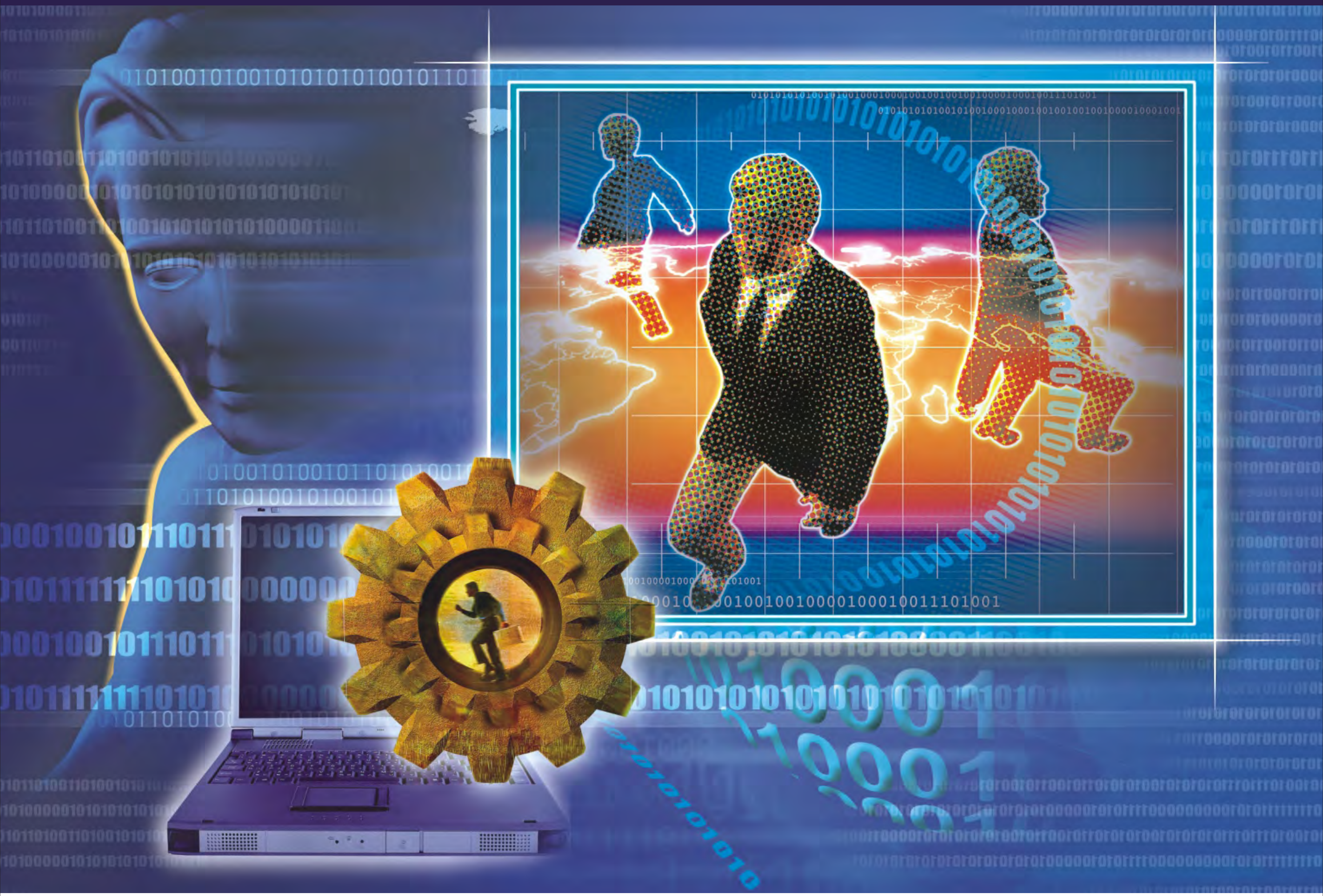
ISSN: 2150-8402 Volume 2, Number 3, August 2010

Scientific
Research

Journal of Intelligent Learning Systems and Applications

Editor-in-Chief: Prof. Xin Xu



ISSN: 2150-8402

9 772150 840003 03

www.scirp.org/journal/jilsa

Journal Editorial Board

ISSN: 2150-8402 (Print), 2150-8410 (Online)

http://www.scirp.org/journal/jilsa

J. Intelligent Learning Systems & *Applications*, 2010, 2, 117-166
Published Online August 2010 in SciRes (http://www.SciRP.org/journal/jilsa/)

TABLE OF CONTENTS

Volume 2 Number 3 August 2010

Journal of Intelligent Learning Systems and Applications (JILSA)

Journal Information

SUBSCRIPTIONS

The *Journal of Intelligent Learning Systems and Applications* (Online at Scientific Research Publishing, www.SciRP.org) is published quarterly by Scientific Research Publishing, Inc., USA.

Subscription rates:

Print: $50 per issue.
To subscribe, please contact Journals Subscriptions Department, E-mail: sub@scirp.org

SERVICES

Advertisements

Advertisement Sales Department, E-mail: service@scirp.org

Reprints (minimum quantity 100 copies)

Reprints Co-ordinator, Scientific Research Publishing, Inc., USA.
E-mail: sub@scirp.org

COPYRIGHT

Statements and opinions expressed in the articles and communications are those of the individual contributors and not the statements and opinion of Scientific Research Publishing, Inc. We assumes no responsibility or liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained herein. We expressly disclaim any implied warranties of merchantability or fitness for a particular purpose. If expert assistance is required, the services of a competent professional person should be sought.

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:
E-mail: jilsa@scirp.org

J. Intelligent Learning Systems & Applications, 2010, 2, 117-125
doi:10.4236/jilsa.2010.23015 Published Online August 2010 (http://www.SciRP.org/journal/jilsa)

A SOM-Based Document Clustering Using Frequent Max Substrings for Non-Segmented Texts

Todsanai Chumwatana, Kok Wai Wong, Hong Xie

School of Information Technology, Murdoch University, South St, Murdoch, Australia.
Email: {T.Chumwatana, K.Wong, H.Xie}@Murdoch.edu.au

Received March 25th, 2010; revised July 15th, 2010; accepted July 30th, 2010.

ABSTRACT

This paper proposes a non-segmented document clustering method using self-organizing map (SOM) and frequent max substring technique to improve the efficiency of information retrieval. SOM has been widely used for document clustering and is successful in many applications. However, when applying to non-segmented document, the challenge is to identify any interesting pattern efficiently. There are two main phases in the propose method: preprocessing phase and clustering phase. In the preprocessing phase, the frequent max substring technique is first applied to discover the patterns of interest called Frequent Max substrings that are long and frequent substrings, rather than individual words from the non-segmented texts. These discovered patterns are then used as indexing terms. The indexing terms together with their number of occurrences form a document vector. In the clustering phase, SOM is used to generate the document cluster map by using the feature vector of Frequent Max substrings. To demonstrate the proposed technique, experimental studies and comparison results on clustering the Thai text documents, which consist of non-segmented texts, are presented in this paper. The results show that the proposed technique can be used for Thai texts. The document cluster map generated with the method can be used to find the relevant documents more efficiently.

Keywords: *Frequent Max Substring, Self-Organizing Map, Document Clustering*

1. Introduction

Document clustering has been an important issue [1] due to the rapid growth in the number of electronic documents. Document clustering, sometimes can be generalized as text clustering, indentifies the similarity of documents and summarize a large number of documents using key attributes of the clusters. Document clustering uses unsupervised learning techniques and may assist fast information retrieval or filtering [2]. This is because clustering technique categorizes documents into groups based on their similarity in term of their member occurrences. Thus clustering can be used to categorize document databases and digital libraries, as well as providing useful summary information of the categories for browsing purposes. In information retrieval, a typical search on document database or the World Wide Web can return several thousands of documents in response to the user's queries. It is often very difficult for users to identify their documents of interest from such a huge number of documents. Clustering the documents enables the user to have a clear and easy grasp of the relevant documents from the collection of documents which are similar to each other and could be relevant to the user's queries.

For text clustering in information retrieval, a document is normally considered as a bag of words, even though a document actually consists of a sequence of sentences and each sentence is composed from a sequence of words. Very often the positions of words are ignored when performing document clustering. Words, also known as indexing terms, and their weights in documents are usually used as important parameters to compute the similarity of documents [3]. Those documents that contain similar indexing terms and frequencies will be grouped under the same cluster. This process is straightforward for European languages where words are clearly defined by word delimiter such as space or other special symbols. European texts are explicitly segmented into word tokens that are used as indexing terms. Many algorithms have been developed to calculate the similarity of documents and to build clusters for fast information retrieval [4]. In contrast, document clustering can be a challenging task for many Asian languages such as Chinese, Japanese, Korean and Thai, because these languages are non-segmented languages, *i.e.*, a sentence is written continuously as a sequence of characters without explicit word boundary delimiters. Due to this characteristic, texts in a non-segmented document cannot be directly used to calculate the similarity. Some preprocessing needs to be performed first to discover keywords for Asian documents before clustering. As a result, most approaches for clustering non-segmented documents consist of two phases: a text mining process to extract the keywords, and a document clustering process to compute the similarity between the input documents.

2. Keyword Extraction

Keywords are usually regarded as an important key to identifying the main content of the documents. Most of the semantics are usually carried by nouns, although a sentence in a natural language text is composed of nouns, pronouns, articles, verbs, adjectives, adverbs, and connectives. Keyword extraction is one of the main applications of text mining. The objective of text mining is to exploit useful information or knowledge contained in textual documents [5]. Information Extraction (IE) is an essential task in text mining that describes a process of discovering interesting keywords underlying unstructured natural-language texts. Most keyword extraction methods proposed in the literature were accomplished by constructing a set of words from given texts. Keywords will then be selected from the set of words during the preprocessing step. Many approaches have been proposed to extract keywords from non-segmented documents such as Chinese [6], Japanese [7] or Thai documents [8]. Most techniques are based on word segmentation which is one of the most widely used information extraction techniques in Natural language Processing (NLP). However, most word segmentation approaches involve complex language analysis and require long computational time. After keyword extraction is performed, keywords are then transformed into feature vector of the words that appear in the documents. The term-weights (usually term-frequencies) of the words are also contained in each feature vector. The vector space model (VSM) has been a standard model of representing documents by containing the set of words with their frequencies [1]. In the VSM, each document is replaced by the vector of the words. The vector size is dependent on the number of keywords that appear in the documents. For instance, let w_{ik} be the weight of keyword k that appear in the document i, and $D_i = (w_{i1}, w_{i2},\ldots, w_{it})$ is the feature vector for document i, where t is the number of unique words of all documents. Therefore, the size of the feature vector is equal to t dimension as shown in **Figure 1**.

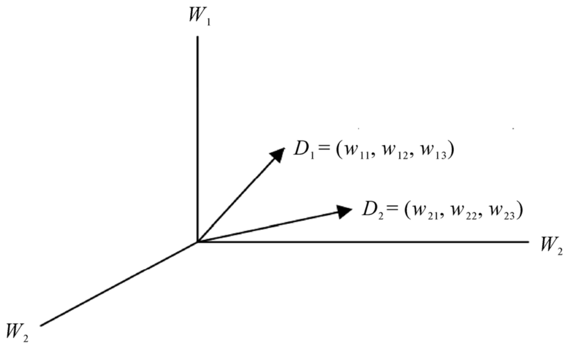

W_1
$D_1 = (w_{11}, w_{12}, w_{13})$
$D_2 = (w_{21}, w_{22}, w_{23})$
W_2
W_2

Figure 1. The example of the document vectors in 3-dimension

From **Figure 1**, the similarity between two documents can be computed with one of several similarity measures based on two corresponding feature vectors, e.g., cosine measure, Jaccard measure, and Euclidean distance measure [9].

3. Document Clustering Algorithms

In document clustering, there are two main approaches: hierarchical and partitional approaches [10,11,4]. The hierarchical approach produces document clusters by using a nested sequence of partitions that can be represented in the form of a tree structure called a dendrogram. The root of the tree contains one cluster covering all data points, and singleton cluster of individual data point are shown on the leaves of the tree. There are two basic methods when performing hierarchical clustering: agglomerative (bottom up) and divisive (top down) clustering [4]. The advantages of hierarchical approach are that it can take any form of similarity function, and also the hierarchy of clusters allows users to discover clusters at any level of detail. However, this technique may suffer from the chain effect, and its space requirement is at least quadratic or $O(n^2)$ compared to the k-means algorithm that provide $O(Iknm)$ where I is the number of necessary iterations, k is the number of clusters, n is the number of documents and m is the dimensionality of the vectors. The partitional approach [12], on the other hand, can be divided into several techniques, e.g., k-means [13], Fuzzy c-means [14], QT (quality threshold) [15] algorithms., The k-means algorithm is more widely used among all clustering algorithms because of its efficiency and simplicity. The basic idea of k-means algorithm is that it separates a given data into k clusters where each cluster has the center point, also called centroid, which can be used to represent the cluster. The main advantages of k-means algorithm are its efficiency and simplicity. Its weaknesses are that it is only applicable to data sets where the notion of the mean is defined, the number of clusters can be identified by users, and it is sensitive to data points that are very far away from other points called outliers [1]. Furthermore, Self-organizing map (SOM) [16,17] can be used as one of the clustering algorithms in the family of an artificial neural network. The self-organizing map is unsupervised neural network, capable of ordering high dimensional data in such a way that similar inputs are grouped spatially close to each other. To use SOM in document clustering, text documents are described by features with high dimensionality, and SOM based techniques have been successfully applied to document clustering. Some of the successful applications of SOM in document clustering are described in the next section.

4. Related Works

Many clustering techniques have been developed and can be applied to clustering English documents Most of these traditional approaches use documents as the basis for clustering [18,19]. The Vector Space Document (VSD) model is a very widely used data representation model for document clustering [20]. This data model starts with a representation of any document as a feature vector of the words that appear in documents. The term-weights of the words are also contained in the feature vectors. The similarity measures are used to compute the similarity of two document vectors. An alternative approach of document clustering is phrase-based document clustering. Zamir and Etzioni [21] introduced the notion of phrase-based document clustering. They proposed to use a generalized suffix-tree to obtain information about the phrases between two documents and use common phrases to cluster the documents. According to [22], Bakus, Hussin, and Kamel used a hierarchical phrase grammar extraction procedure to identify phrases from documents and used these phrases as features for document clustering. The self-organizing map (SOM) method was used as the clustering algorithm. An improvement in clustering performance was demonstrated when using phrases rather than single words as features.

Mladenic and Grobelnik used a Naive Bayesian method to classify documents based on word sequences of different length [23]. Experimental results show that using the word sequences whose length is no more than 3 words can improve the performance of a text classification system. But when the average length of used word sequences is longer than 3 words, there will be no difference between using word sequences or single words.

However, there are not many research works on phrase-based document clustering for Asian languages, primarily due to the fact that most Asian language texts are non-segmented and it is difficult to separate words and phrases from the non-segmented texts. Most document clustering approaches require a preprocessing stage where word segmentation, stopword removal or semantic analysis are performed. NLP techniques provide good support for this step. Word segmentation is very important step involved in most NLP processing tasks. A text is separated into a sequence of tokens by using word segmentation techniques. Many approaches have been proposed for Asian languages such as Chinese, Japanese, Korea and Thai languages.

In [24], a Chinese document clustering method using data mining technique and neural network model was proposed. This technique was divided into two main parts: the preprocessing part which provides Chinese sentence segmentation method, and the clustering part that adopts the dynamical SOM model with a view to dynamically clustering data. In addition, this method uses term vectors clustering process instead of document vectors clustering process.

In Thai language, Canasai and Chuleerat propose a parallel algorithm for clustering text documents based on spherical *k*-means [25]. They implemented an algorithm on the PIRUN Linux Cluster, which is a parallel computer using cluster computing technology. Experimental results show that the use of parallel algorithm can significantly improve clustering performance.

5. A SOM Based Clustering Using Frequent Max Substrings for Non-Segmented Texts

In this section, we describe a new method that combines Kohonen's SOM and frequent max substring technique to process the non-segmented text documents into clusters. SOM is one of the main unsupervised learning methods in the family of artificial neural networks (ANN) that was first developed by Teuvo Kohonen in 1984 [26]. The SOM can be visualized as a regular two-dimensional array of cells or nodes (neurons). The SOM algorithm defines a mapping from the input vector onto a two-dimensional array of nodes. When the input vector $x(t) \in R^n$ is given, it is connected to all neurons in the SOM array denoted as vector $m_i(t) \in R^n$,which are associated by each neuron and is gradually modified in the learning process. The input vector $x(t) \in R^n$ is the input data sets where t is the indexing terms of the input documents. These input data sets have to be mapped with all neurons in the map that is denoted as two-dimensional network of cells or the model vector $m_i(t) \in R^n$.

In mapping, the node where vector m_i is most similar to the input vector x will be activated. This node is often called a best-matching node or a winner. The winner and a number of its neighboring nodes in the SOM array are then turned towards the input vector x according to the learning principle.

The frequent max substring technique is an information extraction technique used to identify the terms called frequent max substrings from non-segmented texts where the word boundary and characteristic are not clearly defined. This technique was first introduced in 2008 [27] and has been proposed as an alternative language-independent technique for keyword extraction for non-segmented texts [28,29]. It also has been applied in application for indexing for non-segmented languages [8,30] as this technique provides good significant in term of the efficiency of storage space which could be able to support the rapid growth in the number of electronic non-segmented documents. The frequent max substring technique classifies indexing terms, known as frequent max substrings, from the non-segmented texts where the word boundaries are not clearly defined. The frequent max substrings refer to the substrings that appear frequently (at a given frequency threshold value *f*) and have the maximum length on the given string, so these terms are likely to be the patterns of interest. We extract the set of frequent max substrings or *FMAX* by using the frequent max substring technique. This technique uses Frequent Suffix Trie or FST data structure to explore the indexing terms [27]. The FST data structure is similar to suffix trie structure, that is an efficient substring enumeration method. However, FST data structure enumerates substrings with their frequencies and positions information while suffix trie structure enumerates only substrings without their frequencies information. Therefore, we employ FST data structure in order to support extracting the frequent max substrings. In this technique, the parameter and the predetermined frequency are applied to reduce the number of the indexing terms. This method uses the two reduction rules: 1) reduction rule using the predetermined frequency to check extracting termination, 2) reduction rule using superstring definition to reduce the number of indexing terms extracted. This technique also uses heap data structure to support computation [27].

In this paper, we use a set of non-segmented documents (Thai documents) as an input to train a map using SOM. We will describe the process of clustering as follows.

Let *D* be a document collection consisting of *n* documents, d_1, d_2, ..., d_n. Firstly, we use the frequent max substring technique [27] to generate a set of frequent max substrings at the given frequency threshold value *f* from the document collection to be used as the set of indexing terms for the document collection.

Assuming the above process produces *m* frequent max substrings from the document collection, denoted $FMAX = (fm_1, fm_2, ..., fm_m)$. where fm_i is the *i*th frequent max substring generated from the document collection. These *m* substrings are used as our indexing terms.

We will then calculate the weight w_{ij} which represents the frequency of indexing term fm_i occurring in document d_j for each indexing term and each document. Finally we construct an $m \times n$ matrix of such weights. In this matrix, row *i* represents the frequencies of occurrence of the *i*th indexing term fm_i in the *n* documents, while *j*th column represents the document vector for document *j*, as depicted in **Figure 2**.

Figure 2 shows an example of document matrix. In a document matrix, each element w_{ij} is at least at *f* if fm_i occurs in the document d_j or 0 if fm_i does not appear in the document d_j, *i.e.*,

$$w_{ij} = \begin{cases} \geq f & \text{if } fm_i \text{ occurs in } d_j \\ 0 & \text{otherwise} \end{cases}$$

After the document matrix is obtained, the document vectors are presented to SOM for clustering. These documents can be labeled into neurons according to the similarity of their document vectors. Two documents containing the same or similar document vectors will map to the same neuron. In contrast, the documents may map to distant neurons if they contain different or non-overlapping frequent max substrings. Furthermore, the documents with similar frequent max substrings may map to neighbouring neurons. This means that the neurons can form the document clusters by examining mapped neurons in the document cluster map. We depict the organization of the document map that clusters similar documents into the same neuron as shown in the boxes. *fms* in the boxes represent the content of documents in the collection.

$$\begin{array}{c} \qquad\qquad\qquad\qquad\qquad fm_{\mathrm{i}} \\ V = \left(\begin{array}{cccccccc} 4 & 3 & 7 & 0 & 0 & 0 \ldots & 0 & 0 \\ 3 & 3 & 0 & 0 & 2 & 0 & 2 & 0 \\ 6 & 4 & 5 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 \\ 3 & 3 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 4 & 7 & 4 & 9 \\ \vdots & & & & & & \vdots & \\ 0 & 0 & 0 & 3 & 0 & 4 \ldots & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 4 & 3 & 4 & 3 & 3 \\ 0 & 0 & 2 & 3 & 4 & 3 & 4 & 0 \end{array}\right) \begin{array}{l} \text{ผลการแข่งขัน (Competition result)} \\ \text{เป็นอันดับที่ (Competition rank)} \\ \text{ตารางการแข่งขัน (Competition time table)} \\ \text{กรรมการตัดสิน (Umpire)} \\ \text{รอบรอบชนะเลิศ (Semi final round)} \\ \text{อัตราการแลกเปลี่ยนเงินตรา (Currency exchange rate)} \\ \\ \text{ธุรกิจการลงทุน (Investment business)} \\ \text{อสังหาริมทรัพย์ (Real estate)} \\ \text{สถานะทางการเงิน (Financial status)} \\ \text{งบประมาณประจำปี (Yearly budget)} \end{array} \\ \quad d_1\ d_2\ d_3\ d_4\ d_5\ d_i\ \ldots\ d_{n-1}\ d_n \end{array}$$

Figure 2. The example of the document matrix at the given frequency threshold value *f* is equal to 2

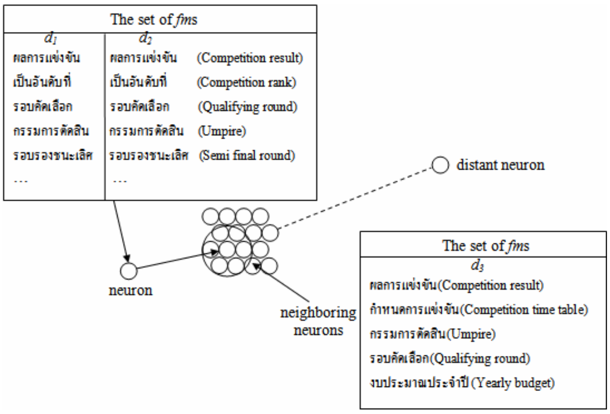

The set of *fms*
d_1
ผลการแข่งขัน
เป็นอันดับที่
รอบคัดเลือก
กรรมการตัดสิน
รอบรองชนะเลิศ
…
d_2
ผลการแข่งขัน (Competition result)
เป็นอันดับที่ (Competition rank)
รอบคัดเลือก (Qualifying round)
กรรมการตัดสิน (Umpire)
รอบรองชนะเลิศ (Semi final round)
…
distant neuron
neuron
neighboring neurons
The set of *fms*
d_3
ผลการแข่งขัน(Competition result)
กำหนดการแข่งขัน(Competition time table)
กรรมการตัดสิน(Umpire)
รอบคัดเลือก(Qualifying round)
งบประมาณประจำปี (Yearly budget)

Figure 3. The example of the document cluster map

After the SOM has been trained, the document clusters are formed by labeling each neuron that contains certain documents of similar type. The documents in the same neuron may not contain exactly the same set of frequent max substrings or *FMAX*, but they usually contain mostly overlapping frequent max substrings. As a result, the document cluster map can be used as a prediction model to generate the different groups of similar documents, and each group will then be used to specify the document type by comparing frequent max substrings of each group with keywords of each area. In **Figure 4**, we depict clustering the documents into different groups, by mapping an input data with neurons in the document cluster map to find the document groups of several types.

From **Figure 4**, the following will describe the process of matching an input data *x* with the neurons in the document cluster map by using SOM.

Let us consider the input vector $x = [x_1, x_2, \ldots, x_n]^t \in R^n$ as the input data sets where *t* is the frequent max substrings of the input documents. These input data sets have to be matched with all neurons in the map that is denoted as two-dimensional network of cells or the model vector $m_i = [m_{i1}, m_{i2}, \ldots, m_{in}]^t \in R^n$ depicted in **Figure 5**. Each neuron *i* in the network contains the model vector m_i, which has the same number of indexing terms as the input vector *x*.

From **Figure 5**, the input vector *x* is compared with all neurons in the model vector m_i to find the best matching node called the winner. The winner unit is the neuron on the map where the set of the frequent max substrings of the input vector *x* is the same or similar to the set of the frequent max substrings of the model vector m_i by using some matching criterion e.g. the Euclidean distances between *x* and m_i. As a result, this method can be used to cluster documents into different groups, and also suggested that this can use to reduce the search time for the relevant document.

6. Experimental Studies and Comparison Results

In this section, we describe an experiment for clustering non-segmented documents (Thai documents) based on the proposed SOM and frequent max substring technique. We also compare the proposed technique with the SOM based documents clustering using single words as features and hierarchical clustering technique using single words on a group of documents. 50 Thai documents were used as an input dataset to train a map. All Thai documents used are from Thai news websites that consist of different categories of contents: sport, travel, education and political news. The documents have varying lengths. The set of documents contains 103,287 characters, and average document length is 2,065 characters or 78 words per document. The basic statistics for the text collection are shown in **Table 1**.

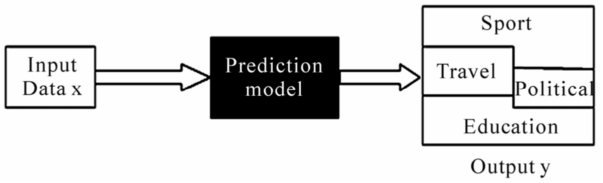


Figure 4. Neuron network architecture

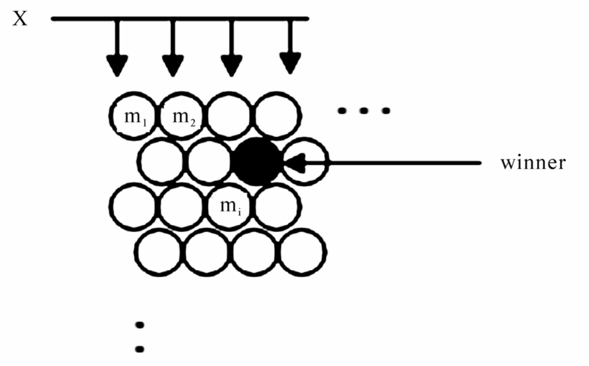


Figure 5. Self-organizing map

Table 1. Basic statistics for Thai text collection

	No. of Docs	No. of Chars	No. of Words	Avg. Chars./ Docs	Avg. Words/ Docs
Sport news	15	24727	997	1648.46	66.46
Travel news	15	29096	1022	2078.28	73
Political news	15	38017	1398	2534.46	93.2
Education news	5	9445	336	1889	67.2

In our proposed technique, the set of frequent max substring was first generated by frequent max substring technique at the given frequency threshold value, which is equal to 2 from the document dataset and 35 frequent max substrings, the long and frequently occurring terms in sport, travel, political and education documents, were used as the set of indexing terms for this document collection. The 50 input documents are then transformed to a document matrix of weighted frequent max substring occurrence. Hence, these 35 indexing terms and 50 input documents to form a 50 × 35 matrix, where each document vector was represented by each column of the matrix, and the rows of the matrix correspond to the indexing terms. We use this 50 × 35 matrix to train a map using SOM, and the number of neurons was set as 9 in SOM program as shown in **Figure 6**.

In the experimental study, 9 was set as the number of neurons because the several numbers of neurons were investigated, and 9 neurons provided the best result among them. From experimental studies, the group of political, sport, and education documents provided good results as similar documents of each type were mapped onto the neuron. It can be observed that some errors occurred within the group of travel documents. The travel documents were mapped onto several neurons due to overlapping terms that appeared across different type of documents.

The **Figure 6** showed the map containing 9 neurons and 50 Thai documents. Each neuron contains the group of similar documents.

From **Figure 6**, the experimental result showed that SOM can cluster 50 documents into 5 neurons on the map, and the similar documents were grouped into the same neuron as shown in **Table 2**.

As observed from the results, this technique can be used to cluster non-segmented documents into several groups according to their similarity. The accuracy of this technique is up to 83.25%. However, from this experiment, we have found that the groups of education and sport documents are mapped onto the same neuron (Neuron5) because they both contain mostly overlapping frequent max substrings such as ผลการแข่งขัน (competition result), การจัดอันดับ (position ranking), ได้รับรางวัล (getting award), etc. Furthermore, the contents of documents and generated indexing terms are also the main factors that impact the accurate value. The content of one document may have overlapping terms from two different types of documents. For instance, Education5, Travel 1, Travel 3, Travel 11 and Travel 14 documents are mapped onto the neuron 3 because they are presenting information on ecotourism, containing overlapping terms from education and travel documents.

The methodology to measure the clustering approaches is to compare the group of documents occurrences. In this paper, we compare our technique with the SOM based documents clustering using single words [31] as features and hierarchical based document clustering using single words [4,12]. In the SOM based documents clustering using single words, single words were first extracted by using Thai word segmentation from the same document data set and 35 single words, most frequent occurring single keywords in education, sport, travel and political documents, and 50 input documents are used to form a 50 × 35 matrix, which is the same matrix size as our earlier experiment to train a map using SOM. From experimental studies, it can be observed that only the groups of politic, and travel documents provided fairly good results as shown in **Table 3**.

The group of similar travel documents was mapped onto the neuron as well as the group of political documents. Meanwhile, the education and sport documents were distributed onto several neurons because most single words extracted from education and sport documents are general terms. These general terms can be shared in many types of documents.

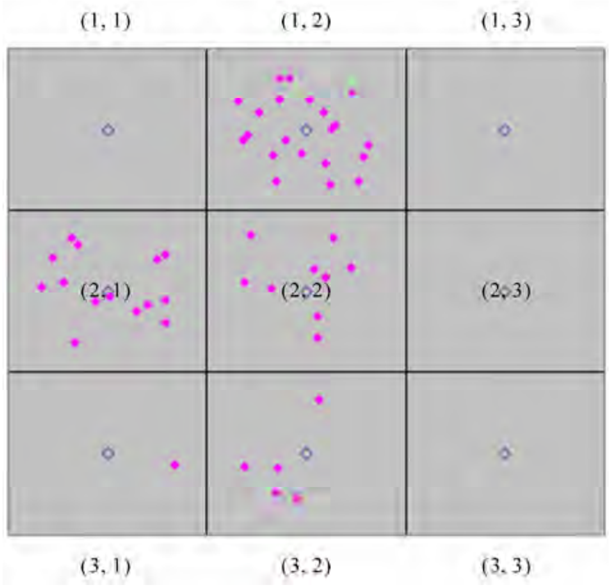


Figure 6. SOM contains 9 neurons and the group of similar documents from 50 Thai document collection

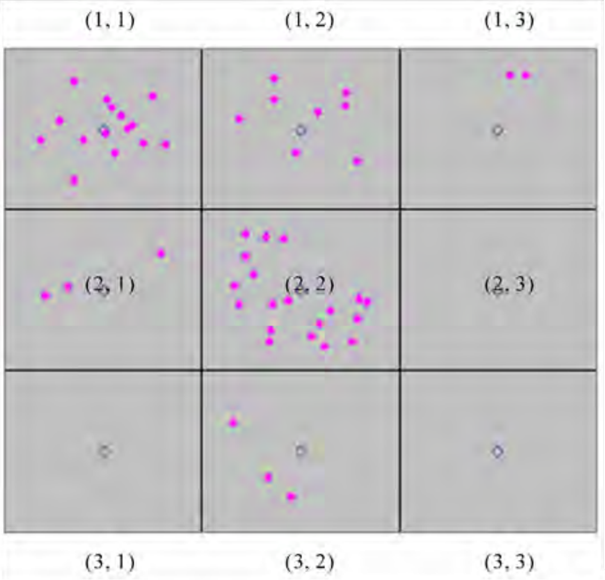


Figure 7. SOM contains 9 neurons and the group of similar documents from 50 Thai document collection by using single words as features

To depict the experiment results, **Figure 7** showed the map containing 9 neurons and 50 Thai documents, and the documents were grouped into the neurons as shown in **Table 3**.

It can be observed that the accuracy of the SOM based documents clustering using single words is 72.21%. Moreover, from the experiment results, we have found that the SOM based documents clustering using frequent max substrings provides better result than the SOM based documents clustering using single words because the frequent max substrings can be used to describe the

Table 2. Clustering results of using SOM and frequent max substring technique

Neuron ID	Row	Column	Document ID
Neuron 5	1	2	Political 1, Education 1, Education 2, Education 3, Education 4, Sport 1, Sport 2, Sport 3, Sport 4, Sport 5, Sport 6, Sport 7, Sport 8, Sport 9, Sport 10, Sport 11, Sport 12, Sport 13, Sport 14, Sport 15, Travel 10
Neuron 2	2	1	Political 2, Political 3, Political 4, Political 5, Political 6, Political 7, Political 8, Political 9, Political 10, Political 11, Political 12, Political 13, Political 14, Political 15,
Neuron 4	2	2	Travel 2, Travel 4, Travel 5, Travel 6, Travel 7, Travel 8, Travel 9, Travel 13, Travel 15
Neuron 1	3	1	Travel 12
Neuron 3	3	2	Education 5, Travel 1, Travel 3, Travel 11, Travel 14

Table 3. Clustering results of SOM based documents clustering using single words

Neuron ID	Row	Column	Document ID
Neuron 2	1	1	Education 1, Sport 1, Sport 2, Sport 5, Sport 6, Political 1, Political 2, Political 3, Political 4, Political 6, Political 8, Political 9, Political 10, Political 11, Political 12
Neuron 5	1	2	Education 3, Education 5, Sport 7, Sport 8, Sport 9, Sport 11, Sport 13, Travel 7
Neuron 6	1	3	Sport 12, Sport 14,
Neuron 1	2	1	Sport 15, Travel 15, Politica l7
Neuron 4	2	2	Education 2, Sport 3, Sport 4, Travel 1, Travel 2, Travel 3, Travel4, Travel 5, Travel 6, Travel 8, Travel 9, Travel 10, Travel 11, Travel 12, Travel 13, Travel 14, Political 13, Political 14, Political 15
Neuron 3	3	2	Education 4, Sport 10, Politcal 5

content of the documents more specifically than using single words, as the frequent max substrings can be referred to the frequently and long terms rather than individual words. Moreover, many researches have also shown an improvement in clustering performance when using phrases rather than single words as features [21,22].

Additionally, we also compare our technique with the hierarchical based document clustering using single words. The hierarchical based document clustering is used because it has been widely used and has been applied successfully in many applications in the area of document clustering [12]. This method has also been used to perform Thai documents clustering. To use this technique with Thai language, single words were first extracted by using Thai word segmentation techniques from the same document dataset used in our experiment that is discussed earlier. After word segmentation is performed, single words are then transformed into feature vector of the words that appear in documents. The term-frequencies of the words are also contained in each feature vector. The feature vector of the words is then used to compute the similarity of the documents by using the hierarchical clustering approach. In the hierarchical clustering program, the feature vector of the words with their frequencies was used as an input data, and the number of clusters was set to 9 after trial-and-error. From the experimental results, the group of political, travel and education documents provided good results. However, travel and education documents were grouped into the same cluster (cluster 1). It can be observed that the travel and education documents are sharing overlapping words that appeared across different type of documents. In addition, some of travel, education, sport and travel document were distributed across several small clusters. In **Table 4**, the experimental result showed that the hierarchical clustering program can cluster 50 documents into 9 clusters.

As can be observed from the results, the accuracy of the proposed method is up to 83.25%, meanwhile using the SOM based documents clustering using single words and the hierarchical clustering approach provide the accuracies 72.21% and 79.75% respectively. The hierarchi-

Table 4. Clustering results of using the hierarchical clustering approach

Cluster ID	Document ID
Cluster 1	Education 1, Education 2, Education 3, Education 4, Sport 9, Sport 13, Travel 1, Travel 2, Travel 3, Travel 4, Travel 5, Travel 7, Travel 8, Travel 9, Travel 10, Travel 11, Travel 12, Travel 13, Travel 14, Travel 15, Political 15
Cluster 2	Education 5
Cluster 3	Sport 1
Cluster 4	Sport 2, Sport 3, Sport 4, Sport 5, Sport 6, Sport 7, Sport 8, Sport 15
Cluster 5	Sport 10, Sport 11
Cluster 6	Sport 12
Cluster 7	Sport 14
Cluster 8	Travel 6
Cluster 9	Political 1, Political 2, Political 3, Political 4, Political 5, Political 6, Political 7, Political 8, Political 9, Political 10, Political 11, Political 12, Political 13, Political 14

cal clustering approach also created many small clusters that containing only a few documents. As a result, an improvement was demonstrated using frequent max substrings rather than single words as features. This proposed technique also does not require any pre-processing technique to extract the frequent max substrings. Meanwhile, the SOM based documents clustering using single words and the hierarchical based document clustering using single words require word segmentation to extract the single words.

7. Conclusions

This paper describes a non-segmented document clustering method using self- organizing map (SOM) and frequent max substring technique to improve the efficiency of information retrieval. We first use the frequent max substring technique to discover patterns of interest, called frequent max substrings, rather than individual words from Thai text documents, and these frequent max substrings are then used as indexing terms with their number of occurrences to form a document vector. SOM is then applied to generate the document cluster map by using the document vector. The experiment studies and comparison results on clustering the 50 Thai text documents is presented in this paper. We compare the proposed technique with the SOM based documents clustering using single words and hierarchical based document clustering technique with the use of single words for grouping the document occurrences. From the experimental results, our technique can be used to cluster 50 Thai documents into different clusters with more accuracy than using the SOM based documents clustering using single words and the hierarchical clustering approaches. As a result, the generated document cluster map from our technique can be used to find the relevant documents according to a user's query more efficiency.

REFERENCES

[1] B. Liu, "Web Data Mining: Exploring Hyperlinks, Con-tents, and Usage Data," 1st Edition, Springer-Verlag, New York Berlin Heidelberg, 2007.

[2] D. R. K. R. D. Cutting, J. O. Pedersen, J. W. Tukey, "Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections," *Proceedings of ACM Special Interest Group on Information Retrieval* '92, Copenhague, 1992, pp. 318-329.

[3] I. Matveeva, "Document Representation and Multilevel Measures of Document Similarity," *Irina Matveeva, Document representation and multilevel measures of document similarity, Proceedings of the* 2006 *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*: *companion volume*: *doctoral consortium*, New York, 2006, pp. 235-238.

[4] G. K. M. Steinbach and V. Kumar, "A Comparison of Docu-ment Clustering Techniques," *KDD Workshop on Text Mining*, Boston, 2000.

[5] A.-H. Tan, "Text Mining: The state of the art and the challenges," *Proceedings of the PAKDD Workshop on Knowledge Discovery from Advanced Databases*, Beijing, 1999, pp. 65-70.

[6] Q. L. H. Jiao and H.-B. Jia, "Chinese Keyword Extraction Based on N-Gram and Word Co-occurrence, 2007 *International Conference on Computational Intelligence and Security Workshops* (*CISW* 2007), Harbin, 2007, pp. 124-127.

[7] J. Mathieu, "Adaptation of a Keyphrase Extractor for Japanese Text," *Proceedings of the* 27*th Annual Conference of the Canadian Association for Information Science* (*CAIS*-99), Sherbrooke, Quebec, 1999, pp. 182-189.

[8] T. Chumwatana, K. W. Wong and H. Xie "An Automatic Indexing Technique for Thai Texts Using Frequent Max Substring," 2009 *Eight International Symposium on Natural Language Processing*, Bangkok, 2009, pp. 67-72.

[9] R. Feldman and J. Sanger, "The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data," Cambridge University Press, Cambridge, 2006.

[10] A. K. Jain and R. C. Dubes, "Algorithms for Clustering Data," Prentice Hall, New Jersey, 1988.

[11] L. Kaufman and P. J. Rousseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis," John Wiley and Sons, New York, 1990.

[12] G. K. Y. Zhao, "Comparison of Agglomerative and Partitional Document Clustering Algorithms," The SIAM workshop on Clustering High-dimensional Data and Its Applications, Washington, DC, April 2002.

[13] Z. Huang, "Extensions to the K-means Algorithm for Clustering Large Datasets with Categorical Values," *Data Mining and Knowledge Discovery*, Vol. 2, No. 3, 1998, pp. 283-304.

[14] D. Dembele and P. Kastner, "Fuzzy C-Means Method for Clustering Microarray Data," *Bioinformatics*, Vol. 19, No. 8, 2003, pp. 973-980.

[15] L. J. Heyer, S. Kruglyak and S. Yooseph, "Exploring Expression Data: Identification and Analysis of Coexpressed Genes," *Genome Research*, Vol. 9, No. 11, 1999, pp. 1106-1115.

[16] C. C. Fung, K. W. Wong, H. Eren, R. Charlebois and H. Crocker, "Modular Artificial Neural Network for Prediction of Petrophysical Properties from Well Log Data," *IEEE Transactions on Instrumentation & Measurement*, Vol. 46, No. 6, December 1997, pp. 1259-1263.

[17] D. Myers, K. W. Wong and C. C. Fung, "Self-organising Maps Use for Intelligent Data Analysis," *Australian Journal of Intelligent Information Processing Systems*, Vol. 6 No. 2, 2000, pp. 89-96.

[18] D. R. Hill, "A Vector Clustering Technique," In: Samuelson, Ed., *Mechanized Information Storage*, *Retrieval and Dissemination North-Holland*, Amsterdam, 1968.

[19] J. J. Rocchio, "Document Retrieval Systems — Optimization and Evaluation," Doctoral Thesis, Harvard University, Boston, 1966.

[20] A. W. G. Salton and C. S. Yang, "A Vector Space Model for Automatic Indexing," *Communication of ACM*, Vol. 18, No. 11, 1975, pp. 613-620.

[21] O. Zamir, "Clustering Web Documents: A Phrase-Based Method for Group Search Engine Results," *Computer Science & Engineering*, Ph.D. Thesis, University of Washington, 1999.

[22] M. F. H. J. Bakus and M. Kamel, "A SOM-Based Document Clustering Using Phrases," *Proceeding of the* 9*th International Conference on Neural Information Processing* (*ICONIP*'02), Vol. 5, 2002, pp. 2212-2216.

[23] D. Mladenic and M. Grobelnik, "Word Sequence as Features in Text-learning," *Proceedings of the* 17*th Electro technical and Computer Science Conference* (*ERK*-98) Ljubljana, Slovenia, 1998.

[24] K.-H. Tsai, C.-M. Tseng, C.-C. Hsu and H.-C. Chang, "On the Chinese Document Clustering Based on Dynamical Term Clustering," *Asia Information Retrieval Symposium* 2005, Jeju Island, October 2005, pp. 534-539.

[25] C. Kruengkrai and C. Jaruskulchai, "Thai Text Document Clustering Using Parallel Spherical K-means Algorithm on PI-RUN Linux Cluster (in Thai)," *The* 5*th National Computer Science and Engineering Conference*, Chiang Mai University, Chiang Mai, 2001, pp. 7-9

[26] T. Kohonen, "Self-Organization and Associative Memory," *Springer Series in Information Sciences*, Springer-Verlag, Berlin, 1984, p. 125.

[27] T. Chumwatana, K. W. Wong and H. Xie "Frequent max substring mining for indexing," *International Journal of Computer Science and System Analysis* (*IJCSSA*), India, 2008, pp. 179-184.

[28] T. Chumwatana, K. W. Wong and H. Xie "An Efficient Text Mining Technique," 9th *Postgraduate Electrical Engineering & Computing Symposium* (*PEECS*2008), Perth, Australia, 2008, pp. 147-152.

[29] T. Chumwatana, K. W. Wong and H. Xie, "Using Frequent Max Substring Technique for Thai Keyword Extraction used in Thai Text Mining," 2*nd International Conference on Soft Computing*, *Intelligent System and Information Technology* (*ICSIIT* 2010), Bali, 1-2 July 2010, pp. 309-314.

[30] T. Chumwatana, K. W. Wong and H. Xie, "Thai Text Mining to Support Web Search for E-Commerce," *The* 7*th International Conference on e-Business* 2008 (*INCEB* 2008), Bangkok, 2008, pp. 66-70.

[31] J. E. Hodges and Y. Wang, "Document Clustering using Compound Words," *Proceedings of the* 2005 *International Conference on Artificial Intelligence* (*ICAI* 2005), Las Vegas, Nevada, 2005, pp. 307-313.

J. Intelligent Learning Systems & Applications, 2010, 2, 126-138
doi:10.4236/jilsa.2010.23016 Published Online August 2010 (http://www.SciRP.org/journal/jilsa)

A New Multilevel Thresholding Method Using Swarm Intelligence Algorithm for Image Segmentation

Sathya P. Duraisamy, Ramanujam Kayalvizhi

[1]The Department of Electrical Engineering, Faculty of Engineering and Technology, Annamalai University, Chidambaram, India; [2]Department of Instrumentation Engineering, Faculty of Engineering and Technology, Annamalai University, Chidambaram, India.
Email: pd.sathya@yahoo.in, mithuvig.knr@gmail.com

Received June 11th, 2010; revised June 29th, 2010; accepted July 20th, 2010.

ABSTRACT

Thresholding is a popular image segmentation method that converts gray-level image into binary image. The selection of optimum thresholds has remained a challenge over decades. In order to determine thresholds, most methods analyze the histogram of the image. The optimal thresholds are often found by either minimizing or maximizing an objective function with respect to the values of the thresholds. In this paper, a new intelligence algorithm, particle swarm optimization (*PSO*)*, is presented for multilevel thresholding in image segmentation. This algorithm is used to maximize the Kapur's and Otsu's objective functions. The performance of the PSO has been tested on ten sample images and it is found to be superior as compared with genetic algorithm* (*GA*)*.*

Keywords: *Image Segmentation, Multilevel Thresholding, Particle Swarm Optimization*

1. Introduction

In many image processing applications, the gray levels of pixels belonging to an object are substantially different from those belonging to the background. As such, thresholding techniques can be used to extract the objects from their background. Indeed, thresholding is a major operation in many image processing applications such as document processing, image compression, particle counting, cell motion estimation and object recognition. The effect of many image processing applications strongly depends on the effect of image thresholding.

Thresholding techniques provide an efficient way, in terms of both the implementation simplicity and the processing time to perform image segmentation. However, the automatic selection of a robust optimum threshold has remained a challenge in image segmentation. Besides being segmentation on its own, thresholding is frequently used as one of the steps in many advanced segmentation methods. In these applications, thresholding is not applied on the original images, but applied in a space generated by the segmentation method. For example, in fuzzy connectedness segmentation [1], a threshold is applied on the strength of connectedness among image elements to produce a final segmentation. Thus, the methods to determine effective thresholds have wide-spread applications. However, automatic determination of the optimum threshold value is often a difficult task. While a number of approaches for automatic threshold determination have been proposed over the past several decades, applying new ideas and concepts to image thresholding remains an interesting and challenging research area.

Excellent reviews on early thresholding methods can be found in [2,3], whereas the latest development in this topic was summarized in [4]. Comparative performance studies of global thresholding techniques were presented by Lee *et al.* [5]. Otsu [6] proposed a method that maximizes between-class variance. Tao *et al.* [7] proposed a thresholding method for object segmentation based on fuzzy entropy theory and ant colony optimization algorithm. An image histogram thresholding approaches using fuzzy sets was proposed by Tobias and Seara [8].

Methods based on optimizing an objective function include maximization of posterior entropy to measure homogeneity of segmented Classes [9-11], maximization of the measure of seperability on the basis of between-class variance [6], thresholding based on index of fuzziness and fuzzy similarity measure [12,13], minimization of Bayesian error [14,15], etc. several such methods have originally been developed for bi-level thresholding and later extended to multilevel thresholding.

Bi-level thresholding divides the pixel into two groups, one including those pixels with gray levels above a certain threshold, the other including the rest. Multilevel thresholding divides the pixels into several groups; the pixels of the same group have gray levels within a specified range. However the problem gets more complex when the segmentation is achieved with greater details by employing multilevel thresholding. Then the image segmentation problem becomes a multiclass classification problem where pixels having gray levels within a specified range are grouped into one class. Usually it is not simple to determine exact locations of distinct valleys in a multimodal histogram of an image, that can segment the image efficiently and hence the problem of multilevel thresholding is regarded as an important area of research interest among the research communities worldwide.

A great number of thresholding methods of parametric or non-parametric type have been proposed in order to perform bi-level thresholding [16] and later extended to multilevel thresholding [17]. In [18], the Otsu's function is modified by a fast recursive algorithm along with a look-up-table for multilevel thresholding. In [19], Lin has proposed a fast thresholding computation using Otsu's function. Another fast multilevel thresholding technique has been proposed by Yin [20].

In recent years, several heuristic optimization techniques such as differential evolution (DE), Ant Colony Optimization (ACO) and Genetic Algorithms (GA) were introduced into the field of image segmentation because of their fast computing ability. Erik Cuevas *et al.* [21] applied the differential evolution (DE) algorithm to solve the multilevel thressholding problem. The algorithm fills the 1-D histogram of the image using a mix of Gaussian functions whose parameters are calculated using the differential evolution method. Each Gaussian function approximating the histogram represents a pixel class and therefore a threshold point. Tao *et al.* [22] proposed the Ant Colony Optimization (ACO) algorithm to obtain the optimal parameters of the entropy-based object segmentation approach.

Several techniques using genetic algorithms (GAs) have also been proposed to solve the multilevel thresholding problem [23,24]. Yin [23] introduced a neighborhood searching strategy in to the GA to speed up the multilevel thresholds optimization. Though GA-based approaches perform well for complex optimization problems, recent research has identified certain deficiencies [25], particularly for problems in which variables are highly correlated. In such cases, the GA crossover and mutation operators do not generate individuals with better fitness of offspring as the chromosomes in the population pool have some structure towards the end of the search.

PSO, first introduced by Kennedy and Eberhart [26] is a flexible, robust, population based stochastic search/optimization algorithm with inherent parallelism. This method has gained popularity over its competitors and is increasingly gaining acceptance for solving many image processing problems [27-29]. Compared with other population-based stochastic optimization methods such as DE, ACO and GA, PSO gives superior search performance with faster and more stable convergence rates [26].

This paper presents a new optimal multilevel thresholding algorithm; Particle Swarm Optimization (PSO) for solving the multilevel thresholding problem in image segmentation. The validity of the proposed method is tested on ten sample images and compared with the GA method.

2. Problem Formulation

In this paper, two broadly used optimal thresholding methods namely entropy criterion (Kapur's) method and between-class variance (Otsu's) method are used.

Kapur has developed the algorithm for bi-level thresholding and this bi-level thresholding can be described as follows:

Let there be L gray levels in a given image and these gray levels are in a given image and these gray levels are in the range {0, 1, 2,………,(L-1)}. Then one can define $P_i = h(i)/N$, $(0 \le i \le (L\text{-}1))$ where $h(i)$ denotes number of pixels for the corresponding gray-level L and N denotes total number of pixels in the image which is equal to $\sum_{i=0}^{L-1} h(i)$.

Then the objective is to maximize the fitness function

$$f(t) = H_0 + H_1 \tag{1}$$

where $H_0 = \sum_{i=0}^{t-1} \frac{P_i}{w_0} \operatorname{In} \frac{P_i}{w_0}$, $w_0 = \sum_{i=0}^{t-1} P_i$ and

$$H_1 = -\sum_{i=t}^{L-1} \frac{P_i}{w_1} \operatorname{In} \frac{P_i}{w_1}, \quad w_1 = \sum_{i=t}^{L-1} P_i$$

The optimal threshold is the gray level that maximizes Equation (1). This Kapur's entropy criterion method tries to achieve a centralized distribution for each histogram-based segmented region of the image.

This Kapur's entropy criterion method has also been extended to multilevel thresholding and can be described as follows: The optimal multilevel thresholding problem can be configured as a m-dimensional optimization problem, for determination of m optimal thresholds for a given image $[t_1, t_2 \ldots t_m]$, where the aim is to maximize the objective function:

$$f([t_1, t_2, \ldots\ldots t_m]) = H_0 + H_1 + H_2 + \ldots. + H_m \tag{2}$$

where

$$H_0 = \sum_{i=0}^{t_1-1} \frac{P_i}{w_0} \text{In} \frac{P_i}{w_0}, \quad w_0 = \sum_{i=0}^{t_1-1} P_i$$

$$H_1 = -\sum_{i=t_1}^{t_2-1} \frac{P_i}{w_1} \text{In} \frac{P_i}{w_1}, \quad w_1 = \sum_{i=t_1}^{t_2-1} P_i$$

$$H_2 = -\sum_{i=t_2}^{t_3-1} \frac{P_i}{w_2} \text{In} \frac{P_i}{w_2}, \quad w_2 = \sum_{i=t_2}^{t_3-1} P_i, \ldots..$$

$$H_m = -\sum_{i=t_m}^{L-1} \frac{P_i}{w_m} \text{In} \frac{P_i}{w_m}, \quad w_m = \sum_{i=t_m}^{L-1} P_i\ .$$

As Kapur based entropy criterion method, the Otsu based between-class variance method has also been employed in determining whether the optimal thresholding can provide histogram-based image segmentation with satisfactory desired. The Otsu based between-class variance algorithm can be described as follows:

If an image can be divided into two classes, C_0 and C_1, by a threshold at a level t, class C_0 contains the gray levels from 0 to t-1 and class C_1 consists of the other gray levels with t to L-1. Then, the gray level probabilities (w_0 and w_1) distributions for the two classes are as follows:

$$C_0 : \frac{P_0}{w_0}, \ldots\ldots \frac{P_{t-1}}{w_0} \text{ and } C_1 : \frac{P_t}{w_1}, \ldots\ldots \frac{P_{L-1}}{w_1}.$$

where, $w_0 = \sum_{i=0}^{t-1} P_i$ and $w_1 = \sum_{i=t}^{L-1} P_i$

Mean levels μ_0 and μ_1 for classes C_0 and C_1 are as follows:

$$\mu_0 = \sum_{i=0}^{t-1} \frac{i \times P_i}{w_0}, \quad \mu_1 = \sum_{i=t}^{L-1} \frac{i \times P_i}{w_1}.$$

Let μ_T be the mean intensity for the whole image, it is easy to show that

$$w_0 \mu_0 + w_1 \mu_1 = \mu_T \quad \text{and} \quad w_0 + w_1 = 1$$

Using discriminant analysis, Otsu based between-class variance thresholded image can be defined as follows:

$$f(t) = \sigma_0 + \sigma_1$$

where $\sigma_0 = w_0 (\mu_0 - \mu_T)^2$ and $\sigma_1 = w_1 (\mu_1 - \mu_T)^2$

For bi-level thresholding, Otsu selects an optimal threshold t^* that maximizes the between-class variance $f(t)$;
that is

$$t^* = \arg \ \max \{ f(t) \} \quad 0 \le t \le L\text{-}1$$

The above formula can be easily extended to multilevel thresholding of an image. Assuming that there are m thresholds, (t_0, t_1, …., t_m), which divide the original image into m classes: C_0 for [0, …., t_1-1], C_1 for [t_1, …., t_2−1] ….. and C_m for [t_m, …., L−1], the optimal thresholds $(t_0^*, t_1^*, ..., t_m^*)$ are chosen by maximizing $f(t)$ as follows:

$$(t_0^*, t_1^*, ..., t_m^*) = \arg \ \max \{ f(t) \} \quad 0 \le t_1 \le .... \le t_m \le L\text{-}1 \tag{3}$$

where $f(t) = \sigma_0 + \sigma_1 + \sigma_2 ..... + \sigma_m$

with $\sigma_0 = w_0 (\mu_0 - \mu_T)^2$,

$$\sigma_1 = w_1 (\mu_1 - \mu_T)^2,$$

$$\sigma_2 = w_2 (\mu_2 - \mu_T)^2, \ldots..$$

$$\sigma_m = w_m (\mu_m - \mu_T)^2.$$

The Kapur and Otsu methods have been proven as an efficient method for bi-level thresholding in image segmentation. However, when these methods are extended to multilevel thresholding, the computation time grows exponentially with the number of thresholds. It would limit the multilevel thresholding applications. To overcome the above problem, this paper proposes the Kapur and Otsu based PSO algorithm for solving multilevel thresholding problem. The aim of this proposed method is to maximize the Kapur's and Otsu's objective function using Equations (2) and (3).

3. Particle Swarm Optimization (PSO)

PSO is a simple end efficient population-based optimization method proposed by Kennedy and Eberhart [24]. It is motivated by social behavior of organisms such as fish schooling and bird flocking. In PSO, potential solutions called particles fly around in a multi-dimensional problem space. Population of particles is called swarm. Each particle in a swarm flies in the search space towards the optimum solution based on its own experience, experience of nearby particles, and global best position among particles in the swarm.

3.1 Advantages of PSO

1) PSO is easy to implement and only few parameters have to be adjusted.

2) Unlike the GA, PSO has no evolution operators such as crossover and mutation.

3) In GAs, chromosomes share information so that the whole population moves like one group, but in PSO, only global best particle (gbest) gives out information to the others. It is more robust than GAs.

4) PSO can be more efficient than GAs; that is, PSO often finds the solution with fewer objective function evaluations than that required by GAs.

Unlike GAs and other heuristic algorithms, PSO has the flexibility to control the balance between global and local exploration of the search space.

3.2 PSO Algorithm

Let X and V denote the particle's position and its corresponding velocity in search space respectively. At iteration K, each particle i has its position defined by $X_i^K = [X_{i,1}, X_{i,2} \ldots X_{i,N}]$ and a velocity is defined as $V_i^K = [V_{i,1}, V_{i,2} \ldots\ldots V_{i,N}]$ in search space N. Velocity and position of each particle in the next iteration can be calculated as

$$V_{i,n}^{k+1} = W \times V_{i,n}^{k} + C_1 \times \text{rand}_1 \times (\text{pbest}_{i,n} - X_{i,n}^{k}) + C_2 \times \text{rand}_2 \times (\text{gbest}_n - X_{i,n}^{k})$$

$$i = 1, 2 \ldots\ldots\ldots m$$

$$n = 1, 2 \ldots\ldots\ldots N \quad (4)$$

$$\begin{aligned} X_{i,n}^{k+1} &= X_{i,n}^{k} + V_{i,n}^{k+1} \text{ if } X_{\min,i,n} \le X_i^{k+1} \le X_{\max\, i,n} \\ &= X_{\min\, i,n} \text{ if } X_{i,n}^{k+1} < X_{\min\, i,n} \\ &= X_{\max\, i,n} \text{ if } X_{i,n}^{k+1} > X_{\max\, i,n} \end{aligned} \quad (5)$$

The inertia weight W is an important factor for the PSO's convergence. It is used to control the impact of previous history of velocities on the current velocity. A large inertia weight factor facilitates global exploration (*i.e.*, searching of new area) while small weight factor facilitates local exploration. Therefore, it is better to choose large weight factor for initial iterations and gradually reduce weight factor in successive iterations. This can be done by using

$$W = W_{\max} - (W_{\max} - W_{\min}) \times \text{Iter}/\text{Iter}_{\max}$$

Where $W_{\max}$ and $W_{\min}$ are initial and final weight respectively, Iter is current iteration number and Iter $_{\max}$ is maximum iteration number.

Acceleration constant C_1 called cognitive parameter pulls each particle towards local best position whereas constant C_2 called social parameter pulls the particle towards global best position. The particle position is modified by Equation (4). The process is repeated until stopping criterion is reached.

4. Implementation of PSO for Multilevel Thresholding Problem

This paper presents a quick solution to the multilevel image thresholding problems using the PSO algorithm. The number of threshold levels is the dimension of the problem. For example, if there are 'm' threshold levels, the ith particle is represented as follows:

$$X_i = (X_{i1}, X_{i2}, \ldots\ldots\ldots, X_{im})$$

Its implementation consists of the following steps.

Step 1. *Initialization of the swarm*: For a population size p, the particles are randomly generated between the minimum and the maximum limits of the threshold values.

Step 2. *Evaluation of the objective function*: The objective function values of the particles are evaluated using the objective functions given by Equation (2) or (3).

Step 3. *Initialization of pbest and gbest*: The objective values obtained above for the initial particles of the swarm are set as the initial pbest values of the particles. The best value among all the pbest values is identified as gbest.

Step 4. *Evaluation of velocity*: The new velocity for each particle is computed using Equation (4).

Step 5. *Update the swarm*: The particle position is updated using Equation (5). The values of the objective function are calculated for the updated positions of the particles. If the new value is better than the previous pbest, the new value is set to pbest. Similarly, gbest value is also updated as the best pbest.

Step 6. *Stopping criteria*: If the stopping criteria are met, the positions of particles represented by gbest are the optimal threshold values. Otherwise, the procedure is repeated from step 4.

5. Experimental Results and Discussions

In this section, the effectiveness and feasibility of the proposed PSO method for multilevel thresholding is demonstrated. Comparisons are performed with the results provided by GA based multilevel thresholding method. **Tables 1** and **2** represent the various parameters chosen for the implementation of GA and PSO algorithms respectively. Ten well-known images namely lena, pepper, baboon, hunter, map, cameraman, living room, house, airplane and butterfly are taken as the test images, and are gathered with their histograms in **Figure 1**.

The quality of the thresholded images for Kapur based

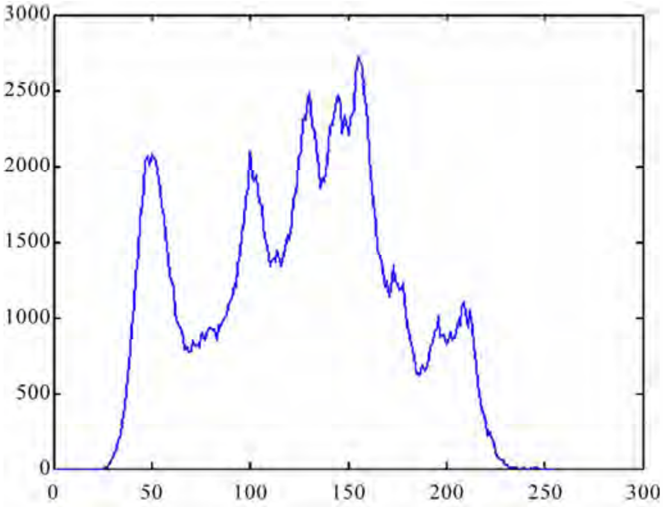
Table 1. Parameters chosen for GA implementation

Parameter	Value
Population size	20
No. of Iterations	100
Crossover probability	0.9
Mutation probability	0.1
Selection operator	Roulette Wheel Selection

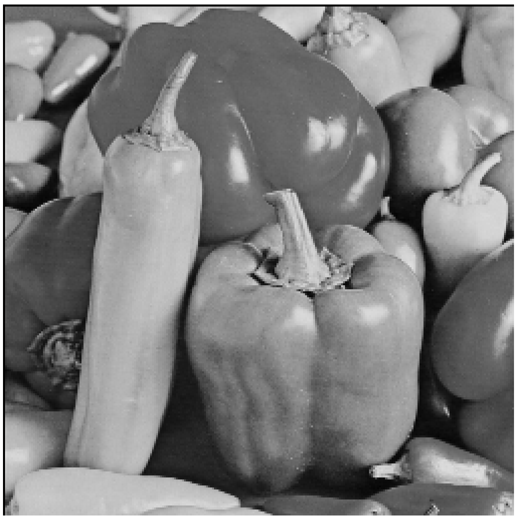
Table 2. Parameters chosen for PSO implementation

Parameter	Value
Swam Size	20
No. of Iterations	100
W_{max}, w_{min}	0.4,0.1
C_1,C_2	2

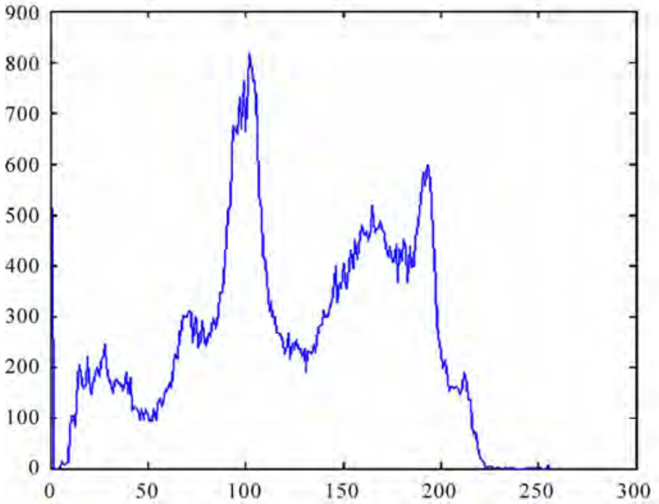
(a)


3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250
300

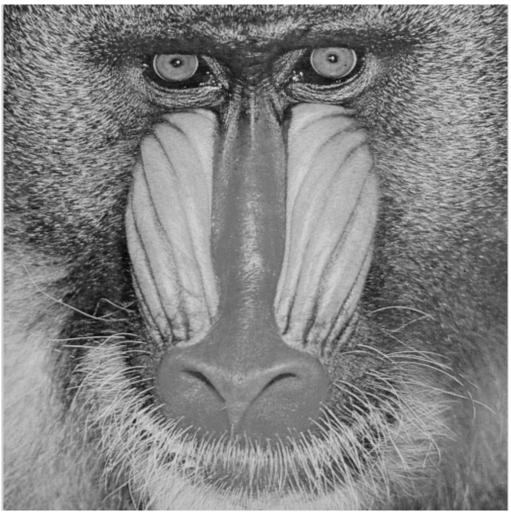
(a')



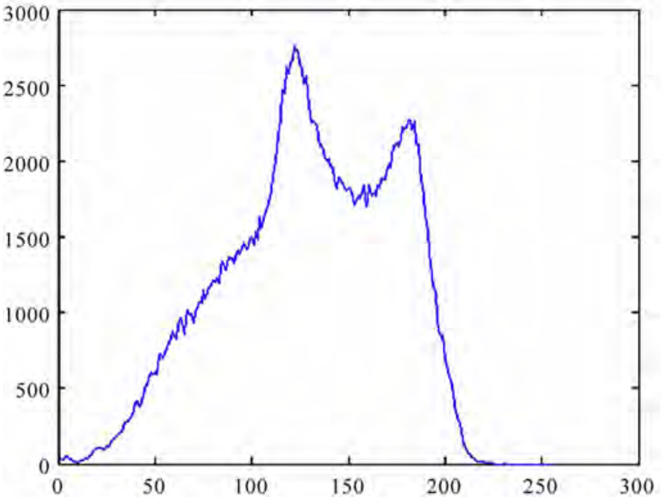
(b)


900
800
700
600
500
400
300
200
100
0
0
50
100
150
200
250
300

(b')

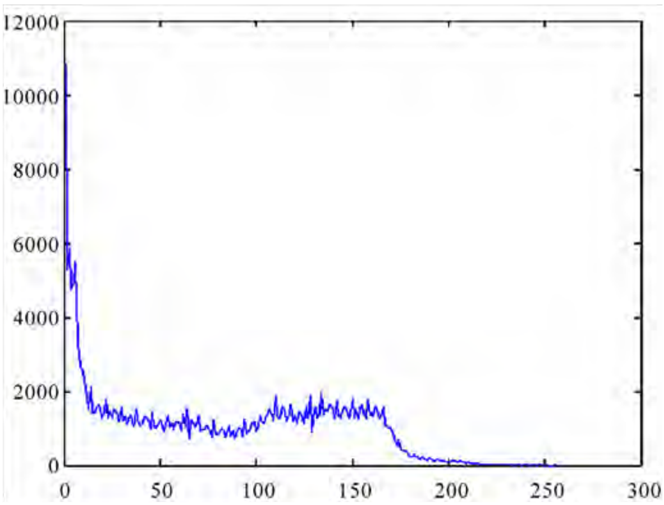


(c)


3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250
300

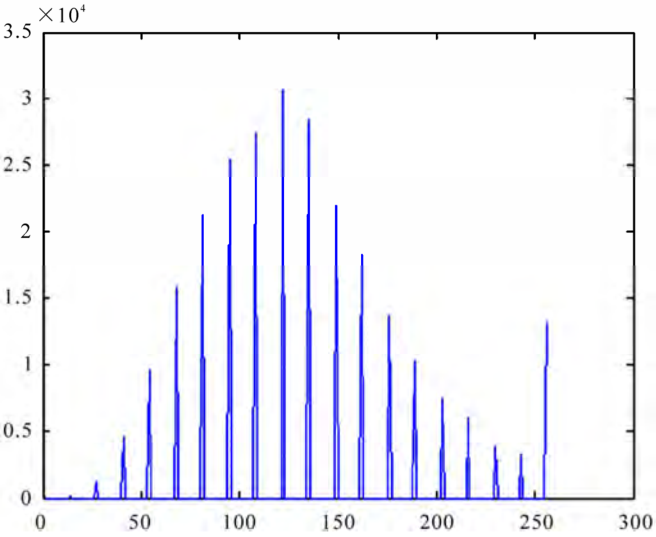
(c')

(d)


12000
10000
8000
6000
4000
2000
0
0
50
100
150
200
250
300

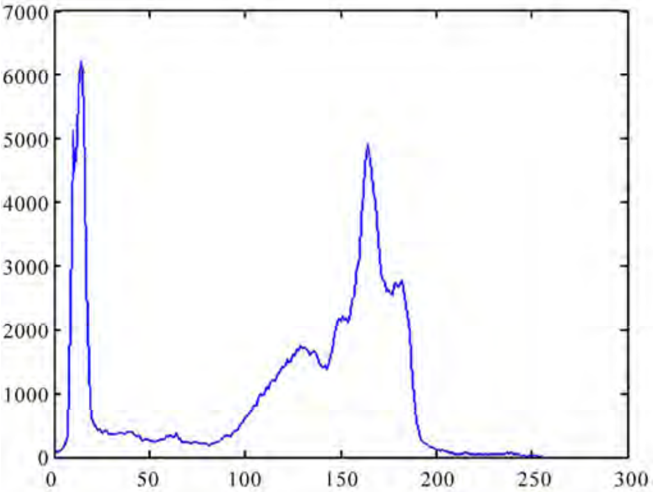
(d’)

(e)

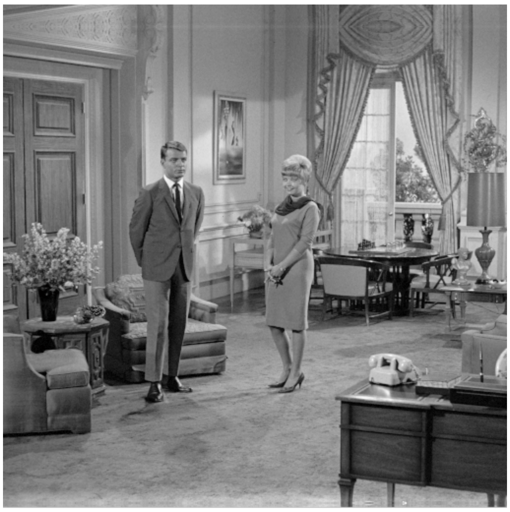

$\times 10^4$
3.5
3
2.5
2
1.5
1
0.5
0
0
50
100
150
200
250
300

(e’)

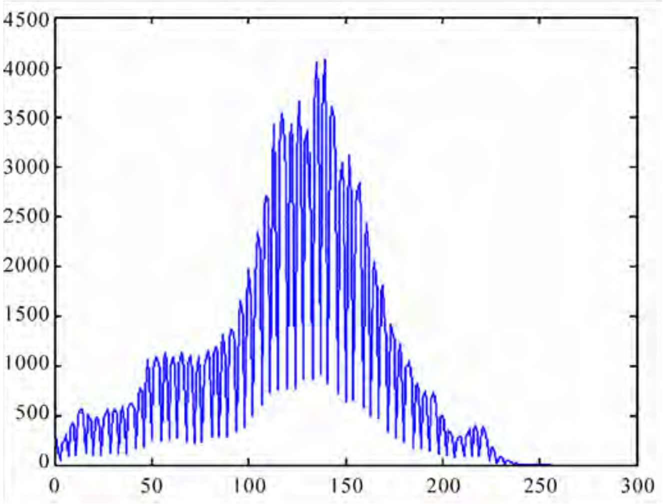
(f)


7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
300

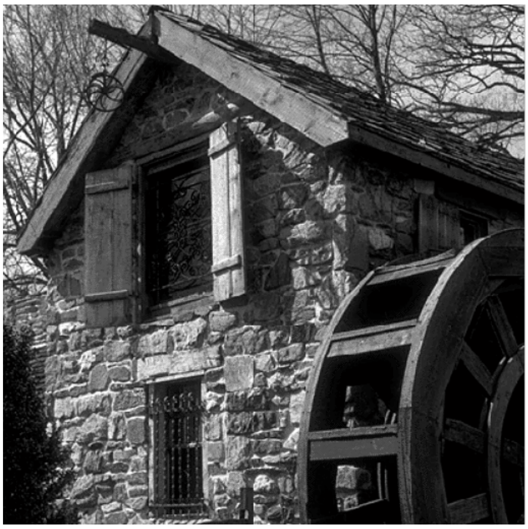
(f’)



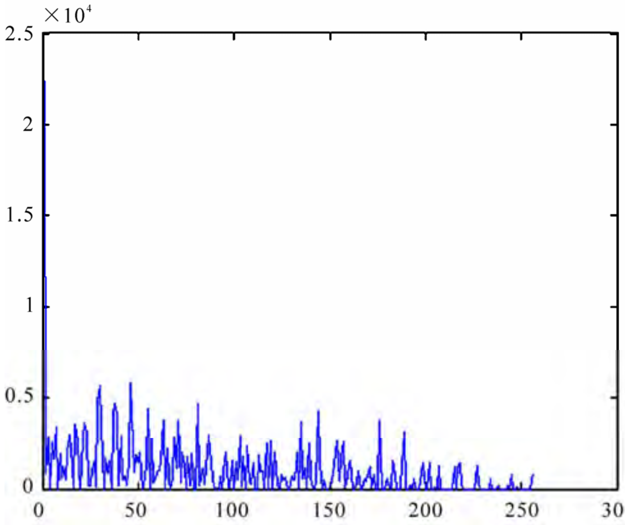
(g)


4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250
300

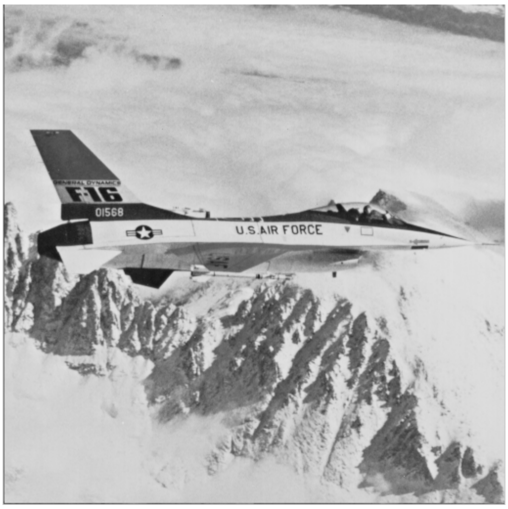
(g')



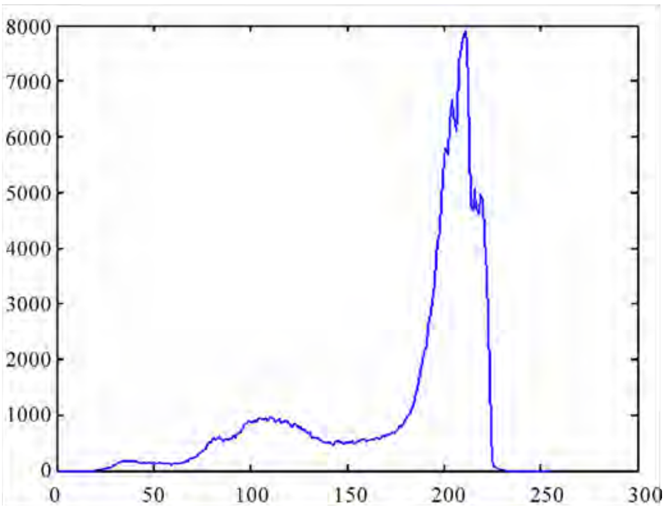
(h)


$\times 10^4$
2.5
2
1.5
1
0.5
0
0
50
100
150
200
250
300

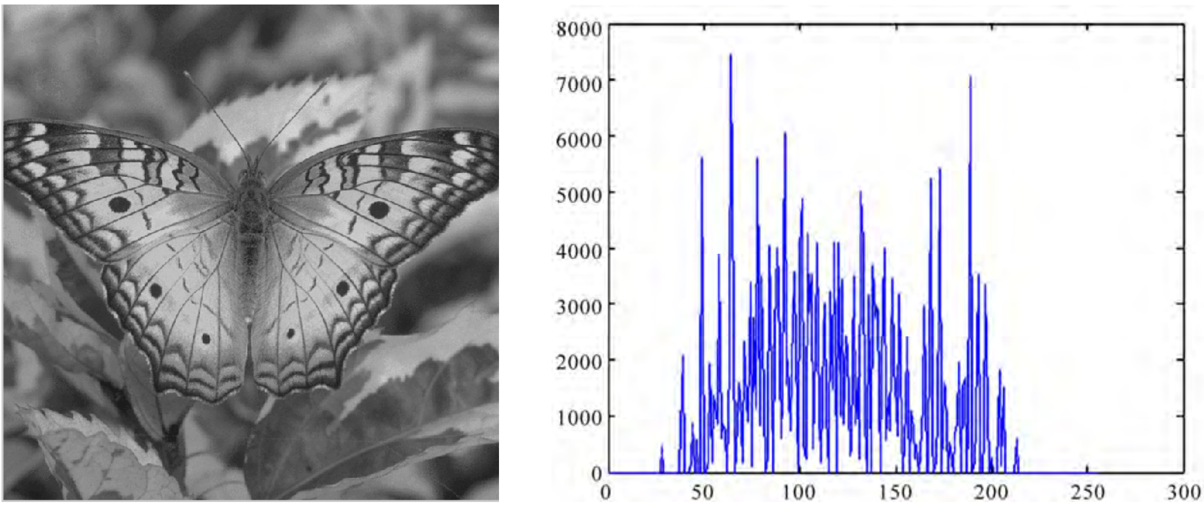
(h')


F16
01568
U.S.AIR FORCE

(i)

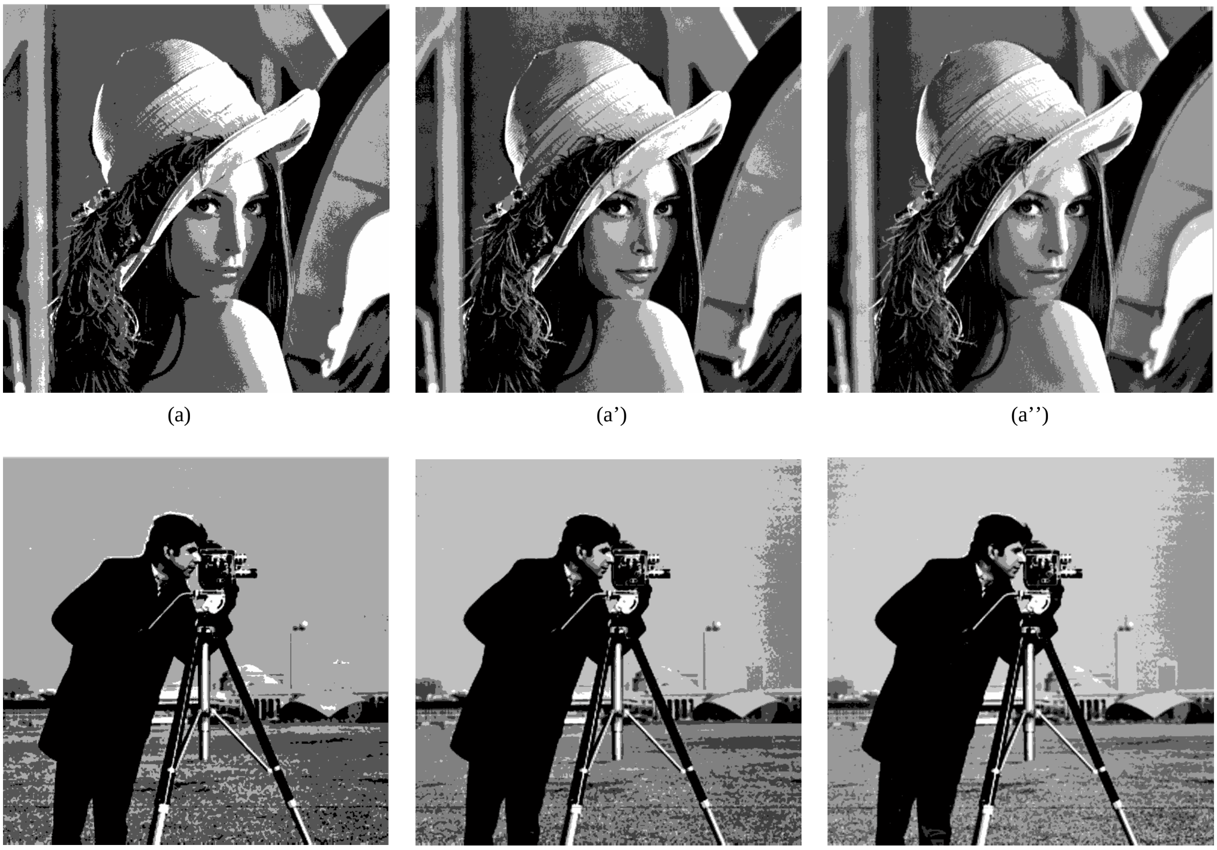

8000
7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
300

(i')



(j) (j’)

Figure 1. Test Images and their histograms (a) Lena, (b) Pepper, (c) Baboon, (d) Hunter, (e) Map, (f) Cameraman, (g) Living room, (h) House,(i) Airplane, (j) Butterfly



(a) (a’) (a’’)

(b) (b’) (b’’)

Figure 2. Thresholded images obtained by Kapur-PSO method ((a), (b) represents 3-level thresholding, (a’), (b’) represents 4-level thresholding, (a’’), (b’’) represents 5-level thresholding)

and Otsu based methods has been evaluated in **Tables 3** and **4**. The tables show the number of thresholds and the optimal threshold values with the corresponding objective value for PSO and GA methods. It is observed from the table that in each case, the PSO could perform well as compared with the GA method. These two methods use

Table 3. Comparison of optimal threshold values and objective values obtained by Kapur method

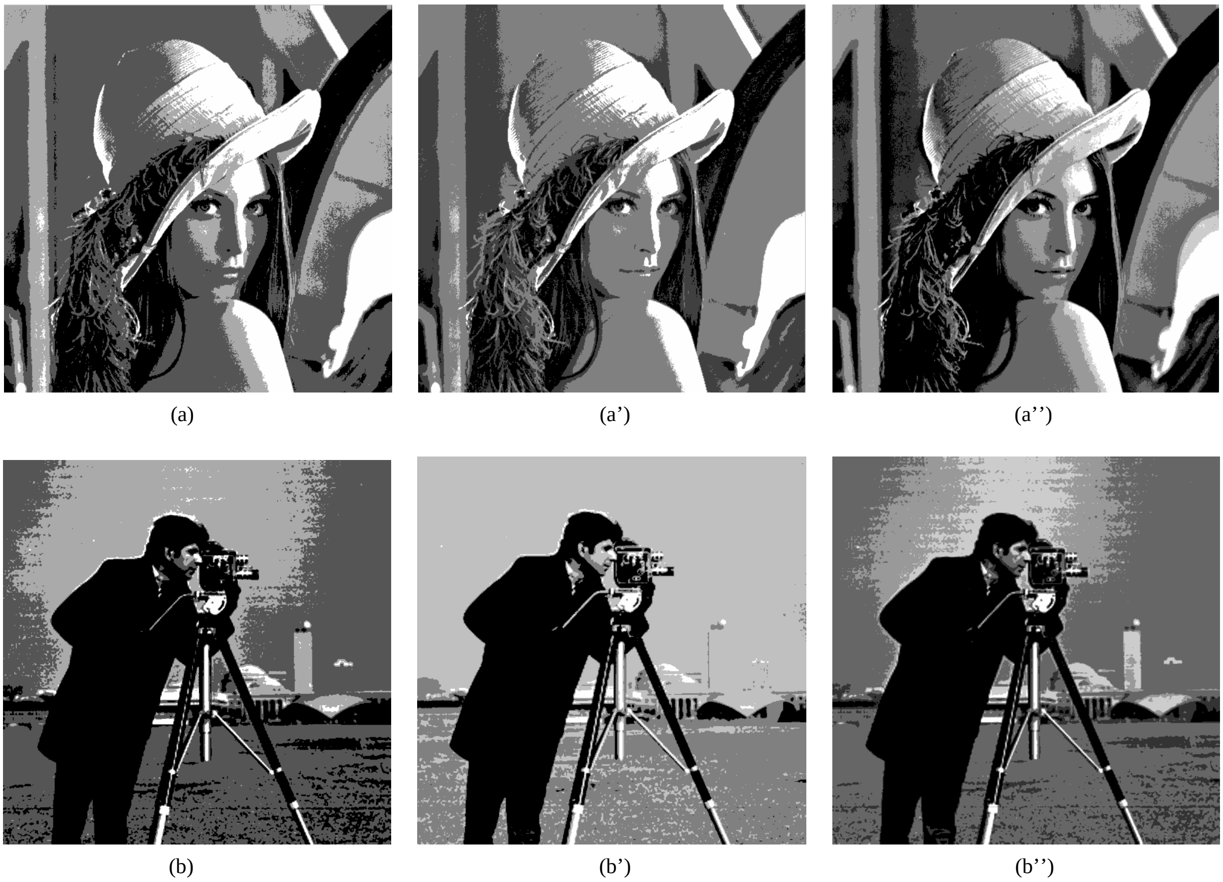
Test Images	m	Optimal threshold values		Objective values	
		PSO	GA	PSO	GA
LENA	2	99,165	104,167	12.3459	12.3344
	3	86,151,180	72,151,180	15.1336	14.9956
	4	92,129,162,191	57,110,178,184	17.8388	17.0892
	5	74,115,145,170,197	96,112,151,186,198	20.4427	19.5492
PEPPER	2	79,146	82,146	12.5168	12.5133
	3	104,141,180	108,127,186	15.0939	14.7122
	4	57,110,162,199	72,102,172,204	18.0974	17.6959
	5	70,116,138,166,200	77,107,124,178,209	20.7338	20.0691
BABOON	2	76,144	93,152	12.2134	12.1847
	3	72,130,181	64,151,181	15.0088	14.7457
	4	65,121,153,180	90,106,152,188	17.5743	16.9356
	5	73,110,142,166,192	96,126,150,172,197	20.2245	19.6622
HUNTER	2	83,179	75,178	12.3708	12.3496
	3	85,128,166	70,148,167	15.1286	14.8381
	4	74,131,174,200	64,100,189,200	18.0401	17.3189
	5	90,120,164,190,219	87,96,128,196,213	20.5339	19.5635
MAP	2	97,181	84,174	4.9789	4.9610
	3	74,140,181	62,94,156	5.5030	5.1351
	4	92,128,152,207	96,113,186,218	5.6903	5.0740
	5	66,109,121,150,195	85,114,159,192,211	5.9165	5.4302
CAMERAMAN	2	115,196	76,195	12.2595	11.9414
	3	96,138,191	111,165,189	15.2110	14.8278
	4	77,116,151,202	71,80,141,192	18.0009	17.1665
	5	64,95,121,156,198	66,110,169,180,209	20.9631	19.7950
LIVINGROOM	2	86,175	84,171	12.4000	12.3923
	3	73,158,187	74,138,160	15.2123	14.9700
	4	59,124,172,202	74,137,164,175	18.1410	17.2063
	5	72,97,119,158,197	60,120,148,155,200	20.6752	19.8410
HOUSE	2	81,144	91,145	10.8321	10.7436
	3	81,116,155	96,134,164	13.1006	12.8473
	4	75,123,154,193	83,135,170,193	15.1027	14.6588
	5	48,97,139,159,189	81,107,132,157,189	17.2517	16.9452
AIRPLANE	2	80,175	90,176	12.1503	12.1153
	3	72,121,191	75,110,199	15.2925	14.8059
	4	74,129,162,188	87,124,154,187	18.0300	17.8923
	5	81,118,144,167,192	95,121,141,151,196	20.3964	19.4465
BUTTERFLY	2	95,141	93,142	10.4743	10.4707
	3	63,126,172	96,103,167	12.3130	11.6280
	4	71,113,162,184	111,149,155,173	14.2317	13.3144
	5	92,116,142,157,182	75,105,140,179,198	16.3374	15.7566

Table 4. Comparison of optimal threshold values and objective values obtained by Otsu method

Test Images	m	Optimal threshold values		Objective values	
		PSO	GA	PSO	GA
LENA	2	94,152	91,149	1961.4149	1960.9603
	3	79,127,170	80,124,173	2127.7771	2126.4107
	4	78,112,134,175	80,126,159,185	2180.6868	2173.7148
	5	79,110,140,167,188	80,116,146,179,213	2212.5555	2196.2745
PEPPER	2	76,144	84,144	2469.5788	2457.1517
	3	72,124,171	65,116,175	2623.2739	2614.0841
	4	57,92,130,172	62,108,142,177	2695.8867	2682.8391
	5	56,84,115,150,179	52,90,128,166,191	2733.5097	2725.8750
BABOON	2	96,149	98,151	1547.9977	1547.6588
	3	85,126,166	86,125,155	1635.3623	1633.5220
	4	79,105,140,174	82,122,146,173	1684.3363	1677.7052
	5	74,104,134,161,180	73,106,140,167,199	1712.9582	1699.3909
HUNTER	2	52,116	51,115	3064.0688	3064.0156
	3	39,86,135	36,89,133	3212.0585	3211.7947
	4	36,84,130,157	39,93,142,163	3257.1767	3231.1313
	5	37,85,125,154,177	39,94,130,169,204	3276.3173	3244.7387
MAP	2	113,177	81,173	2340.3950	2252.3864
	3	81,145,197	83,132,181	2526.3034	2503.7932
	4	92,133,162,206	90,110,158,204	2618.4894	2617.9534
	5	79,116,139,162,204	68,106,138,170,214	2665.4116	2660.8599
CAMERAMAN	2	71,143	72,145	3609.3703	3609.0761
	3	71,134,166	71,143,196	3677.1783	3643.2153
	4	65,121,147,172	59,119,155,203	3722.6447	3710.7311
	5	45,78,121,146,172	51,106,141,167,194	3764.9571	3755.5529
LIVINGROOM	2	88,145	89,155	1627.7966	1627.0537
	3	81,127,165	83,132,174	1757.4664	1748.6885
	4	69,110,143,178	71,116,150,182	1822.1136	1816.0692
	5	56,98,128,156,190	65,104,133,160,189	1865.4766	1858.0959
HOUSE	2	57,127	56,124	3420.9868	3418.4387
	3	48,104,165	50,119,182	3617.9836	3592.1268
	4	40,88,140,194	41,98,149,184	3702.2895	3686.1240
	5	32,74,129,158,188	48,106,136,169,199	3752.1468	3700.3010
AIRPLANE	2	117,174	116,175	1837.7222	1837.7144
	3	99,158,193	86,133,204	1905.7664	1844.5642
	4	84,125,168,201	71,119,164,200	1953.8872	1950.5919
	5	60,101,138,177,204	84,124,164,188,204	1977.9742	1973.0894
BUTTERFLY	2	99,150	100,151	1553.0687	1552.4129
	3	79,119,164	74,115,155	1665.7589	1662.6963
	4	80,113,145,177	82,119,154,184	1702.9069	1696.6940
	5	75,106,129,157,180	77,107,134,171,185	1730.7879	1716.0428

Table 5. Comparison of standard deviation and CPU time (in seconds) for Kapur and Otsu methods

Test Images	m	Standard Deviation				Computation time			
		Kapur method		Otsu method		Kapur method		Otsu method	
		PSO	GA	PSO	GA	PSO	GA	PSO	GA
LENA	2	0.0033	0.0049	0.1423	0.2077	7.8594	8.5469	3.5781	3.9688
	3	0.0390	0.1100	0.4155	0.5555	8.3594	8.8594	4.4031	5.2969
	4	0.1810	0.2594	2.3601	3.0640	9.1719	9.5156	4.7500	5.6094
	5	0.2181	0.3043	4.5341	5.7362	9.4063	10.1250	5.2031	5.8938
PEPPER	2	0.0012	0.0031	0.0956	0.1455	7.1358	8.6492	3.4010	3.8569
	3	0.0764	0.1750	0.1629	0.2891	7.6250	9.1056	4.3125	4.9787
	4	0.1080	0.2707	2.1102	3.9721	8.1254	9.6406	4.6719	5.5156
	5	0.1758	0.3048	3.2057	4.9999	8.4844	9.9688	4.8125	5.9844
BABOON	2	0.0077	0.0567	0.1040	0.2224	8.0016	8.3563	3.8469	4.3969
	3	0.0816	0.1580	0.5720	1.5317	8.7188	9.3750	4.3125	4.7969
	4	0.0853	0.1765	2.1501	3.0653	9.1084	9.6750	4.9063	5.6094
	5	0.1899	0.2775	3.4447	4.6721	9.7813	10.1875	5.3281	6.0109
HUNTER	2	0.0068	0.0148	0.2282	0.3283	8.000	8.6406	3.8438	4.4063
	3	0.0936	0.1741	0.8203	1.8080	8.7031	9.9844	4.4844	4.8625
	4	0.1560	0.2192	2.9836	6.3644	9.0313	9.6219	4.8125	5.3906
	5	0.2720	0.3466	7.3030	11.1247	10.1406	10.6094	5.3031	6.1563
MAP	2	0.0023	0.0030	1.2241	1.8856	6.8906	7.4625	3.6094	4.2000
	3	0.1153	0.1226	1.2298	2.1368	7.1563	7.6563	4.4219	4.9688
	4	0.1366	0.1849	2.2333	4.5790	8.1250	8.9094	4.8750	5.5156
	5	0.1521	0.1901	3.4511	6.3580	8.3594	9.7969	5.7500	6.4188
CAMERAMAN	2	0.1001	0.1270	0.0908	0.3812	8.4844	9.2500	3.4844	3.9531
	3	0.1107	0.2136	6.3502	9.4711	9.0625	9.7000	4.1250	4.8125
	4	0.2005	0.2857	2.4498	4.5059	9.1250	9.9844	4.7406	5.2500
	5	0.2734	0.3528	8.9650	11.0079	10.1094	10.9688	5.2656	6.0025
LIVINGROOM	2	0.0022	0.0039	0.2637	0.5425	7.5844	8.2156	3.3281	3.7656
	3	0.0718	0.1364	1.0446	2.4428	8.7188	9.6250	4.0469	4.9531
	4	0.2286	0.3220	2.0787	3.0313	9.1001	9.7656	4.5000	5.1056
	5	0.2619	0.3805	2.2655	4.3189	10.1719	10.5631	5.7969	6.6094
HOUSE	2	0.0224	0.0637	0.8001	1.7181	7.9063	8.3656	3.6252	4.4313
	3	0.0805	0.1549	3.1018	6.2939	8.2626	9.2500	4.2969	4.9844
	4	0.1324	0.2555	3.7038	8.2156	8.8438	9.5938	4.6094	5.3750
	5	0.1824	0.2696	6.5478	9.9390	9.6406	10.0938	5.7344	6.6963
AIRPLANE	2	0.0106	0.0305	1.1731	2.7001	7.9844	8.7188	3.4688	4.0000
	3	0.1248	0.1958	2.5107	5.0948	8.9688	10.4844	4.5938	5.1875
	4	0.1424	0.3011	3.4728	7.0157	9.2031	9.9531	4.7969	5.3594
	5	0.2760	0.3369	4.7571	8.6500	9.9688	10.4031	5.0781	5.8125
BUTTERFLY	2	0.0025	0.0872	1.6744	2.3493	7.7188	8.4906	3.5313	3.9219
	3	0.1880	0.2021	2.2356	3.4016	8.5469	9.4656	4.1875	4.9531
	4	0.2473	0.2596	4.2227	5.2383	9.0000	9.8659	4.8281	5.5156
	5	0.2821	0.3977	5.1212	6.2719	9.3813	10.2469	5.4594	6.1313


(a) (a') (a'')

(b) (b') (b'')

Figure 3. Thresholded images obtained by Otsu-PSO method ((a), (b) represents 3-level thresholding, (a'), (b') represents 4-level thresholding, (a''), (b'') represents 5-level thresholding)

the objective function to decide whether the number of thresholds has reached the optimal value or not. The higher value of the objective function results in better segmentation.

For a visual interpretation of the segmentation results, the segmented lena and cameraman images for both Kapur-PSO and Otsu-PSO with m = 3, 4 and 5 are presented in **Figures 2** and **3** respectively. It can be easily seen that the quality of segmentation is better, in each case, when m = 5 is chosen.

The standard deviation values and computation time obtained from Kapur and Otsu based evolutionary algorithms are given in **Table 5**. The higher value of standard deviation shows that the results of experiment are unstable. From the tables, it is seen that the PSO method is more stable than the GA method. It is also observed from the table that, even though the Kapur-based method gives lower standard deviation than the Otsu's method, the computation time of Kapur based PSO is higher than the Otsu based PSO.

6. Conclusions

In this paper, particle swarm optimization (PSO) based multilevel thresholding has been presented for image segmentation. In order to verify the efficiency and effectiveness of the proposed PSO approach, ten standard test images are investigated. The performance of this approach has been compared with the GA method, and it is found that PSO outperforms GA approach in terms of solution quality, convergence and robustness. Compared with all the cases, the Kapur-PSO gives lower standard deviation value. Even though the Kapur-PSO gives lower standard deviation, the Otsu-PSO method converges quickly than the Kapur method. Hence, the Otsu-PSO approach is an efficient tool for finding optimized threshold values.

REFERENCES

[1] J. K. Udupa and S. Samarasekera, "Fuzzy Connectedness and Object Definition: Theory, Algorithms and Applications in Image Segmentation," *Graphical Models and Image Processing*, Vol. 58, No. 3, 1996, pp. 246-261.

[2] P. K. Sahoo, S. Soltani and A. K. C. Wong, "A Survey of Thresholding Techniques," *Computer Vision, Graphics and Image Processing*, Vol. 41, No. 2, 1998, pp. 233-260.

[3] N. P. Pal and S. K. Pal, "A Review on Image Segmenta-

tion Techniques," *Pattern Recognition*, Vol. 26, No. 9, 1993, pp. 1277-1294.

[4] M. Sezgin and B. Sankar, "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation," *Journal of Electronic Imaging*, Vol. 13, No. 1, 2004, pp. 146-165.

[5] S. U. Lee, S. Y. Chung and R. H. Park, "A Comparative Performance Study of Several Global Thresholding Techniques for Segmentation," *Computer Vision*, *Graphics and Image Processing*, Vol. 52, No. 2, 1990, pp. 171-190.

[6] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transaction on Systems, Man and Cybernetics*, Vol. 9, No. 1, 1979, pp. 62-66.

[7] W. Tao, H. Jin and L. Liu, "Object Segmentation Using Ant Colony Optimization Algorithm and Fuzzy Entropy," *Pattern Recognition Letters*, Vol. 28, No. 7, 2007, pp. 788-796.

[8] O. J. Tobias and R. Seara, "Image Segmentation by Histogram Thresholding Using Fuzzy Sets," *IEEE Transaction on Image Processing*, Vol. 11, No. 12, 2002, pp. 1457-1465.

[9] J. N. Kapur, P. K. Sahoo and A. K. C. Wong, "A New Method for Gray-Level Picture Thresholding Using the Entropy of The Histogram," *Computer Vision*, *Graphics and Image Processing*, Vol. 29, No. 3, 1985, pp. 273-285.

[10] T. Pun, "Entropy Thresholding: A New Approach," *Computer Vision*, *Graphics and Image Processing*, Vol. 16, No. 3, 1981, pp. 210-239.

[11] A. D. Brink, "Minimum Spatial Entropy Threshold Selection," *IEEE Proceedings*, *Vision Image and Signal Processing*, Vol. 142, No. 3, 1995, pp. 128-132.

[12] X. Li, Z. Zhao and H. D. Cheng, "Fuzzy Entropy Threshold Approach to Breast Cancer Detection," *Information Sciences*, Vol. 4, No. 1, 1995, pp.49-56.

[13] L. K. Huang and M. J. Wang, "Image Thresholding by Minimizing the Measure of Fuzziness," *Pattern Recognition*, Vol. 28, No. 1, 1995, pp. 41-51.

[14] J. Kittler and J. Illingworth, "Minimum Error Thresholding," *Pattern Recognition*, Vol. 19, No. 1, 1986, pp. 41-47.

[15] Q. Ye and P. Danielsson, "On Minimum Error Thresholding and its Implementation," *Pattern Recognition Letters*, Vol. 7, No. 4, 1988, pp. 201-206.

[16] U. Gonzales-Baron and F. Butler, "A Comparison of Seven Thresholding Techniques with the K-Means Clustering Algorithm for Measurement of Bread-Crumb Features by Digital Image Analysis," *Journal of Food Engineering*, Vol. 74, No. 2, 2006, pp. 268-278.

[17] P. Y. Yin and L. H. Chen, "A Fast Iterative Scheme For Multilevel Thresholding Methods," *Signal Processing*, Vol. 60, No. 3, 1997, pp. 305-313.

[18] P. S. T. Liao, S. Chen and P. C. Chung, "A Fast Algorithm for Multilevel Thresholding," *Journal of Information Science and Engineering*, Vol. 17, No. 5, 2001, pp. 713-727.

[19] K. C. Lin, "Fast Image Thresholding by Finding Zero(S) of the First Derivative of between Class Variance," *Machine Vision and Applications*, Vol. 13, No. 5-6, 2003, pp. 254-262.

[20] P.-Y. Yin and L.-H. Chen, "A Fast Iterative Scheme for Multilevel Thresholding Methods," *Signal Processing*, Vol. 60, No. 3, 1997, pp. 305-313.

[21] E. Cuevas, D. Zaldivar and M. Perez-Cisneros, "A Novel Multi-Threshold Segmentation Approach Based on Differential Evolution Optimization," Expert Systems with Applications, Vol. 37, No. 7, 2010, pp. 5265-5271.

[22] W. B. Tao, H. Jin and L. M. Liu, "Object Segmentation Using Ant Colony Optimization Algorithm and Fuzzy Entropy," *Pattern Recognition Letters*, Vol. 28, No. 7, 2008, pp. 788-796.

[23] P. Y. Yin, "A Fast Scheme for Optimal Thresholding Using Genetic Algorithms," *Signal Processing*, Vol. 72, No. 2, 1999, pp. 85-95.

[24] C. C. Lai and D. C. Tseng, "A Hybrid Approach Using Gaussian Smoothing and Genetic Algorithm for Multilevel Thresholding," *International Journal of Hybrid Intelligent Systems*, Vol. 1, No. 3, 2004, pp. 143-152.

[25] D. B. Fogel, "Evolutionary Computation: Toward a New Philosophy of Machine Intelligence," 2nd Edition, IEEE Press, Piscataway, 2000.

[26] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Proceedings of the IEEE Conference on Neural Networks—ICNN*'95, Perth, Vol. 4, 1995, pp. 1942-1948.

[27] Y.-T. Kao, E. Zahara and I-W. Kao, "A Hybridized Approach to Data Clustering," *Expert Systems with Applications*, Vol. 34, No. 3, 2008, pp. 1754-1762.

[28] Z.-J. Lee, S.-W. Lin, S.-F. Su and C.-Y. Lin, "A Hybrid Watermarking Technique Applied to Digital Images," *Expert Systems with Applications*, Vol. 8, No. 1, 2008, pp. 789-808.

[29] C.-C. Tseng, J.-G. Hsieh and J.-H. Jeng, "Fractal Image Compression Using Visual-Based Particle Swarm Optimization," *Image and Vision Computing*, Vol. 26, No. 8, 2008, pp. 1154-1162.

J. Intelligent Learning Systems & Applications, **2010, 2, 139-146**
doi:10.4236/jilsa.2010.23017 Published Online August 2010 (http://www.SciRP.org/journal/jilsa)

Chunk Parsing and Entity Relation Extracting to Chinese Text by Using Conditional Random Fields Model

Junhua Wu, Longxia Liu

College of Electronics and Information Engineering, Nanjing University of Technology, Nanjing, China.
Email: wujh@njut.edu.cn

Received March 19th, 2010; revised July 20th, 2010; accepted July 30th, 2010.

ABSTRACT

Currently, large amounts of information exist in Web sites and various digital media. Most of them are in natural language. They are easy to be browsed, but difficult to be understood by computer. Chunk parsing and entity relation extracting is important work to understanding information semantic in natural language processing. Chunk analysis is a shallow parsing method, and entity relation extraction is used in establishing relationship between entities. Because full syntax parsing is complexity in Chinese text understanding, many researchers is more interesting in chunk analysis and relation extraction. Conditional random fields (CRFs) model is the valid probabilistic model to segment and label sequence data. This paper models chunk and entity relation problems in Chinese text. By transforming them into label solution we can use CRFs to realize the chunk analysis and entities relation extraction.

Keywords: *Information Extraction, Chunk Parsing, Entity Relation Extraction*

1. Introduction

At present, information is presented in various digital media. Many of them are organized in natural language, such as information in Web pages, text document in digital library etc. They are non structural or semi-structural and difficult to understand by computer. Further processing to the information is blocked. It makes large amounts of information wasted. So research on semantic Web, natural language understanding is developed in order to structure and retrieve information from Web pages or other natural language documents. And information extraction is important task in the work.

Information extraction is a process to retrieve information from large text set. It may be concerned with identifying named entity, extracting relationship and label properties of sentence etc. It is a subfield of natural language understanding. There are some methods for information extraction including methods based on rules [1,2] and statistical model [3-6].

Chunk analysis and relation extraction play the important roles in information extraction. It is a simplified syntax paring technology to define and label chunk based on syntax and semantics [7]. Comparing with full parsing this method only identifies the partial structure in a sentence, such as noun phrase or verb phrase. Through which, the simple syntax parsing can be implemented and information extraction may be more effective and simple.

The objective of entity relation extraction is identifying the relationship between entities in text. Miller et al. considered the problem of relation extraction in the context of natural language parsing and augmented syntactic parses with semantic relation-specific attributes [8]. It will be critical in events detecting and describing for research on information extraction. Entity relation may be explicit and implicit. Some encountered problems make studying on entities relation hard such as few dataset, difficult extraction of implicit relation and immature parsing to Chinese.

Conditional random fields model is a valid probabilistic model to segment and label sequence data [9]. In Chinese understanding, some research use CRFs in Chinese part-of-speech and word segmentation [10,11], but seldom in chunk parsing and entity relation extraction.

Compared with other statistical model CRFs can represent long-range dependences and multiple interacting features. Our innovation is that we analyze Chinese characteristics and then model chunk and entity relation problems as label problem. Moreover using CRFs realizes the chunk analysis and relation extraction.

2. Related Work

A number of approaches currently have been used for natural language tasks as part of speech tagging and entity extraction. They are usually based on rules or statistic models.

Text chunk divides a text in syntactically correlated parts of words. Steven introduced chunks [12] firstly. Many machine learning approaches, such as Memory-based Learning (MBL) [13], Transformation-based Learning (TBL) [14], and Hidden Markov Models (HMMs), have been applied to text chunking [15] for parsing.

Named entity is important linguistic unit. So there are many works such as named entity recognition, disambiguation, and relationship extraction on it [16-20]. The problem of relation extraction is starting to be addressed within the natural language processing and machine learning communities. Since it is proposed, many methods have been suggested. Methods based on knowledge base were used in decision of relation extraction firstly. But it is difficult to construct knowledge base. Therefore some methods based on machine learning were emerged, such as feature-based [16], kernel-based [17] method. Approach kernel-based is a valid one for relation extracting, but its training and testing time is long for large amounts of data.

2.1 Model for Information Extraction

A lot of research to information extraction is based on machine learning methods using statistic model because by the model sentence can be segmented and labeled. Statistical language model is a probability model which estimate probability of expected text sequence by computing probability. These models are concerned with Hidden Markov Model (HMM), Maximum Entropy Model (ME), Maximum Entropy Markov Model (MEMM) and conditional random fields model CRFs. Our method is also based on statistic model.

Hidden Markov models (HMMs) are a powerful probabilistic tool for modeling sequential data, and have been applied with success to many text-related tasks, such as part-of-speech tagging, text segmentation and information extraction [6]. HMM can be considered as a finite state machine that presents states and transition chains of an application. The model is built either by manual or training. Usually extracting text information is concerned with training and labeling. Maximum likelihood and Baum-Welch algorithm are used to learning sample data labeled or unlabeled. And then Viterbi algorithm is used to label state sequence with maximum probability in text needed processing.

HMM is easy to build. It needn’t large dictionary or rule sets with well flexibilities. There are many improved HMM model and their application in information extraction. Freitag and McCallum’s paper [3] uses stochastic optimization to search the fittest HMM. Souyma Ray and Mark Crave [4] choose HMM to represent sentence structure. Scheffer T, Decomain C and Wrobel S [5] proposes a method which uses active learning to minimize the label data for HMM training. But HMM is a generative model and independent hypothesis is needed, so it will ignore the context of information and lead to an unexpected result.

Maximum Entropy (ME) method [21] converts the sequence label into data classifying. Its principle can be stated as follows [22]:

1) Reformulate the different information sources as constraints to be satisfied by the target (combined) estimate.

2) Among all probability distributions that satisfy these constraints, choose the one that has the highest entropy.

The advantage of ME is [21]: It makes the least assumptions about the distribution being modeled other than those imposed by constraints and given by the prior information. The framework is completely general in that almost any consistent piece of probabilistic information can be formulated a constraint. Moreover, if the constraints are consistent, that is there exists a probability function which satisfies them, then amongst all probability functions which satisfy the constraints, there is a unique maximum entropy.

This ME method will lost sequence properties. So a model combining ME and MM (Markov Model) is emerged, that is MEMM [6].

In MEMM, the HMM transition and observation functions are replaced by a single function $P(s|s',o)$ that provides the probability of the current state s given the previous state s' and the current observation o. In this model, as in most applications of HMMs, the observations are given—reflecting the fact that we don’t actually care about their probability, only the probability of the state sequence (and hence label sequence) they induce.

Conditional probability of transition between states is introduced in MEMM, which makes the arbitrary choice of properties possible. But MEMM is partial model which needs normalization for each node. Therefore only a localized optimization value is obtained. Also the problem named length bias and label bias [9] may be caused. It means the method will ignore those not in training dataset.

2.2 Label Bias

Classical discriminative Markov models, maximum entropy taggers (Ratnaparkhi, 1996), and MEMMs, as well as non-probabilistic sequence tagging and segmentation models with independently trained next-state classifiers are all potential victims of the label bias problem [9].

Consider a MEMM model shown in **Figure 1** which is a finite-state acceptor for shallow parsing of two sentences:

The robot wheels Fred round.

The robot wheels are round.

Here [B-NP] etc. are labels for sentence. NP, VP, ADJP and PP mean Noun Phrase, Verb Phrase, Adjective Phrase and Prep Phrase. B or I stand for word location, begin or inter of a phrase.

It is obvious that sum of transition probability is 1 from a state i to other adjacent states. Because there is only one transition in state 3 and 7, while current state and observed value *Fred* are specified, conditional probability of next state is:

$$p(4\,|\,3, Fred) = p(8\,|\,7, Fred) = 1$$

But this equation will face to some problems if there isn't existing a transition from state 7 to state 8 while observe value is *Fred* in training dataset. Generally a low probability is specified if an unknown event exists in training dataset. But for state with single output, the follow equation have to be given:

$$\sum_{\substack{allstates\\ fromstate7to}} p(s\,|\,7, Fred) = 1$$

It means that the observed value *Fred* is ignored. This will result in that label sequence is not related to observed sequence. That is label bias.

Proper solutions require models that account for whole state sequences at once by letting some transitions "vote" more strongly than others depending on the corresponding observations [9].

Lafferty suggests a global model CRFs that can solve the problems discussed before. Instead of local normalizing CRFs can realize global processing, so a global optimization value will be produced. CRFs is a new graph model of probability which can represent the long-range dependences and multiple interacting features. Domain knowledge is represented conveniently by the model. McCallum use this model to process named entity recognition [23]. His experiments shows F value is 84.04% while processing English, F value is 68.11% while processing German. Hong mingcai uses CRFs to label Chinese part-of speech [11]. But information extraction of Chinese is still a difficult task presented in many subfields such as chunk analysis and entity relation extraction. So this paper explores the methods about chunk analysis and entity relation extraction to Chinese text based on CRFs.

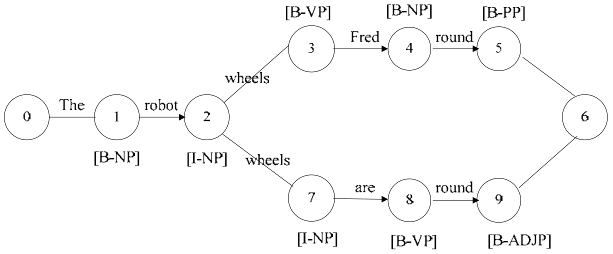


Figure 1. Finite-state acceptor for shallow parsing of two sentences

3. Conditional Random Fields (CRFs) Model

Conditional random fields model is a probabilistic model to segment and label sequence data based on statistic. It is a non-directional graph model that can compute conditional probability of output sequence when conditioned on input sequence of model.

Definition 1 [9]. Let $G = (V,E)$ be a graph such that $Y = (Y_v)\ v \in 2V$, so that Y is indexed by the vertices of G. Then (X,Y) is a conditional random field in case, when conditioned on X, the random variables Y_v obey the Markov property with respect to the graph: $P(Y_v\,|X, Y_w, w \neq v) = P(Y_v\,|X, Y_w, w \sim v)$, where $w \sim v$ means that w and v are neighbors in G.

CRFs is a random field globally conditioned on the observation X. if $X = \{x_1, x_2, \dots x_n\}$ is specified as data sequence needed label then $Y = \{y_1, y_2, \dots y_n\}$is the result data which have been segmented or labeled by the model. The model computes the joint distribution over the label sequence Y given X instead of only defining next state in terms of current state.

The conditional probability of label sequence Y depends on the global interactional features with different weight.

Assume $\Lambda = \{\lambda_1, \dots, \lambda_k\}$ is a vector of features, conditional probability, for a given X, $P_\Lambda(Y|X)$ is defined as follow:

$$p \wedge (Y\,|\,X) = \frac{1}{Z_X} \exp\left\{ \sum_{t=1}^{T} \sum_{k} \lambda_k f_k\left(y_{t-1}, y_t, X, t\right) \right\} \quad (1)$$

$$Z_X = \sum_{Y} \exp\left\{ \sum_{t=1}^{T} \sum_{k} \lambda_k f_k\left(y_{t-1}, y_t, X, t\right) \right\} \quad (2)$$

Z_x is a normalized value that makes the total probability of all state sequence is 1 for given X. $f_k(y_{t-1}, y_t, X, t)$ is a feature function to mark the feature at position t and t-1 for observed X. Its value is between 0 and 1. $\Lambda = \{\lambda_1, \dots, \lambda_k\}$ is corresponding to the context of data sequence and is a weight set of $f_k(y_{t-1}, y_t, X, t)$.

If we want to use the CRFs model to obtain expected result the critical task is training model. A model trained can produce optimization $P(Y|X)$, that is $Y^* = \arg\max_Y p(Y\,|\,X)$. It also means $\Lambda = \{\lambda_1, \dots, \lambda_k\}$ will be determined. Training may use log-likehood algorithm that is independent of applications.

In this paper chunk analysis and entity relation extraction will be converted into the label solution. Data sequence X is made up of some words. For each word W_0 there are some words ahead or back of it. It is represented $W = \{W_{-n}, \ldots W_{-1}, W_0, W_{+1}, \ldots W_{+n}\}$. W_{-n}, stands for nth word previous to W_0 and W_{+n} is the nth one following W_0. $\Lambda = \left\{\lambda_1, ..., \lambda_k\right\}$, in model, can be thought as feature weights related to $W = \{W_{-k}, \ldots W_{-1}, W_0, W_{+1}, \ldots W_{+k}\}$. Each λ is specified in model after training, and label sequence can be produced by Viterbi algorithm through running the model .

4. Chunk Analysis Based on CRFs

Chunk is firstly proposed by Abney [12]. He thinks chunk is the syntax element between word and sentence and with non-recursive properties.

Chunk analysis is partial parsing, also named shallow parsing, relative to complete parsing with simplified policy [15]. It is a new technology of natural language processing. Full parsing can produce a complete parse tree finally by series analysis process to sentence, which needs large cost. But chunk analysis only needs to identify some structures of the sentences such as non-recursive noun phrase, verb phrases etc, called chunk. By dividing sentence into different chunks in syntax or semantics and labeling chunks we can improve the efficiency of information extraction. It is a policy between lexical analysis and syntax analysis. Chunk partitioning and identifying are completed by chunk parsing in natural language processing.

4.1 The Definition and Label of Chunks

Definition 2. Chunk is a structure that is non-recursive phrase meet syntax. Each chunk has a head word and begins or ends at this word.

Non-recursive phrase means nested structure not exist. That is, all chunks are the same level.

Conference on Computational Natural Language Learning (CoNLL-2000) developed a dataset of English chunk which provided a platform to evaluating and test chunk analysis algorithms. There are 11 types chunks defined. They are NP, VP, ADVP, ADJP, PP, SBAR, CONJP, PRT, INTJ, LST, UCP [24].

Most of Chinese chunks present the same properties compared with English. But there is some difference. By analyzing the properties of Chinese we defined some chunk types: noun chunk(NP), verb chunk(VP), adjective chunk(AP), adverb chunk(DP), preposition chunk(SP), time chunk(TP), quantifier chunk(MP), conjunction chunk (CONJP) and other chunk(UCP).

In fact chunk analysis based on CRFs has become a process of labeling chunk like tagging part-of speech. Generally there are two kinds of standard method to label: Inside/Outside and Start/End methods. Inside/Outside policy, named IOB1, uses tag set {I,O,B} [25] to label internal, outside and first word of a chunk. Combining it with chunk type we will have chunk labeled. Such as B-VP, it shows that is a first word of a verb chunk. O means the word doesn’t belong to any chunk. Start/End method, named IOBES, uses tag set {I,O,B,E,S}. When chunk only includes one word, S tag is used. E labels the last word of a chunk. Other tags are the same as Inside/Outside. For example S-NP means a chunk is constructed by one word. **Table 1** presents the label chunks of a sentence. The first column of table is Chinese words and the second column is corresponding to English for reader understanding. Next two columns are notations used IOB1 and IOBES.

In the table, row 4-6 represent a verb chunk which consist of three Chinese words labeled B-VP, I-VP and E-VP if use IOBES method. By these label chunk analysis is considered as chunk label which can be implemented by training CRFs model.

4.2 Model Training

CRFs model must be trained using labeled dataset to determine the model parameters. That trained model can be used to realize processing text which expects to be segmented and labeled. If X is sentences that have been labeled and Y is corresponding label sequence of chunk CRFs model training will make label sequence $Y^* = \arg\max_Y p(Y \mid X)$ optimal.

Here we use CRF++0.50 as training and testing tool. CRF++0.50 is a string learning tool based on CRFs principle. The training sample file and feature template file are needed in training process. Training will result in a CRfs model which will be used in labeling chunk to Chinese text.

Table 1. An example of label chunks

Chinese	English	IOB1	IOBES
因而	So	I-CONJP	S-CONJP
我们	we	B-NP	S-NP
可能	may	B-VP	B-VP
会	be	I-VP	I-VP
面临	Face to	I-VP	E-VP
一个	a	B-MP	S-MP
不	un	B-NP	B-NP
稳定	stable	I-NP	I-NP
时期	period	I-NP	E-NP
。	.	O	O

The training sample file is made up of some blocks and each block represents a sentence. The block form of training sample file is presented in **Table 2**.

There is a blank row between blocks. Each block includes some tokens and each token is a label word in one row. First column is the Chinese word and next column is the English word to help understanding. Third column lists the properties of the word (may be more than one column). Last column is the tag notations.

In section 3 we know some feature weights used in representing context of a word W_0. So it is important to select feature set. Generally context of a word and their properties are very useful for decision of feature. That means we can use some words which are previous or succeed to word W_0 as features. Features may be N-gram. Features {…W_{-2}, W_{-1}, W_0, W_1, W_2…} is named Uni-gram basic features and {…W_{-2}, W_{-1}, W_{-1}, W_0, W_0, W_1, W_1, W_2…} is named Bi-gram basic features. Here W_n stands for a word. In addition, advanced features {… WP_{-2}, WP_{-1}, WP_0, WP_1, WP_2 …} which combine the word and its property together are also used to improve result of analysis. *P* means property of a word. **Table 3** shows various features of word "赤字" (deficit) in **Table 2**. The last column of **Table 3** is English word corresponding to Chinese word.

So an observing window of token W_0 need to be given for training. The window includes W_0 and some words before and after it, that is $W = \{ W_{-n}, W_{-(n-1)}, \ldots, W_0, \ldots, W_{n-1}, W_n\}$. **Table 4** is an example about observing window. It is used as feature source for training vector $\Lambda = \{\lambda_1, ..., \lambda_k\}$. Larger window provides more context feature, but it will increase the cost of processing. Too small window may lose important features. So we define windows size as 5, that is $W = \{W_{-2}, W_{-1}, W_0, W_1, W_2\}$.

Table 2. Form of experiment dataset sample

Chinese token	English token	Property	Notation
他	He	PRP	B-NP
认为	reckons	VBZ	B-VP
当前的	current	JJ	I-NP
赤字	deficit	NN	I-NP
将	will	MD	B-VP
缩小	narrow	VB	I-VP
到	to	TO	B-PP
仅	only	RB	B-NP
1800	18000	CD	I-NP
万	thousands	CD	I-NP
9 月	september	NNP	B-NP
.		.	O

Table 3. Feature instance

Feature	Feature item	Feature value	Value in English
Uni-gram basic features	W_{-2}	认为	reckons
	W_{-1}	当前的	current
	W_0	赤字	deficit
	W_1	将	will
	W_2	缩小	narrow
Bi-gram basic features	$W_{-1}W_0$	当前的/赤字	current/ deficit
	W_0W_1	赤字/将	deficit/will
Uni-gram advanced features	WP_{-1},	当前的/JJ	current/JJ
	WP_1	将/MD	Will/MD

Table 4. Observing window of features

Feature position	Description	Chinese example
$W = W_0$	Token	当前的
$W = W_{-1}$	Last word of token	认为
$W = W_{+1}$	Next word of token	赤字
$W = W_0W_{+1}$	Token and next word	当前的 赤字
$W = W_{-1}W_0W_{+1}$	Last word, token and next word	认为 当前的 赤字

The template file of features defines the feature item for training. After training $\Lambda = \{\lambda_1, ..., \lambda_k\}$ is produced, that is CRFs model has been available.

5. Entity Relation Extraction

Entity is the basic element in natural text, such as place, role, organization, thing etc. Entities play important roles in natural language text. Generally there are some relationships between them. Such as locating, belong to, adjacent and so on. These relationships may be explicit or implicit. Implicit relationship needs reasoning by knowledge. Entity relation extraction is the process of identifying the relationship between entities in text and labeling them. It is not only an important work in information extraction but also useful in automatic answer or semantic network.

Testing from MUC shows that many systems are able to process named entity to large of English document [26]. But entity relation extraction to Chinese may be difficult. As we known machine learning is the valid method for extracting, but it needs dataset labeled. Currently, "People's Daily" labeled by Beijing university is perhaps a better choice. This dataset has been labeled in part-of speech, properties of word, named entity. But extracting implicit relationship needs more external knowledge.

5.1 The Definition and Label of Entity Relation

ACE 2006 defines six classes and 18 subclasses relationships between two entities. They are shown in **Table 5**. The dataset provided by ACE covers English, Chinese and Arabie.

This paper focuses on Chinese entity relation. Here we only illustrate two kinds of relationship of physical class, and only implement extraction based on the definition of these two relationships.

Definition 3: Relation 1 (M, N) is defined as located relationship. With Entity M, N ∈ geographical entity and N ∈ M.

Definition 4: Relation 2(M, N) is defined as near relationship. With M, N ∈ geographical entity and 2(M, N) = 2 (N, M).

In this paper the same principle as chunk analysis is used to realize extraction which training CRFs model through label sample dataset. Then using CRfs model realizes the extraction. Here we suggest nine kinds of notation to label the position relationship in dataset. **Table 6** presents these nine notations.

In the table each row presents one or two entities and their relationship. For a notation 1-B, 1 stands for Relation 1 (M, N) and B stands for the first entity M. The third column lists the sentence including entities. The end column shows the instance corresponding to notation.

5.2 Experiment of Entity Relation Extraction

Experiment is designed based on the definition of Relation 1 (M,N) and Relation 2 (M,N). The paper uses label dataset of “people’s daily” on January 1998 as sample dataset named M, size 3.2 MB. It is divided into 10 subsets from M1 to M10. M1 includes text on Jan 1st and M2 ranges from 1st to 2nd. By the way M10 ranges from 1st to 10th. That is Mn means newspaper contents of *n* days.

M1-M9 is considered as training dataset and M10 as test set. **Table 7** is the dataset and result of experiment.

In the table Recall R, Precision P and F-Score are metrics to information extraction. Let r_1 be numbers of relationships extracted correctly, r_2 as numbers of relationships extracted actually, r_3 as numbers of original relationships in text. Then:

Table 5. Named entity relation types and instances

Class	Subclass
Physical	Located, Near
Part-Whole	Artifact, Geographical, Subsidiary
Personal-Social	Business, Family, Lasting-Personal
ORG-Affiliation	Employment, Founder, Ownership, Student-Alum, Sports-Affiliation, Investor-Shareholder, Membership
Agent-Artifact	User-Owner-Investor-Manufacturer
General-Affiliation	Citizen-Resident-Religion-Ethnicity, Org-Location

Table 6. Nine kinds of labeling and instances

Notation	Description	Sentences for example	Instance
1-B	Entity M in Relation1(M,N)	中国首都北京 (Beijing of china capital)	中国 (China)
1-E	Entity N in Relation1(M,N)	中国首都北京 (Beijing of china capital)	北京 (Beijing)
2	Entity M,N in Relation 2(M,N)	城市北京和天津 (City Beijing and Tianjin)	北京 ,天津 (Beijing,Tianjin)
1-E-1-B	Entity N in Relation 1(M,N), 1(N,S)	位于中国首都北京的西单 (Xidan in Beijing of China capital)	北京 (Beijing)
1-E-1-E	Entity S in Relation 1(M,N), 1(N,S)	位于中国首都北京的西单 (Xidan in Beijing of China capital)	西单 (Xidan)
1-B-2	Entity M, S in Relation 1(M,N), 2（M,S)	美国总统访问中国 (The president of America visits China)	美国，中国 (America, China)
1-E-2	Entity N,S in 1(M,N), 2（N,S)	中国城市北京和天津 (City Beijing and Tianjin of China)	北京，天津 (Beijing, Tianjin)
2-1-B	Entity N in 2(M,N), 1（N,S)	美国总统访问中国北京 (The president of America visits China)	中国 (China)
2-1-E	Entity S in 2(M,N), 1（N,S)	美国总统访问中国北京 (The president of America visits China)	北京 (Beijing)

Table 7. Experiment result of entity relation extraction

Dataset	CRFs trained	Time (s)	Precision (%)	Recall R(%)	F-Score (%)
M1	Model 1	48	39.8	45.3	42.4
M2	Model 2	112	38.2	57.2	45.8
M3	Model 3	203	45.3	59.6	51.5
M4	Model 4	293	50.2	65.5	56.8
M5	Model 5	547	63.1	69.8	66.3
M6	Model 6	923	73.1	74.9	74.0
M7	Model 7	1982	76.9	84.7	80.8
M8	Model 8	2439	87.9	89.4	88.6
M9	Model 9	3010	90.5	95.8	93.1

$$P = \frac{r_1}{r_2} \times 100\%$$

$$R = \frac{r_1}{r_3} \times 100\%$$

$$F = \frac{2 \times P \times R}{P + R}$$

The data in table is the result of using M10 to test each model trained through M1-M9. Experiment shows that *P*, *R* and *F* increase with the more dataset used in training model. When we use M1 training the model F-Score is 42.4%. This is a disappointed value. But it only uses few sample dataset (newspaper of one day) for training. When using data of nine days we have obtained *P*,*R*,*F* values as 90.5%, 95.8% and 93.1% by use Model10. It illustrates it's a valid method. If we provide enough sample dataset we may win better result.

6. Conclusions

This paper discusses the information extraction of Chinese text based on CRFs which aims at the chunk parsing and relation extraction. Processing Chinese text is a complex system. This is an exploration because we haven't enough sample dataset for training CRFs model by now. But we think it's an effective method by experiments since CRFs model possesses working with global features.

At present we are developing a prototype of information extraction so a lot of work will be continued. Absence of training database is the common problem for many kinds of language. But manual label will be large cost. Therefore there are some researches on automatic or semi-automatic constructing dataset. In addition, how to select suitable feature set and improve precision is also future works.

REFERENCES

[1] E. C. Mary and J. M. Raymond, "Relational Learning of Pattern-match Rules for Information Extraction," Ph.D. Thesis, University of Texas, Austin, 1998.

[2] S. Stephen, "Learning Information Extraction Rules for Semi-Structured and Free Text," *Machine Learning,* Vol. 34, No. 13, 1999, pp. 233-272.

[3] D. Freitag and A. McCallum, "Information Extraction with HMM Structures Learned by Stochastic Optimization," *Proceedings of* 18*th Conference on Artificial Intelligence*, AAAI Press, Edmonton, 2002, pp. 584-589.

[4] R. Souyma and C. Mark, "Representing Sentence Structure in Hidden Markov Models for Information Extraction," *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, Washington, 2001, pp. 1273-1279.

[5] T. Scheffer, C. Decomain and S. Wrobel, "Active Hidden Markov Models for Information Extraction," *Proceedings of the Fourth International Symposium on Intelligent Data Analysis*, Springer, Lisbon, 2001, pp. 301-109.

[6] D. Freitag, A. McCallum and F. Pereira, "Maximum Entropy Markov Models for Information Extraction and Segmentation," *Proceedings of the Seventeenth International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, 2000, pp. 591-598.

[7] H. L. Sun and S. W. Yu, "Shallow Parsing: An Overview," Contemporary Linguistics, 2000.

[8] S. Miller, M. Crystal, H. Fox, L. Ramshaw, R. Schwartz, R. Stone and R. Weischedel, "Algorithms that Learn to Extract Information-BBN: Description Of The SIFT System as Used for MUC-7, *Proceedings of MUC*-7, Fairfax, 1998.

[9] J. Lafferty, A. McCallum and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," *Proceedings of the International Conference on Machine Learning* (*ICML*), 2001, pp. 282-289.

[10] Y. Y. Luo and D. G. Huang, “Chinese Word Segmentation Based on the Marginal Probabilities Generated by CRFs,” *Journal of Chinese Information Processing*, Vol. 23, No. 5, 2009, pp. 3-8.

[11] M.-C. Hong, K. Zhang, J. Tang and J.-Z. Li “A Chinese Part-of-Speech Tagging Approach Using Conditional Random Fields,” *Computer Science*, Vol. 33, No. 10, 2006, pp. 148-152.

[12] S. P. Abney and C. Tenny, “Parsing by Chunks. Principle based Parsing: Computation and Psycholinguistics,” Kluwer Academic Publishers, Dordrecht, 1991, pp. 257-278.

[13] F. Erik, “Tjong Kim Sang and Sabine Buch holz. Introduction to the Conll-2000 Shared Task: Chunking,” *Proceedings of CoNLL*-2000 *and LLL*2000, Lisbin, 2000, pp. 127-132.

[14] L. Ramshaw and M. Marcus, “Text Chunking Using Transformation-Based Learning,” In: D. Yarovsky and K. Church, Eds., *Proceedings of the Third Workshop on Very Large Corpora*, Association for Computational Linguistics, Somerset, 1995, pp. 82-94.

[15] J. Hammerton, M. Osborne, S. Armstrong and W. Daelemans, “Introduction to Special Issue on Machine Learning Approaches to Shallow Parsing,” *Journal of Machine Learning Research*, Vol. 2, No. 3, 2002, pp. 551-558.

[16] K. Nanda, “Combining Lexical, Syntactic and Semantic Features with Maximum Entropy Models for Extracting Relations,” *Proceedings of the ACL* 2004 *on Interactive poster and demonstration sessions*, Barcelona, 2004, pp. 22-25.

[17] D. Zelenko, C. Aone and A. Richardella, “Kernel Methods for Relation Extraction,” *Journal of Machine Learning Research*, Vol. 3, 2003, pp. 1083-1106.

[18] C. Whitelaw, A. Kehlenbeck, N. Petrovic, *et al*., “Web-Scale Named Entity Recognition,” *Proceeding of ACM* 17*th Conference on Information and Knowledge Management*, Napa Valley, 2008, pp. 123-132.

[19] Z. Q. Chen, D. V. Kalashnikov and S. Mehrotra, “Adaptive Graphical Approach to Entity Resolution,” *Proceedings of ACM IEEE Joint Conference on Digital Libraries*, Vancouver, 2007, pp. 204-213.

[20] X. P. Han and J. Zhao, “Person Name Disambiguation Based on Web-Based Person Mining and Categorization,” 2*nd Web People Search Evaluation Workshop in conjunction with WWW*2009, Madrid, 2009.

[21] S. D. Pietra, R. L. Mercer and S. Roukos, “Adaptive Language Modeling Using Minimum Discriminate Estimation,” *Proceedings of the Speech and Natural Language DARPA Workshop*, San Francisco, 1992, pp. 103-106.

[22] R. Rosenfeld, “Adaptive Statistical Language Modeling: A Maximum Entropy Approach,” Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1994.

[23] A. McCallum and W. Li, “Early Results for Named Entity Recognition with Conditional Random Fields Feature Induction and Web-Enhanced Lexicons,” *Proceedings of CoNLL*-2003 *Association for Computational Linguistics*, Daelemans, 2003, pp. 188-191.

[24] K. Tjong, E. F. Sang and S. Buchholz, “Introduction to the CoNLL-2000 Shared Task: Chunking,” *Proceedings of CoNLL*-2000 *and LLL*-2000 *Association for Computational Linguistics*, Lisbon, 2000, pp. 127-132.

[25] K. Tjong, E. F. Sang and J. Veenstra, “Representing Text Chunks,” *Proceedings of EACL*’99, Association for Computational Linguistics, Bergen, 1995, pp. 173-179.

[26] J. Zhao, “A Survey on Named Entity Recognition, Disambiguation and Cross 2 Lingual Conference Resolution,” *Journal of Chinese Information Processing*, Vol. 23, No. 2 March 2009, pp. 3-17.

J. Intelligent Learning Systems & Applications, **2010, 2, 147-155**
doi:10.4236/jilsa.2010.23018 Published Online August 2010 (http://www.SciRP.org/journal/jilsa)

Identification and Prediction of Internet Traffic Using Artificial Neural Networks

Samira Chabaa[1], Abdelouhab Zeroual[1], Jilali Antari[1,2]

[1]Department of Physics Cadi Ayyad University, Faculty of Sciences Semlalia, Marrakech, Morocco; [2]Ibn Zohr University-Agadir, Polydisciplinaire Faculty of Taroudant, Morocco.
Email: s.chabaa@ucam.ac.ma

Received March 16th, 2010; revised July 16th, 2010; accepted July 25th, 2010.

ABSTRACT

This paper presents the development of an artificial neural network (ANN) model based on the multi-layer perceptron (MLP) for analyzing internet traffic data over IP networks. We applied the ANN to analyze a time series of measured data for network response evaluation. For this reason, we used the input and output data of an internet traffic over IP networks to identify the ANN model, and we studied the performance of some training algorithms used to estimate the weights of the neuron. The comparison between some training algorithms demonstrates the efficiency and the accuracy of the Levenberg-Marquardt (LM) and the Resilient back propagation (Rp) algorithms in term of statistical criteria. Consequently, the obtained results show that the developed models, using the LM and the Rp algorithms, can successfully be used for analyzing internet traffic over IP networks, and can be applied as an excellent and fundamental tool for the management of the internet traffic at different times.

Keywords: *Artificial Neural Network, Multi-Layer Perceptron, Training Algorithms, Internet Traffic*

1. Introduction

Recently, much attention has been paid to the topic of complex networks, which characterize many natural and artificial systems such as internet, airline transport systems, power grid infrastructures, and the World Wide Web [1-3]. Indeed, Traffic modeling is fundamental to the network performance evaluation and the design of network control scheme which is crucial for the success of high-speed networks [4]. This is because network traffic capacity will help each webmaster to optimize their website, maximize online marketing conversions and lead campaign tracking [5,6]. Furthermore, monitoring the efficiency and performance of IP networks based on accurate and advanced traffic measurements is an important topic in which research needs to explore a new scheme for monitoring network traffic and then find out its proper approach [7]. So, a traffic model with a simple expression is significant, which is able to capture the statistical characteristics of the actual traffic accurately. Since the 1950s, many models have been developed to study complex traffic phenomena [5]. The need for accurate traffic parameter prediction has long been recognized in the international scientific literature [8].

The main motivation here is to obtain a better understanding of the characteristics of the network traffic. One of the approaches used for the preventive control is to predict the near future traffic in the network and then take appropriate actions such as controlling buffer sizes [9]. Several works developed in the literature are interested to resolve the problem of improving the efficiency and effectiveness of network traffic monitoring by forecasting data packet flow in advance. Therefore, an accurate traffic prediction model should have the ability to capture the prominent traffic characteristics, e.g. short- and long- range dependence, self-similarity in large-time scale and multifractal in small-time scale [10]. Several traffic prediction schemes have been proposed [11,19]. Among the proposed schemes on traffic prediction, neural network (NN) based schemes brought our attention since NN has been shown more than acceptable performance with relatively simple architecture in various fields [19-17]. Neural networks (NNs) have been successfully used for modeling complex nonlinear systems and forecasting signals for a wide range of engineering applications [20-26]. Indeed, the literature has shown that neural networks are one of the best alternatives for modeling and predicting traffic parameters possibly because they can approximate almost any function regardless of its degree of nonlinearity and without prior knowledge of its functional form [27]. Several researchers have demonstrated that the structure of neural network is characterized by a large degree of uncertainty which is presented when trying to select the optimal network structure. The most distinguished character of a neural network over the conventional techniques in modeling nonlinear systems is learning capability [19]. The neural network can learn the underlying relationship between input and output of the system with the provided data [19-26]. Among the various NN-based models, the feed-forward neural network, also known as the Multi Layer Perceptron Type Neural Network (MLPNN), is the most commonly used and has been applied to solve many difficult and diverse problems [27-30].

The aim of this paper is to use artificial neural networks (ANN) based on the multi-layer perceptron (MLP) for identifying and developing a model that is able to analyze and predict the internet traffic over IP networks by comparing some training algorithms using statistical criteria.

2. Artificial Neural Networks

Artificial neural networks (ANN) are an abstract simulation of a real nervous system that consists of a set of neural units connected to each other via axon connections which are very similar to the dendrites and the axons in biological nervous systems [31].

Furthermore, artificial neural networks are a large class of parallel processing architecture which can mimic complex and nonlinear processing units called neurons [32]. An ANN, as function approximator, is useful because it can approximate a desired behavior without the need to specify a particular function. This is a big advantage of artificial neural networks compared to multivariate statistics [33]. ANN can be trained to reach, from a particular input, a specific target output using a suitable learning method until the network output matches the target [34]. In addition, neural networks are trained by experience, when an unknown input is applied to the network it can generalize from past experiences and product a new result [35-37].

ANN is constituted by a tree layer: an input layer, an output layer, and an intermediate hidden layer, with their corresponding neurons. Each layer is connected to the next layer with a neuron giving rise to a large number of connections. This architecture allows ANNs to learn complicated patterns. Each connection has a weight associated with it. The hidden layer learns to provide a representation for the inputs. The output of a neuron in a hidden or output layer is calculated by applying an activation function to the weighted sum of the input to that neuron [20] (**Figure 1**). ANN model must first be "trained" by using cases with known outcomes and it will then adjust its weighting of various input variables over time to refine the output data [10]. The validation data are used for evaluating the performance of the ANN model.

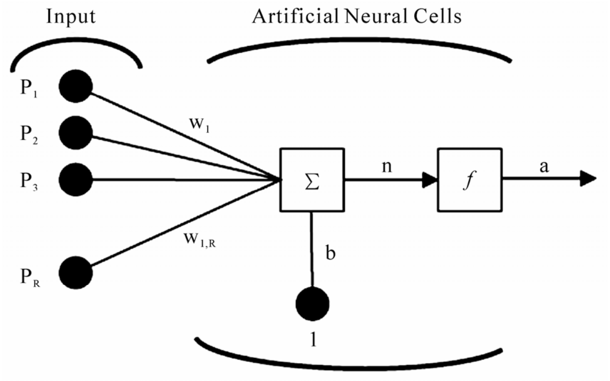


Figure 1. Neural network model

In this work, we used the back-propagation based Multi Layer Perceptron (MLP) neural network. The multi layer perceptron is the most frequently used neural network technique, which makes it possible to carry out the most various applications. The identification of the MLP neural networks requires two types of stages. The first is the determination of the network structure. Different networks with one layer hidden have been tried, and the activation function used in this study is the sigmoid function described as:

$$f(x) = \frac{1}{1 + exp(-x)} \tag{1}$$

The second stage is the identification of parameters (learning of the neural networks).

The suite of the used back-propagation neural networks are part of the MATLAB neural network toolbox which assisted in appraising each of the above individual neural network models for predictive purposes [38,39]. In this study various training algorithms are used.

3. Training Algorithms

The MLP network training can be viewed as a function approximation problem in which the network parameters (weights and biases) are adjusted during the training, in an effort to minimize (optimize) error function between the network output and the desired output [40]. The issue of learning algorithm is very important for MLP neural network [41]. Most of the well known ANN training algorithms are based on true gradient computations. Among these, the most popular and widely used ANN training algorithm is the Back Propagation (BP) [42,43]. The BP method, also known as the error back propagation algorithm, is based on the error correlation learning rule [44]. The BP neural networks are trained with different training algorithms. In this section we describe some of these algorithms. The BP algorithm uses the gradients of the activation functions of neurons in order to backpropagate the error that is measured at the output of a neural network and calculate the gradients of the output error over each weight in the network. Subsequently, these gradients are used in updating the ANN weights [45].

3.1 Gradient Descent Algorithm

The standard training process of the MLP can be realized by minimizing the error function E defined by:

$$E = \sum_{p=1}^{p} \sum_{j=1}^{N_M} \left(y_{j,p}^{M} - t_{j,p} \right)^2 = \sum_{p=1}^{p} E_p \qquad (2)$$

where $y_{j,p}^{M} - t_{j,p}$ is the squared difference between the actual output value at the j^{th} output layer neuron for pattern p and the target output value. The scalar *p* is an index over input–output pairs. The general purpose of the training is to search an optimal set of connection weights in the manner that the errors of the network output can be minimized [46].

In order to simplify the formulation of the equations, let w be the n-dimensional weight vector of all connection weights and biases. Accordingly, the weight update equation for any training algorithm has the iterative form. In each iteration, the synaptic weights are modified in the opposing direction to those of the gradient of the cost function. The on-line or off-line versions are applied where we use the instantaneous gradient of the partial error function E_p, or on the contrary the gradient of the total error function E respectively.

To calculate the gradient for the two cases, the Error Back Propagation (EBP) algorithm is applied. The procedure in the mode off-line sums up as follows

$$w(k+1) = w(k) + \alpha_k d_k \qquad (3)$$

where, $w(k) = \left(w_1(k), \ldots \ldots \ldots, w_n(k) \right)^T$ is the weight vector in *k* iterations, n is the number of synaptic connections of the network, *k* is the index of iteration, α_k is the learning rate which adjusts the size of the step gradient, and d_k is a search direction which satisfies the descent condition.

The steepest descent direction is based to minimize the error function, namely $d_k = -g_w$

where $g_w(k) = \left(\frac{\partial E}{\partial w_1(k)}, \ldots \ldots \ldots \ldots, \frac{\partial E}{\partial w_n(k)} \right)^T$ is the gradient of the estimated error in *w*. throughout the training with the standard steepest descent, the learning rate is held constant, which makes the algorithm very sensitive to the proper setting of the learning rate. Indeed, the algorithm may oscillate and become unstable, if the learning rate is set too high. But, if the learning rate is too small, the algorithm will take a long time to converge.

3.2 Conjugate Gradient Algorithm

The basic back propagation algorithm adjusts the weights in the steepest descent direction in which the performance function decreases most rapidly. Although, the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence [44]. In the conjugate gradient algorithms, a search is made along the conjugate gradient direction to determine the step size which will minimize the performance function along that line [41].

The conjugate gradient (CG) methods are a class of very important methods for minimizing smooth functions, especially when the dimension is large [47].

The principal advantage of the CG is that they do not require the storage of any matrices as in Newton's method, or as in quasi-Newton methods, and they are designed to converge faster than the steepest descent method [46].

There are four types of conjugate gradient algorithms which can be used for training.

All of the conjugate gradient algorithms start out by searching in the steepest descent direction (negative of the gradient) on the first iteration [41,44,46]:

$$p_0 = -g_w(0)$$

where p_0 is the initial search gradient and g_0 is the initial gradient.

A line search is then performed to determine the optimal distance to move along the current search direction:

$$w(k+1) = w(k) + \alpha_k d_k$$

The next search direction is determined so that it is conjugated to previous search directions. The general procedure for determining the new search direction is to combine the new steepest descent direction with the previous search direction:

$$d_k = -g_w(k) + \beta_k d_{k-1}$$

where β_k is a parameter to be determined so that d_k becomes the k-the conjugate direction.

The way in which the β_k constant is computed distinguishes the various versions of conjugate gradient, namely Fletcher-Reeves updates (Cgf), Conjugate gradient with Polak-Ribiere updates (Cgp), Conjugate gradient with Powell-Beale restarts (Cgb) and scaled conjugate gradient algorithm (Scg).

1) ***Conjugate gradient with Fletcher-Reeves updates***

The procedure to evaluate the constant β_k with the Fletcher-Reeves update is [48]

$$\beta_k = \frac{g_w^T(k) g_w(k)}{g_w^T(k-1) g_w(k-1)}$$

β_k represents the ratio of the norm squared of the current gradient to the norm squared of the previous gradient.

2) *Conjugate gradient with Polak-Ribiere updates*

The constant β_k is computed by the Polak-Ribiére update as [49]:

$$\beta_k = \frac{g_w^T(k)\,y_k}{g_w^T(k-1)\,g_w(k-1)}$$

where $y_k = g_w(k) - g_w(k-1)$ is the inner product of the previous change in the gradient with the current gradient divided by the norm squared of the previous gradient.

3) *Conjugate gradient with Powell-Beale restarts*

In conjugate gradient algorithms, the search direction is periodically reset to the negative of the gradient. The standard reset point occurs when the number of iterations is equal to the number of network parameters (weights and biases), but there are other reset methods that can improve the efficiency of the training process [50]. This technique restarts if there is a very little orthogonality left between the current and the previous gradient:

$$\left| g_w^T(k-1)\,g_w(k) \right| \geq 0.2 \left\| g_w(k) \right\|^2$$

If this condition is satisfied, the search direction is reset to the negative of the gradient [41,44,51].

4) *Scaled conjugate gradient* (*Scg*)

The scaled conjugate gradient algorithm requires a line search at any iteration which is computationally expensive since it requires computing the network response for all training inputs at several times for each search.

The Scg combines the model-trust region approach (used in the Levenberg-Marquardt algorithm described in the following section), with the conjugate gradient approach. This algorithm was designed to avoid the time-consuming line search. It is developed by Moller [52], where the constant β_k is computed by:

$$\beta_k = \frac{\left| g_w(k+1) \right|^2 - g_w^T(k+1)\,g_w(k)}{\left| g_w(k) \right|^2}$$

3.3 One Step Secant

Battiti proposes a new memory-less quasi-Newton method named one step secant (OSS) [53], which is an attempt to bridge the gap between the conjugate gradient algorithms and the quasi-Newton (secant) algorithms. This algorithm does not store the complete Hessian matrix. It assumes that at any iteration, the previous Hessian was the identity matrix. This has the additional advantage that the new search direction can be calculated without computing a matrix inverse [41,44].

3.4 Levenberg-Marquardt Algorithm

The Levenberg-Marquardt (LM) algorithm [54,55] is the most widely used optimization algorithm. It is an iterative technique that locates the minimum of a multivariante function that is expressed as the sum of squares of non linear real valued functions [56-58]. The LM is the first algorithm shown to be blend of steepest gradient descent and Gauss-Newton iterations. Like the quasi-Newton methods, the LM algorithm was designed to approach second-order training speed without having to compute the Hessian matrix [44]. The LM algorithm provides a solution for non linear least squares minimization problems. When the performance function has the form of a sum of squares, then the Hessian matrix can be approximated as [44]:

$$H = \mathrm{J}^{\mathrm{T}}\mathrm{J}$$

where J is the Jacobian matrix that contains the first derivates of network errors and the gradient can be computed as:

$$g_w = \mathrm{J}^{\mathrm{T}} e$$

where the Jacobian matrix contains the first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors.

The Levenberg–Marquardt (LM) algorithm uses the approximation to the Hessian matrix in the following Newton-like update [41,44]:

$$w_{k+1} = w_k - \left[\mathrm{J}^{\mathrm{T}}\mathrm{J} + \mu I \right]^{-1} \mathrm{J}^{\mathrm{T}} e$$

where I is the identity matrix and μ is a constant.

μ decreases after each successful step (reduction in performance function) and increases only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm [44].

3.5. Resilient back Propagation (Rp) Algorithm

There has been a number of refinements made to the BP algorithm with arguably the most successful in general being the Resilient Back Propagation method or Rp [59-61]. Furthermore, the goal of the algorithm of Rp is to eliminate the harmful effects of the magnitudes of the partial derivatives. Therefore, only the sign of the derivative is used to determine the direction of the weight update. Indeed, the Rp modifies the size of the weight step that is adaptively taken. The adaptation mechanism in Rp does not take into account the magnitude of the gradient ($g_w(k)$) as seen by a particular weight, but only the sign of the gradient (positive or negative) [44,61].

The Rp algorithm is based on the modification of each weight by the update value (or learning parameter) in such a way as to decrease the overall error. The update value for each weight and bias is increased whenever the derivative of the performance function with respect to that weight has the same sign for two successive iterations.

The principle of this method is as follows:

$$w_k - w_{k-1} = -sign\left(g_w\left(k-1\right)\right)\Delta_k$$

$$\Delta_k = n^{+}\Delta_{k-1} \; if \; g_w\left(k-1\right) * g_w\left(k\right) > 0$$

$$\Delta_k = n^{-}\Delta_{k-1} \; if \; g_w\left(k-1\right) * g_w\left(k\right) < 0$$

else $\Delta_k = \Delta_{k-1}$ where $0 < n^{-} < 1 < n^{+}$

Δ_k is the update value of the weight, which evolves according to changes of sign of the difference ($w_k - w_{k-1}$) of the same weight in k iterations. The update values and the weights are changed after each iteration.

All update values are initialized to the value D0. The update value is modified in the following manner: if the current gradient ($g_w\left(k\right)$) multiplied by the gradient of the previous step is positive (that is the gradient direction has remained the same), then the update value is multiplied by a value n^{+} (which is greater than one). Similarly, if the gradient product is negative, the update value is multiplied by the value n^{-} (which is less than one) [61].

4. Results and Discussion

In this part, we are interested in appling the MPL neural networks for developing a model able to identify and predict the internet traffic. The considered data are composed of 1000 points (**Figure 2**). The databases were divided in two parts training (750 points) and testing (250 points) data as required by the application of MLP. Additionally, the training data set is used to train the MLP and must have enough size to be representative for overall problem. The testing data set should be independent of the training set and are used to assess the classification accuracy of the MLP after the training process [62,63].

The error analysis was used to check the performance of the developed model. The accuracy of correlations relative to the measured values is determined by various statistical means. The criteria exploited in this study were the Root Mean Square Error (RMSE), the Scatter Index (SI), the Relative Error and Mean Absolute Percentage Error (MAPE) [64-66] given by:

$$\mathrm{R}_{error} = E\left[\left\{\left(\hat{y}_m - y_m\right)^2 / y_m^2\right\}^{1/2}\right] \tag{2}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum\nolimits_{m=1}^{N}\left(y_m - \hat{y}_m\right)^2} \tag{3}$$

$$SI = \frac{RMSE}{\overline{y}_m} \tag{4}$$

$$MAPE = \frac{100}{N}\sum\nolimits_{i=1}^{N}\left|y_m - \hat{y}_m\right| \tag{5}$$

where y_m and $\hat{y}_m$ represent respectively real and estimated data, $\overline{y}_m$ is the mean values of real data and N represents the sample size. **Table 1** shows the obtained results of each statistical indicator for the different algorithms:

From these results, we conclude that the Levenberg-Marquardt (LM) and the Resilient back propagation (Rp) algorithms give more precision using the statistical criteria than the other training algorithms. The Gd, Scg, Cgf, Cgp, Cgb and Oss training algorithm give big values in term of the used statistical criteria, which prove that these training algorithms are not significant for prediction purpose. For this reason, we used the LM and the Rp training algorithms in the next paragraph.

To agreement the efficiency of the developed model based on the LM and the Rp training algorithms, we draft in **Figure 3** the evolution of measured and predicted time series of the internet traffic for the two algorithms where we represent just 100 points. We notice that the two time series have the same behaviour.

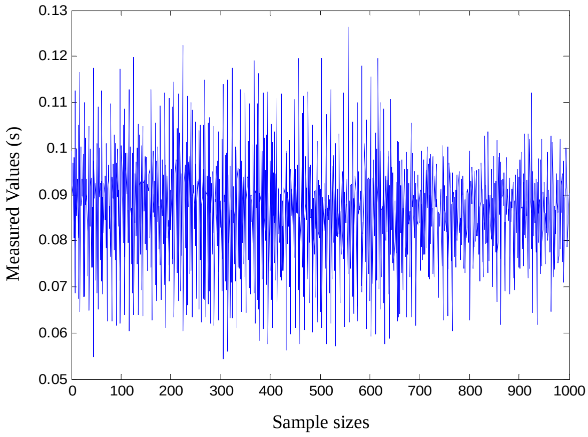
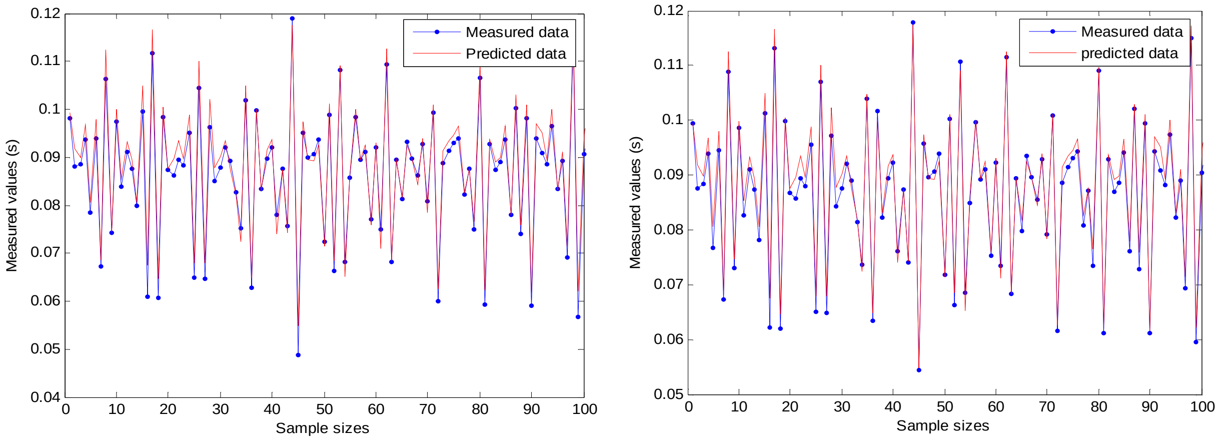


Figure 2. Real data

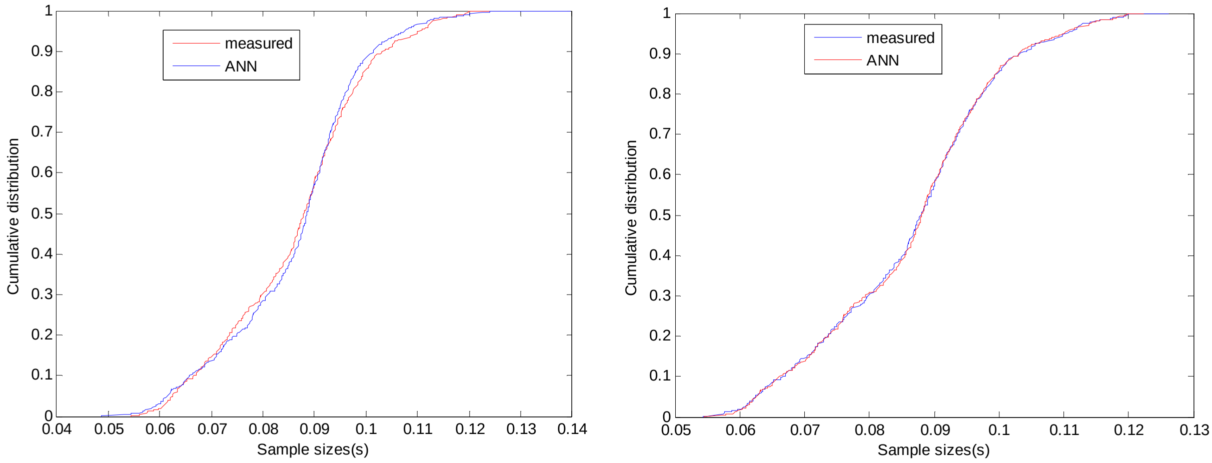
Table 1. Values of different statistical indicators for different algorithms

Training algorithms	R$_{\text{error}}$	RMSE	SI	MAPE
LM	0.0230	0.0019	0.0222	4.2563
Gd	0.1666	0.0142	0.1642	4.2580
Rp	0.0371	0.0031	0.1327	4.3584
Scg	0.1279	0.0128	0.0357	4.1235
Cgf	0.1448	0.0128	0.1300	4.2528
Cgp	0.1339	0.0118	0.1485	4.2246
Cgb	0.1480	0.0128	0.1480	4.2621
Oss	0.1480	0.0128	0.1480	4.2622



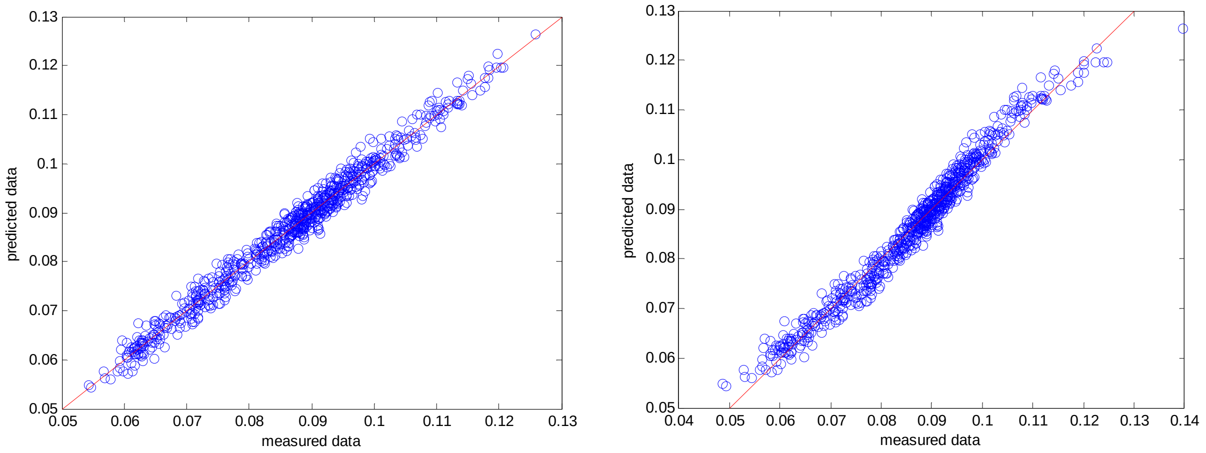
(a) (b)

Figure 3. Comparison between measured and predicted data (a) Rp algorithm, (b) LM



(a) (b)

Figure 4. Cumulative distribution of measured and predicted data (a) Rp algorithm, (b) LM algorithm



(a) (b)

Figure 5. Scattering diagram of measured and predicted data for (a) Rp algorithm, (b) LM algorithm

On the other hand, we present in **Figure 4** the cumulative distributions of measured and predicted data. Figure 4 demonstrates clearly the similarity between measured and predicted values. So, the identified ANN model can be used for predicting data of internet traffic. Furthermore, the scattering diagram (**Figure 5**) presents a comparison between measured and predicted data using ANN model which constitutes another means to test the performance of the model.

5. Conclusions

In this paper we present an artificial neural network (ANN) model based on the multi-layer perceptron (MLP) for analyzing internet traffic over IP networks. We used the input and output data to describe the ANN model, and we studied the performance of the training algorithms which are used to estimate the weights of the neuron. The comparison between some training algorithms demonstrates the efficiency of the Levenberg-Marquardt (LM) and the Resilient back propagation (Rp) algorithms using statistical criteria. Consequently, the obtained model using the LM and the Rp can successfully be used as an adequate model for the identification and the management of internet traffic over IP networks. In addition, it can be applied as an excellent fundamental tool to management of the internet traffic at different times, and as a practical concept to install the computer material in a high industrial area.

REFERENCES

[1] J.-J. Wu, Z.-Y. Gao and H.-J. Sun, "Statistical Properties of Individual Choice Behaviors on Urban Traffic Networks," *Journal of Transportation Systems Engineering and Information Technology*, Vol. 8, No. 2, 2008, pp. 69-74.

[2] R. Albert and A. L. Barabási, "Statistical Mechanics of Complex Networks," Review Modern Physics, Vol. 74, No. 1, 2002, pp. 47-97.

[3] J.-J. Wu, H.-J. Sun and Z.-Y. Gao, "Cascade and Breakdown in Scale-Free Networks with Community Structure", *Physical Review E-Statistical, Nonlinear, and Soft Matter Physics*, Vol. 74, No. 6, 2006.

[4] W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, "On the Self Similar Nature of Ethernet Traffic," *Proceedings of ACM Sigcomm*, San Francisco, 1993, pp. 183-193.

[5] R. Yunhua, "Evaluation and Estimation of Second-Order Self-Similar Network Traffic," *Computer Communications*, Vol. 27, No. 9, 2004, pp. 898-904.

[6] B. R. Chang and H. F. Tsai, "Novel Hybrid Approach to Data-Packet-Flow Prediction for Improving Network Traffic Analysis," *Applied Soft Computing*, Vol. 9, No. 3, 2009, pp. 1177-1183.

[7] B. R. Chang and H. F. Tsai, "Improving Network Traffic Analysis by Foreseeing Data-Packet-Flow with Hybrid Fuzzy-Based Model Prediction," *Expert Systems with Applications*, Vol. 36, No. 3, 2009, pp. 6960-6965.

[8] A. Stathopoulos and M. G. Karlaftis, "A Multivariate State Space Approach for Urban Traffic Flow Modeling and Prediction," Transportation Research Part C, Vol. 11, No. 2, 2003, pp. 121-135.

[9] D.-C. Park, "Prediction of MPEG video traffic over ATM Networks Using Dynamic Bilinear Recurrent Neural Network," *Applied Mathematics and Computation*, Vol. 205, No. 2, 2008, pp. 648-657.

[10] B. Zhou, D. He, Z. Sun and W. H. Ng, "Network Traffic Modeling and Prediction with ARIMA/GARTH," *HET-NETs*' 06 *Conference*, Ilkley, 11-13 September 2006, pp. 1-10.

[11] A. Taraf, I. Habib and T. Saadawi, "Neural Networks for ATM Multimedia Traffic Prediction," *Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications*, Princeton, 1993, pp. 85-91.

[12] P. Chang and J. Hu, "Optimal Non-Linear Adaptive Prediction And Modeling Of MPEG Video in ATM Networks Using Pipelined Recurrent Neural Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 6, 1997, pp. 1087-1100.

[13] A. Abdennour, "Evaluation of Neural Network Architectures for MPEG-4 Video Traffic Prediction," *IEEE Transactions on Broadcasting*, Vol. 52, No. 2, pp. 184-192, 2006.

[14] A. F. Atiya, M. A. Aly and A. G. Parlos, "Sparse Basis Selection: New Results and Application to Adaptive Prediction of Video Source Traffic," *IEEE Transactions on Neural Networks*, Vol. 16, No. 5, 2005, pp. 1136-1146.

[15] Z. Fang, Y. Zhou and D. Zou, "Kalman Optimized Model for MPEG-4 VBR Sources," *IEEE Transactions on Consumer Electronics*, Vol. 50, No. 2, 2004, pp. 688-690.

[16] V. Alarcon-Aquino and J. A. Barria, "Multi Resolution Fir Neural Network-Based Learning Algorithm Applied to Network Traffic Prediction," *IEEE Transactions on Systems*, *Man and Cybernetics-Part C*: *Applications and Reviews*, Vol. 36, No. 2, 2006, pp. 208-220.

[17] E. S. Yu and C. Y. R. Chen, "Traffic Prediction Using Neural Networks," *Proceedings of the IEEE Global Telecommunications Conference* (*GLOBCOM*), Vol. 2, 1993, pp. 991-995.

[18] H. Lin and Y. Ouyang, "Neural Network Based Traffic Prediction for Cell Discarding Policy," *Proceedings of IJCNN*'97, Vol. 4, 1997.

[19] A. Tarraf, I. Habib and T Saadawi, "Characterization of Packetized Voice Traffic In ATM Networks Using Neural Networks," *Proceeding of the IEEE Global Telecommunications Conference* (*GLOBCOM*), Vol. 2, 1993, pp 996-1000.

[20] W. M. Moh, M.-J. Chen, N.-M. Chu and C.-D. Liao, "Traffic Prediction and Dynamic Bandwidth Allocation over ATM: A Neural Network Approach," *Computer Communications*, Vol. 18, No. 8, 1995, pp. 563-571.

[21] C. Looney, "Pattern Recognition Using Neural Net-

works," Oxford Press, Oxford, 1997.

[22] V. Vemuri and R. Rogers, "Artificial Neural Networks: Forecasting Time Series," The IEEE Computer Society Press, Los Alamitos, 1994.

[23] D. C. Park, M. A. El-Sharkawi and R. J. Marks II, "Adaptively Trained Neural Network," *IEEE Transactions on Neural Networks*, Vol. 2, No. 3, 1991, pp. 34-345.

[24] D. C. Park and T. K. Jeong, "Complex Bilinear Recurrent Neural Network for Equalization of a Satellite Channel," *IEEE Transactions on Neural Networks*, Vol. 13, No. 3, 2002, pp. 711-725.

[25] D. C. Park, M. A. El-Sharkawi, R. J. Marks II, L. E. Atlas and M. J. Damborg, "Electronic Load Forecasting Using an Artificial Neural Network," *IEEE Transactions on Power Systems*, Vol. 6, No. 2, 1991, pp. 442-449.

[26] D.-C. Park, "Structure Optimization of Bi-Linear Recurrent Neural Networks and its Application to Ethernet Network Traffic Prediction," *Information Sciences*, 2009, in press.

[27] E. I. Vlahogianni, M. G. Karlaftis and J. C. Golias, "Optimized and Meta-Optimized Neural Networks for Short-Term Traffic Flow Prediction: A Genetic Approach," *Transportation Research Part C*, Vol. 13, No. 3, 2005, pp. 211-234.

[28] A. Eswaradass, X.-H. Sun and M. Wu, "A Neural Network Based Predictive Mechanism for Available Bandwidth," *Proceeding of* 19*th IEEE International. Conference on Parallel and Distributed Proceeding Symposium*, Denver, 2005.

[29] N. Stamatis, D. Parthimos and T. M. Griffith, "Forecasting Chaotic Cardiovascular Time Series with an Adaptive Slope Multilayer Perceptron Neural Network," *IEEE Transactions on Biomedical Engineering*, Vol. 46, No. 2, 1999, pp. 1441-1453.

[30] H. Yousefi'zadeh, E. A. Jonckheere and J. A. Silvester, "Utilizing Neural Networks to Reduce Packet Loss in Self-Similar Tele Traffic," *Proceeding of IEEE International. Conference on Communications*, Vol. 3, 2003, pp. 1942-1946.

[31] R. Muñoz, O. Castillo and P. Melin, "Optimization of Fuzzy Response Integrators in Modular Neural Networks with Hierarchical Genetic Algorithms: The Case of Face, Fingerprint and Voice Recognition," *Evolutionary Design of Intelligent Systems in Modeling, Simulation and Control*, Vol. 257, 2009, pp. 111-129.

[32] Y. C. Lin, J. Zhang and J. Zhong, "Application of Neural Networks to Predict the Elevated Temperature Flow Behavior of a Low Alloy Steel," *Computational Materials Science*, Vol. 43, 2008, pp. 752-758.

[33] R. Wieland and W. Mirschel, "Adaptive Fuzzy Modeling Versus Artificial Neural Networks," *Environmental Modeling & Software*, Vol. 23, No. 2, 2008, pp. 215-224.

[34] H. M. Ertunc and M. Hosoz, "Comparative Analysis of an Evaporative Condenser Using Artificial Neural Network and Adaptive Neuro Fuzzy Inference System," *International Journal of Refrigeration*, Vol. 31, No. 8, 2009, pp. 1426-1436.

[35] C. L. Zhang, "Generalized Correlation of Refrigerant Mass Flow Rate Through Adiabatic Capillary Tubes Using Artificial Neural Network," *International Journal of Reference*, Vol. 28, No. 4, 2005, pp. 506-514.

[36] A. Sencan and S. A. Kalogirou, "A New Approach Using Artificial Neural Networks for Determination of the Thermodynamic Properties of Fluid Couples," *Energy Conversion and Management*, Vol. 46, No. 15-16, 2005, pp. 2405-2418.

[37] H. Esen, M. Inalli, A. Sengur and M. Esen, "Artificial Neural Networks and Adaptive Neuro-Fuzzy Assessments for Ground-Coupled Heat Pump System," *Energy and Buildings*, Vol. 40, No. 6, 2008, pp. 1074-1083.

[38] A. Sang and S.-Q. Li, "A Predictability Analysis of Network Traffic," *Computer Networks*, Vol. 39, No. 1, 2002, pp. 329-345.

[39] M. Çınar, M. Engin, E. Z. Engin and Y. Z. Ateşçi, "Early Prostate Cancer Diagnosis by Using Artificial Neural Networks And Support Vector Machines," *Expert Systems with Applications*, Vol. 36, No. 3, 2009, pp. 6357-6361.

[40] R. Pasti and L. N. de Castro, "Bio-Inspired and Gradient-Based Algorithms to Train Mlps: The Influence of Diversity," *Information Sciences*, Vol. 179, No. 10, 2009, pp. 1441-1453.

[41] A. Ebrahimzadeh and A. Khazaee, "Detection of Premature Ventricular Contractions Using MLP Neural Networks: A Comparative Study," *Measurement*, Vol. 43, No. 1, 2010, pp. 103-112.

[42] D. E. Rumelhart and J. L. McClelland, "Parallel Distributed Processing Foundations," MIT Press, Cambridge, 1986.

[43] Y. Chauvin, D. E. Rumelhart, (Eds.), "Backpropagation: Theory, Architectures, and Applications," Lawrence Erlbaum Associates, Inc., Hillsdale, 1995.

[44] J. Ramesh, P. T. Vanathi and K. Gunavathi, "Fault Classification in Phase-Locked Loops Using Back Propagation Neural Networks," *ETRI Journal*, Vol. 30, No. 4, 2008, pp. 546-553.

[45] D. Panagiotopoulos, C. Orovas and D. Syndoukas, "A Heuristically Enhanced Gradient Approximation (HEGA) Algorithm for Training Neural Networks," *Neurocomputing*, Vol. 73, No. 7-9, 2010, pp. 1303-1323.

[46] A. E. Kostopoulos and T. Grapsa, "Self-Scaled Conjugate Gradient Training Algorithms," *Neurocomputing*, Vol. 72, No. 13-15, 2009, pp. 3000-3019.

[47] R. Fletcher and C. M. Reeves, "Function Minimization by Conjugate Gradient," *Computer Journal*, Vol. 7, No. 2, 1964, pp. 149-154.

[48] J. Nocedal and S. J. Wright, "Numerical Optimization", Series in Operations Research, Springer Verlag, Heidelberg, Berlin, New York, 1999.

[49] E .Polak and G. Ribiéres, "Note sur la convergence de méthodes de directions conjuguées," *Revue Française*

d'Informatique et de Recherche Opérationnelle, Vol. 16, 1969, pp. 35-43.

[50] M. J. D. Powell, "Restart Procedures for the Conjugate Gradient Method," *Mathematical Programming*, Vol. 12, No. 1, 1977, pp. 241-254.

[51] D. Yuhong and Y. Yaxiang, "Convergence Properties of Beale-Powell Restart Algorithm," *Science in China* (*Series A*), Vol. 41, No. 11, 1998, pp. 1142-1150.

[52] M. Moller, "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning," *Neural Networks*, Vol. 6, No. 4, 1993, pp. 525-533.

[53] S. M. A. Burney, T. A. Jhilani and C. Adril, "Levenberg Mauquardt Algorithm for Karachi Stock Exchange Share Rates Forecasting," *Proceeding of World Academy of Science*, *Engineering*, *and Technology*, Vol. 3, 2005, pp. 171-176.

[54] R. Battiti, "First- and Second-Order Methods for Learning: Between Steepest Descent and Newton's Method," *Neural Computation*, Vol. 4, No. 2, 1992, pp. 141-166.

[55] M. T. Hagan and M. B. Menhaj, "Training Feed-Forward Networks with the Marquardt Algorithm," *IEEE Transaction Neural Networks*, Vol. 5, No. 6, 1994, pp. 989-993.

[56] M. I. A. Lourakis and A. A. Argyros, "The Design and Implementation of a Generic Sparse Buddle Adjustment Software Package Based on the Levenberg-Marquardt Algorithmy," *Technical Report FORTH-ICS/TR*, No. 340, Institute of Computer Science, August 2004.

[57] K. Levenberg, "A Method for the Solution of Certain Non-linear Problems in Least Squares," *Quarterly of Applied Mathematics*, Vol. 2, No. 2, 1944, pp. 164-168.

[58] D. W. Marquardt, "An Algorithm for the Least-Squares Estimation of Non linear Parameters", *SIAM Journal of Applied Mathematics*, Vol. 11, No. 2, 1963, pp. 431-441.

[59] M. Riedmiller and H. Braun, "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm," In: H. Ruspini, Ed., *Proceeding of the IEEE international conference on neural networks* (*ICNN*), San Francisco, 1993, pp. 586-591.

[60] M. Riedmiller, "Rprop—Description and Implementation Details," Technical Report, University of Karlsruhe, Karlsruhe, 1994.

[61] N. K. Treadgold and T. D. Gedeon, "The SARPROP Algorithm: A Simulated Annealing Enhancement to Resilient Back Propagation," *Proceedings International Panel Conference on Soft and Intelligent Computing*, Budapest, 1996, pp. 293-298.

[62] M. M. Mostafa, "Profiling Blood Donors in Egypt: A Neural Network Analysis," *Expert Systems with Applications*, Vol. 36, No. 3, 2009, pp. 5031-5038.

[63] S. Chabaa, A. Zeroual and J. Antari, "MLP Neural Networks for Modeling non Gaussian Signal," *Workshop STIC Wotic*'09 *in Agadir*, 24-25 December 2009, p. 58

[64] H. Demuth and M. Beale, "Neural Network Toolbox for Use with MATLAB**:** Computation, Visualization, Programming, User's Guide, Version 3.0," The Mathworks Inc., Asheboro, 2001.

[65] V. Karri, T. Ho and O. Madsen, "Artificial Neural Networks and Neuro-Fuzzy Inference Systems as Virtual Sensors for Hydrogen Safety Prediction," *International Journal of Hydrogen Energy*, Vol. 33, No. 11, 2008, pp. 2857-2867.

[66] J. Antari, R. Iqdour and A. Zeroual, "Forecasting the Wind Speed Process Using Higher Order Statistics and Fuzzy Systems," *Review of Renewable Energy*, Vol. 9, No. 4, 2006, pp. 237-251.

J. Intelligent Learning Systems & Applications, **2010, 2, 156-166**
doi:10.4236/jilsa.2010.23019 Published Online August 2010 (http://www.SciRP.org/journal/jilsa)

Knowledge Discovery for Query Formulation for Validation of a Bayesian Belief Network

Gursel Serpen, Michael Riesen

Electrical Engineering and Computer Science, College of Engineering, University of Toledo; School of Law, University of Toledo, Toledo, USA.
Email: gserpen@eng.utoledo.edu, riesen@fraser-ip.com

Received February 23th, 2010; revised July 6th, 2010; accepted July 20th, 2010.

ABSTRACT

This paper proposes machine learning techniques to discover knowledge in a dataset in the form of if-then rules for the purpose of formulating queries for validation of a Bayesian belief network model of the same data. Although domain expertise is often available, the query formulation task is tedious and laborious, and hence automation of query formulation is desirable. In an effort to automate the query formulation process, a machine learning algorithm is leveraged to discover knowledge in the form of if-then rules in the data from which the Bayesian belief network model under validation was also induced. The set of if-then rules are processed and filtered through domain expertise to identify a subset that consists of "interesting" and "significant" rules. The subset of interesting and significant rules is formulated into corresponding queries to be posed, for validation purposes, to the Bayesian belief network induced from the same dataset. The promise of the proposed methodology was assessed through an empirical study performed on a real-life dataset, the National Crime Victimization Survey, which has over 250 *attributes and well over* 200,000 *data points. The study demonstrated that the proposed approach is feasible and provides automation, in part, of the query formulation process for validation of a complex probabilistic model, which culminates in substantial savings for the need for human expert involvement and investment.*

Keywords: *Rule Induction, Semi-Automated Query Generation, Bayesian Net Validation, Knowledge Acquisition Bottleneck, Crime Data, National Crime Victimization Survey*

1. Introduction

Query formulation is an essential step in the validation of complex probabilistic reasoning models that are induced from data using machine learning or statistical techniques. Bayesian belief networks (BBN) have proven to be computationally viable empirical probabilistic models of data [1]. Advances in machine learning, data mining, and knowledge discovery and extraction fields greatly aided in maturation of Bayesian belief networks, particularly for classification and probabilistic reasoning tasks. A Bayesian belief network can be created through a multitude of means: it can be induced solely from data, handcrafted by a domain expert, or a combination of these two techniques can be leveraged. A Bayesian belief network model essentially approximates the full joint probability distribution in the domain of interest. The development of a Bayesian belief network model is followed by a rigorous validation phase to ascertain that the model in fact approximates the full joint probability distribution reasonably well, even under the set of independence assumptions made. Validation is a comprehensive, multi-part process and often requires costly domain expert involvement and labor.

When a BBN model is used as a probabilistic reasoning engine, the validation requires a complex and challenging approach, wherein a multitude of validation-related activities must be performed [2-5] and as part of one such activity, queries must be formed and posed to the network. Any subset of variables might be considered as evidence in such a query, which leads to the need to formulate an inordinate number of queries based on various subsets of variables. During validation by querying, a value assignment to some variables in the network is made and the posterior marginal probability or expectation of some other variables is desired. In other words, marginal probabilities and expectations can be calculated conditionally on any number of observations or evidence supplied to the network. It is also desirable, given that certain evidence is supplied, to ask for the values of non-evidence variables that result in the maximum possible posterior probability for the evidence, *i.e.*, an *ex-*

planation for the available evidence. One can specify a group of variables in the network to be estimated or estimate all variables in the network collectively. The existing literature for validation of BBNs as probabilistic reasoning tools is sparse and mainly promotes ad hoc approaches or mechanisms.

The formulation of an appropriate "query" requires the use of extrinsic methods in order to discover relationships among attributes. More specifically, in forming a query, access to a specific domain expertise can prove to be an efficient method in choosing which attributes to include as evidence and which attributes to identify for explanation or estimation. Experts in the domain of the focus data can prove to be a useful resource in forming the queries. However, there are many challenges in utilizing domain experts in manual formulation of queries and these challenges are in addition to the shear cost and resources needed.

Conducting interviews with one or preferably more experts in the relevant field of interest is one of the preliminary steps in manual query formulation. Such interviews typically expose many issues and challenges associated with relying on experts in the field to focus and to form queries. Experts interviewed are likely to demonstrate an interest in forming unique queries that would parallel their own expertise or interest, which might not fully overlap with the specific domain on which the model was built [6]. The list of potential queries suggested by the domain experts could prove to be inapplicable as the specific dataset employed to develop the BBN model might not include all the attributes sought by the domain experts. In other circumstances, experts may be interested in applying local and regional attributes rather than the global attributes or the national attributes used in the dataset.

It is highly desirable to develop an automated procedure that formulates queries by leveraging the same dataset that was employed to induce the Bayesian belief network model. In similar terms, exploration of other, and possibly automated, 'options' in generating useful and possibly non-obvious queries would be attractive. Data mining and machine learning techniques can be employed, through an inductive process, to discover automatically "queries" from a given dataset. More specifically, rule discovery and extraction algorithms can prove useful in "query formation". Examples of specific such algorithms are PART [7] and APRIORI [8].

1.1 Problem Statement

Validation of a complex Bayesian belief network, *i.e.*, one that has on the order of hundreds of variables, induced from a large dataset, like the National Crime Victimization Survey (NCVS), is a highly challenging task since it requires major investment of resources and domain expertise, while also being labor-intensive. The data mining and knowledge discovery algorithms are poised to offer a certain degree of relief from this challenge, and hence can be leveraged to automate segments of the overall process of query formation for validation. A machine learning or data mining algorithm can be leveraged to mine for rules in a dataset from which the Bayesian belief network model was induced, wherein these rules can be formulated as queries for validation purposes. The proposed study envisions processing a large and complex dataset through a rule-generation algorithm 1) to discover embedded knowledge in the form of if-then rules, and subsequently 2) to identify, through expert involvement, a subset of "interesting" and "significant" rules that can be formulated as queries for validation of the Bayesian belief network model of the dataset.

The next section discusses and elaborates on validation of a Bayesian belief network (BBN) model of a dataset, automatic query generation through a specific knowledge discovery tool, the NCVS dataset leveraged for this study, and the development of a BBN model on the same dataset. The subsequent section will demonstrate application of the proposed methodology to discover rules in the data set, filtering of rules to identify an interesting and significant subset, mapping of chosen rules into queries, and demonstration of application of such queries for validation purposes on a specific BBN model of a real-life size dataset that has over 250 attributes and 200,000 data points, namely the National Crime Victimization Survey.

2. Background

This section discusses fundamental aspects of the problem being addressed. Elaborations on validating Bayesian belief networks when employed as probabilistic reasoning models, query formulation with the help of machine learning and data mining, the dataset used for the study, National Crime Victimization Survey (NCVS), and the development of the Bayesian belief network model of the dataset are presented.

2.1 The NCVS Dataset

The National Crime Victimization Survey (NCVS) [9-10], previously the National Crime Survey (NCS), has been collecting data on personal and household victimization through an ongoing survey of a nationally representative sample of residential addresses since 1973. The geographic coverage is 50 United States. The 'universe' is persons in the United States aged 12 and over in "core" counties within the top 40 National Crime Victimization Survey Metropolitan Statistical Areas (MSA). The sample used was a stratified multistage cluster sample. The NCVS MSA Incident data that was chosen for this study contains select household, person, and crime incident variables for persons who reported a violent crime within any of the core counties of the 40 largest MSAs from January 1979 through December 2004. Household, person, and incident information for persons reporting non-violent crime are excluded from this file. The NCVS, which contains 216,203 instances and a total of 259 attributes, uses a labeling system for the attributes represented by letters and numbers. A typical attribute of interest is labeled by a five character (alpha-numeric) tag, e. g., V4529.

2.2 Bayesian Belief Network Model of NCVS Data

A Bayesian belief network (BBN) expresses a view of the joint probability distribution of a set of variables, given a collection of independence relationships. This means that a Bayesian belief network will correctly represent a joint probability distribution and simplify the computations if and only if the conditional independence assumptions hold. The task of determining a full joint distribution, in a brute-force fashion, is daunting. Such calculations are computationally expensive and in some instances impossible. In order to address this formidable computational challenge, Bayesian belief networks are built upon conditional independence assumptions that appear to hold in many domains of interest.

A Bayesian belief network enables the user to extract a posterior belief. All causal relationships and conditional probabilities are incorporated into the network and are accessible through an automated inference process. A once tedious and costly (in terms of computation) method of extracting posterior beliefs in a given domain is now space-efficient and time-efficient. It is also possible to make queries on any attribute of one's choosing as long as it is one of those included in the model. One can easily adjust the prior evidence in the same manner enabling him to effectively compare and contrast posterior probabilities of a given attribute based on prior knowledge. The introduction of such a method has increased the breadth and depth of statistical analysis exponentially.

The BBN creation process consists of multiple phases. Following any preprocessing needed on a given dataset, the learning or training phase starts, wherein appropriate structure learner and parameter learner algorithms need to be selected by means of empirical means [11-17]. Learning a Bayesian belief network is a two stage process: first learn a network structure and then learn the probability tables. There are various software tools, some in the public domain and open source, to accomplish the development of a BBN through induction from data. For instance, the open-source and public-domain software tool WEKA [7], a machine learning tool that facilitates empirical development of clustering, classification, and functional approximation algorithms, has been leveraged to develop a BBN from the NCVS dataset for the study reported herein.

The validation phase can best be managed through a software tool that can implement the "probabilistic inferencing" procedure applicable for Bayesian belief networks. Another open-source and public-domain software tool, the JavaBayes [18] was used for this purpose, which is able to import an already-built BBN model, and facilitate through its graphical user interface querying of any attribute for its posterior probability value among many other options. A BBN model developed in WEKA can easily be imported into the JavaBayes. Once imported, the JavaBayes allows the user to identify and enter the evidence, and query a posterior belief of any attribute.

In this study, the BayesNet tool of the WEKA has been used to induce a classifier with the "Victimization" attribute in the NCVS dataset as the class label [19]. The NCVS dataset has been split into training and test subsets with 66% and 33% ratios, respectively. Simulations were run for a variety of structure and parameter learning options. Results suggest that a number of BBN models performed exceptionally well as classifiers for the "Victimization" attribute in the NCVS dataset. All WEKA versions of the local hill climbers and local K2 search algorithms led to classification performances on the test subset with 98% or better accuracy. Since the classification accuracy rates were so close to each other, the value of parameter "number of parent nodes" became significant given that it directly relates to the approximation capability of the BBN to the full joint distribution. Accordingly, the BBN model generated through the local K2 algorithm with Bayes learning and four parent nodes (the command-line syntax is "Local K2-P4-N-S BAYES" in WEKA format) was selected as the final network. This model, which, upon request, can be obtained in BIF format from the authors, has been used exclusively in the validation experiments reported in the following sections.

2.3 Validation of Bayesian Belief Networks

Validation of a Bayesian belief network is a comprehensive process. Once the Bayesian belief network (BBN) is induced from the data and subsequently tuned by the domain experts, the next step is the testing for validation of the premise that the network faithfully represents the full joint probability distribution subject to conditional independence assumptions [5,20,21]. As part of the validation task, values computed by the BBN are compared with those supplied by the domain experts, statistical analysis, and the literature. Another distinct activity for validation entails querying any variable for its posterior distribution or posterior expectation, and to obtain an explanation for a subset of or all of the variables in the network. In that respect, knowledge discovery and data mining tools, in conjunction with the domain experts, are leveraged to formulate a set of so-called "interesting" and "significant" queries to pose to the BBN. Validating a BNN is no trivial task and necessitates ad hoc and empirical elements. More specifically, a comprehensive and rigorous process of evaluation and validation of a BBN model entails the following:

1) Perform elicitation review that consists of reviewing the graph structure for the model, and reviewing and comparing probabilities with each other [22].

2) Carry out sensitivity analysis that measures the effect of one variable on another [3].

3) Implement validation using the data that entails analysis of predictive accuracy and expected value calculations.

4) Conduct case-based evaluations that may include the following: run the model on test cases, compare the model output with the expert judgment, and finally, compare the model predictions with the “ground truth” or accepted trends currently relied upon by experts in the domain of interest.

The case-based evaluations validation step is the most costly and challenging since it requires substantial human expertise. In particular, elicitation of expert judgment to be leveraged for the validation of the Bayesian belief network poses a serious obstacle since numerous test cases or “queries” must be generated and applied to the Bayesian belief network model. The expected values must be defined in advance by human experts to form a basis for comparison with those calculated by the network itself.

2.4 Query Formulation

Machine learning and data mining techniques may be leveraged to automatically discover “queries” for a given dataset. A query is the calculation of the posterior probabilities of any attribute or variable based upon the given prior evidence. When a user provides that a specific attribute is observed to have a (discrete) value, this ‘evidence’ may be used in calculating the posterior probability of a dependent variable. This is best understood by an example. Assume that the user makes a query for the posterior probability that a person will be a victim of burglary. This query is dependent upon the values observed for relevant attributes like the gender of the potential victim. If burglary is shown to be dependent upon the gender of the victim, then the prior observed value of male or female for the potential victim’s gender will need to be supplied by the user in order to calculate the conditional probability of this incident. This is analogous to an if-then rule: such a rule is a candidate for a query. One rule could postulate that

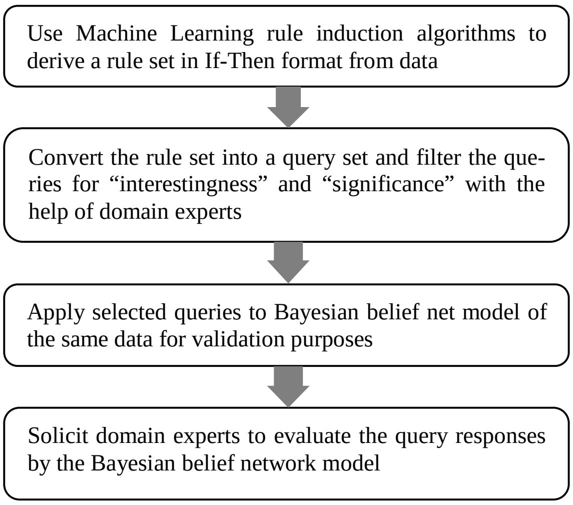
“**If** *the gender of the victim is female* **Then** *the probability of burglary will be greater than* 0.60.”

By having such a rule at one’s disposal, the process of making valid and knowledgeable queries can be streamlined. One does not necessarily have to solely rely on an expert for help to formulate “interesting” and “significant” queries. A rule set may be generated using one of many knowledge discovery algorithms, which can be structured to produce a set of if-then rules. Machine learning and data mining techniques prove useful for discovering knowledge that can be modeled as a set of if-then rules. Among the viable algorithms, PART [23], C4.5 or C5 [24], and RIPPER [25] from machine learning, and APRIORI [8] and its derivatives from the data mining fields are prominent.

3. Automation of Query Generation

This section presents application of machine learning algorithms for knowledge discovery in the form of if-then rules on the NCVS dataset for the purpose of formulating queries to the Bayesian belief network model of the same dataset. Although data mining algorithms are also appropriate for knowledge extraction and subsequent automation of the query formulation process [26], their computational cost may quickly become prohibitive if care is not exercised. Decision tree or list based algorithms within the domain of machine learning are appealing in that they can generate a rule set for a given single attribute of interest often within reasonable spatio-temporal cost bounds. Accordingly, the machine learning algorithm PART is chosen for the rule discovery and extraction task given its desirable algorithmic and computational properties. The PART algorithm [23] combines two approaches, C4.5 [24] and RIPPER [25] in an attempt to avoid their respective disadvantages. The main steps for validation of a Bayesian belief net model of data through automated query generation are shown in **Figure 1**.

The rule induction algorithm PART is applied to the NCVS dataset in order to extract a set of rules. The same rules are leveraged, following further processing by domain experts, as queries to the BBN model of the NCVS



Use Machine Learning rule induction algorithms to derive a rule set in If-Then format from data

Convert the rule set into a query set and filter the queries for “interestingness” and “significance” with the help of domain experts

Apply selected queries to Bayesian belief net model of the same data for validation purposes

Solicit domain experts to evaluate the query responses by the Bayesian belief network model

Figure 1. Generic overview of steps for Bayesian belief net validation through automated query generations

dataset for validation purposes. Initially, a subset of rules is labeled as “interesting” and “significant” by the domain experts, wherein “interesting” is a subjective labeling by a particular domain expert based upon the relationship of the evidence and the resultant projected probability of the THEN consequent variable. Next, these rules are formulated as queries and evidence associated with each query supplied to the BBN model on JavaBayes. Posterior probability calculations performed by the JavaBayes reasoning or inferencing engine for the attribute(s) of interest, which can be any subset from the list, are compared to expected values. This is done to infer if, in fact, the BBN model approximates reasonably well the joint probability distribution for the set of attributes entailed by the NCVS dataset.

3.1 PART Algorithm and Rules on NCVS Data

Rules that are derived from a dataset through a machine learning algorithm like PART expose the relationship between a subset of attributes and a single attribute of interest (or the class label), *i.e.* in this case the class label is designated as the “Victimization” due to its significance in the domain. Any attribute can be designated as the class label and would require a separate run of the PART algorithm to generate the set of rules whose consequents are the class label. Through the PART algorithm, the knowledge entailed by the dataset is captured into a framework with a set of if-then rules. Specifically, the format for a rule complies with the following: IF *premise* THEN *consequent*, where the premise is a statement of the form of a logical conjunction of a subset of attribute-value pairs, and the consequent represents a certain type of victimization. We have used the WEKA implementation of the PART algorithm throughout this study. Available options for the PART as implemented in the WEKA package and their associated default settings are shown in **Table 1**.

The NCVS Incident dataset was preprocessed prior to the rule induction step: the attribute count was reduced from 259 to 225 through removal of those that were not deemed to be relevant for the study. The attributes in the NCVS Incident dataset are represented, with a few exceptions, by a label that has four numeric characters preceded by the letter “V”. The PART algorithm was applied to the NCVS dataset with default parameter values and the V4529 (Victimization) as the class attribute. Values for the V4529 attribute are shown in **Table 2**. The algorithm was trained on a 66%-33% training-testing split of the NCVS dataset, and generated a list of 176 rules [27]. The rules output are in the traditional IF-THEN format, where the premise is the logical conjunction of a set of attribute-value pairs (*i.e.*, evidence) followed by the consequent which is a specific value of the class attribute. **Table 3** illustrates one of the rules discovered by the PART algorithm on the NCVS data and its interpretation.

Table 1. Parameter options and default values for the WEKA PART algorithm.

PART Option	Explanation	Default Values
-C number	Confidence threshold for pruning	0.25
-M number	Minimum number of instances per leaf	2
-R	Use reduced error pruning	False
-N number	Number of folds for reduced error pruning	3
-B	Use binary splits for nominal attributes	False
-U	Generate unpruned decision list	False
-Q <seed>	Seed for random data shuffling	1

Table 2. Values for the NCVS attribute V4529

V4529 Label	Description of Values for “Victimization” Attribute V4529
x60	Completed/Attempted rape
x61	Sexual attack/assault/serious assault
x62	Attempted/completed robbery with injury from serious assault
x63	Attempted/completed robbery with injury from minor assault
x64	Attempted/completed robbery without injury
x65	Attempted/completed aggravated assault
x66	Threatened assault with weapon
x67	Simple assault completed with injury
x68	Assault without weapon without injury
x69	Verbal threat of rape/sexual assault
x70	Verbal threat of assault
x71	Attempted/Completed purse snatching and pocket picking
x72	Burglary
x73	Attempted forcible entry
x74	Attempted/completed motor vehicle theft
x75	Attempted/completed theft

3.2 Query Formulation Based on PART Rules

The process of query formulation using the PART rules and posing the queries to the BBN model entails human expert involvement and is the focus of the discussion in this section. A PART rule, which is captured through the “IF-*premise*-THEN-*consequent*” framework, readily lends itself to the query formation: the premise becomes the prior evidence for a query, where posterior probability value calculation is desired for the rule consequent. Such queries may be employed to validate, among other uses,

Table 3. A sample rule generated by the PART algorithm and its interpretation

PART Rule	Interpretation

V4113 = 0 &
V4094 = 0 &
V4119 = 0 &
V4117 = 0 &
V4118 = 0 &
V4096 = 9:67
	If the victim
	· did not receive injuries from an attempted rape (V4113 = 0), and
	· was not attacked in the form of rape (V4094 = 0), and
	· was not knocked unconscious (V4119 = 0), and
	· did not have broken bones or teeth as a result of incident (V4117 = 0), and
	· did not sustain any internal injuries (V4118 = 0), and
	· could not answer if (s)he was or was not a victim of sexual assault (V4096 = 9),
	Then
	· there is a high probability that this person will be a victim of "Simple Assault Completed with Injury" (V4529 = x67)

the Bayesian belief network model of the full joint probability distribution of the 225 attributes in the NCVS dataset. The list of 176 rules generated by the PART algorithm was manually processed by domain experts, Gabrielle Davis [28] and Michael Riesen [27], to identify those that are interesting and significant for query formation to serve as the validation set through the domain specialist's somewhat subjective perspective. The list of 49 rules identified accordingly to be leveraged as queries to the BBN model of the NCVS dataset are listed in [27].

Conversion of PART rules to queries and posing resulting queries to the JavaBayes realization of the BBN model is a straightforward process and will be illustrated next. The middle column in **Table 4** displays (in JavaBayes format) the posterior probability for the victimization attribute V4529 with no prior evidence observed before any query is posed as provided by the BBN model. One of the simple rules generated by the PART that will be used as an example query is shown in **Table 4**. The premise part of the rule, *i.e.*, V4127 = 2 AND V4095 = 1, is considered as prior evidence and supplied to the BBN model as such. Next, the JavaBayes is asked to perform "reasoning" or "inference" using the supplied prior evidence through the BBN model of the NCVS data. Once the inferencing calculations are complete, the updated posterior probabilities for all discrete values of the victimization attribute are as shown in the rightmost column in **Table 4**. As an example, the probability value for the *x*60 value of the victimization attribute is now 0.612, a marked increase compared to the no-evidence case. Translating the NCVS notation of the above comparison, this rule indicates that when a victim is attacked in such a way that the victim perceived the incident as an attempted rape (V4095 = 1) and the victim was not injured to the extent that the victim received any medical care, including self treatment (V4127 = 2), there is a 61% chance that this victim would be a victim of a completed rape or attempted rape (V4529 = *x*60).

Next, another and relatively more complex rule generated by the PART algorithm as shown in **Table 5** was presented as a query to the BBN model on JavaBayes. In Table 6, the process of supplying the evidence as provided from this PART rule is shown. First, the prior evidence that the victim suffered no injuries that are related to attempted rape (V4113 = 0) is supplied. Then, further prior evidence is supplied through V4052 = 0, meaning that the offender did not use a rifle, shotgun or any other gun different from a handgun. More prior evidence is added in the form of V4050 = 3, indicating that there was a weapon used, but the specific type is not applicable as reported in the NCVS. In the final step, V4241 = 1 as prior evidence is provided. However, with this addition of V4241 = 1 the JavaBayes running in the Java Runtime Environment generated an OutOfMemory exception, although the heap size was set to 3.5 GB. Nevertheless, for each of the reportable cases, the corresponding posterior probability table for the NCVS Victimization attribute V4529 is displayed. As shown in **Table 6**, inclusion of each further evidence has a direct affect on the posterior probability of the consequent (*i.e.*, the so-called "Then" part of a rule), which can be observed through the value of *x*65 discrete label for the class attribute V4529.

Table 4. A PART rule (V4127 = 2 & V4095 = 1: 60), associated JavaBayes query, and updated posterior probability values for V4529 with increasing evidence

Conditional Probabilities of V4529 Labels	Posterior Probabilities for V4529 with No Evidence	Posterier Probabilities with Evidence due to V4127 = 2 & V4095 = 1
p(x60|evidence)	0.004	0.612
p(x61|evidence)	0.001	0.005
p(x62|evidence)	0.005	0.006
p(x63|evidence)	0.005	0.009
p(x64|evidence)	0.025	0.003
p(x65|evidence)	0.036	0.003
p(x66|evidence)	0.006	0.069
p(x67|evidence)	0.022	0.008
p(x68|evidence)	0.055	0.003
p(x69|evidence)	0.000	0.007
p(x70|evidence)	0.019	0.227
p(x71|evidence)	0.018	0.010
p(x72|evidence)	0.113	0.007
p(x73|evidence)	0.032	0.009
p(x74|evidence)	0.053	0.008
p(x75|evidence)	0.598	0.008

Table 5. PART rule and associated JavaBayes query

PART Rule	Corresponding JavaBayes Query Syntax

V4113 = 0 AND
V4052 = 0 AND
V4050 = 3 AND
V4241 = 1:65
	Posterior distribution:
	probability (“V4529”|V4113 = 0, V4052 = 0, V4050 = 3, V4241 = 1)

Table 6. Posterior probabilities for the Victimization attribute V4529 with progressively increasing prior evidence (fraction truncated beyond third significant digit)

	Posterior Distributions		
V4529 Values	*probability* (*V*4529| *V*4113 = 0)	*probability* (*V*4529| *V*4113 = 0, *V*4052 = 0)	*probability* (*V*4529| *V*4113 = 0, *V*4052 = 0, *V*4050 = 3)
p(x60|evidence)	0.032	0.023	0.028
p(x61|evidence)	0.004	0.005	0.003
p(x62|evidence)	0.064	0.195	0.210
p(x63|evidence)	0.066	0.003	0.002
p(x64|evidence)	0.083	0.073	0.086
p(x65|evidence)	0.206	0.624	0.630
p(x66|evidence)	0.010	0.024	0.010
p(x67|evidence)	0.259	0.003	0.002
p(x68|evidence)	0.245	0.001	0.001
p(x69|evidence)	0.000	0.000	0.000
p(x70|evidence)	0.020	0.037	0.021
p(x71|evidence)	0.000	0.001	0.000
p(x72|evidence)	0.000	0.000	0.000
p(x73|evidence)	0.000	0.000	0.000
p(x74|evidence)	0.000	0.000	0.000
p(x75|evidence)	0.001	0.000	0.000

3.3 Validation of NCVS BBN Model through PART-Induced Queries

Each of 49 rules that were identified as “interesting” and “significant” by the domain experts was carefully considered as a test query. In light of the memory limitation encountered earlier, original rules had to be altered in order for the system to be able compute the posterior probabilities within the memory constraints of the system available. Accordingly, some of the rules were eliminated due to memory limitations: a total of 22 rules were selected, revised and included in the query list. **Table 7** shows a revised version of the rules supplied by the PART algorithm, which were computable and hence was applied as queries to the BBN model of the NCVS data. The attributes or evidence variables in each rule was ranked by domain experts [28-29], in order of interest (*i.e.* importance to study of the domain). The domain experts were able to classify two general groups of “interesting” and “significant” rules: 1) rules listing IF premises that produced an unexpected result; and 2) rules that were in direct alignment with the accepted standards in the domain.

Some attributes that are originally appearing in a specific rule and were ranked low by the experts were excluded from the corresponding query due to memory constraints. As a result of exclusion of certain attributes-value pairs from many of the 22 rules used as query, it is expected that the consequent attribute value is likely to be affected and possibly change from the value as indicated by the original rule induced by the PART rule discovery algorithm. Each revised rule in **Table 7** is indicated with an (R) next to the number of the rule.

The posterior probabilities of each rule in **Table 7** upon being posed as a query and as computed by the JavaBayes are displayed in **Table 8**, where only significant probability values are denoted for the sake of presentation clarity. **Table 9** represents the rules recovered from computed probabilities in **Table 8** to comparatively demonstrate the differences between the revised rules in **Table 7** and those computed by the BBN model of the NCVS data in **Table 9**. In formulating rules in **Table 9**, any consequent attribute value that has a comparatively significant probability value was included. Due to revision of the original rules induced from the NCVS data, there are differences between the consequents of rules in **Tables 7** and **9**.

Although there are discrepancies between the consequents of the rules in **Tables 7** and **9**, knowledge exposed by the PART rules is still present to a large degree. The “x75” represents the crime of attempted or completed theft and is a dominant value for the victimization attribute. With no evidence being presented, “x75” will represent nearly 60% of all crimes reported in the NCVS. Interestingly, the PART rules have extracted a second layer of usable information. The revised rules are not necessarily “incorrect” but are showing how a particular set of values can drastically affect the outcome of the victimization attribute. For example, rule 10 in unrevised form provides that the victimization attribute should have a large value for “x71”. As noted in Tables 8 and 9, “x71” is not the dominant value for the revised rule 10. However, the change in posterior probability for the variable “x71” from 1.8% to 18% is nevertheless noteworthy. Where the rules generated by the PART algorithm are queried exactly as they appear, the consequents of the rule hold true as the dominant variable. Since certain queries fail due to memory error, rules had to be revised to demonstrate at least a portion of the knowledge extracted by the original PART-induced rules.

Table 7. Revised query list based on PART rules

Rule No	If	Then V4529 =	Rule No	If	Then V4529 =
1 (R)	V4065 = 1 & V4026 = 9 & V3018 = 1 & V3024 = 2	75	12 (R)	V4322 = 9 & V4065 = 1 & V4024 = 7 &	71
2 (R)	V4052 = 0 & V4083 = 9 & V4094 = 0 & V4095 = 0 & V4024 = 7	65	13 (R)	V4322 = 9 & V4065 = 1 & V4307 = 0 & V4024 = 8	71
3 (R)	V4052 = 0 & V4112 = 0 & V4113 = 0 & V4095 = 0 & V4094 = 0 & V4024 = 1	65	14	V4322 = 9 & V4065 = 1 & V4285 = 9 & V4307 = 0 & V4024 = 7 & MSACC = 35	71
4 (R)	V4052 = 0 & V4094 = 0 & V4095 = 0 & V4111 = 0 & V4024 = 2	65	15 (R)	V4322 = 9 & V4065 = 1 & V4024 = 3	71
5 (R)	V4322 = 9 & V4065 = 1 & V4024 = 5	71	16 (R)	V3024 = 2 & V3020 = 23 & V2045 = 1	71
6 (R)	V4322 = 9 & V4065 = 1 & V4024 = 7 & V3018 = 2 & MSACC = 17	71	23	V4073 = 0 & V4029 = 9 & V3018 = 2 & V4152 = 9 & V2045 = 2 & V3019 = 2	75
7 (R)	V4322 = 9 & V4065 = 1 & V4024 = 7 & V3018 = 2 & MSACC = 26	71	45 (R)	V4065 = 1 & V4029 = 9 & V3018 = 2 &	75
8 (R)	V4322 = 9 & V4065 = 1 & V4024 = 2	71	46 (R)	V3020 = 8	71
9 (R)	V4322 = 9 & V4065 = 1 & V4024 = 7 & MSACC = 4	71	47 (R)	V3020 = 24 & V3014 = 3	75
10 (R)	V4322 = 9 & V4065 = 1 & V4024 = 7 & V3015 = 5	71	48 (R)	V4113 = 0 & V4052 = 0 & V4050 = 3 &	65
11 (R)	V4322 = 9 & V4128 = 1 & V4094 = 0 & V4095 = 0 & V4052 = 0 & V4051 = 0 & V4289 = 2 &	65	35	V4322 = 9 & V4052 = 0 & V4081 = 9 & V4095 = 0 & V4094 = 0 & V4096 = 9 & V4036 = 9 & V4024 = 5	65

The query results for revised PART rules were reviewed by two domain experts [28,29]. In the majority of the cases, both experts found the predicted posterior probabilities to be reasonable and in accord with the current statistical trends provided by conventional means. As an example, the Bureau of Justice Statistics (BJS) provides periodic statistical reports [9]. BJS reported that, based upon violent crimes statistics from 1973-2005, beginning with the 25-34 age category, the rate at which persons were victims of violent crimes declined significantly as the age category increased [30]. The BJS also reports that in general, males experienced higher victimization rates than females for all types of violent crime except rape/sexual assault [9]. Where the generated rules included attributes (e.g. V3014 (Age), V3018 (Gender), and V4024 (location of incident)) that were consistent with known and generally accepted trends, the experts were not surprised with the values predicted and agreed that the posterior probabilities based upon each set of the evidence attributes were not in the extremes, based upon current publications in the field. The values were not unexpectedly high and thus did not trigger a shocking response. Conversely, the posterior values were not inordinately low compared to expected results, and thus the validity of the predicted value was not drawn into question.

Rules 11, 35 and 48 were highlighted by the experts as the strongest rules, having the most sensible values for posterior prediction as compared to the generally accepted statistical values presented in currently available publications and studies. In particular, the experts easily identified a known relationship or correlation between the IF premise and consequent for each of the rules 11, 35, and 48. In each of these three strongest rules, experts found the prior evidence values clearly set the stage for the associated posterior victimization predictions. Overall, both experts indicated that the responses computed by the BBN model of the NCVS data to all queries posed were expected and reasonable in generality, suggesting that the model is realistic, and accordingly is a good approximation to the joint probability distribution.

As an exception to the generally positive feedback, rule 10 was found to be somewhat extraordinary. Rule 10 included the attribute that the victim was never married (V3015 = 5). A value of 5 for V3015 shows a distinct increase for the probability of a purse snatching or pickpocketing. Domain experts were surprised to find that this evidence value would have such an impact on the posterior probability of pick pocketing. Although the posterior prediction was not necessarily discounted, experts were skeptical, outside a more thorough explanation of the increased victimization. However, the skepticism did not detract from the intriguing prospect that the generated rule might have exposed "new" knowledge. As the experts reviewed the list of rules, the inclusion of certain "unusual" or unexpected attributes similar to the attribute uncovered by rule 10 stimulated the most feedback from the domain experts. The experts were interested in further investigation of the "new" and "unusual"

Table 8. Query results as probability values for revised PART rules in Table 7 (only highest probability values are shown and fractions are truncated beyond the second significant digit)

Rule No	x61	x62	x64	x65	x66	x67	x68	x69	x70	x71	x72	x73	x74	x75
1									0.04	0.20			0.03	0.68
2		0.26	.09	**0.61**	0.01									
3		0.22	.06	**0.67**										
4		0.23		**0.74**										
5					0.05				0.13	0.06		0.02	**0.36**	**0.32**
6					0.04				0.09	**0.28**			0.12	**0.41**
7					0.03			0.01	0.08	**0.36**			0.09	**0.38**
8					0.02				0.14	0.02		0.01	**0.30**	**0.46**
9					0.04				0.15	**0.21**			0.12	**0.41**
10					0.07				0.17	0.18		0.01	0.07	**0.43**
11				**0.98**	0.01									
12					0.05				0.15	0.19			0.09	**0.44**
13					0.01					0.12			0.07	**0.69**
14					0.01				0.03	**0.35**			0.08	**0.49**
15	0.01				0.07				0.16	0.04		0.02	0.14	**0.47**
16			0.02	**0.03**							0.11	0.03	0.05	**0.59**
23									0.01	**0.20**	0.11		0.02	**0.63**
35		0.09	0.06	**0.78**	0.03									
45									0.02	0.20	0.10		0.03	**0.62**
46			0.03	**0.04**		0.03	0.07				0.08		0.04	**0.59**
47							0.05				0.10	0.03	0.05	**0.60**
48		0.21	0.08	**0.63**					0.02					

combination of attribute-value pairs presented in generated rules, stating that the rules could provide a starting point for further research of factors that may not have been fully developed with conventional methods.

The implications of using a rule generating algorithm such as the PART to essentially generate queries are potentially profound. Limitations associated with user bias and limited domain knowledge may impede the self-generation of useful and interesting queries. Using PART as an automatic query generation tool could potentially uncover a not-so-obvious relationship between prior evidence and the resulting posterior probability of another attribute. Applying this principle to the NCVS data, the practical significance means uncovering the specific attributes of a victim or circumstance that makes them more or less probable to be a victim of a specific crime. As an example of practical implementation within the context of criminal justice, by identifying these relationships that have the greatest impact on posterior probability, resources can be channeled into areas that would be most effective in combating violent crime.

Domain experts indicated that automatic query generation using the PART algorithm or an equivalent would be helpful in not only discovering any hidden or novel relationships between attributes, but more practically as a method to reinforce trends and relationships already relied upon in the field. A second group of domain experts[1] were independently interviewed and asked to provide a list of self-generated queries that would be of personal interest. None of the second group was able to provide a list of more than three potential queries. The second group was then presented with the automatically generated queries. All experts in the second group found that

[1] Six Professors at the University of Toledo College of Law

Table 9. Rules reconstructed from probability values in Table 8 (only modified rules are shown)

Rule No	If	Then V4529 =	Rule No	If	Then V4529 =
5 (R)	V4322 = 9 & V4065 = 1 & V4024 = 5	74 & 75	14	V4322 = 9 & V4065 = 1 & V4285 = 9 & V4307 = 0 & V4024 = 7 & MSACC = 35	71 & 75
6 (R)	V4322 = 9 & V4065 = 1 & V4024 = 7 & V3018 = 2 & MSACC = 17	71 & 75	15 (R)	V4322 = 9 & V4065 = 1 & V4024 = 3	704 & 74 & 75
7 (R)	V4322 = 9 & V4065 = 1 & V4024 = 7 & V3018 = 2 & MSACC = 26	71 & 75	16 (R)	V3024 = 2 & V3020 = 23 & V2045 = 1	72 & 75
8 (R)	V4322 = 9 & V4065 = 1 & V4024 = 2	74 & 75	23	V4073 = 0 & V4029 = 9 & V3018 = 2 & V4152 = 9 & V2045 = 2 & V3019 = 2	7 1 & 75
9 (R)	V4322 = 9 & V4065 = 1 & V4024 = 7 & MSACC = 4	71 & 75	35	V4322 = 9 & V4052 = 0 & V4081 = 9 & V4095 = 0 & V4094 = 0 & V4096 = 9 & V4036 = 9 & V4024 = 5	62 & 65
10 (R)	V4322 = 9 & V4065 = 1 & V4024 = 7 & V3015 = 5	70 & 71 & 75	45 (R)	V4065 = 1 & V4029 = 9 & V3018 = 2 &	71 & 75
12 (R)	V4322 = 9 & V4065 = 1 & V4024 = 7 &	70 & 71 & 75	46 (R)	V3020 = 8	75
13 (R)	V4322 = 9 & V4065 = 1 & V4307 = 0 & V4024 = 8	71 & 75			

the collection of automatically generated queries was relatively easy to review compared to the alternative of postulating the-defined list of rules and queries.

Each of the experts in the second group agreed that it is sometimes difficult to consider the impact of a particular variable, especially if the particular variable is not one that has been extensively researched using other known techniques. In this way, the automatic rule generation may also be used as a reliable method to test prior hypotheses. Each member of the second group also agreed that an automatically generated list of rules provided a catalyst to the generation of user-defined rules and queries. At a minimum the relationships of the attributes presented in the generated rules caused members in the second group to reflect upon their own conception of trends in victimization, which ultimately resulted in a wholesale request for more information on the resultant effect of certain unexpected attributes on the posterior probability of victimization.

4. Conclusions

This paper presented an approach to address the acquisition bottleneck problem in generating human expert-formulated queries for validation of a Bayesian belief network model. A machine learning based approach for rule discovery from a dataset to serve as potential queries was proposed. The proposed technique employs machine learning (and potentially data mining) algorithms to generate a set of classification or association rules that can be converted into corresponding queries with minimal human intervention and processing in the form of filtering for interestingness and significance by domain experts. The application and utility of proposed methodology for semi-automated query formulation based on rule discovery was demonstrated on validation of a Bayesian belief network model of a real life size dataset from the domain of criminal justice.

REFERENCES

[1] D. Heckerman, "Bayesian Networks for Data Mining," *Data Mining and Knowledge Discovery*, Vol. 1, No. 1, 1997, pp. 79-119.

[2] K. B. Laskey and S. M. Mahoney, "Network Engineering for Agile Belief Network Models," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 12, No. 4, 2000, pp. 487-498.

[3] K. B. Laskey, "Sensitivity Analysis for Probability Assessments in Bayesian Networks," *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence*, Washington, D.C., 1993, pp. 136-142.

[4] M. Pradham, G. Provan, B. Middleton and M. Henrion, "Knowledge Engineering for Large Belief Networks," *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence*, Seattle, Washington, 1994, pp. 484-490.

[5] O. Woodberry, A. E. Nicholson and C. Pollino, "Parameterising Bayesian Networks," In: G. I. Webb and X. Yu Eds., *Lecture Notes in Artificial Intelligence*, Springer-Verlag, Berlin, Vol. 3339, 2004, pp. 1101-1107.

[6] S. Monti and G. Carenini, "Dealing with the Expert Inconsistency in Probability Elicitation," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 12, No. 4, 2000, pp. 499-508.

[7] H. Witten and E. Frank, "Data Mining: Practical Machine Learning Tools and Techniques," 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[8] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," *Proceedings of the* 20*th International Conference on Very Large Data Bases*, Santiago, 1994,

pp. 487-499.

[9] US Department of Justice, Bureau of Justice Statistics. National Crime Victimization Survey: Msa Data, 1979-2004. Ann Arbor, MI: Inter-university Consortium for Political and Social Research, 2007-01-15. http://www.icpsr.umich. edu/cocoon/NACJD/STUDY/04576.xml

[10] T. C. Hart and C. Rennison, Bureau of Justice Statistics, "Special Report", March 2003, NCJ 195710. http://www.ojp.usdoj.gov/bjs/abstract/rcp00.html

[11] R. Blanco, I. Inza and P. Larrañaga, "Learning Bayesian Networks in the Space of Structures by Estimation of Distribution Algorithms," *International Journal of Intelligent Systems*, Vol. 18, No. 1, 2003, pp. 205-220.

[12] R. Bouckaert, "Belief Networks Construction Using the Minimum Description Length Principle," *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Vol. 747, 1993, pp. 41-48.

[13] L. M. de Campos, J. M. Fernández-Luna and J. M. Puerta, "An Iterated Local Search Algorithm for Learning Bayesian Networks with Restarts Based on Conditional Independence Tests" *International Journal of Intelligent Systems*, Vol. 18, No. 2, 2003, pp. 221-235.

[14] J. Cheng, R. Greiner, J. Kelly, D. A. Bell and W. Liu, "Learning Bayesian Networks from Data: An Information—Theory Based Approach," *Artificial Intelligence*, Vol. 137, No. 1-2, 2002, pp. 43-90.

[15] D. Heckerman, D. Geiger and D. M. Chickering, "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data," *Machine Learning*, Vol. 20, No. 3, 1995, pp. 197-243.

[16] T. V. Allen and R. Greiner, "Model Selection Criteria for Learning Belief Nets: An Empirical Comparison," *Proceedings of International Conference on Machine Learning*, Stanford, 2000, pp. 1047-1054.

[17] Y. Guo and R. Greiner, "Discriminative Model Selection for Belief Net Structures," *Proceedings of the Twentieth National Conference on Artificial Intelligence*, Pittsburgh, 2005, pp. 770-776.

[18] F. G. Cozman**, "**JavaBayes Software Package," University of São Paulo, Politécnica**,** cited 2006. http://www.cs.cmu.edu/~fgcozman/home.html

[19] R. Bouckaert, "Bayesian Network Classifiers in Weka," Technical Report, Department of Computer Science, Waikato University, Hamilton, 2005.

[20] M. J. Druzdzel and L. C. van der Gaag, "Building probabilistic Networks: Where do the Numbers Come from?" *IEEE Transactions on Knowledge and Data Engineering*, Vol. 12, No. 4, 2000, pp. 481-486.

[21] T. Boneh, "Visualisation of Structural Dependencies for Bayesian Network Knowledge Engineering," Masters Thesis, University of Melbourne, Melbourne, 2002.

[22] M. J. Druzdzel and L. C. van der Gaag, "Elicitation of Probabilities for belief Networks: Combining Qualitative and Quantitative Information," *Proceedings of the Tenth Annual Conference on Uncertainty in AI*, Seattle, 1995, pp. 141-148.

[23] H. Witten and E. Frank, "Generating Accurate Rule Sets without Global Optimization," *Proceedings of the Fifteenth International Conference on Machine Learning*, Madison, 1998, pp. 144-151.

[24] J. R. Quinlan, "C4.5: Programs for Machine Learning," Morgan Kaufmann Publishers, San Mateo, 1993.

[25] W. W. Cohen, "Fast Effective Rule Induction," *Proceedings of the* 12*th International Conference on Machine Learning*, Lake Tahoe, 1995, pp. 115-123.

[26] J. Hipp, U. Guntzer and G. Nakaeizadeh, "Algorithms for Association Rule Mining—A General Survey and Comparison," *ACM SIGKDD Explorations*, Vol. 2, No. 1, 2000, pp. 58-64.

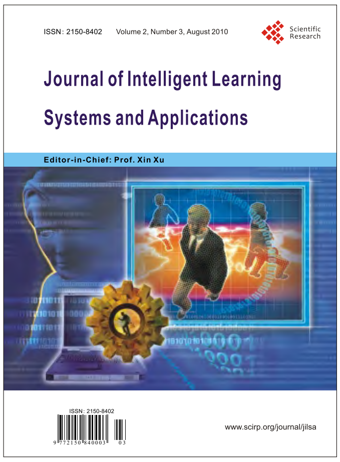
[27] M. Riesen, "Development of a Bayesian Belief Network Model of NCVS Data as a Generic Query Tool," Masters Project, Engineering, University of Toledo, Toledo, 2007.

[28] G. Davis, Private communications, College of Law, University of Toledo, Toledo, 2008.

[29] P. Ventura, "Private Communications, Criminal Justice," University of Toledo, Toledo, 2008.

[30] S. M. Catalano, Crime Victimization 2005, NCJ 214644. http://www.ojp.usdoj.gov/bjs/abstract/cv05.html

Call for Papers


ISSN: 2150-8402
Volume 2, Number 3, August 2010
Scientific Research
Journal of Intelligent Learning Systems and Applications
Editor-in-Chief: Prof. Xin Xu
ISSN: 2150-8402
www.scirp.org/journal/jilsa

Journal of Intelligent Learning Systems and Applications

ISSN 2150-8402 (print) ISSN 2150-8410 (online)

www.scirp.org/journal/jilsa

The Journal of Intelligent Learning Systems and Applications (JILSA) is a peer reviewed international journal with a key objective to provide the academic and industrial community a medium for presenting original cutting-edge research related to intelligent learning systems and their applications. JILSA invites authors to submit their original and unpublished work that communicates current research on intelligent learning systems both in the theoretical and methodological aspects, as well as various applications in real-world applications.

Papers are invited on the topics including, but not limited to:

- Approximate Dynamic Programming
- Autonomic Computing
- Autonomous Learning Systems
- Bio-inspired Learning Method
- Data Mining
- Evolutionary Computation
- General Theory on Intelligent Learning Systems
- Learning Control Systems
- Multi-agent Learning
- Network Security
- Neural Networks
- Pattern Recognition Based on Learning Techniques
- Reinforcement Learning
- Robotics
- Statistical Learning Theory
- Supervised Learning
- Unsupervised Learning

Editors in Chief

Dr. Xin Xu	**National University of Defense Technology, China**
Dr. Haibo He	**University of Rhode Island, USA**

Website and E-Mail

http://www.scirp.org/journal/jilsa **E-Mail:jilsa@scirp.org**

J. Intelligent Learning Systems & Applications, 2010, 2, 117-166
Published Online August 2010 in SciRes (http://www.SciRP.org/journal/jilsa/).

TABLE OF CONTENTS

Volume 2 Number 3 **August 2010**