B	17	1	1	1	5	68.00
E	0	15	0	4	1	75.00
P	2	0	19	1	0	86.36
S	1	3	1	18	1	75.00
T	5	1	0	1	13	65.00
Recall (%)	68.00	75.00	90.48	72.00	65.00	

Table 8 Confusion matrix for the classification results using DT(CART) classifier on BBC News Dataset

	Truth data					Precision (%)
	B	E	P	S	T	
Classifier result						
B	17	0	1	1	2	80.95
E	0	15	0	3	2	75.00
P	3	1	15	0	0	78.95
S	1	4	3	19	3	63.33
T	4	0	2	2	13	61.91
Recall (%)	68.00	75.00	71.43	76.00	65.00	

[1] http://www.flipkart.com.

Table 9 Confusion matrix for the classification results using SVM classifier on BBC News Dataset

	Truth data					Precision (%)
	B	E	P	S	T	
Classifier result						
B	19	1	1	1	3	76.00
E	0	17	0	3	1	80.95
P	2	0	19	0	0	90.48
S	0	2	1	21	2	80.77
T	4	0	0	0	14	77.78
Recall (%)	76.00	85.00	90.48	84.00	70.00	

Table 10 Confusion matrix for the classification results using NB classifier on BBC Sport News Dataset

	Truth data					Precision (%)
	A	C	F	R	T	
Classifier result						
A	3	0	1	0	0	75.00
C	0	5	1	1	1	62.50
F	1	0	9	0	0	90.00
R	1	0	2	6	0	66.67
T	0	1	0	1	4	66.67
Recall (%)	60.00	83.33	69.23	75.00	80.00	

Table 11 Confusion matrix for the classification results using DT (CART) classifier on BBC Sport News Dataset

	Truth data					Precision (%)
	A	C	F	R	T	
Classifier result						
A	3	0	1	1	0	60.00
C	0	4	3	0	1	50.00
F	2	1	8	0	0	72.73
R	0	0	1	7	0	87.50
T	0	1	0	0	4	80.00
Recall (%)	60.00	66.67	61.54	87.50	80.00	

2. Product Review 2: This dataset contains 3300 product reviews collected from Flipkart. Class labels are air conditioner (AC), refrigerator (R), tablet (T), and washing machine (WM).

Table 12 Confusion matrix for the classification results using SVM classifier on BBC Sport News Dataset

	Truth data					Precision (%)
	A	C	F	R	T	
Classifier result						
A	3	0	0	0	0	100.00
C	0	5	1	1	1	62.50
F	1	0	10	0	0	90.91
R	1	0	2	7	0	70.00
T	0	1	0	0	4	80.00
Recall (%)	60.00	83.33	76.92	87.50	80.00	

Table 13 Confusion matrix for the classification results using NB classifier on 20 News Groups_subset Dataset

	Truth data						Precision (%)
	C	SA	P	R	SC	SP	
Classifier result							
C	60	2	0	0	7	0	86.96
SA	4	10	3	0	1	3	47.62
P	1	0	33	2	0	2	86.84
R	0	0	2	37	0	0	94.87
SC	9	2	0	0	50	8	72.46
SP	1	1	7	6	2	47	73.44
Recall (%)	80.00	66.67	73.33	82.22	83.33	78.33	

Table 14 Confusion matrix for the classification results using DT (CART) classifier on 20 News Groups_subset Dataset

	Truth data						Precision (%)
	C	SA	P	R	SC	SP	
Classifier result							
C	58	2	0	0	9	0	84.06
SA	4	10	1	0	2	3	50.00
P	1	0	33	3	0	4	80.49
R	0	0	5	35	0	0	87.50
SC	10	3	0	0	47	9	68.12
SP	2	0	6	7	2	44	72.13
Recall (%)	77.33	66.67	73.33	77.78	78.33	73.33	

Details of $D, D_{labelled}, D_{test},$ and D_{input} of five different experimental datasets can be found in Table 1. Detailed descriptions of five test datasets can be found in Tables 2, 3, 4, 5, and 6. Performance of NB classifier on the five experimental datasets can be found in Tables 7, 10, 13, 16, and 19. Tables 8, 11, 14, 17, and 20 describe the performance of DT (CART). Performance of SVM classifier can be found in Tables 9, 12, 15, 18, and 21. Comparisons among three different classifiers (i.e. NB, DT (CART), and SVM) can be found in Table 22.

Table 15 Confusion matrix for the classification results using SVM classifier on 20 News Groups_subset Dataset

	Truth data						Precision (%)
	C	SA	P	R	SC	SP	
Classifier result							
C	66	2	0	0	5	0	90.41
SA	2	11	1	0	0	2	68.75
P	0	0	38	2	0	3	88.37
R	0	0	1	41	0	0	97.62
SC	7	1	0	0	53	4	81.54
SP	0	1	5	2	2	51	83.61
Recall (%)	88.00	73.33	84.44	91.11	88.33	85.00	

Table 16 Confusion matrix for the classification results using NB classifier on Product review 1 Dataset

	Truth data				Precision (%)
	C	TV	M	L	
Classifier result					
C	38	0	8	10	67.86
TV	5	41	3	1	82.00
M	7	7	57	8	72.15
L	2	4	11	43	71.67
Recall (%)	73.01	78.85	72.00	69.36	

Table 17 Confusion matrix for the classification results using DT (CART) classifier on Product review 1 Dataset

	Truth data				Precision (%)
	C	TV	M	L	
Classifier result					
C	36	1	9	6	69.23
TV	4	43	5	3	78.18
M	7	5	54	9	72.00
L	5	3	11	44	69.84
Recall (%)	69.23	82.69	68.00	70.97	

Overall accuracy of SVM, NB, and DT (CART) classifiers can be found in Table 22. We also have measured Kappa value to compare the performance of these three classifiers.

It can be noted from Table 22 that Kappa value is greater than 0.61 in every case. Kappa statistic or value represents the difference between accuracy of the classification system to the accuracy of a random system (e.g. Kappa of 0.7 means a classification system is 70% better than a random system). According to Landis and Koch [40] Kappa value of greater than 0.6 indicates a decent or

Table 18 Confusion matrix for the classification results using SVM classifier on Product review 1 Dataset

	Truth data				Precision (%)
	C	TV	M	L	
Classifier result					
C	46	0	6	7	77.97
TV	0	47	3	0	94.00
M	2	2	61	5	87.14
L	4	3	9	50	75.76
Recall (%)	88.46	90.39	77.00	80.65	

Table 19 Confusion matrix for the classification results using NB classifier on Product review 2 Dataset

	Truth data				Precision (%)
	AC	R	T	WM	
Classifier result					
AC	33	4	3	4	75.00
R	8	18	0	8	52.94
T	0	0	47	0	100.00
WM	4	3	0	33	82.50
Recall (%)	73.33	72.00	94.00	73.33	

Table 20 Confusion matrix for the classification results using DT (CART) classifier on Product review 2 Dataset

	Truth data				Precision (%)
	AC	R	T	WM	
Classifier result					
AC	32	7	5	5	65.31
R	8	16	0	8	50.00
T	0	0	45	0	100.00
WM	5	2	0	32	82.05
Recall (%)	71.11	64.00	90.00	71.11	

Table 21 Confusion matrix for the classification results using SVM classifier on Product review 2 Dataset

	Truth data				Precision (%)
	AC	R	T	WM	
Classifier result					
AC	37	4	2	1	84.09
R	6	19	0	3	67.86
T	0	0	48	0	100.00
WM	2	2	0	41	91.11
Recall (%)	82.22	76.00	96.00	91.11	

Table 22 Comparison of the classifiers

Name of the dataset	Classifier	Overall accuracy (%)	Kappa
BBC News Dataset	NB	73.87	0.673
	DT (CART)	71.17	0.639
	SVM	81.08	0.763
BBC Sport News Dataset	NB	72.97	0.654
	DT (CART)	70.27	0.617
	SVM	78.38	0.720
20 News groups_subset	NB	79.00	0.741
	DT (CART)	75.67	0.700
	SVM	86.67	0.835
Product Review 1 Dataset	NB	73.06	0.637
	DT (CART)	72.25	0.627
	SVM	83.27	0.775
Product Review 2 Dataset	NB	79.39	0.723
	DT (CART)	75.76	0.674
	SVM	87.88	0.836

significant classifier. Thus, we can conclude that our proposed method of labelling a large size of unlabeled data using a relatively small size of labelled data is efficient enough to provide an acceptable label of accuracy for classification.

We have reported our experimental results using labelled data ($D_{labelled}$) of size 10% of that of total dataset for first three datasets (i.e. BBC News Dataset, BBC Sport News Dataset, and 20 News groups_subset) in Table 22. It may be noted that these three datasets were already labelled. We have manually labelled 10% of that of total dataset ($D_{labelled}$) for last two datasets (i.e. Product Review 1, Product Review 2 Dataset). It may be noted that, these datasets were not labelled. We have also carried out experiment with different size of manually labelled data from 1 to 25% of the size of total dataset for two datasets (i.e. Product Review 1 and Product Review 2) to study the effect of size of manually labelled data on classification accuracy. Figures 2 and 3 present the findings of the said experiments.

The changes in overall accuracy w.r.t. initial number of labeled data in the total dataset, can be found in Figs. 2 and 3. As you can see from the Figs. 2 and 3, changes in the overall accuracies are very sharp when the number of labeled data is increasing from small but gradually it became stagnant with increment of the labeled data.

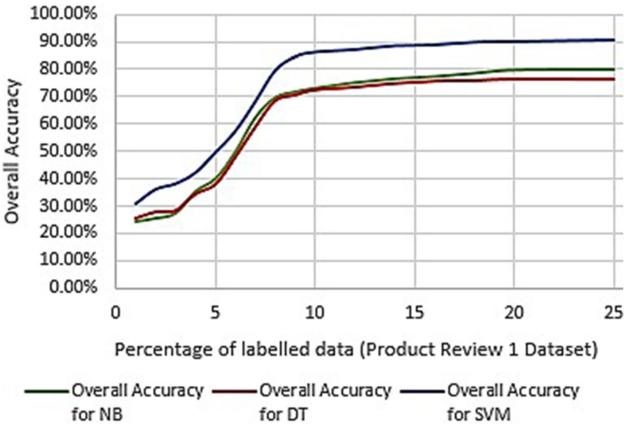

Fig. 2 Change in overall accuracy w.r.t. percentage of labeled data (Product Review 1 dataset)

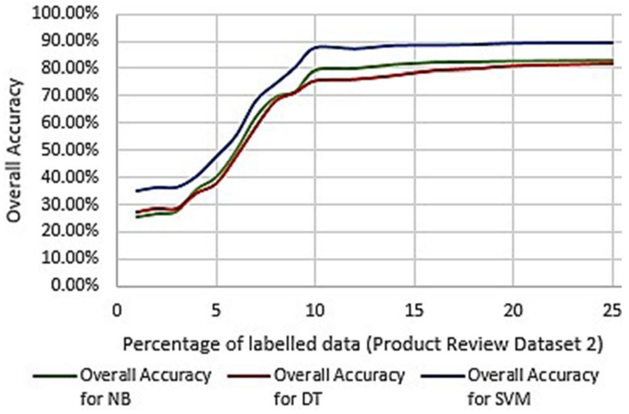

Fig. 3 Change in overall accuracy w.r.t. percentage of labeled data (Product Review 2 dataset)

5 Conclusion

In this paper we have proposed a semi supervised method for text document classification. Note that, adequate number of labelled data required for training of classifiers are not always easily available. Here our intention is to use a small amount of labelled data to label a relatively large amount of unlabeled data. The proposed method uses Kohonen SOM for labelling the unlabeled data and three classifiers namely SVM, NB, and DT (CART) for observing the accuracy of classification. We have selected five datasets for experimentation. In all five datasets SVM delivers better performance than NB and DT (CART) classifiers. We have concluded that our proposed method can efficiently assign label to a set of large unlabeled data with the help of very small labelled dataset. In future, we will try to use various other clustering algorithms which can efficiently handle high-dimensional data to improve our system's performance. Various feature selection techniques can be used to reduce dimensionality of the data.

References

1. Torkkola K (2004) Discriminative features for text document classification. Formal Pattern Anal Appl 6(4):301–308
2. Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv (CSUR). 34(1):1–47
3. Han EHS, Karypis G (2000) Centroid-based document classification: analysis and experimental results. In: European conference on principles of data mining and knowledge discovery. Springer, Berlin, pp 424–431
4. Basili R, Moschitti A (2001) A robust model for intelligent text classification. In: Proceedings of the 13th international conference on tools with artificial intelligence. IEEE, pp 265–272
5. Ruiz ME, Srinivasan P (1999) Hierarchical neural networks for text categorization (poster abstract). In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 281–282
6. Selamat A, Yanagimoto H, Omatu S (2002) Web news classification using neural networks based on PCA. In: SICE 2002. Proceedings of the 41st SICE annual conference, vol 4. IEEE, pp 2389–2394
7. Robertson SE, Jones KS (1976) Relevance weighting of search terms. J Am Soc Inf Sci 27(3):129–146
8. Lewis DD (1992) An evaluation of phrasal and clustered representations on a text categorization task. In: Proceedings of the 15th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 37–50
9. Kalt T, Croft WB (1996) A new probabilistic model of text classification and retrieval. Technical Report IR-78, University of Massachusetts Center for Intelligent Information Retrieval
10. Larkey LS, Croft WB (1996) Combining classifiers in text categorization. In: Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 289–297
11. Sahami M (1996) Learning limited dependence Bayesian classifiers. In: KDD, vol 96, pp 335–338
12. Lewis DD, Gale WA (1994) A sequential algorithm for training text classifiers. In: Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval. Springer, New York, pp 3–12
13. Guthrie L, Walker E, Guthrie J (1994) Document classification by machine: theory and practice. In: Proceedings of the 15th conference on computational linguistics. Association for Computational Linguistics, vol 2, pp 1059–1063
14. Li H, Yamanishi K (1997) Document classification using a finite mixture model. In: Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the European chapter of the association for computational linguistics, pp 39–47. Association for Computational Linguistics
15. Nigam K, McCallum A, Thrun S, Mitchell T (1998) Learning to classify text from labeled and unlabeled documents. In: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence, pp 792–799
16. McCallum A, Rosenfeld R, Mitchell TM, Ng AY (1998) Improving text classification by shrinkage in a hierarchy of classes. In: ICML, vol 98, pp 359–367
17. Carpenter GA, Grossberg S, Reynolds JH (1991) ARTMAP: supervised real-time learning and classification of nonstationary data by a self-organizing neural network. Neural Netw 4(5):565–588
18. Lam SL, Lee DL (1991) Feature reduction for neural network based text categorization. In: Proceedings 6th international conference on database systems for advanced applications, 1999. IEEE, pp 195–202
19. Wermter S (2000) Neural network agents for learning semantic text classification. Inf Retr 3(2):87–103
20. Fuhr N, Hartmann S, Lustig G, Schwantner M, Tzeras K, Knorz G (1991) AIR/X-a rule based multistage indexing system for large subject fields. In: RIAO, vol 91, pp 606–623
21. Apte C, Damerau F, Weiss, S (1998) Text mining with decision rules and decision trees. IBM Thomas J. Watson Research Division
22. Moulinier I (1997) Is learning bias an issue on the text categorization problem. Technical Report, LAFORIA-LIP6, Universite Paris VI
23. Lewis DD, Ringuette M (1994) A comparison of two learning algorithms for text categorization. In: Third annual symposium on document analysis and information retrieval, vol 33, pp 81–93
24. Lam W, Ho CY (1998) Using a generalized instance set for automatic text categorization. In: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 81–89
25. Masand B, Linoff G, Waltz D (1999) Classifying news stories using memory based reasoning. In: Proceedings of the 15th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 59–65
26. Yang Y(1994) Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. Springer, New York, pp 13–22
27. Yang Y, Pedersen JO (1996) Feature selection in statistical learning of text categorization. Center for Machine Translation, Carnegie Mellon University
28. Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. In: European conference on machine learning. Springer, Berlin, pp 137–142
29. Leopold E, Kindermann J (2002) Text categorization with support vector machines. How to represent texts in input space? Mach Learn 46(1–3):423–444
30. Beil F, Ester M, Xu X (2002) Frequent term-based text clustering. In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 436–442
31. Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice-Hall Inc, Upper Saddle River
32. Kaufman L, Rousseeuw PJ (2009) Finding groups in data: an introduction to cluster analysis, vol 344. Wiley, New York
33. Murtagh F (1983) A survey of recent advances in hierarchical clustering algorithms. Comput J 26(4):354–359
34. Murtagh F (1984) Complexities of hierarchic clustering algorithms: state of the art. Comput Stat Q 1(2):101–113
35. Voorhees EM (1986) Implementing agglomerative hierarchical clustering for use in information retrieval. Technical Report TR86–765, Cornell University, Ithaca
36. Willett P (1988) Recent trends in hierarchic document clustering: a critical review. Inf Process Manag 24(5):577–597
37. Slonim N, Tishby N (2000) Document clustering using word clusters via the information bottleneck method. In: Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 208–215
38. Aggarwal CC, Gates SC, Yu PS (2004) On using partial supervision for text categorization. IEEE Trans Knowl Data Eng 16(2):245–255
39. An English stop word list (2016) http://snowball.tartarus.org/algorithms/english/stop.txt. Accessed 25 July 2016
40. Landis JR, Koch GG (1977) The measurement of observer agreement for categorical data. Biometrics 33(1):159–174

41. Chowdhury N, Saha D (2005) Unsupervised text classification using Kohonen's self organizing network. In: International conference on intelligent text processing and computational linguistics. Springer, Berlin, pp 715–718
42. Greene D, Cunningham P (2006) Practical solutions to the problem of diagonal dominance in kernel document clustering. In: Proceedings of the 23rd international conference on machine learning. ACM, pp 377–384
43. Newsgroup Datasets. http://qwone.com/∼jason/20Newsgroups/. Accessed 25 July 2016
44. Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on computational learning theory. ACM, pp 144–152
45. Vapnik VN, Vapnik V (1998) Statistical learning theory, vol 1. Wiley, New York
46. Hsu CW, Chang CC, Lin CJ (2003) A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, pp 1–16
47. Milgram J, Cheriet M, Sabourin R (2006) "One Against One" or "One Against All": which one is better for handwriting recognition with SVMs?. In: Tenth international workshop on frontiers in handwriting recognition. Suvisoft
48. Chang YW, Hsieh CJ, Chang KW, Ringgaard M, Lin CJ (2010) Training and testing low-degree polynomial data mappings via linear SVM. J Mach Learn Res 11(Apr):1471–1490