Two Dimensional Feature Extraction and Blog Classification using Artificial Neural Network

Aruna Devi K
Research Scholar, Mother Teresa Women’s University, Kodaikanal 624102, India.
Assistant Professor in Computer Science (PG), Kristu Jayanti College, Bengaluru 560077, India.

Kathirvalavakumar T
Research Center in Computer Science, V.H.N.Senthikumara Nadar College, Virudhunagar 626001, India.

Abstract
A two dimensional feature reduction for automatic blog classification using artificial neural network is proposed. A blog can be manually categorized using the tags provided in the page itself. In spite of the tag, the blog contents can diverge to some other topic as it can be written by novice content writers also. In the proposed method the contents are represented as a collection of features. Significant features are identified using term weighting technique and information gain in the first phase. The leader algorithm selects only representatives of the clusters in the next phase, thus minimizes the number of patterns from the large amount of blogs. The selected patterns are fed to ANN classifier for training that scale down the training time of the classifier and also lead to better performance. The proposed method is implemented and validated on a live dataset. The results outperform the existing methods in terms of accuracy and training time.

Keywords: Blog Classification; Term Weighting; Information Gain; Leader Algorithm; Artificial Neural Networks; Dimensionality Reduction

INTRODUCTION
The Web infrastructure is flooded with augmented structured and unstructured data in various domains. With the rapid growth of contents in the web it is hard to leverage the maximum useful information appropriate to the user’s requirement on a specific domain. A directory is a powerful tool that automatically categorizes the web pages and blogs under different classes of interest. A Blog is kind of website but differ in the way of content presentation. Unlike the web, the contents are written by a single author or a discussion on a specific topic. Bloggers write whatever is on their mind, sometimes inventing new vocabulary and grammar. Some blogs deliberately deviate from rules of language and decorum to attract larger audience followers for their blogs [1]. Blogs are difficult to classify automatically as the author can write on any topic and present in his own style. Many of the blog directories are human edited and list the links after manual review. The exploding growth of heterogeneous contents on the web leads the automatic blog classification awfully difficult and huge time consuming. Several challenges are apparent in classifying the Blogs for a Blog directory, which include informal presentation of contents, type and structure of data, volume of the information and availability of millions of blogs etc.

The blogs can be classified according to their contents like political blogs, science blogs, fiction blogs etc. It can also be categorized by identifying the emotions of the author behind the article. The blog contents can be classified using document classification method and it can be recommended to users after user preference analysis. Since the readers may be interested in specific kind of articles, they have to filter the articles manually. For the effective information retrieval, a two-layer SVM classification mechanism to classify blog articles is proposed by Guo-Heng Luo, Jia-Chiam Liu and Shyan-Ming Yuan [2]. Mita K. Dalal and Mukesh A. Zaveri[3] have classified unstructured blog posts using a semi-supervised machine learning approach. In their research they concluded that blogs can be classified with good accuracy using the multi-step classification strategy. The TF-IDF combined with Multi-word heuristics can be an effective statistical feature set extractor. They tested the classification of unstructured blog text using Naïve Bayes algorithm and basic artificial neural networks. The challenges in their research were larger and more varied datasets. Elisabeth Lexet al. [4] have classified the blogs into common newspaper categories using German News Corpus. The supervised text classification algorithms are generally applied to classify blogs into topics or other categories. The challenge in supervised text classifiers is it needs a sufficient large amount of labeled data to learn a good model. For blogs, data labeled with terms that capture current and actual topics are not available and data labeled in the past is not applicable owing to topic drifts. Their approach is to exploit the labeled data from the news corpus and use this knowledge to perform cross-domain classification on the unlabeled blogs. They evaluated their approach with a number of text classification algorithms with different parameter settings by means of accuracy and complexity. Their proposed Class-Feature-Centroid classifier (CFC) achieved a good accuracy.

In text categorization, the term weighting is used to discriminate the terms by assigning proper weights to the terms that result in better performance. The distinction between supervised and unsupervised methods is that the former use categorical information and the later are computed
Feature selection is a combinatorial optimization problem that selects the most important features from an original feature set [9]. It plays a pivotal role in categorizing the document effectively and efficiently. The traditional text categorization is based on term matching where a document is represented as the high dimensional vector space model. The rows in VSM represent the text documents and columns represent the words in the documents. The term matching categorization technique does not consider the semantic relationship between terms thereby result in a poor categorization. A two-stage feature selection method is proposed by Jiana Meng et al.[5] to categorize spam blogs, that reduce the dimension of terms and then build a new semantic space between terms, based on the latent semantic indexing method. Bing Xue et al. [10] have listed and stated in their study that the evolutionary computation (EC) paradigms can be used as search techniques in feature selection. It can be categorized as evolutionary algorithms, swarm intelligence and others. The evolutionary algorithms include genetic algorithm and genetic programming. Examples of swarm intelligence are particle swarm optimization and ant colony optimization. The EC paradigms include Differential evolution, memetic algorithms, evolutionary strategy and artificial bee colony etc.

Feature extraction [11] synthesizes a set of new set of features from the original features and it is smaller than the original feature set. Its outcome may not be a subset of the original feature. It can be result of combinations or transformations of the original feature space. Harun [12] in his methodology ranked each term in the document with respect to their importance in the document using Information Gain (IG). Then the Genetic Algorithm (GA) and Principal Component Analysis (PCA) are applied separately to the ranked terms and then the dimension is reduced based on their ranking. The k-nearest neighbor and C4.5 decision tree classification algorithm was used at the final stage to validate the dimension reduction methods. A novel clustering based feature subset selection framework was proposed by Sivakumar [13]. Initially clusters are formed using minimum variance method that is used to reduce the number of features. The cluster pair which has maximum number of votes is chosen and a member with the highest priority is chosen from each cluster using Information Gain (IG) and the attributes with less priority voting were removed resulted in dimensionality reduction.

In this paper classifying the Blogs of various categories with two phase feature extraction and pattern selection method using artificial neural network is proposed. The blogs are represented as feature sets. To determine the significance of each feature, they are weighted using term weighting method. The weighted terms are then ranked using Information Gain, which is a supervised feature selection method. Patterns with the salient features are given to the second phase for reducing the training patterns using leader algorithm. The article is organized as given. The Blog dataset representation and preprocessing techniques are discussed in Section 2. The term weighting schemes are listed in Section 3. The Information Gain for the feature extraction is explained in Section 4. The dataset reduction method using Leader algorithm is illustrated in Section 5. The Classifier with its training method is explained in Section 6. The Blog Classification framework and the proposed algorithm are defined in Section 7. The experiments and results are discussed in detail and compared with existing schemes in Section 8.

**BLOG REPRESENTATIONS**

The first step in the Blog classification is to transform a Blog page, which consists of Characters, Images, Hyperlinks and HTML Tags into a feature vectors. The preprocessing steps are carried out as shown in Figure 1.
A document can be represented as a document vector. The tokenization is a process of word segmentation. The words are chopped at white spaces in the sentences of the document and removes punctuations. Then the special characters and numbers are removed [17] as they don’t play significant role in the blog content classification. All the words are uniformly converted into lower case since it is easy to compare them with the word in dictionary for stopping and stemming. Stop words are frequently occurred, most common words like ‘the’, ‘in’, ‘and’, ‘whose’, that carries less priority in categorization. There is no specific standard list of English stopwords, the common range of stop words are between 100 and 1,000 terms [18]. Most of the text analysis software packages make use of a default list for the stop word removal process. In addition to the default stop words list, the domain specific stop word dictionary can be constructed for yielding better performance. Stemming is a process that relates semantically similar indexing and search terms. Stemming is used to reduce the size of index terms [19]. Stemming can be done in different approaches that include affix removal, successor variety, table lookup and n-gram. The process is a vocabulary reduction technique intended to map a word to its most basic form. For example the words “sleep”, “slept”, “sleeping” all share a common stem “sleep”. Consequently the large numbers of redundant and least important words are removed.

Thus each Blog is represented by vector of ‘m’ unique words called as ‘bag of words’, \(d = \{t_1, t_2, ... t_m\}\). A term document matrix which is the vector space model of the dataset is constructed with each blogs as the rows, the index words or terms as column, and the cell values as the term frequency of the term in the respective document.

### TERM WEIGHTING

The term value for a certain document specifies how much it influences the semantics of the document. The significance of a term cannot be measured by only with the frequency of occurrences. There are various schemes available to measure the weight of a term. They are listed in the Table 1 [20].

<table>
<thead>
<tr>
<th>Term Weighting Method</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term Frequency (TF)</td>
<td>(TF_{t,d} = \frac{fr_{t,d}}{\sqrt{\sum_{t=1}^{n} fr_{t,d}^2}})</td>
</tr>
<tr>
<td>Document Frequency (DF)</td>
<td>(DF_t = \sum_{d=1}^{N} { \begin{cases} 1 &amp; t \in d \ 0 &amp; t \notin d \end{cases} })</td>
</tr>
<tr>
<td>Term Frequency – Inverse Document Frequency (TF-IDF)</td>
<td>(TF - IDF_{t,d} = TF_{t,d} \cdot IDF_t) where (IDF_t = \log\left(\frac{N}{DF_t}\right) + 1)</td>
</tr>
<tr>
<td>Entropy</td>
<td>(w_{t,d} = L_{t,d} \times G_t) where (G_t = \frac{1 + \sum_{i=1}^{N} \frac{fr_{t,d}}{F_t} \log\left(\frac{fr_{t,d}}{F_t} + 1\right)}{\log N}) and (L_{t,d} = { \begin{cases} 1 + \log fr_{t,d} &amp; fr_{t,d} &gt; 0 \ 0 &amp; fr_{t,d} = 0 \end{cases} })</td>
</tr>
<tr>
<td>Term Weighting[20]</td>
<td>(w_{t,d} = \frac{\log(TF_{t,d} + 1)}{\log\left(\frac{DF_t}{N} + 1\right)})</td>
</tr>
</tbody>
</table>

The symbols in Table 1 denotes,

- \(fr_{t,d}\) is the frequency of term ‘t’ in document ‘d’
- \(F_t\) is the frequency of term ‘t’ at the document collection level
- \(n\) is the number of unique terms in document ‘d’
- \(N\) is the number of documents in the collection
- \(length_d\) is the length of the vector that represents unique terms in document ‘d’
i. Term Frequency (TF): TF is the direct method that states how many times the term ‘t’ occurred in the document ‘d’. In TF scheme, if a term frequency is high it shows that term is more relevant than the term with low frequency.

ii. Document Frequency (DF): Document frequency enumerates how many documents in the document collection has the term ‘t’.

iii. Term Frequency – Inverse Document Frequency (TF-IDF): The Inverse document frequency implies the most occurred terms in the collection are least significant terms. TF-IDF is a prominent global term weighting scheme where the weight is computed with respect to its incidence in the entire collection. In this inevitable ranking measure the term which is less frequent in the collection however most occurred in a specific document is assigned higher weight.

iv. Entropy term weighting: The term weight is computed from two aspects. They are local term weighting and global term weighting based on purity measure.

v. Term weighting [20]: This term weighting method is enhanced from traditional TF-IDF. It majorly focus on three aspects, they are term frequency, collection frequency and document length.

**INFORMATION GAIN**

Information Gain Information Gain [12] is a supervised technique based on Information theory used for ranking the attributes. The Information Gain Ratio is initially introduced to measure the goodness for attributes used in Decision Tree learning algorithm. It measures the no. of bits of information acquired for estimation of a class (C) by knowing the availability of a term (t) in a document. The information gain of term ‘t’ is defined as,

\[
IG(t) = \sum_{i=1}^{C} \sum_{j=1}^{t} \sum_{p(C_j | t)} \log \frac{P(C_j | t)}{P(C_j)} + \sum_{i=1}^{C} \sum_{j=1}^{t} \sum_{p(C_j | \neg t)} \log \frac{P(C_j | \neg t)}{P(C_j)}
\]

(1)

Where \(C_i\) represents the ith Class and \(P(C_i)\) is the Probability of the ith Class. \(P(t)\) and \(P(\neg t)\) are the probabilities that the term ‘t’ is present or not present in the documents respectively. \(P(C_i | t)\) and \(P(C_i | \neg t)\) are the conditional probabilities of the ‘ith’ class given that the term ‘t’ does not appear.

**LEADER ALGORITHM**

Leader algorithm [21] is an unsupervised clustering technique that groups the data sets according to the similarity. The similarity is computed by standard distance measure. Clustering is formulated as an optimization problem to create a subset of clusters. If \(D = \{d_1, d_2, d_3...d_n\}\) is the collection of documents; ‘n’ is the maximum no. of documents in the dataset; ‘k’ is the maximum no. of clusters formed by Leader Cluster algorithm and \(C = \{c_1, c_2, ...c_k\}\) are the centroids of the clusters. Leader algorithm is an incremental clustering algorithm used to cluster large data sets. This algorithm is order dependent and may form different clusters according to the input order, the data set is provided to the algorithm. The algorithm consists of the following steps.

**Algorithm:** Leader Cluster

**Input:** A Collection of documents ‘D’

**Output:** New Subset of clusters ‘K’

1. Read the first data item, \(d_1\) and allocate it to the first cluster \(C_1\). This data is the leader of the cluster \(C_1\).
2. Increment no. of clusters to 1
3. Read the next data item \(d_2\) and calculate its distance from the leader \(d_1\).
   The distance is computed using Euclidean distance measure as in (2)

\[
dist = \sqrt{(d_1 - d_2)^2}
\]

(2)

4. If the distance between \(d_1\) and leader \(d_1 < \) threshold \(t\), then data point \(d_2\) is assigned to cluster \(C_1\).
5. If the distance between \(d_1\) and leader \(d_1 > \) threshold \(t\), then form a new cluster \(C_2\) and assign \(d_2\) to this new cluster and \(d_2\) will be the leader of the cluster \(C_2\).
6. Repeat the steps 6 – 10 for all the remaining data items.
7. Calculate the distance between the data point and the leader of the all the clusters
8. If the distance between the data items and the any of the leader \(<\) threshold \(t\), the data point is assigned to that cluster.
9. If the computed distance for all the clusters is greater than the threshold, a new cluster is created and the data point is assigned to that cluster. Now this data is the leader of the new cluster.
10. Increment no. of clusters to 1

**CLASSIFIER**

Artificial Neural Network (ANN) is prominently used as a classifier on implementing two major steps. First, Construction of architecture of ANN secondly, training the ANN. The network can be constructed by connecting neurons or the functional elements in different layers. The most frequently used topology is feedforward neural network which can be constructed by having minimum of three layers as shown in Figure 2. The multiple layers include input layer, output layer and hidden layer. The neurons in the input layer are determined by the number of input features and similarly number of output neurons is determined by the no. of outputs. The hidden neuron can be adjusted according to the application and the training. All the neurons are connected in a standard manner using weighted links. The training of ANN
is a classification problem where the weights of the links are adjusted in order to receive better accurate output. Backpropagation training algorithm is used for training the network. With the algorithm the network is continuously observed and learnt by gradually reaching to the specified Mean Square Error (MSE) value. The MSE is computed as

\[
MSE = \frac{1}{p} \sum_{i=1}^{p} (d_i - o_i)^2
\]

where ‘p’ denote the total number of patterns, ‘j’ denote the jth neuron of the output layer, ‘d’ is the desired value and ‘o’ is the computed output of the network for the given pattern.

The basic idea of the standard backpropagation algorithm is the recurrent use of the chain rule to calculate the impact of each weight in the network with respect to an arbitrary error function.

![Feedforward Neural Network](image)

**Figure 2: Feedforward Neural Network**

**PROPOSED METHOD**

A blog has contents more about personal opinions, activities, and experience. The massive contents of the blog lead to dimensionality problem and will have the adverse effects on the performance of the classifier. The blogs can be classified according to the contents; emotions of the blogger; professionalism etc. In this paper the blogs are classified according to their category as shown in the Figure 3. The blogs which are retrieved from WWW has contents in the form of text, video, images etc., are arranged in chronological order. For the classification of blogs according to category, first, contents alone are extracted by removing all html tags and links. Then in the preprocessing phase, the text is tokenized and then stop words are removed. Then Stemming is also applied to map the modulated or derived words to their word stem which reduces the term index to a considerable amount. The result of the preprocessing phase is a collection of unique term index and they are formulated as the document-term frequency matrix as in Table 2.

**Table 2: Sample DTM**

<table>
<thead>
<tr>
<th>Blog Term</th>
<th>t₁</th>
<th>t₂</th>
<th>t₃</th>
<th>...</th>
<th>tₖ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blog₁</td>
<td>21</td>
<td>7</td>
<td>10</td>
<td>...</td>
<td>8</td>
</tr>
<tr>
<td>Blog₂</td>
<td>17</td>
<td>11</td>
<td>5</td>
<td>...</td>
<td>9</td>
</tr>
<tr>
<td>Blog₃</td>
<td>8</td>
<td>6</td>
<td>3</td>
<td>...</td>
<td>4</td>
</tr>
<tr>
<td>Blogₙ</td>
<td>2</td>
<td>14</td>
<td>5</td>
<td>...</td>
<td>15</td>
</tr>
</tbody>
</table>

The rows and columns of the table represent the documents and the terms respectively. In the table, m denotes the total number of blog documents and k is the total number of terms present in the document collection. The values in the cells of the table depict the number of times the term tᵢ available in Blogᵢ.

The Vector Space Model (VSM) of the given data has too many terms out of those very few terms are significantly relevant. Thus, the terms are weighted using the term weighting formula and weights are sorted in an ascending order. There can be specific number of features given to the next level of feature ranking, which is done using Information gain, a supervised technique. Now top ranked p numbers of features are selected for the next phase. The selected features undergo normalization so that the data sets are scaled within the range of [1, −1]. Later the feature reduction, the pattern reduction is done using Leader cluster algorithm. The algorithm significantly reduces the patterns and forms ‘k’ clusters depend on the threshold distance. The leader patterns are given as inputs to train ANN classifier. The feedforward artificial neural network categorizes the given dataset.

**BLOG CLASSIFICATION ALGORITHM**

Step 0: Start

Step 1: Consider p no. of features; k no. of words in the collection; m no. of blog documents in the collection

Step 2: Retrieve Blogs; Extract only blog text contents

Step 3: Perform the text preprocessing techniques

Step 4: Construct Vector Space Model for the preprocessed text

![Blog Classification](image)

**Figure 3: Blog Classification**
Step 5: Input the VSM of the blog documents in matrix A (Dimension m x k)

Step 6: Transform the matrix A to Term weight matrix B using the improved term weighting function \[20\]

Step 7: Sort the matrix B in descending order and rank the columns with respect to the sum of term weight

Step 8: From the sorted matrix B, extract the top ranked 5000 features to matrix C.

Step 9: Perform supervised ranking using information gain

Step 10: Extract the top ranked p no. of features

Step 11: Normalize the feature reduced matrix with dimension m x p

Step 12: The normalized output E is given as input to Leader algorithm

Step 13: Perform Leader clustering that reduces the rows

Step 14: Feed Leaders of the clusters to ANN

Step 15: Initialize the network parameters for FNN

Step 16: Train the ANN, until the MSE reaches 0.001

Step 17: Feed the testing documents as input to the network

Step 18: Compute the classifier performance by measuring Accuracy, Precision, Recall and F1

EXPERIMENTS AND RESULTS

Experiments have been conducted on the dataset contains blog posts from MarginalRevolution.com \[16\]. It has the collection of posts from Jan. 1, 2010 to 9/17/2016, with the attributes as Author Name, Post Title, Post Date, Post content (words), Number of Words in post, Number of Comments in post, and categories. This dataset has about 13,000 blogs, subdivided into 15 classes. 1612 documents (m=1612) of 5 classes are used for the analysis of the proposed method. Table 3 shows the details of various blog posts taken from the dataset for the experiment. The number of features are limited to 5000 (k=5000). The data is cleaned by applying text preprocessing techniques using R language. The Hardware and Software specifications for the execution of the experiment are shown in Table 4.

<table>
<thead>
<tr>
<th>Category</th>
<th>No. of Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Economics</td>
<td>397</td>
</tr>
<tr>
<td>Web Technology</td>
<td>468</td>
</tr>
<tr>
<td>Books</td>
<td>349</td>
</tr>
<tr>
<td>Food &amp; Drinks</td>
<td>225</td>
</tr>
<tr>
<td>Political Science</td>
<td>173</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1612</strong></td>
</tr>
</tbody>
</table>

Table 4: Hardware and Software Configuration

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor: Intel Core Duo 2.1GHz</td>
<td>Platform: MS Windows 7</td>
</tr>
<tr>
<td>Memory: 3GB RAM; 32 bit OS</td>
<td>Software: Matlab R2014a, R Studio</td>
</tr>
</tbody>
</table>

At the end of the preprocessing phase there are ‘m’ documents and ‘n’ terms in the document term matrix. The terms are weighted using term weighting techniques and sorted in the descending order of the term values. The top ranked 5000 columns are fetched for the Information gain ranking. The weighted matrix is ranked with supervised ranking and ‘p’ no. of significant features are selected. The outcome of the Feature selection and extraction is a matrix of ‘m’ documents and ‘p’ columns. At this juncture, it is necessary to assign the optimal value for ‘p’ in the first phase and ‘distance’ for leader cluster in the next phase of the proposed framework. The parameter ‘p’ is given with different values (p=25, 30, 35 and 40) in accordance with leader distance (dist= 1.0, 1.5, 2.0, 2.5 and 3.0) and the accuracy results are tabulated in Table 5. It is found that better results are yield for p=25 and dist=2.0 results in compact and efficient dimensionality reduction. The p is chosen as 25 and the resultant matrix with reduced columns undergoes an unsupervised clustering, Leader clustering for the row reduction. The distance for leader clustering is chosen as 2.0. The leaders of the clusters are extracted for the classification. The feedforward neural network is configured with 25 neurons in input layer, 8 neurons in hidden layer and 5 neurons in the output layer. The network is trained using backpropagation algorithm. The parameters of the feedforward ANN are given in Table 6. On trial basis, the learning rate is fixed for minimum cost. Initially the weights are chosen randomly between 0 and 1. The learning curve of the configured network is shown in Figure 4.

<table>
<thead>
<tr>
<th>p</th>
<th>Leader distance</th>
<th>No. of Leaders</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>1.0</td>
<td>78</td>
<td>55.1</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>34</td>
<td>73.5</td>
</tr>
<tr>
<td>30</td>
<td>2.0</td>
<td>18</td>
<td><strong>92.9</strong></td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>99</td>
<td>51.9</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>45</td>
<td>68.9</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>23</td>
<td>89.5</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>35</td>
<td>1.0</td>
<td>122</td>
<td>53.3</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>50</td>
<td>62.0</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>28</td>
<td>85.0</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>5</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3: Marginal revolution Blogbost

Table 5: No. of features, leader distance Vs Accuracy
### Table 6: ANN parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Rate</td>
<td>0.1</td>
</tr>
<tr>
<td>Mean Squared Error</td>
<td>0.001</td>
</tr>
<tr>
<td>Input Neurons</td>
<td>p [25]</td>
</tr>
<tr>
<td>Hidden Neurons</td>
<td>8</td>
</tr>
<tr>
<td>Output Neurons</td>
<td>5</td>
</tr>
</tbody>
</table>

#### Figure 4: Epoch Vs MSE Learning Curve

Once the classifier is trained, it is tested with a set of patterns and the results are evaluated using standard information retrieval measurement tools that are precision (P), recall (R), accuracy (Acc) and F1.

\[
P = \frac{a}{a+b} \quad (4)
\]

\[
R = \frac{a}{a+c} \quad (5)
\]

\[
F1 = \frac{2PR}{P+R} \quad (6)
\]

\[
Acc = \left( \frac{Total \ Correct}{Total \ No. \ of \ Documents} \right) \times 100\% \quad (7)
\]

The Figure 5 illustrate the confusion matrix of blog classification using proposed method with the p value as 25 and cluster distance as 2.0. It shows that out of 14 patterns 13 patterns are predicted correctly. In the five classes of patterns, only Economics class dataset is predicted wrongly. A pattern in the Economics dataset is predicted as Food and drinks class, due to similarity in the keywords. The confusion matrix for blog classification using proposed method and TF-IDF is shown for web technology class in Figures 6 and 7.
The Figure 6 shows the confusion matrix of web technology class using the proposed method, where 25 patterns are predicted correctly and 5 patterns are predicted wrongly out of 30 patterns. The 4 patterns were false positive and 1 was false negative. Figure 7 shows the misclassification results of TFIDF scheme for web technology class. 7 patterns are predicted wrongly and 14 patterns are predicted correctly.

<table>
<thead>
<tr>
<th>Class</th>
<th>TF</th>
<th>DF</th>
<th>TF-IDF</th>
<th>Entropy</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Economics</td>
<td>90.4%</td>
<td>86.1%</td>
<td>85.7%</td>
<td>94.6%</td>
<td>96.7%</td>
</tr>
<tr>
<td>Web Technology</td>
<td>68.5%</td>
<td>64.3%</td>
<td>66.7%</td>
<td>74.6%</td>
<td>83.3%</td>
</tr>
<tr>
<td>Books</td>
<td>85.4%</td>
<td>81.2%</td>
<td>87.7%</td>
<td>90.3%</td>
<td>95.2%</td>
</tr>
<tr>
<td>Food &amp; Drinks</td>
<td>80.4%</td>
<td>77.7%</td>
<td>85.6%</td>
<td>88.6%</td>
<td>90.4%</td>
</tr>
<tr>
<td>Political Science</td>
<td>82.6%</td>
<td>81.1%</td>
<td>83.5%</td>
<td>83.75%</td>
<td>85.9%</td>
</tr>
</tbody>
</table>

Similarly for the remaining four classes the experiments were conducted and tabulated in Table 7. The patterns predicted under false positive are more than false negative in both the methods. It is observed that even though the patterns are reduced more compactly using TFIDF the accuracy is less compared to the proposed scheme. The classification accuracy of the proposed method by varying different term weighting methods is tabulated in Table 7. The accuracy of the proposed scheme is found to be remarkable on comparing all the weighting schemes. The classifier performance in terms of accuracy, precision, recall and F-measure is tabulated in Table 8.

### REFERENCES


