Expert Systems with Applications 36 (2009) 9584–9591

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Using the self organizing map for clustering of text documents

Dino Isa[1], V.P. Kallimani *, Lam Hong Lee[2]

Faculty of Engineering and Computer Science, University of Nottingham, Malaysia Campus, 43500 Semenyih, Malaysia

ARTICLE INFO

Keywords:
Bayesian
Self organizing maps
Clusters similarity

ABSTRACT

An increasing number of computational and statistical approaches have been used for text classification, including nearest-neighbor classification, naïve Bayes classification, support vector machines, decision tree induction, rule induction, and artificial neural networks. Among these approaches, naïve Bayes classifiers have been widely used because of its simplicity. Due to the simplicity of the Bayes formula, the naïve Bayes classification algorithm requires a relatively small number of training data and shorter time in both the training and classification stages as compared to other classifiers. However, a major short coming of this technique is the fact that the classifier will pick the highest probability category as the one to which the document is annotated too. Doing this is tantamount to classifying using only one dimension of a multi-dimensional data set. The main aim of this work is to utilize the strengths of the self organizing map (SOM) to overcome the inadvertent dimensionality reduction resulting from using only the Bayes formula to classify. Combining the hybrid system with new ranking techniques further improves the performance of the proposed document classification approach. This work describes the implementation of an enhanced hybrid classification approach which affords a better classification accuracy through the utilization of two familiar algorithms, the naïve Bayes classification algorithm which is used to vectorize the document using a probability distribution and the self organizing map (SOM) clustering algorithm which is used as the multi-dimensional unsupervised classifier.

1. Introduction

Document classification can be defined as the task of learning methods for categorizing collections of electronic documents into their automatically annotated classes, based on its contents. For several decades now, document classification in the form of text classification systems have been widely implemented in numerous applications such as spam filtering (Cunningham, Nowlan, Delany, & Haahr, 2003; Delany, Cunningham, & Coyle, 2005; Delany, Cunningham, Tsymbal, & Coyle, 2004; O′Brien & Vogel, 2002; Provost, 1999; Sahami, Dumais, Heckerman, & Horvitz, 1998; Wei, 2003), e-mails categorization (Kamens, 2005; Xia, Liu, & Guthrie, 2005; Brucher, Knowlmayer, & Mittermayer, 2002), formation of knowledge repositories (Hartley, Isa, Kallimani, & Lee, 2006), and ontology mapping (Su, 2002). An increasing number of statistical approaches have been developed for document classification, including k-nearest-neighbor classification (Han, Karypis, & Kumar, 1999), naïve Bayes classification (McCallum & Nigam, 2003), support vector machines (Chakrabarti, Roy, & Soundalgekar, 2003; Joachims, 1998), maximum entropy (Nigam, Lafferty, & McCallum, 1999), decision tree induction, rule induction, and artificial neural networks.

Each one of the document classification schemes mentioned previously has unique properties. The decision tree induction algorithm and rule induction algorithm are simple to understand and interpret after a brief explanation. However, these algorithms do not work well when the number of distinguishing features is large (Quinlan, 1993). The k-nearest-neighbor algorithm is easy to implement and has show its effectiveness in a variety of problem domains (Han et al., 1999). However, a major drawback of the k-NN algorithm is that it is computationally intensive, especially when the size of the training set grows (Han et al., 1999). Support vector machines can be used as a discriminative document classifier, and these have been shown to be more accurate in classification tasks. The good generalization property of the SVM is due to the implementation of structural risk minimization which entails finding a hyper-plane which guarantees the lowest classification error (Vapnik, 1995). An ability to learn which is independent of the dimensionality of the feature space (Joachims, 1998) is also an advantage of the SVM. However, the usage of SVMs in many real world applications is relatively complex due to its convoluted training and categorizing algorithms as compared to the naïve Bayes classifier (Chakrabarti et al., 2003; Isa, Prasad, Lee, & Kallimani, 2007; Kim, Rim, Yook, & Lim, 2002).

Among these approaches, the naïve Bayes text classifier has been widely used because of simplicity in both the training and

* Corresponding author. Tel.: +60 3 89248141; fax: +60 3 89248017.
E-mail addresses: Dino.Isa@nottingham.edu.my (D. Isa), VP.Kallimani@nottingham.edu.my (V.P. Kallimani), kcx4lhl@nottingham.edu.my (L.H. Lee).

[1] Tel.: +60 3 89248116.

[2] Tel.: +60 3 89248141.

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.07.082

classifying stage although this generative method has been reported less accurate than discriminative methods such as SVM (Chakrabarti et al., 2003; Joachims, 1998). However, some researchers have proven that the naïve Bayes classification approach provides an intuitively simple text generation model and performs surprisingly well in many other domains, under specific "ideal" conditions (McCallum & Nigam, 2003). A naive Bayes classifier is a simple probabilistic classifier based on Bayes' Theorem with strong independence assumptions but this assumption severely limits its applicability (Flach, Gyftodimos, & Lachiche, 2002). In real life applications, the probability values associated with an event are seldom "independent". For example, even tossing a coin will not have the expected 50:50 chance of a result being either "heads" or "tails" due to factors which are associated with machining the coin, the different surface textures, different environments and different ways and methods used to toss the coin among other things. If we are lucky, these factors even out over time, if we are not, then the naïve Bayes formula will misclassify frequently. However, depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently and requires a relatively small amount of training data to estimate the parameters necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

Naïve Bayes classification is a probabilistic inference approach which has been implemented in mail repositories to remove spam e-mails (Cunningham et al., 2003; Delany et al.,2004, 2005; O′Brien & Vogel, 2002; Provost, 1999; Sahami et al., 1998; Wei, 2003). In this work, the traditional naïve Bayes classification approach is implemented to classify electronic documents into one or more categories, by calculating the probabilistic distribution of the text body of the document in a vector space of features. In the context of classification, the Bayes theorem emphasizes that the probability of a particular document being annotated to a particular category given that the document contains certain words in it, is equal to the probability of finding those certain words in that particular category, times the probability that any document is annotated to that category, divided by the probability of finding those words in any document, as illustrated in equation below:

$$\Pr(Category|Word) = \frac{\Pr(Word|Category) \cdot \Pr(Category)}{\Pr(Word)}$$

Each document contains words which are given probabilities based on its number of occurrence within that particular kind of documents. Naïve Bayes classification is predicated on the idea that electronic documents can be classified based on the probability that certain keywords will correctly identify a piece of text document to its annotated category. At the basic level, a naïve Bayes classifier examines a set of text documents that have been well organized and categorized into categories, and compares the content in all categories in order to build a list of words and their occurrence. This list of word occurrence is used to identify or classify new documents to their right categories, according to the probability of occurrence of certain words in the document (Fig. 1b).

The naive Bayes classifier is attractive as compared to other classification methods due to its simplicity. This is due to the fact that it can "make do" with a small amount of training data to estimate the parameters necessary for classification. The Bayesian classification approach arrives at the correct classification as long as the correct category gives the highest probability value as compared to the others. A category′s probability does not have to be estimated very well. In other words, the overall classifier is robust enough to ignore serious deficiencies in its underlying naive probability model (Haykin, 1999). However, the major drawback of the Bayesian classification approach is its relatively low classification performance compare to other discriminative algorithms due to its "single dimensional" nature (classifying by highest probability category). Therefore, much active research has been carried out to improve the naïve Bayes classifier, enhancing this approach through the implementation of techniques which add a method of ranking the potential candidates through a tournament structure in the classification task (Isa, Lee, & Kallimani, in press; McCallum & Nigam, 2003).

The self organizing map (SOM) is a clustering method which clusters data, based on a similarity measure related to the calculation of Euclidean distances. The idea of this principle is to find a winner-takes-all neuron to find the most closely matching case. The SOM was proposed by Kohonen, and is based on the idea that systems can be design to emulate the collective co-operation of the neurons in the human brain. Collectivism can be realized by feedback and thus can also be realized in the network, where many neighboring neurons react collectively upon being activated by events. If neurons are activated in the learning process, the neighboring neurons are also affected. The network structure is defined by synapses and has a similar total arrangement after a phase of self organization as the input data of the event space (Negnevitsky, 2002). Consequently, the SOM is an established paradigm in AI and cognitive modeling being the basis of unsupervised learning. This unsupervised machine learning method is widely used in data mining, visualization of complex data, image processing, speech recognition, process control, diagnostics in industry and medicine, and natural language processing (Michalski, Bratko, & Kubat, 1999).

As a summary, the simplicity of implementation of the naïve Bayes classifier is in stark contrasts with its poor performance in classification tasks. In this work, this poor performance is improved using the SOM as the multi-dimensional classifier and the Bayes formula as the feature extractor or vectorizer. Our previous work has introduced some specialized algorithms to improve the performance of naïve Bayes classifier when handling different types of knowledge domains and thus guarantees a lower error rate s compared to using only the Bayes theorem to classify (Isa et al., in press). A so called flat ranking classification algorithm and a series of tournament structures ranking algorithms have been designed and implemented. Besides this, additional techniques are introduced with the hope of obtaining a higher classification accuracy, such as the high relevance keywords extraction (HRKE) facility and the automatically computed document dependent (ACDD) weighting factors (not covered in this paper), in order overcome some of the weakness of the traditional naïve Bayes classification algorithms (Isa et al., in press). We have implemented here, a practical system which uses the Bayes formula and various ranking algorithms along with the SOM to automatically classify any electronic document for any database, with 100% accuracy.

2. The hybrid classification approach

We propose, design, implement and evaluate a hybrid classification approach by integrating the naïve Bayes classification (with tournament ranking methods) and SOM utilizing the simplicity of the naïve Bayes to vectorize raw text data based on probability values and the SOM to automatically cluster based on the previously vectorized data. The naïve Bayes classifier vectorizes the raw text documents into numerical values, so that the SOM can use the vectorized data in both the training and the categorizing stages. The structure of the proposed classification approach is illustrated in Fig. 1a.

In the training stage of the classifier, the training dataset which contains well organized training documents are used by the front-end naïve Bayes classifier. After the naïve Bayes classifier has been

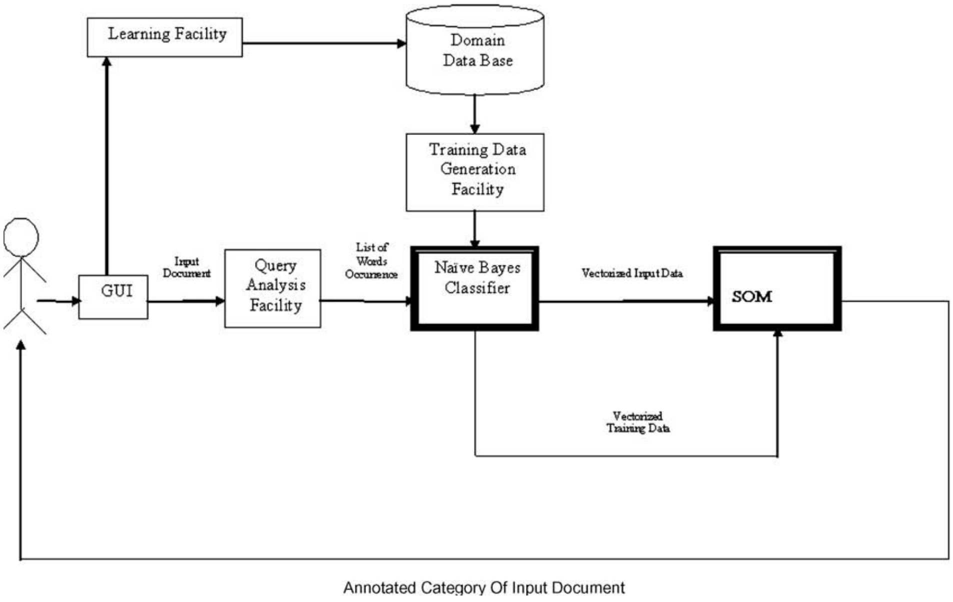


Fig. 1a. Proposed hybrid approach block diagram.

trained, each training document is vectorized by the trained naïve Bayes classifier through the calculation of the posterior probability value for each existing category based on the Bayes formula. For example, the probability value for a document X to be annotated to a category C is computed as $\Pr(C|X)$. Assuming that we have a category list as $Cat1, Cat2, Cat3, Cat4, Cat5, \ldots, CatN$, thus, each document has N associated probability values, where document X will have $\Pr(Cat1|X), \ldots, \Pr(Cat2|X), \ldots, \Pr(Cat3|X), \ldots, \Pr(Cat4|X), \ldots, \Pr(\text{Cat5}|\text{X}), \ldots, \Pr(CatN|X)$. All the probability values for a document are combined to construct a multi-dimensional numerical array. In this way, all the documents in the training dataset are vectorized into multi-dimensional numerical values to be used for the construction of separate vectorized training dataset for the SOM.

As for the classification stage, the categorizing processes are similar to the processes used during the text document vectorization in the training stage. The input to the trained naïve Bayes classifier during the classifying stage is the raw text document which is to be classified. The output from the naïve Bayes classifier, which is vectorized data in the format of multi-dimensional numerical probability values (an "address") is used as the input for the SOM for the final classification step (Fig. 1b). In this example the address sent to the SOM interface program is 5311, relating to the 50%, 30% and 10% probabilities.

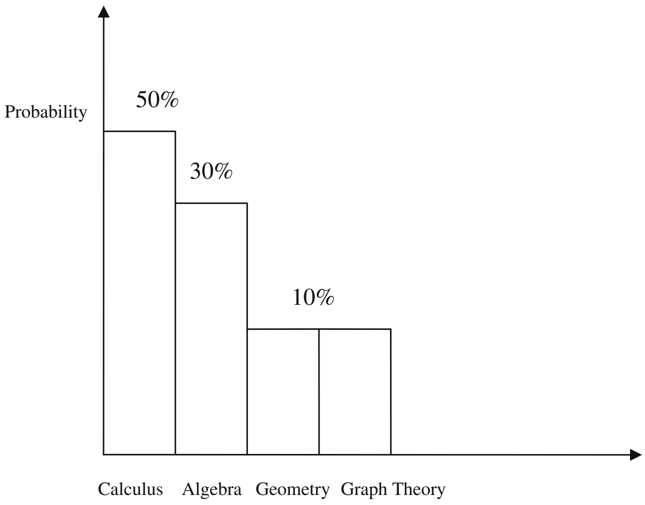


Fig. 1b. An example of the vectorization results using the na Bayes Classifier. The document has a 50% chance of being related to the subtopic "Calculus".

2.1. The naïve Bayes classification approach

Our proposed naïve Bayes classifier (Isa et al., in press) performs its classification tasks starting with the initial step of analyzing the text document by extracting words from the document. To perform this analysis, a simple word extraction algorithm is used to extract each individual word from the document to generate a list of words. This list is used when the naïve Bayes classifier calculates the probability of each word being annotated to a particular category. The list of words is constructed with the assumption that the input document contains words $w_1, w_2, w_3, \ldots, w_{n-1}, w_n$, and the length of the description is n.

This list of words is then used to generate a calculation table of words and their probabilities to be annotated to each category for the input text document, which consists of columns of words, and probability of the word belonging to each category. The column "Word" is filled with words which are extracted from the input document. For the columns of probabilities of a particular word for each category, the values to be filled is be calculated by the naïve Bayes classifier in the following stage. The tables below illustrate the use of this method for the input document in Table 1.

Before the naïve Bayes classifier performs the calculation of word probabilities for each category, it needs to be trained with a set of well categorized documents. Each individual word in all documents from the same category are extracted and listed in a list of words occurrence, by using a simple data structure algorithm, based on the computation of the frequency of word occurrence.

Based on the list of words occurrence, the trained classifier is able to calculate the posterior probability values of words which are extracted from the input document individually, by using the same formula for calculating the posterior probability which is derived from Bayes' theorem, given by the formula which is shown as below, since each single word from the input document

Table 1
Table of words occurrence and probabilities

Word	Probability category 1	Probability category 2	Probability category 3	......	Probability category $k-1$	Probability category k
w_1	$\Pr(C_1\|w_1)$	$\Pr(C_2\|w_1)$	$\Pr(C_3\|w_1)$	...	$\Pr(C_{k-1}\|w_1)$	$\Pr(C_k\|w_1)$
w_2	$\Pr(C_1\|w_2)$	$\Pr(C_2\|w_2)$	$\Pr(C_3\|w_2)$	...	$\Pr(C_{k-1}\|w_2)$	$\Pr(C_k\|w_2)$
w_3	$\Pr(C_1\|w_3)$	$\Pr(C_2\|w_3)$	$\Pr(C_3\|w_3)$	...	$\Pr(C_{k-1}\|w_3)$	$\Pr(C_k\|w_3)$
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
w_{n-1}	$\Pr(C_1\|w_{n-1})$	$\Pr(C_2\|w_{n-1})$	$\Pr(C_3\|w_{n-1})$	...	$\Pr(C_{k-1}\|w_{n-1})$	$\Pr(C\ k\|w\ n\text{-}1)$
w_n	$\Pr(C_1\|w_n)$	$\Pr(C_2\|w_n)$	$\Pr(C_3\|w_n)$	...	$\Pr(C_{k-1}\|w_n)$	$\Pr(C\ k\|wn)$
Total	$\sum\Pr(C_1\|W)$	$\sum\Pr(C_2\|W)$	$\sum\Pr(C_3\|W)$	...	$\sum\Pr(C_{k-1}\|W)$	$\sum\Pr(C\ k\|W)$
Probability of input document	$\frac{\sum\Pr(C_1\|W)}{\sum f}$	$\frac{\sum\Pr(C_2\|W)}{\sum f}$	$\frac{\sum\Pr(C_3\|W)}{\sum f}$		$\frac{\sum\Pr(C_{k-1}\|W)}{\sum f}$	$\frac{\sum\Pr(C_k\|W)}{\sum f}$

contributes to the probability values of a document to be annotated to every existing category,

$$\Pr(Category|Word) = \frac{\Pr(Word|Category)\cdot\Pr(Category)}{\Pr(Word)}$$

The derived equation above shows that by observing the value of *Word*, the posterior probability, $\Pr(Category|Word)$, which represent the probability of the state of nature being a particular category given that feature value can be calculated based on the Bayes formula. The prior probability, $\Pr(Category)$ can be compute from the equation below:

$$\Pr(Category) = \frac{Total\_of\_Words\_in\_Category}{Total\_of\_Words\_in\_Training\_Dataset} = \frac{Size\_of\_Category}{Size\_of\_Training\_Dataset}$$

Meanwhile, the evidence, which is also known as normalizing constant, $\Pr(Word)$ is calculated by using the equation:

$$\Pr(Word) = \frac{\sum occurrence\_of\_Word\_in\_every\_Category}{\sum occurrence\_of\_all\_words\_in\_every\_Category}$$

The total occurrence of a particular word in every category can be retrieved by searching the relevance data from the training data base, which are the lists of words occurrence for every category, generated from the analysis of all training files in the particular category during the stage of initial training. The sum of occurrence of all words in every category can also been calculated from the lists of words occurrence for every category.

To calculate the likelihood of a particular category with respect to a particular word, the lists of words occurrence from the training data base is referred to retrieve the occurrence of the particular word in the particular category, and the sum of all words in that particular category. These information will contribute to the value of $\Pr(Word|Category)$ with the equation:

$$\Pr(Word|Category) = \frac{occurrence\_of\_Word\_in\_Category}{\sum occurrence\_of\_all\_words\_in\_Category}$$

Based on the derived Bayes' formula for text classification, with the value of the prior probability $\Pr(Category)$, the likelihood $\Pr(Word|Category)$, and the evidence $\Pr(Word)$, the posterior probability, $\Pr(\ Category|Word)$ of each word in the input document annotated to each category can be measured. The posterior probability of each word annotated to each category is then used to fill the appointed cells in the table as illustrated in Table 1. After all the cells of "Probability" have been filled, the overall probability for an input document to be annotated to a particular category is calculated by dividing the sum of each of the "Probability" column with the total number of words in the input document, *n*, which is shown in the equation below:

$$\Pr(Category|Document) = \frac{\Pr(Category|w_1, w_2, w_3, \ldots, w_{n-1}, w_n)}{n},$$

where $w_1, w_2, w_3, \ldots, w_{n-1}, w_n$, are the words which are extracted from the input document.

Typically, the ordinary naïve Bayes classifier is able to determine the right category of an input document based on the Bayes Classification Rule, by referring to the associated probability values calculated by the trained classifier based on the Bayes formula. The right category is represented by the category which has the highest posterior probability value, *Pr(Category|Document)* (Kontakanen, Myllymaki, Silander, & Tirri, 1997). It is this simplicity that is both an advantage (simple algorithm) and a disadvantage (poor generalization). Since the ordinary naïve Bayes classification algorithm has been proven to be one of the poorest performing of classifiers, we propose SOM clustering for the purposes of increasing generalization and classification accuracy.

2.2. Clustering thru the use of the self organizing map

Knowledge discovery tasks can be broken down into two general steps: pre-processing and classification. In the pre-processing step, data is transformed into a format which can be processed by a classifier. The self organizing map (SOM) can be used to carry out the classification tasks effectively, especially for the analysis and visualization of a variety of economical, financial, scientific, and manufacturing data sets (Petrushin, 2005; Wang, 2001). The first step in designing the SOM is to decide on what prominent features are to be used in order to effectively cluster the data into groups. The criterion for selecting the main features plays an important role in ensuring that the SOM clusters properly and thus supports goal based decision making. Traditionally statistical cluster analysis is an important step in improving feature extraction and is done iteratively. An alternative to these statistical methods is the SOM (Vesanto, Alhonieme, Himberg, Kiviluoto, & Pervienen, 1999). Kohonen′s principle of topographic map formation, states that the spatial location of an output neuron in the topographic map corresponds to a particular feature of the input pattern. The SOM model, which is shown in Fig. 2, provides a map which places a fixed number input patterns from an input layer into the so called Kohonen layer (Kriegel, Brechesen, Kroger, Pfeifle, & Schbert, 2003; Wang, 2001). The system learns through self organization of random neurons whose weights are attached to the layers of neurons. These weights are altered at every epoch during the training session. The change depends upon the similarity or neighborhood between the input pattern and the map pattern (Michalski et al., 1999). The topographic feature maps reduce the dimensions of data to two dimensions simplifying viewing and interpretation.

In the SOM, certain trends in clustering can be observed by changing some of the training parameters. After the incremental

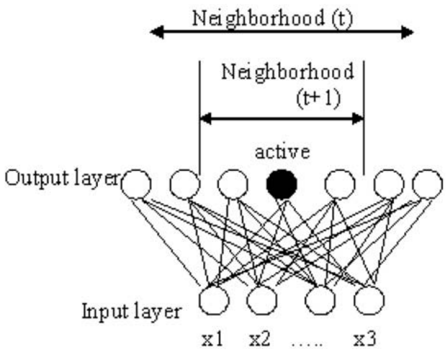


Fig. 2. The SOM model.

training of the map, the application saves the weight vectors of the map and these weights can be used as the starting weights. Once the training is over, the output mean and variance of each cluster is reported. Furthermore, the location of each cluster is also reported. Speed is a big concern in SOM clustering. By reducing dimensionality thru the use of a probability distribution of categories in feature space, instead of a raw word occurrence count we reduce computation time in both training and classification. The concerns associated with data pre-processing before the training starts and final drawing of the map once the training is over (Pal & Shiu, 2004) is also addressed by our hybrid system.

The SOM is trained iteratively. In each training step, a sample vector, x from the input data set is chosen randomly and the distance between x and all the weight vectors of the SOM, is calculated by using a Euclidean distance measure. The neuron with the weight vector which is closest to the input vector x is called the Best Matching Unit (BMU). The distance between x and weight vectors, is computed by using the equation below:

$$\|x - m_c\| = \min\{\|x_i - m_i\|\}$$

where ||.|| is the distance measure, typically Euclidean distance. After the BMU is found, the weight vectors of the SOM are updated so that the BMU is moved closer to the input vector in the input space Miyamoto, 2007. The topological neighbors of the BMU are treated similarly. The update rule for the weight vector of i is

$$x_i(t+1) = m_i(t) + \alpha(t)h_{ci}(t)[x(t) - m_i(t)]$$

where $x(t)$ is a vector which is randomly drawn from the input data set, and function $\alpha(t)$ is the learning rate and t denotes time (Deboeck & Kohonen, 1998). The function $h_{ci}(t)$ is the neighborhood kernel around the winner unit c. The dataset of manufacturing details are fed into the input layer of SOM. Learning parameter is selected between 0.0 and 0.9, and the SOM is trained. The training steps will be in the range of 100,000 epochs in order to obtain a trained map. These training datasets are coded with reference to their prominent features (Wang, 2001).

In this work, the naïve Bayes classifier and the SOM are trained independently; the Bayes classifier with raw documents and the SOM with address vectors arising from the execution of the Bayes classifier. Once a new document is to be identified and categorized, the naïve Bayes classifier is executed and outputs an "address" consisting of the probability distribution of the document being annotated to various pre-defined categories. At this point various enhancements are added via tournament ranking algorithms (and the HRKE facility) with the intention to improve on the plain vanilla naïve Bayes classifier (in our case we call this plain naïve Bayes option, "Flat Ranking"). This address is then fed into the SOM interface program which is then executed to find the best matching unit (BMU), which is the neuron corresponding to the document most closely related to the input document described by the address given by the naïve Bayes classifier. The top five similar documents are then retrieved and presented to the user. The original document is then classified as the same as other documents closest to the BMU. In this way, a multi-dimensional classification system is obtained, the SOM adding robustness and increasing generalization to the overall approach with the naïve Bayes classifier providing a way to vectorize the documents and reduce dimensionality thus resulting in faster training and classification time. In summary, this combination gives "enough" generalization (multi-dimensional as opposed to single dimensional classification of the naïve Bayes alone) but not so much so as to make the classifier overfit and detrimentally increase training and classification time.

3. Experimental results

The objective of this evaluation is to determine whether our proposed approach results in better classification accuracy and performance when compared to the naïve Bayes classifier (both with and without tournament ranking methods). As mentioned in the sections above, the hybrid approach utilizes the simplicity, low requirements of the naïve Bayes classifier as a raw text document vectorizer, and uses the SOM to cluster the vectorized documents into groups. The evaluations are made by comparing the classification accuracy of the ordinary naïve Bayes classifier, along with some specialized techniques which was presented in our previous work (Isa et al., in press), with the hybrid naïve Bayes vectorizer (plus specialized techniques) and SOM clustering approach which is proposed in this paper. In particular these specialized techniques include the naïve Bayes with flat ranking that is, the system computes the probability distribution by considering all categories in a single round of competition. The round robin version on the other hand, calculates the probability distribution by considering only two categories at a time iteratively until each category has competed with all the other categories and the winner is determined by the highest winning score. The single elimination method entails finding a winner that has not lost even once within the competition. The HRKE algorithm culls out words such as "a", "the", etc., which have a low effect on the classification task because it appears in every document. The algorithms mentioned above determine the right category for input documents by referring to the associated probability values calculated by the trained classifier based on the Bayes formula. The right category is represented by the category which has the highest posterior probability value, *Pr(Category|Document)*.

A dataset of vehicle characteristics is tested in the prototype system for the evaluation of classification performance in handling the case with four categories which have low degrees of similarity. Our selected dataset contains four categories of vehicles: Aircrafts, Boats, Cars, and Trains. All the four categories are easily differentiated and every category has a set of unique keywords. We have collected 110 documents for each category, with the total of 440 documents in the entire dataset. 50 documents from each category are extracted randomly to build the training dataset for the classifier. The other 60 documents for each category are used as the testing dataset to test the classifier.

Initially, we perform the experiment by implementing different pure naïve Bayes classification algorithms: the naïve Bayes with flat ranking algorithm, the naïve Bayes with round robin tournament ranking algorithm, the naïve Bayes with single elimination tournament ranking algorithm and the naïve Bayes with high relevance keywords extraction (HRKE) algorithm. These classification algorithms have been discussed in our previous work (Isa et al., in press) and have also been briefly described above. The algorithms mentioned above determine the right category for input documents by referring to the associated probability values calculated by the trained classifier based on the Bayes formula. According to the Bayes Classification Rule, the right category is represented by the category which has the highest posterior probability value, *Pr(Category|Document)*.

To evaluate the hybrid approach which is proposed in this paper, we implement the naïve Bayes classification algorithms mentioned above in the front-end to vectorize raw text data into the associated probability values calculated based on the Bayes formula. In the right category determination stage, we do not perform the same method as the ordinary naïve Bayes classifier by implementing the Bayes Classification Rule which selects the category with the highest posterior probability value, *Pr(Category|Document)*. We implement the SOM to cluster the vectorized documents into groups. The details processes of the SOM in performing the right category determination steps have been discussed in Section 2.2.

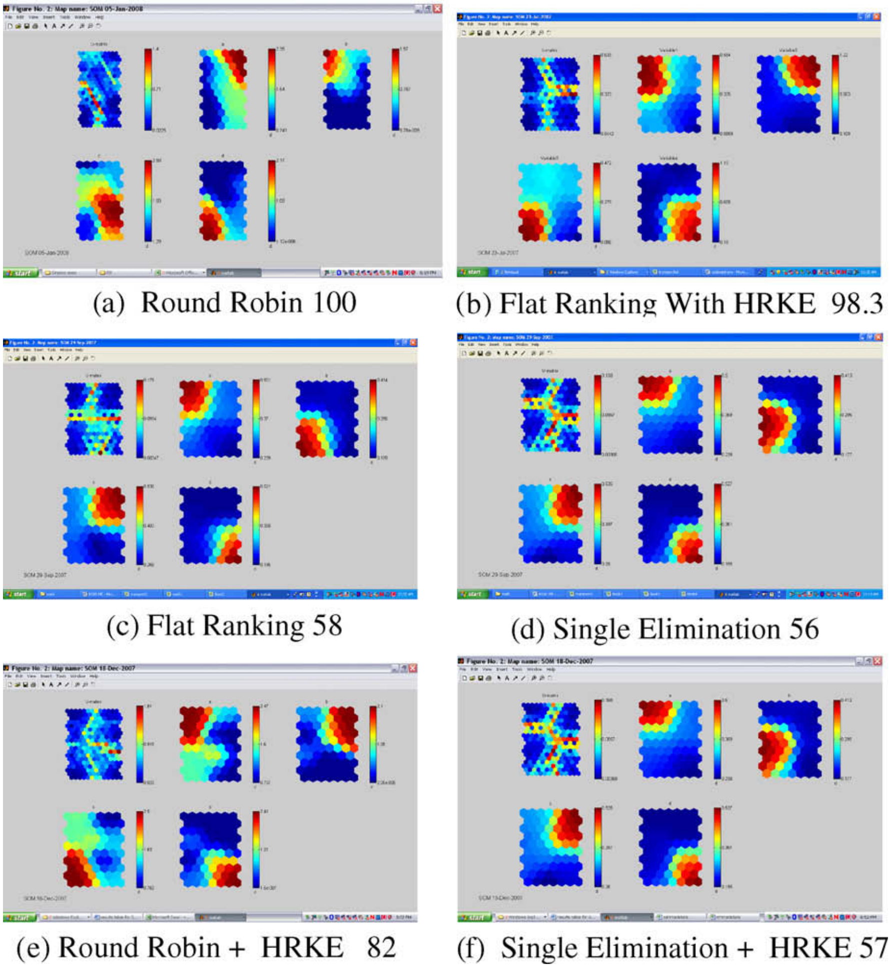
By submitting the entire training data matrix to the SOM, a U-matrix, which represents the discovery of SOM, is obtained as shown in Fig. 3. Each hexagon represents a node on a map or a processing unit of the output layer of the neural network. The shade of each node on this map indicates a pattern among the attributes which are used for this map. In this experiment, the map is clustered into five clusters. Two hundred and forty (60 documents per category) test vectors are chosen for testing purpose. Table 5 illustrates the details of data and cluster nodes and their distributions.

In this experiment, we present a large data matrix for the dataset with 200 training documents. The dataset has four dimensional of information, which is categorized by different methods as shown in Table 4. The listing of results shows the performances are carried in conjunction with SOM. The vectorized data by the front-end naïve Bayes vectorizer are considered as the input data to the SOM for further clustering purposes.

Different maps are created for different levels and the best matching unit (BMU) is calculated by using the Euclidian distance (Deboeck & Kohonen, 1998). The results show an improvement in recognition rate of case retrieval process. The visualization of the trained data is shown in Fig. 3. The min, max and average value of the attributes of the neuron map is shown in Table 3. The training parameters are chosen as shown in Table 2. The training parameters considered in this case are: initial radius of 3 and final as 1, in rough training, where as in fine tune it is chosen as 1 and 1.

Table 2
Training parameters

	Training	Fine tune
Map size 10 × 7		
Radius initial	3	1
Radius final	1	1
Final length	30,000	20


(a) Round Robin 100

(b) Flat Ranking With HRKE 98.3

(c) Flat Ranking 58

(d) Single Elimination 56

(e) Round Robin + HRKE 82

(f) Single Elimination + HRKE 57

Fig. 3. Map-visualization of clusters.

The comparison table in Table 4 illustrates that the recognition rate is poor, when the SOM is trained in combination with the single elimination tournament ranking. The recognition rate is only 56.66%. A similar situation was found when the hybrid approach is used with the flat ranking naïve Bayes classifier. The recognition rate is only 58.00%.

The hybrid approach with the naïve Bayes vectorizer when enhanced by high relevance keywords extraction facility shows good results, with 98.33% recognition rate, a slight improvement over the pure naïve Bayes classifier classification rate. The highest result of 100% recognition is obtained for the hybrid approach with the naïve Bayes vectorizer with Round Robin ranking method. Table 2 above provides the parameters chosen in training the SOM. The results are tested for 30,000 epochs of training cycle. Initial radius of 3 is considered in rough training and radius of 1 is considered in the final training. Table 3 provides the statistical values with minimum and maximum values for each one of the six vectorization techniques presented here.

With reference to Figs. 3a and b we see that clustering is better defined and has more detail as indicated by the lesser amount of yellows and greens as opposed to Figs. 3c and d where there is a spread of yellows and greens throughout the map. Table 5 shows the relative spread of colors between the round robin and (round robin + HRKE) techniques displaying the relationship between non primary colors and classification error.

We further draw the conclusion that the best performing algorithms, i.e., the round robin and (flat + HRKE) combination provides better delineation between clusters as high lighted by the minimal non primary green and yellow colors. This may be seen as due to an action akin to “filtering” out the noise in the input data to the SOM by means of the HRKE facility and through the iterative competition afforded by the round robin tournament method. Out of 240 test documents (represented by vectors) which have been presented to the organized map trained using different documents from the same data base, all 240 vectors were recognized accurately. This combination of the naïve Bayes vectorizer with round robin tournament ranking structure enhanced with HRKE combined with the SOM gives 100% classification accuracy for the data sets we have tested.

We see from Table 4, however, that the SOM classification accuracy for (round robin + HRKE) has dropped from 100% (without HRKE) to 82%. This is due to the fact that implementing HRKE artificially reduces the effectiveness of multi-dimensional classification (clustering) through the elimination of words such as “a” and “the” (stop words) which is indirectly considered by the SOM in its classification task. This is due to the fact that the vectorizer calculates the probability distribution using all words in the document including the stop words and the elimination of these words effects the input address to the SOM. For the naïve Bayes classifier on the other hand, the elimination of these stop words is less detrimental because only the highest probability is taken as the right category, i.e. a single dimension is used for classification instead of a multi-dimensional approach.

Table 4
Classification accuracy of the na Bayes algorithms and hybrid algorithms with and without HRKE

Tournament ranking enhancements	Pure na Bayes		With SOM	
	No HRKE (%)	With HRKE (%)	No HRKE (%)	With HRKE (%)
Flat ranking	81.25	96.25	58.00	98.33
Round robin tournament ranking	69.58	79.5	100	82
Single elimination tournament ranking	64.58	76.82	56.66	57

Table 5
Cluster formation according to distance measurement (round robin)

Color code	No. of neurons	% Distribution	Values
Dark red	6	8	2.15–2.36
Red	5	7	1.85–2.10
Yellow	5	7	1.65–1.80
Green	13	18	1.45–1.40
Light blue	17	24	1.14–1.40
Blue	26	36	0.741–1.10

4. Conclusion

A hybrid text document classification approach is proposed. Through the implementation of an enhanced naïve Bayes classifica-

Table 3
Statistics of the maps for different techniques

Distance	RR (100%)				RR + HRKE (82%)			
	*D*1	*D*2	*D*3	*D*4	*D*1	*D*2	*D*3	*D*4
Average	1.376223	0.368416	1.949174	0.647098	1.458709	0.693082	1.548414	0.621877
Maximum	2.34731	1.57367	2.57733	2.17383	2.46863	2.09739	2.49728	2.41367
Minimum	0.740616	5.78E-09	1.28699	1.12E-06	0.736841	2.05E-05	0.762738	1.5E-07
Variance	0.22436	0.186627	0.135178	0.503128	0.292686	0.56394	0.230363	0.56791
Std Dev	0.491255	0.432003	0.367665	0.709315	0.554666	0.75096	0.479961	0.753598
Distance	Flat (58%)				Flat and HRKE (98.33%)			
	*D*1	*D*2	*D*3	*D*4	*D*1	*D*2	*D*3	*D*4
Average	0.331384	0.243228	0.355904	0.268384	0.293388	0.434992	0.245219	0.457635
Maximum	0.504098	0.421988	0.538897	0.537959	0.583702	1.21695	0.471976	1.19018
Minimum	0.234698	0.17502	0.257436	0.192912	0.085879	0.109249	0.085964	0.179638
Variance	0.006616	0.006526	0.006062	0.010637	0.020681	0.131244	0.009334	0.108925
Std Dev	0.081341	0.080782	0.07786	0.103135	0.146401	0.361481	0.099921	0.330275
Distance	SE (57%)				SE + HRKE (57%)			
	*D*1	*D*2	*D*3	*D*4	*D*1	*D*2	*D*3	*D*4
Average	0.320067	0.249916	0.358831	0.266348	0.320067	0.25086	0.358789	0.266276
Maximum	0.499788	0.412554	0.534841	0.527001	0.499788	0.412554	0.534841	0.527001
Minimum	0.237894	0.176811	0.259863	0.195361	0.237894	0.176811	0.256949	0.195361
Variance	0.005397	0.006509	0.006549	0.009856	0.005397	0.006585	0.006557	0.009861
Std Dev	0.081941	0.08068	0.080925	0.09928	0.081941	0.081148	0.080977	0.099305

tion method at the front-end for raw text data vectorization, in conjunction with a SOM at the back-end to determine the right cluster for the input documents, better generalization, lower training and classification time, and a 100% classification accuracy can be obtained.

References

Brucher, H., Knowlmayer, G., & Mittermayer, M. A. (2002). *Document classification methods for organizing explicit knowledge*. University of Bern, Institute of Information System, Research Group Information Engineering, Engehaldenstrasse 8, CH-3012 Bern, Switzerland.

Chakrabarti, S., Roy, S., & Soundalgekar, M. V. (2003). Fast and accurate text classification via multiple linear discriminant projection. *The VLDB Journal The International Journal on Very Large Data Bases*, 170–185.

Cunningham, P., Nowlan, N., Delany, S. J., & Haahr, M. (2003). A case-based approach in spam filtering that can track concept drift. In *The ICCBR'03 workshop on long-lived cbr systems, Trondheim, Norway*.

Deboeck, G., & Kohonen, T. (1998). *Visual explorations in finance with self organizing maps*. Springer-Verlag.

Delany, S. J., Cunningham, P., & Coyle, L. (2005). An assessment of case-based reasoning for spam filtering. *Artificial Intelligence Review Journal, 24*(3–4), 359–378.

Delany, S. J., Cunningham, P., Tsymbal, A., & Coyle, L. (2004). A case-based technique for tracking concept drift in spam filtering. *Journal of Knowledge Based Systems, 18*(4–5), 187–195.

Flach, P. A., Gyftodimos, E., & Lachiche, N. (2002). *Probabilistic reasoning with terms*. Strasbourg: University of Bristol, Loius Pasteur University.

Han, E. H., Karypis, G., & Kumar, V. (1999). *Text categorization using weight adjusted k-nearest neighbour classification*. Department of Computer Science and Engineering, Army HPC Research Center, University of Minnesota.

Hartley, M., Isa, D., Kallimani, V. P., & Lee, L. H. (2006). *A domain knowledge preserving in process engineering using self-organizing concept, ICAIET 06*. Sabah, Malaysia: Kota Kinabalu.

Haykin, S. (1999). *Neural Networks. A Comprehensive Foundation* (2nd ed.). Prentice Hall.

Isa, D., Prasad, R., Lee, L. H., & Kallimani, V. P. (2007). Pipeline defect prediction using support vector machines. In *CSECS 2007, Cairo, Egypt*.

Isa, D., Lee, L. H., & Kallimani, V. P. (in press). A polychotomizer for case-based reasoning beyond the traditional bayesian classification approach. *Journal of Computer and Information Science*.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Machine learning: ECML-98, 10th European conference on machine learning* (pp. 137–142).

Kamens, B. (2005). *Bayesian filtering: Beyond binary classification*. Fog Creek Software, Inc.

Kim, S. B., Rim, H. C., Yook, D. S., & Lim, H. S. (2002). Effective methods for improving naïve bayes text classifiers. In *Proceeding of the 7th Pacific rim international conference on artificial intelligence* (Vol. 2417).

Kontkanen, P., Myllymaki, P., Silander, T., & Tirri, H. (1997). A Bayesian approach for retrieving relevant cases. In *Proceedings of the EXPERSYS-97 conference, Sunderland, UK* (pp. 67–72).

Kriegel, H. P., Brechesen, S., Kroger, P., Pfeifle, M., & Schbert, M. (2003). Using sets of feature vectors for similarity search on voxelized CAD objects. In *Proceedings of the ACM SIGMOD 2003 international conference on management of data, San Diago, 2003*.

McCallum, A., & Nigam, K. (2003). A comparison of event models for naïve Bayes text classification. *Journal of Machine Learning Research, 3*, 1265–1287.

Michalski, R. S., Bratko, I., & Kubat, M. (1999). *Machine learning and data mining methods and applications*. Wiley.

Miyamoto, S. (2007). *Data clustering algorithms for Information Systems*. Berlin: Springer-Verlag.

Negnevitsky, M. (2002). *Artificial intelligence. A guide to intelligent systems*. Addison Wesley.

Nigam, K., Lafferty, J., McCallum, A. (1999). Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering* (pp. 61–67).

O'Brien, C., & Vogel, C. (2002). Spam filters: Bayes vs Chisquared; letters vs words. In *Proceedings of the 1st international symposium on Information and communication technologies*.

Pal, S. K., & Shiu, C. K. (2004). *Foundation of soft case-based reasoning*. Wiley.

Petrushin, V. A. (2005). Mining rare and frequent events in multi-camera surveillance video using self organizing maps. In *Proceeding of the 11th ACM SIGKDD international conference on knowledge discovery in data mining*.

Provost, J. (1999). *Naïve-Bayes vs Rule-Learning in Classification of E-mail*. Department of Computer Science, The University of Austin.

Quinlan, J. R. (1993). *C4.5: program for machine learning*. San Mateo, CA: Morgan Kaufmann.

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. In *AAAI-98 workshop on learning for text categorization*.

Su, X. (2002). *A text categorization perspective for ontology mapping*. Norway: Department of Computer and Information Science, Norweigian University of Science and Technology.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. NewYork: Springer.

Vesanto, J., Alhonieme, E., Himberg, J., Kiviluoto, K., & Pervienen, J. (1999). Self organising maps for data mining in Matlab. In *The SOM toolbox, simulation news Europe* (Vol. 25, p. 54).

Wang, S. H. (2001). Cluster Analysis using a validated Self organizing method: Cases of problem identification. *International Journal of Intelligent systems in Accounting, Finance and Management*, 127.

Wei, K. (2003). *A naïve Bayes spam filter*. Faculty of Computer Science, University of Berkely.

Xia, Y., Liu, W., & Guthrie L. (2005). Email categorization with tournament methods. In *Proceeding international conference on application of natural language*.