International Journal of Computer Theory and Engineering, Vol. 1, No. 5, December, 2009
1793-8201

Classification Of Documents Using Kohonen's Self-Organizing Map

B.H.ChandraShekar, Dr.G.Shoba

Abstract—**Innovative methods that are user friendly and efficient are needed for retrieval of textual information available on the World Wide Web. The self-organizing map (SOM) is one of the most widely used neural network algorithms. SOM can be used to identify clusters of documents with similar context and content. In this paper, we explore and visualize the Self Organizing Map and discuss how to classify text documents. The paper also portrays the capabilities of SOM in text classification. We also discuss about experiments done using 20 news group dataset**.

***Index Terms*— Self-organizing map, stop words, term-document, neurons.**

I. INTRODUCTION

In the current scenario, browsing for exact information has become very tedious job as the number of electronic documents on the Internet has grown gargantuan and still is growing. It is necessary to classify the documents into categories so that retrieval of documents becomes easy and more efficient. For example, we have enough information to classify News articles of ten years, but it is impossible to see what is going on without mechanical assist such assistance. We try to overcome this difficulty by efficiently organizing, the documents into set of related topics or categories, which enable the user query process, to be precise and optimized.

Basically document classification can be defined as content based assignment of one or more predefined categories or topics to documents ie., collection of words determine the best fit category for this collection of words. The goal of all document classifiers is to assign documents into one or more content categories such as technology, entertainment, sports, politics, etc., Classification of any type of text document is possible, including traditional documents such as memos and reports as well as e-mails, web pages, etc.,

Many techniques have been proposed, such as Principal Component Analysis and Singular Value Decomposition. The results of these techniques are difficult to translate and sometimes often they lose accuracy. To overcome these problems, document vectors based on significant words in each category is adopted. Even vector, based documents are of very high dimension, which causes difficulty in understanding results.

Automatic classification of documents can be done using supervised or unsupervised learning system to overcome some of the difficulties stated above.

A. Supervised Learning

In supervised learning, the classifier for a given input is able to classify it with respect to some kind of classification. For a system to be using supervised learning, a teacher must help the system in its model construction by defining classes and providing positive and negative examples of objects belonging to these classes. The system is then to find out common properties of the different classes, and what separates them, in order to make correct classification for other objects. Supervised document classifiers are commonly referred to as statistical document classifiers because they make use of statistical properties of category features during classification.

B. Unsupervised

This learning technique identifies groups or clusters, of related documents as well as the relationships among them. This approach is commonly referred to as clustering; because this approach eliminates the need for tagged training documents and also does not require a preexisting taxonomy or category structure. However, clustering algorithms are not always good at selecting categories that are intuitive to human users. For this reason, clustering generally works hand-in-hand with the previously described supervised learning.

Kohonen's Self Organizing Map is an unsupervised learning technique. By using Kohonen's SOM, we can reduce the dimensionality from a very high dimension data into 2 or 3 dimensional space. This reduction is dimensionality enables us to interpret the results easily and instinctively.

II. DOCUMENT PREPROCESSING

Data preprocessing is a very important and essential phase in an effective document classification. The first part of feature extraction is preprocessing the lexicon and involves removal of stop words, stemming and term weighting [2].

A. Stop Words

This is the first step in preprocessing which will generate a list of terms that describes the document satisfactorily. The document is parsed through to find out the list of all the words. The next process in this step is to reduce the size of the list created by the parsing process, generally using methods of stop words removal and stemming. The stop words

Manuscript received Tue, Jul 7, 2009. This work was is carried out as one of a part of the research work titled "An Intelligent Agent for Efficient Internet Searching" by B.H.Chandrashekar. B.H.ChandraShekar, is working as Lecturer in the Department of Master of Computer Applications, R.V.College of Engineering, Mysore Raod, Bangalore 560059, India. (e-mail: chandrashekarbh@gmail.com)

Dr. G.Shobha is the Director of Master of Computer Applications, R.V.College of Engineering, Mysore Road, Bangalaore 560059, India. (e-mail: shobhatilak@rediffmail.com)

removal accounts to 20% to 30% of total words counts while the process of stemming reduce the number of terms in the document. Both the process helps in improving the effectiveness and efficiency of text processing as they reduce the indexing file size.

Stop words are removed from each of the document by comparing the with the stop word list. This process reduces the number of words in the document significantly since these stop words are insignificant for search keywords. Stop words can be pre-specified list of words or they can depend on the context of the corpus.

B. Stemming

The next process in phase one after stop word removal is stemming. Stemming is process of linguistic normalization in which the variant forms of a word is reduced to a common form. For example: the word, connect has various forms such as connect, connection, connective, connected, etc., Stemming process reduces all these forms of words to a normalized word connect. Porter's English stemmer algorithm is used to stem the words for each of the document in our stemming process.

C. Document Representation

A Document is represented by a set of keywords/ terms extracted from the document. The collection or union of all set of terms is the set of terms that represents the entire collection and defines a 'space' such that each distinct term represents one dimension in that space. Since each document is represented as a set of terms, this space is called 'document space" [3].

A term-document matrix can be encoded as a collection of n documents and m terms. An entry in the matrix corresponds to the "weight" of a term in the document; zero means the term has no significance in the document or it simply doesn't exist in the document. The whole document collection can therefore be seen as a m x n-feature matrix A (with m as the number of documents) where the element a_{ij} represents the frequency of occurrence of feature **j** in document **i.** This was of representing the document is called term-frequency method. However the terms that have a large frequency are not necessary more important or have higher discrimination power. So we might, want to weight the terms with respect to the local context, the document or the corpus. The most popular term weighting is the Inverse document frequency, where the term frequency is weighed with respect to the total number of times the term appears in the corpus. There is an extension of this designated the term frequency inverse document frequency (tf-idf). The formulation of tf-idf is given as follows:-

$$W_{ij} = tf_{i,j} * \log (N / df_i)$$

where w_{ij} is the weight of the term **i** in document **j**, $tf_{i,j}$ = number of occurrences of term **i** in document **j**, N is the total number of documents in the corpus, df_i = is the number of documents containing the term **i**.

The development and understanding of the impact of terms and weights on text data mining methodologies is another area where the statisticians can contribute. The encoding scheme is best explained in the recent work by Berry[4].

Figure below show the term document frequency for the title of the books.

D1 - *Data mining techniques*: for marketing sales and *customer relationship managemen*
D2 - Principles *of data mining*: Adaptive computation & *machine learning*
D3 – *Data mining*: practical *machine learning* tools & *techniques* with Java
D4 –*Mastering Data Mining* – the arts and *science of Customer Relationship Management*
D5 - *Mastering Data* Modeling: A user driven approach
D6 - Investigat*e Data mining* for security and *Criminal detection*
D7 – *Science* and *criminal detection*
D8 - *Crime* and Human nature: the definitive study of the causes and *crime*
D9 – Statistics of *crime* and *criminals*: a handbook of primary *data*

Term – Document Matrix

	D1	D2	D3	D4	D5	D6	D7	D8	D9
Crime	0	0	0	0	0	1	1	2	2
Customer	1	0	0	1	0	0	0	0	0
Data	1	1	1	1	1	1	0	0	1
Detection	0	0	0	0	0	1	1	0	0
Learning	0	1	1	0	0	0	0	0	0
Machine	0	1	1	0	0	0	0	0	0
Management	1	0	0	1	0	0	0	0	0
Mastering	0	0	0	0	1	0	0	0	0
Mining	1	1	1	0	0	1	0	0	0
Relationship	1	0	0	1	0	0	0	0	0
Science	0	0	0	1	0	0	1	0	0
Technique	1	0	1	0	0	0	0	0	0

Fig. Shows a small corpus of 9 book titles; each title is a document. To save space, we are using only italicized words in the document list. The ij –th element of the term-document matrix shows the number of times the ith word is repeated in the jth document.

D. Dimensionality Reduction

The space in which the document, reside is typically thousands of dimensions or more. Given the collection of documents along with the associated distance matrix, we

would like to find a convenient lower-dimensional space to perform subsequent analysis. This will certainly facilitate clustering or classification. By dimensionality reduction, one can remove noise from data and better apply our statistical data mining methods to discover subtle relationship that might exist between the documents.

1) Latent Semantic Indexing

A well known theorem from linear algebra to obtain a set of useful projection via singular value decomposition is the best choice popularly known as latent semantic Indexing (analysis) in the field of text data mining and natural language processing [5].

The singular value decomposition allows us to write the term-document matrix as product of 3 matrices.

$$X = T S D^T$$

where T is the matrix of left singular vector, S is the diagonal matrix of singular values, and D is the matrix of right singular vectors. The exact mathematical decomposition is described in detail[6].

2) Principal Component Analysis

The principal component analysis is a popular method, which uses eigenvectors from either covariance or correlation matrix to reduce the dimensionality[7].

PCA is used for dimensionality reduction in a dataset by retrieving those characteristics of the dataset that contributes most of its variance; by keeping lower-order principal component and ignoring higher order ones. Such lower order components often contain the "most important" aspect of the data.

III. Document Clustering

This is the second phase of Document classification. In this phase, we describe about the self-organizing map, its network architecture, learning algorithm and tools for visualizing the results.

A. Self-Organizing map architecture

SOM learn to classify data without supervision. With this approach an input vector is presented to the network and the output is compared with the target vector. If they differ, the weights of the network are altered slightly to reduce the error in the output. This is repeated many times and with many sets of vector pairs until the network gives the desired output.

The network is created from a 2D lattice of 'nodes', each of which is fully connected to the input layer. Fig.3 shows a very small Kohonen's network of 4 X 4 nodes connected to the input layer representing a two dimensional vector.

All neurons in the output layer are well connected to adjacent neurons by a neighborhood relation depicting the structure of the map. Generally the output layer can be arranged in rectangular or hexagonal lattice.

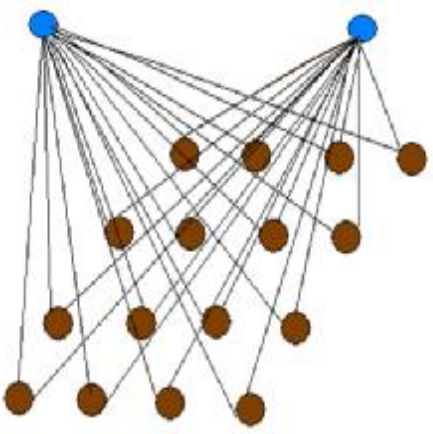

Figure 3

SOM topology - network containing Two layer of nodes an input layer (blue nodes) and an output layer (brown nodes) in the shape of two dimensional

Figure 4 below shows 30 x 30 and 10 x 10 two different neuron hexagonal grid.

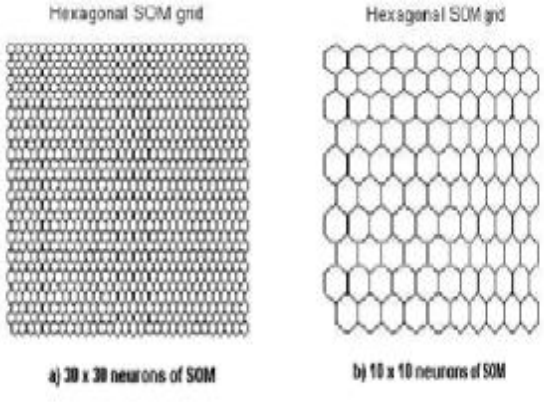

Figure 4: Arrangment of neurons in two different number sizes of hexagonal grid

IV. IMPLEMENTATION

A SOM was used to cluster the input vectors from the preprocessed documents. The SOM is an algorithm used to visualize and interpret large high-dimensional data sets. The map consists of regular grid of processing units, neurons. A document of some multidimensional observations, eventually a vector consisting of features, is associated with each unit. The map attempts to represent all the available observations with optimal accuracy using a restricted set of vector documents. At the same time the vector documents become ordered on the grid so that similar vector documents are close to each other and dissimilar documents far from each other. A sequential regression process usually carries out fitting of the document vectors. In our approach, with the text document from high dimensional input data is finally reduced to a 2-dimensional map of similar documents pattern with different clusters formed from the SOM training process. [8][9].

The data set used was 7000 files, which were grouped into 20 sets taken from 20 Newsgroup dataset. These 20 categories were stored in a directory named trainset.

The implementation was carried out using Java language. The process of classification of documents was carried out in

3 phases.

The first phase is document preprocessing in which the stop words removal, stemming of words, document representation carried on. The second phase is the training process in which learning algorithm was used to build the clusters of documents that are similar. The third phase is the test phase in which a document is classified and the weights of neighboring units are updated.

V. RESULTS AND CONCLUSION

Experiment result:

With Celeron M processor 1.46GHz, 512MB RAM.

	Total Input files			
	350		200	
Mapping time(s)	91		74	
No. of files to test	75	100	75	100
Time taken(s)	39	51	27	40
Accuracy(%)	57	67	46	48

With the above results we can analyze that the following:

1) By training more number of documents the mapping time is reduced.
2) The accuracy for testing a file for classification can be enhanced if the number of trained documents is large.
3) The percent of accuracy to classify and cluster a document grows with more number of trained documents.

The Web has become the largest source knowledge repository. Extracting information and building knowledge from extracted information efficiently and effectively is becoming increasingly important for various reasons. As popularity of the web continues to increase, there is a growing need to develop tools and techniques that will help improve its overall usefulness.

VI. REFERENCES

[1] Blaz Fortuna, Dunja mladenic, Marko Grobelnik, Semi-Automatic Construction of Topic Ontology, Semantics, Web and Mining, Joint International Workshop, EWMF 2005 and KDO 2005.
[2] Wei, C. P, Dong, Y. X. (2001): A Mining-Based Category Evolution, Approach to Managing Online Document Categories, in: Proceedings of the 34th ,Annual Hawaii International Conference on System Sciences.
[3] J. L. Martínez-Fernández , A. García-Serrano, P. Martínez1, J. Villena, "Automatic Keyword Extraction for News Finder", Computer Science Department, Universidad Carlos III de Madrid, Avda. Universidad 30, 28911 Leganés, Madrid, Spain
[4] Berry, M. W. (2003). Survey of Text Mining: Clustering, Classification, and Retrieval (Hardcover). Springer.
[5] Deerwester, S., Dumais, S. T., Furnas, G. W., and Landauer, T. K. (1990). Indexing by latent semantic analysis. Journal of the Am. Soc. for Information Science 41, 6, 391–407.
[6] Hadi, A. (2000). Matrix Algebra as a Tool. CRC.
[7] Jolliffe, I. (1986). Principal Component Analysis. Springer-Verlag.
[8] Vesanto, 1999, SOM-based data visualiation methods. Intelligent-Data-Analysis, v3, 111-126, Vesanto and Alhoniemi,2000.
[9] Kosala, R Blockeel, H. (2000). Web Mining Research: A Survey. ACM SIGKDD, July2000.

B.H.Chandrashekar is working is R.V.College of Engineering from 1990. He has completed his Bachelor of Computer Appications and Master of Computer Applications from Indira Gandhi National Open Univeristy, New Delhi. M.Phil in Computer Science from Alagappa University, Karikudi, Tamil Nadu and is persuing Ph.D in Avinashilingam Univeristy, Coimbatore, Tamil Nadu. He has life memberships in Computer Society of India (CSI) and Indian Society for Technical Eduction (ISTE). His area of interest is Information Retrieval, Information Filtering and Data Mining.

Dr. Shobha is working in R.V.College of Engineering and has experience about 15 years of teaching and 6 years of Research and Development. She has completed her Bachelor of Engineering in Computer Science and Master of Sciences from BITS, Pilani, India. She completed her Doctarate Degree from Mangalore Univeristy, Karnataka, India. She has been awarded Young Teachers Career Award by All India Council of Technical Eduction, has Received certificate of appreciation from ISTE, RVCE chapter for presenting Papers in National/ International Conferences / Journals during the year 2007-2008. Her area of interest is Data Mining.