PART C—SOFT COMPUTING

Multi Objective Travelling Salesman Problem

Kinjal U. Adhvaryu[#1], C.K.Bhensdadia[*2]

[#]*Computer Engineering Department, Silver Oak College of Engineering & Technology*
Ahmedabad, Gujarat ,India
[1]kinjalvk@yahoo.com

[*]*Computer Engineering Department, Dharamsinh Desai Institute of Technology,*
Nadiad,Gujarat,India
[2]ckbhensdadia@yahoo.co.in

***Abstract*— Evolutionary Algorithms (EAs) are often well suited for optimization problems involving several coupled parameters. Since 1985, various evolutionary approaches to multi objective optimization have been developed, capable of searching for multiple solutions in a single iteration. These methods differ in the fitness assignment function obtained, however the decision to which method is best suited for a given problem depends mainly upon the nature of problem and its complexity. This paper utilizes four Evolutionary Algorithms, Vector Evaluated Genetic Algorithm, Multi-objective Genetic Algorithm, Non- dominated Sorting Genetic Algorithm and Nicked Pareto Genetic Algorithm and compared the performance of these algorithms in terms of closeness to the Pareto front and diversity of solutions for Multi Objective Traveling Salesman Problem (MOTSP) using two performance metrics, average set coverage and maximum spread. This paper concluded that NPGA algorithm is comparatively best suited for any Travelling Salesman Problem (TSP) type transportation applications.**

***Keywords*— Diversity of solutions, Genetic algorithms, Optimization problems**

I. INTRODUCTION

The *Traveling Salesman Problem* is one which has commanded much attention of mathematicians and computer scientists specifically because it is so easy to describe and so difficult to solve. The problem can simply be stated as: if a traveling salesman wishes to visit exactly once each of a list of *m* cities and then return to the home city, which is the least costly tour the traveling salesman can take? And the *Multi-Objective Traveling Salesman* Pr*oblem* (MOTSP) is one in which a traveling salesman wishes to visit exactly once each of a list of *m* cities and then return to the home city but with more than one constraint like minimize the distance, cost, time or increase touring attractiveness etc. Here the problem is *symmetric multi-objective traveling salesman problem*.

Different multi-objective genetic algorithm methods [2] have been proposed by the researchers and used for multi-objective optimization for number of years. Yet, in general, no method is superior to others in all the performance aspects. Different methods have their advantages and disadvantages. But still there is a scope for finding out which method is appropriate for particular Multi-Objective Travelling Salesman Problem. A comparative analysis of these methods is a new research area. The methods are mainly compared on the basis of diversity of solutions and closeness to the Pareto front. This paper takes four multi-objective genetic algorithms and compares their performance by applying them to a Multi-objective Traveling Salesman problem. Hence this project explores the performance of these MOEAs.

II. GENETIC ALGORITHM

Search and optimization techniques can be categorized into three classes: calculus based, enumerative, and random. Calculus based approaches usually require the existence of derivatives and the continuity. Therefore it is difficult to apply them to realistic problems where these assumptions often do not hold. Enumerative methods are straightforward search schemes. They can be applied to optimization problems when the numbers of feasible solutions are few. Most optimization problems in the real world, however, have countless possible solutions [13][16]. Therefore they cannot be applied to such complex problems. As for random searches, while they search in solution spaces without any kind of information, it may not be efficient. Therefore the search direction should be specified in order to improve their search ability. Genetic Algorithm (GA) is one of random searches because they use a random choice as a tool in their searching process. While a random choice performs an important role in GAs, the environment directs the search in GAs i.e. they utilize information from the environment in their searching process. The idea of Genetic Algorithms was introduced by John Holland.

III. MULTI OBJECTIVE OPTIMIZATION PROBLEM

Besides having multiple objectives, there are a number of fundamental differences between single-objective and multi-objective optimization as follows:

1) Two goals instead of one;
2) Dealing with two search spaces;
3) No artificial fix-ups.

A striking difference between a classical search and optimization method [2][3][6][7][8] and a Genetic Algorithm is that in the latter a population of solutions is processed in each iteration (or generation). This feature alone gives Genetic Algorithms a tremendous advantage for its use in solving multi-objective optimization problems. Recall that one of the goals of an ideal multi-objective optimization procedure is to find as many Pareto-optimal solutions as possible. Since a Genetic Algorithm works with a population of solutions, in theory we should be able to make some changes to the basic Genetic Algorithm so that a population of Pareto-optimal solutions can be captured in one single simulation run of a Genetic Algorithm. This is the powerful feature of Genetic Algorithms that makes them particularly suitable to solve multi objective optimization problems. We don't need to perform a series of separate runs as in the case of the traditional mathematical programming techniques. Several independent groups of researchers have developed different versions of multi-objective evolutionary algorithms [2][3][4]. Some of them are: Vector Evaluated Genetic Algorithm (VEGA) by David Schaffer (1984),Multi-objective Genetic Algorithm (MOGA) by Fonseca and Fleming (1993),Non-dominated Sorting Genetic Algorithm (NSGA) by Srinivas and Deb (1994),Nicked-Pareto Genetic Algorithm (NPGA) by Horn, Nafploitis and Goldberg (1994).

IV. VECTOR EVALUATED GENETIC ALGORITHM (VEGA)

VEGA is the simplest possible multi-objective Genetic Algorithm [2] and is a straightforward extension of a single-objective Genetic Algorithm for multi-objective optimization. Since a number of objectives (say M) have to be handled, Schaffer thought of dividing the population at every generation into M equal subpopulations randomly. Each subpopulation is assigned a fitness based on a different objective function. In this way, each of the M objective functions is used to evaluate some members in the population. The population at any generation is divided into M equal divisions. Each individual in the first subpopulation is assigned a fitness based on the first objective function only, while each individual in the second subpopulation is assigned a fitness based on the second objective function only, and so on. In order to reduce the positional bias in the population, it is better to shuffle the population before it is partitioned into equal subpopulations. After each solution is assigned fitness, the selection operator, restricted among solutions of each subpopulation, is applied until the complete sub-population is filled. This is particularly useful in handling problems where objective functions take values of different orders of magnitude. Since all members in a sub-population are assigned a fitness based on a particular objective function, restricting the selection operator only within a subpopulation emphasizes good solutions corresponding to that particular objective function. Moreover, since no two solutions are compared for different objective function, disparity in the ranges of different objective functions does not create any difficulty either. Schaffer used the proportionate selection operator.

V. MULTIPLE OBJECTIVES GENETIC ALGORITHM (MOGA)

Fonseca and Fleming have proposed a scheme in which the rank of a certain individual corresponds to the number of chromosomes in the current population by which it is dominated. Consider, for example, an individual x_i at generation t, which is dominated by ni individuals in the current generation. Its current position in the individuals rank can be given by:

$$r_i = 1 + n_i$$

All non-dominated individuals are assigned rank 1, while dominated ones are penalized according to the population density of the corresponding region of the trade-off surface. In any population, there must be at least one solution with rank equal to one and the maximum rank of any population member cannot be more than N (the population size). It is clear that the ranking procedure in MOGA may not assign all possible ranks (between 1 and N) to any population. Once the ranking is performed; a raw fitness to a solution is assigned based on its rank. To perform this, first the ranks are sorted in ascending order of magnitude. Then, a raw fitness is assigned to each solution by using a linear (or any other) mapping function. Usually, the mapping function is chosen so as to assign fitness between N (for the best-rank solution) and 1 (for the worst-rank solution). Thereafter, solutions of each rank are considered at a time and their raw fitnesses are averaged. This average fitness is now called the assigned fitness to each solution of the rank. In this way, the total allocated raw fitness and total assigned fitness to each rank remains identical. Moreover, the mapping and averaging procedure ensures that the better-ranked solutions have a higher assigned fitness. In this way, non-dominated solutions are emphasized in a population.

MOGA Fitness Assignment Procedure

1. Choose a σ_{share}. Initialize μ?(j) = 0 for all possible ranks j = 1, … …, N. Set solution counter i = 1.
2. Calculate the number of solutions (n_i) that dominates solution i. Compute the rank of the i^{th} solution as $r_i = 1 + n_i$. Increment the count for the number of solutions in rank r_i by one, that is, $\mu(r_i) = \mu(r_i) + 1$.
3. If i < N, increment i by one and go to step 1. Otherwise, go to step 4.
4. Identify the maximum rank r* by checking the largest r_i which has $\mu(r_i) > 0$. The sorting according to rank and fitness-averaging yields the following assignment of the average fitness to any solution i = 1, … , N:

   $$= N - \sum_{k=1} (\mu(k) - 0.5\bullet(\mu(ri) - \qquad (1)$$

   To each solution i with rank ri =1, the above equation assigns a fitness equal to Fi = N - 0.5 ((1) - 1), which is the average value of (1) consecutive integers from N to N - (1) + 1. Set a rank counter r = 1.

5 For each solution i in rank r, calculate the niche count nci with other solutions (μ(r) of them) of the same rank by using above equation. Calculate the shared fitness using Fj' = Fj / ncj. To preserve the same average fitness, scale the shared fitness as follows:

$$Fj' \leftarrow \left(Fj\bullet\mu(r) / \sum_{k=1}^{\mu(r)} Fk'\right) \quad \bullet\ Fj' \qquad (2)$$

6 If rank counter r < r*, increment r by one and go to step 5. Otherwise, the process is complete.

VI. NON-DOMINATED SORTING GENETIC ALGORITHM (NSGA)

The Non-dominated Sorting Genetic Algorithm (NSGA) was proposed by Srinivas and Deb [4], and is based on several layers of classifications of the individuals. Before the selection is performed, the population is ranked on the basis of non-domination. All non-dominated individuals are classified into one category. Then this group of classified individuals is ignored and another layer of non-dominated individuals is considered. The process continues until all individuals in the population are classified. This classifies the population P into a number of mutually exclusive equivalent classes called non-dominated sets Pj:

$$P = \bigcup_{j=1}^{\rho} Pj \qquad (3)$$

It is important to realize that any two members from the same class cannot be said to be better than one another with respect to all objectives. The total number of classes (or fronts), denoted as ? in the above equation, depends on the population P and the underlying problem.

NSGA Fitness Assignment

1) Choose sharing parameter σ_{share} and a small positive number ε and initialize $F_{min} = N + \varepsilon$. Set front counter j = 1.
2) Classify population P according to non-domination to create non-dominated sets $P_1, P_2, \ldots, P\rho$.
3) For each $q \in P_j$
   a) Assign fitness $F_j^{(q)} = F_{min} - \varepsilon$.
   b) Calculate niche count nc_q among solutions of P_j only.
   c) Calculate shared fitness $F_j'^{(q)} = F_j^{(q)} / nc_q$. $F_{min} = \min (F_j'^{(q)} : q \in P_j)$ and set j = j + 1.
4) If $j \leq \rho$, go to step 3. Otherwise, the process is complete.

VII. NICHED PARETO GENETIC ALGORITHM (NPGA)

Horn and Nafploitis proposed a tournament selection scheme based on Pareto dominance [2][4]. Instead of limiting the comparison to two individuals, a number of other individuals in the population were used to help to determine dominance (typically around 10). When both competitors were either dominated or non-dominated (i.e., there was a tie), the result of the tournament was decided through fitness sharing. Population sizes considerably larger than usual with other approaches were used so that the emerging niches in the population could tolerate the noise of the selection method.

NPGA Procedure

1) Shuffle P, set i = 1, and set Q = ϕ.
2) Perform the above tournament selection and find the first parent, p1 = NPGA-tournament (i, i + 1, Q).
3) Set i = i + 2 and find the second parent, p2 = NPGA-tournament (i, i + 1, Q).
4) Perform crossover with p1 and p2 and create offspring c1 and c2. Perform mutation on c1 and c2.
5) Update offspring population Q = Q U {c1, c2}.
6) Set i = i + 1. If i < N, go to step 2. Otherwise, if |Q| = N / 2, shuffle P, set i = 1, and go to step 2. Otherwise, the process is complete.

VIII. PERFORMANCE COMPARISON

There are two distinct goals in multi-objective optimization: (1) to discover solutions as close to the Pareto-optimal solutions as possible, and (2) to find solutions as diverse as possible in the obtained non-dominated front [10]. A Multi-objective GA will be termed a good Multi-objective GA, if both goals are satisfied adequately. For that Set Coverage Metric and Maximum Spread Metric are used.

A. Implementation Details

This work describes implementation of different instances of two different numbers of objectives for Traveling Salesman Problem. All the programs were coded in C language. It uses Turbo C in DOS environment.In bi-objective TSP, the objectives are to minimize distance and cost. In three-objective TSP, the objectives are to minimize distance and cost and maximize touring attractiveness which are conflictive.

B. Simulation for Multi-Objective TSP

This paper uses the MOTSP with the problem-instances for 10-cities, 25-cities, 50-cities and 60-cities with 100 population size, maximum number of generation 100, crossover rate 0.8, mutation rate 0.1, niching parameter 0.185 and domination tournament set size 10.This analysis work contains total 32 graphs for bi-objective TSP and three-objective TSP[14]. This paper contains results of 60 cities instance of all four algorithms for two-objective and three-objective TSP. The results are compared on the basis of performance metric. There are two distinct goals in multi-objective optimization: 1) Discover solutions as close as to the pareto -optimal solutions as much as possible and 2) Find solutions as diverse as possible in the obtained non-dominated front. Figure 1 to figure 4 and figure 5 to figure 8 show the solution set for 60-cities instances for two-objective and three-objective TSP respectively for VEGA, MOGA, NSGA and NPGA.

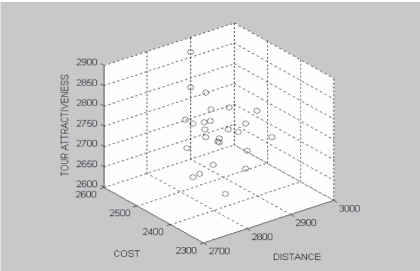


Figure 3 solution set for 60-cities (NSGA)

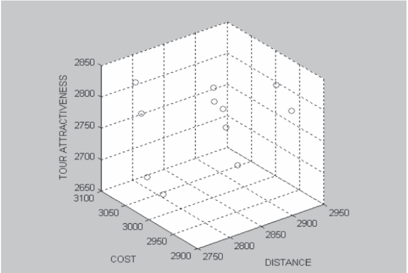
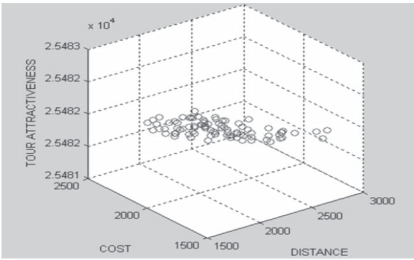
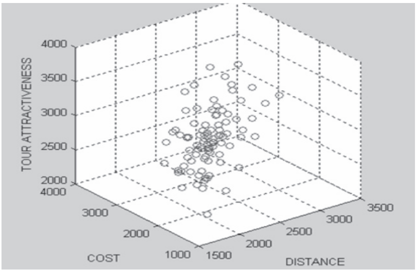


Figure 4 solution set for 60-cities (NPGA)



For bi-objective TSP, the results for average set coverage metric are : VEGA gives 63%, MOGA gives only 81% solutions and NSGA gives 97% and NPGA gives100% for 10-cities instance. For 25-cities, VEGA gives61% MOGA gives 52%, NSGA gives 100% and NPGA gives 69%. For 50-cities VEGA gives41% MOGA gives 47%, NSGA gives 98% and NPGA gives 100%. For 60-cities instances, VEGA gives56% MOGA gives 64%, NSGA gives 100% and NPGA gives 100%. For maximum spread metric also, NPGA gives good results compared to remaining three algorithms. For three-objective TSP, NPGA gives good results compared to all remaining three algorithms. For maximum spread metric, results are: for 10-cities instance VEGA gives 50, MOGA gives 43, NSGA gives 65 and NPGA gives 98. For 25-cities instance VEGA gives 453 ,MOGA gives 282 ,NSGA gives 787and NPGA gives 1540.For 50-cities instance VEGA gives 245 ,MOGA gives 257 ,NSGA gives 897 and NPGA gives 2487 . For 60-cities instance VEGA gives 368, MOGA gives273, NSGA gives1279 and NPGA gives 2657.



IX. CONCLUSIONS

Analysing the drawn results of average set coverage metric, NPGA gives the good results compared to VEGA, MOGA and NSGA since it selects individuals by directly checking them for the non-domination. Also, from the solution sets, we can see that NPGA gives more solution sets. From the results of maximum spread metric also, we can conclude that NPGA gives good results for MOTSP compared to remaining all three algorithms. From these performance metrics, it is clear that there is no effect of number of objectives on performance of NPGA. From all the above reasons we can analyse that NPGA is the best for Multi-objective travelling salesman problem and also for similar type of applications.

X. ACKNOWLEDGEMENTS

I would like to thank Dr. Y.P.Kosta for his helpful comments and suggestions.

XI.REFERENCES

[1] Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Singapore: Pearson Education.

[2] Deb, K. (2002). Multi-objective Optimization using Evolutionary Algorithms. West Sussex: John Wiley and Sons, Ltd.

[3] Coello, Carlos A. and Christiansen, Alan D. An Approach to Multi-objective Optimization using Genetic Algorithms. Department of Computer Science, Tulane University, New Orleans, LA, USA.

[4] S. Johnson and L. A. McGeoch, The Traveling Salesman Problem: A Case Study in Local Optimization, Local Search in Combinatorial Optimization, E. H. L. Aarts and J.K. Lenstra (ed), John Wiley and Sons Ltd, 1997, pp 215-310.

[5] Chang Wook Ahn. Advances in Evolutionary Algorithms. Theory, Design and Practice, Springer, ISBN 3-540-31758-9, 2006

[6] Zitzler, E. and Thiele, L. *An Evolutionary Algorithm for Multi-objective Optimization: The Strength Pareto Approach.* Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Zurich, Switzerland.

[7] Ajith Abraham, Lakhmi Jain and Robert Goldberg(editors).Evolutionary Multiobjective Optimization. Theoretical Advances and Applications, Springer, USA, 2005, ISBN 1-85233-787-7

[8] Coello, Carlos A. and Christiansen, Alan D. Two New GA based Methods for Multi-objective Optimization. Department of Computer Science, Tulane University, New Orleans, LA, USA

[9] G. B. Dantzig, R. Fulkerson, and S. M. Johnson, Solution of a large-scale traveling salesman problem, Operations Research 2 (1954), 393-410.

[10] S. Arora. Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and other Geometric Problems. Journal of ACM, 45(1998), 753-782.

[11] N. Christofides, Worst-case analysis of a new heuristic for the traveling salesman problem, Report 388, Graduate School of Industrial Administration, CMU, 1976.

[12] H. Papadimitriou, S. Vempala: On the approximability of the traveling salesman problem (extended abstract). Proceedings of STOC'2000, 126-133.

[13] G. Gutin, Traveling Salesman and Related Problems, Royal Holloway, University of London, 2003

[14] Gutin, A. Punnen, The Traveling Salesman Problem and its variations, Kluwer Academic Publishers.

Artificial Neural Network, Case Base Reasoning and Rule Based Model for English to Sanskrit Machine Translation

Vimal Mishra[1] and R. B. Mishra[2]

[1]*Research Scholar, Department of Computer Engineering, Institute of Technology*
Banaras Hindu University, (IT-BHU), Varanasi-221005. U.P., India.
Email: vimal.mishra.cse07@itbhu.ac.in , vimal.mishra.upte@gmail.com

[2]*Professor, Department of Computer Engineering, Institute of Technology*
Banaras Hindu University, (IT-BHU), Varanasi-221005. U.P., India.

Abstract— **In machine translation (MT) system, we reuse past translation experience that is encoded into a set of cases, where case is the input sentence and its corresponding translation. A case which is similar to the input sentence will be retrieved and a solution is produced by adapting its target language components. The CBR approach of machine translation is used as a learning technique in the domain of machine translation of English to Sanskrit. In our approach, the syntactical features of English language are part of the cases in the case base. The new input English sentence is matched with old cases from the stored case bases using ANN (Artificial Neural Networks) method. The retrieved case is adapted using rules. In this paper, we present the integration of CBR approach of MT with ANN and rule based model of English to Sanskrit MT, where CBR approach of MT is used for the selection of Sanskrit translation rule of the input English sentence.**

Keywords- case base reasoning, machine translation; learning technique; artificial neural Networks; rule based machine translation; English to Sanskrit machine translation;

1. INTRODUCTION

Sanskrit language is the mother of all Indian languages. If we develop English to Sanskrit machine translation system then it will easier to translate from English to any Sanskrit originating Indian languages.

Our motivation behind use of feed forward ANN in language processing tasks are work of Nemec, Peter (2003) and Khalilov, Maxim et al. (2008).

CBR is an approach to problem solving that uses prior experience during future problem solving. In CBR, new problems are solved by finding the most similar case in the case base and used this as a model for new solution through a process of adaptation (Somers, 2003). A brief review of CBR models, processes and methods are given below.

The various models of CBR have been described in the context of R^5 viz. repartition, retrieval, reuse, revise, and retain (Finnie and Sun, 2005). The various processes that has used in the Hunt's model (1995) are as input, retrieval, adaptation, evaluation and repair. Allen's model (1994) consists of the five processes viz. presentation, retrieval, adaptation, validation and update. The model of Kolodner and Leak (1996) comprised of five steps as retrieval, adaptation or Justify, criticize, evaluate, and store. The R^4 model of CBR contains four processes as retrieve, reuse, revise and retain (Aamdt and Plaza, 1994). Retrieval is the only common process in all of the five models mentioned above. Each of the process has various methods for its implementation viz. mathematical, heuristic and algorithmic ones.

Few literature sources are available which describe the uses of CBR in machine translation viz. Collins, B. (1998) and Zwarts (2003) and Somers (2003). Basically, Somers' work is an extensive review albeit a tutorial on the various processes and methods of CBR to be used in Example based machine translation (EBMT). The Case Based ReVerb system (Collins, 1998) applies CBR technique to EBMT. S. Zwarts et al. (2003) uses CBR as dependency based machine translation which is based strictly on the CBR algorithm of Aamodt and Plaza.

In our system, the new input English sentence is assigned a part-of speech (POS) tagging. After POS tagging, the gender, number and person (GNP) of the input English sentence is detected. The detection modules detect the sentence in terms of structure, form and type of sentence which are encoded into decimal coded form that is tested into ANN system to produce Sanskrit translation rule (Sans_tr_rule). We use CBR in the matching of new sentences to the previous stored cases and apply various adaptation processes viz. addition and deletion. In this paper, we apply CBR for the selection of Sanskrit translation rule of the input English sentence on our ANN and Rule based model for English to Sanskrit machine translation system (Mishra, Vimal et al., 2010c).

The rest of the work in this paper is divided into following sections. Section 2 shows system model of our EST system with diagram that describe the main module of our system. Section 3 describes CBR processes and methods. Section 4 shows the results of our system for the various types of English sentences. Section 5 concludes the paper and gives the future works.

2. SYSTEM MODEL OF OUR EST SYSTEM

Figure 1 shows the system model of EST system. The description of main module of the system model is as follows: The sentence tokenizer module splits the English sentences into tokens (words) using split method of string tokenizer class in Java. The outputs of the sentence tokenizer module are given to POS Tagger module. The POS (Part–of-Speech) tagging is the process of assigning a part–of-speech to each word in a sentence. In POS Tagger module, the Part–of-Speech (POS) tagging is done on each word in the input English sentence. The output of POS tagger module is given to rule base engine. The GNP detection module detects the gender, number and person of the noun in English sentence. The tense of English sentence is determined by using rules. The sentence detection gives the structure, form and type of sentence. The noun and object detection module gives noun for Sanskrit of the equivalent English noun. It uses ANN method for the selection of noun for Sanskrit. The adaptation rules are used to generate the word form. The root dhaatu detection module gives verb (dhaatu) for Sanskrit of the equivalent English verb. In Sanskrit, the root dhaatu denotes the root verb that depends on number and person of the noun used in sentence, e.g., in the Sanskrit sentence, "sah pustakam pathati" ("He reads book" in English), pathati is verb that is formed with "path" root dhaatu which is equivalent to "read" root verb in English. It uses ANN method for the selection of verb for Sanskrit. We apply adaptation rules to generate the required dhaatu form. The Sans_tr_Rule detection module gives the number of modules that is used in the Sanskrit translation. In this, we make input data and corresponding output data. The input data has structure, form and type of English sentence in the decimal coded form. We have stored fifty adverbs for Sanskrit of the equivalent English adverb in a database file.

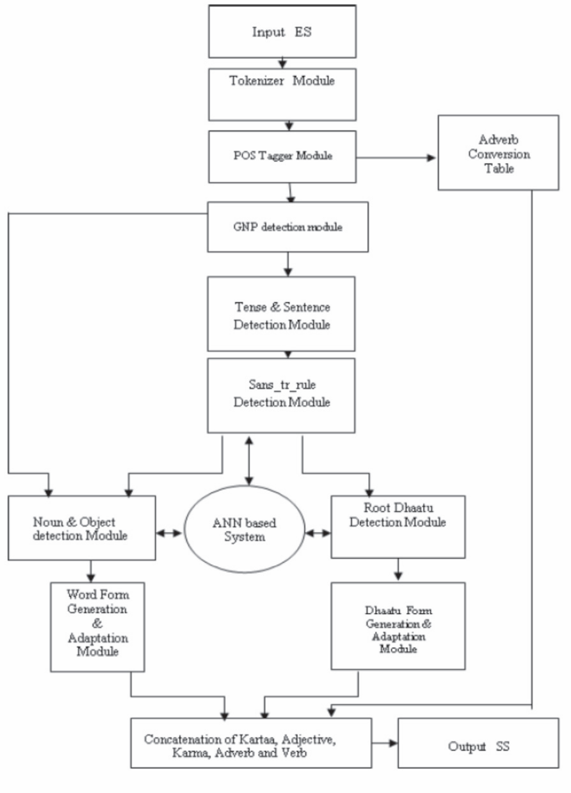

Input ES
Tokenizer Module
POS Tagger Module
Adverb Conversion Table
GNP detection module
Tense & Sentence Detection Module
Sans_tr_rule Detection Module
Noun & Object detection Module
ANN based System
Root Dhaatu Detection Module
Word Form Generation & Adaptation Module
Dhaatu Form Generation & Adaptation Module
Concatenation of Kartaa, Adjective, Karma, Adverb and Verb
Output SS

Figure1 . Information Flow in EST Model

3. CBR PROCESSES AND METHODS

The CBR cycle (Riesbeck and Schank, 1989) consists of the processes that have representation, retrieval, adaptation and reuse processes.

The main components of the knowledge in our EST is the structure of the sentence: SV, SV-Adjective, SV–Adverb, SV-Adjective–Adverb, SVO, SVO–Preposition, SVO–Adverb, SVO–Adjective, SVO–Adjective–Adverb; the form of the sentence; the three type of tense with four form of each type of tense; and four types of sentences and the part of speech. This knowledge is represented in our system using decimal coded form. We use feed forward ANN for the selection of Sanskrit translation rule (San_tr_Rule) of English sentence. We basically perform three steps in ANN based system viz. encoding of input data, input-output generation and decoding of output data. The structure of sentence consists of nine forms which are represented by four bit binary. Each structure of binary form is divided by 16 to obtain the decimal coded form of structure. The total form of sentence is 20 which are represented by five bit binary. For three tenses and four form of each type of tenses, there are twelve active forms of sentences and eight passive forms of sentences. To obtain the decimal coded form of this, we divide each binary form by 32. We use four type of sentence which is represented by three bit binary. We make the type of sentence to decimal coded form which is obtained by dividing each to 4. Each structure of sentence has twenty forms of sentence and each form of sentence has four types of sentences. Each structure of sentence has total eighty data sets which has encoded in our proposed system. After preparing the input-output data set, we train the input data set through feed forward ANN and then test the output data set. We get the translation rule (San_tr_Rule) as the output data which is in decimal coded form. We have developed rules for the decoding of output data.

The ANN method is used for retrieval of the new case from the old case. We make the new case that consists of structure, form of sentence and type of sentence into decimal coded form as we described in section 3.1. This decimal coded form is feed as input data into ANN system and after testing the input data from ANN, we get the Sans_tr_rule in decimal coded form. This Sans_tr_rule indicates the name of modules used for Sanskrit translation. In our system, the retrieval of Sanskrit translation rule (Sans_tr_Rule) of the new sentence is performed through ANN which is in decimal coded form that is based on the minimum difference of the new sentence to stored sentence. We have a data set of 720 input-output pair for input data (which is given in structure, form and type of English sentence) and output data (Sans_tr_Rule). The input, hidden and output values for this is taken 5, 62 and 9. The training is terminated at a training error of 10^{-3} after 300 epochs.

The word form of noun and verb are generated using rules. The word generation of the noun and object depend upon GNP of the English noun and the word generation of the verb depends upon number and person of the English noun (Kale, M. R., 2005; Macdonnel, Arthur A, 2003 and Nautiyal, Chakradhar, 1997). The adaptation processes involves two basic operations: addition and deletion.

In ANN based model, we basically perform three steps viz. encoding of User Data Vector (UDV), input-output generation of UDV and decoding of UDV. The name of our data sets have called UDV here, which is used in feed forward ANN for the selection of equivalent Sanskrit word and verb of English sentence (Mishra, Vimal and Mishra, R. B., 2010a; 2010b).

English alphabet consists of twenty-six characters which can be represented by five bit binary (2^5 =32, it ranges from 00000 to 11111). First, we write alphabet (a-z) into five bit binary in which alphabet "a" as 00001, to avoid the problem of divide by zero and alphabet "z" as 11010. For the training into ANN system, we make the alphabet to decimal coded form which is obtained by dividing each to thirty-two. This gives us input word in decimal coded form and output in corresponding Sanskrit word in roman script as encoded on the basis of table I.

TABLE I: ENCODING OF ENGLISH ALPHABET

S.No	Character of alphabet	5-bit binary	Decimal of binary form divide by 32
1	A	00001	0.031
2	B	00010	0.063
3	C	00011	0.094
.	.	.	.
.	.	.	.
25	Y	11001	0.781
26	Z	11010	0.813

We prepare UDV of noun (subject or object) and verb which is of maximum of five characters. In case of two characters noun or verb etc, we add three dummy values which range between 0.007 and 0.009, as suffix to UDV to make them five characters noun or verb. Similarly, we add two and one dummy values for three and four characters noun or verb etc respectively to make them five characters noun or verb.

After preparing the UDV, we train the UDV through feed forward ANN and then test the UDV. We get the output of Sanskrit word in the UDV form.

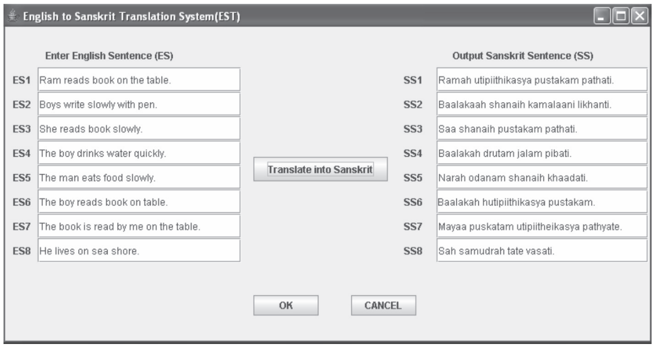
We have developed rules for the decoding of UDV for verb and noun but due to restriction of page limit, we are not describing these at here.

4 IMPLEMENTATION AND RESULTS

Our EST system has been implemented on windows platform using Java. The ANN model is implemented using MATLAB 7.1 neural Networks tool. We use feed forward ANN that gives matching of equivalent Sanskrit word of English word which handles noun and verb. We have a data set of 250 input-output pair for verb. The input, hidden and output values for verb is taken 5, 38 and 6. The training is terminated at a training error of 10^{-3} after 300 epochs. For the noun, we have 250 input-output pair in which the input, hidden and output values are taken 5, 15 and 7. This training is terminated at a training error of 10^{-2} after 300 epochs. Our EST system uses off line mode for knowledge representation. In our system, retrieval of Sanskrit translation rule (Sans_tr_Rule) of the new sentence is performed through ANN which is in decimal coded form that is based on the minimum difference of the new sentence to stored sentence. We use feed forward ANN that gives matching of input English sentence with its Sanskrit transfer rule (Sans_tr_rule) from the stored cases of case base. After applying different adaptation operation on noun and verb, we get the required form of noun and verb. Also, we use feed forward ANN that gives matching of equivalent Sanskrit word of English word which handles noun and verb. Our EST system work satisfactorily on the various features viz. structure, form and type of English sentence. In our method, we combine the explicit method of representation of rule based method and explicit method of reasoning of CBR to make it more effective as a knowledge base system. The ANN method used in our system solves two processes of CBR: one in the retrieval by similarity matching and other learning through training phase of the ANN. Learning process of CBR is as essential as adaptation and becomes very effective when used with adaptation which has deployed in our system. The figure 2 shows the sample output from our EST system for different types of sentences viz. [SVO, Preposition, Active], [SVO, Adverb, Adjective, Active], [SVO, Passive], [SVO, Preposition, Passive] and [SVO, Adverb, Adjective, Passive] class of English sentence with their Sanskrit translation.

5. CONCLUSIONS

Our work is basically integration of three approaches: rule based model, the dictionary matching by ANN model and CBR approach of MT. The CBR approach of MT is used as learning technique for selection of Sanskrit transfer rule (Sans_tr_rule) of the input English sentence. The integration of CBR with rule based model combines the representation and reasoning and ANN adds the retrieval and learning processes in the computation for MT.



	Enter English Sentence (ES)		Output Sanskrit Sentence (SS)
ES1	Ram reads book on the table.	SS1	Ramah utipiithikasya pustakam pathati.
ES2	Boys write slowly with pen.	SS2	Baalakaah shanaih kamalaani likhanti.
ES3	She reads book slowly.	SS3	Saa shanaih pustakam pathati.
ES4	The boy drinks water quickly.	SS4	Baalakah drutam jalam pibati.
ES5	The man eats food slowly.	SS5	Narah odanam shanaih khaadati.
ES6	The boy reads book on table.	SS6	Baalakah hutipiithikasya pustakam.
ES7	The book is read by me on the table.	SS7	Mayaa puskatam utipiitheikasya pathyate.
ES8	He lives on sea shore.	SS8	Sah samudrah tate vasati.

Figure 2: A sample output from EST system for various classes of English Sentences

REFERENCES

[1] Aamodt, A. and Plaza, E., 'Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches', AI Communications, Vol. 7, No. 1, pp.39–59, 1994.

[2] Allen, B.P., 'Case-based reasoning: Business applications', Communications of the ACM, Vol. 37, No. 3, and pp 40–42, 1994.

[3] Collins, B., 'Example based machine translation: adaptation guided retrieval approach', PhD thesis, Trinity college, Dublin, 1998.

[4] Finnie, G. and Sun, Z., 'R 5 model for case-based reasoning', Journal of Knowledge-Based Systems, Vol. 16, pp.59–65, 2003.

[5] Hunt, J., 'Evolutionary case based design', In: Waston, I.D. (ed.), Progress in Case-based Reasoning, LNAI 1020, pp 17-31, Springer, Berlin, 1995.

[6] Kale, M. R., '*A Higher Sanskrit Grammar'*, 4th Ed, Motilal Banarasidas Publishers Pvt. Ltd., 2005.

[7] Khalilov, Maxim et al., '*Neural Network Language Models for Translation with Limited Data*', In proceedings of 20th IEEE International Conference on Tools with Artificial, pp 445-451, 2008.

[8] Kolodner, J.L. and Leake, D.B., 'A tutorial introduction to case-based reasoning', in Leake, D. (Ed.): *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, AAAI Press/MIT Press, pp 31–65, 1996.

[9] Macdonnel, Arthur A, 'A Sanskrit Grammar for Students', 3rd Ed, Motilal Banarasidas Publishers Pvt. Ltd., 2003.

[10] Mishra, Vimal and Mishra, R. B., '*English to Sanskrit Machine Translation System: A Hybrid Model'*, In Proceedings of International Joint Conference on Information and Communication Technology (IJCICT 2010), 9th-10th January, pp 174-180, IIMT, Bhubaneswar, India, 2010a.

[11] Mishra, Vimal and Mishra, R. B., 'ANN and Rule based model for English to Sanskrit Machine Translation', INFOCOMP Journal of Computer Science, Vol. 9(1), pp 80-89, 2010b.

[12] Mishra, Vimal and Mishra, R. B., '*Approach of English to Sanskrit Machine Translation based on Case Based Reasoning, Artificial Neural Networks and Translation Rules',* International Journal of Knowledge Engineering and Soft Data Paradigms (IJKESDP), InderScience Publication, UK, Vol.2, No.4, pp 228-248, 2010c.

[13] Nautiyal, Chakradhar, '*Vrihad Anuvaad Chandrika'*, 4th Ed, Motilal Banarasidas Publishers Pvt. Ltd., India, 1997.

[14] Nemec, Peter, '*Application of Artificial Neural Networks in Morphological Tagging of Czech*', pp 1-8, 2003.

[15] Riesbeck, C.K. and Schank, R.C., *Inside Case-Based Reasoning*, Lawrence Erlbaum Associates, Cambridge MA, 1989.

[16] Somers, H., 'EBMT Seen as case-based Reasoning', Book Chapter in Recent advances in EBMT, Springer Publisher, 2003.

[17] Zwarts, S., 'Using CBR as a learning technique for natural language translation', Master's Thesis, University of Twente, 2003.

Offline Recognition of Handwritten Devnagari Characters

Aarti Desai[#1], Dr. Latesh Malik[#2]
[#] *Department of Computer Science & Engineering*
G. H. Raisoni College Of Engineerin, Nagpur, India

[1] aartikarandikar@gmail.com
[2]lgmalik@rediffmail.com

***Abstract*— Character recognition has long been a critical area in the field of OCR (Optical Character Recognition).In this paper we present a new set of features for recognition of Devnagari characters. modified algorithm for Devnagari character recognition. Devnagari characters are difficult to recognize as compared to English characters. The proposed method consists of 4 major steps : 1) binarization 2) thinning 3) feature extraction 4) recognition. Here the feature vector comprises of number of endpoints for the character, number of branch points for the character, type of spine present, type of shirorekha present, chain codes. We have ensured that the basic characteristics of the image are maintained while keeping the algorithm as simple as possible. The performance of the algorithm after testing is demonstrated .**

***Keywords*- Character recognition , thinning , Devnagari characters.**

1. INTRODUCTION

Character recognition has long been a critical area in the field of OCR (Optical Character Recognition). Offline character recognition has achieved a great attention for many years due to its contribution in the digital library evolution. In this paper we concentrate on the handwritten Devnagari script. Handwriting recognition is described as the ability of a computer to translate human writing into text.

Methods and recognition rates depend on the number of constraints on handwriting. The offline recognition process operates on pictures generated by an optical scanner. The data is two-dimensional and space ordered which means that overlapping characters can cause segmentation problems.

The paper is organized as follows : Section 2 contains a brief introduction to the Devnagari script. Section 3 contains the proposed technique which is followed by section 4 which contains the experimental results which are quite promising.

2. DEVNAGARI SCRIPT

The Devnagari script is the most widely used Indian Script. It is a moderately complex pattern. Unlike simple juxtaposition in Roman script, a word in Devnagari script is composed of composite characters joined by a horizontal line at the top. The basic alphabet set of Devnagari is very large comprising of about 13 vowels, 34 consonants and 14 matras. The number goes up once half letter forms are also considered. It is used as the writing system for over 28 languages including Sanskrit, Hindi, Kashmiri, Marathi and Nepali. Devnagari characters are joined by a horizontal bar (Shirorekha) that creates an imaginary line by which Devnagari text is suspended. A single or double vertical line called a Danda (Spine) was traditionally used to indicate the end of phrase or sentence. Figure 1 below shows the basic and special character set of Devnagari script.

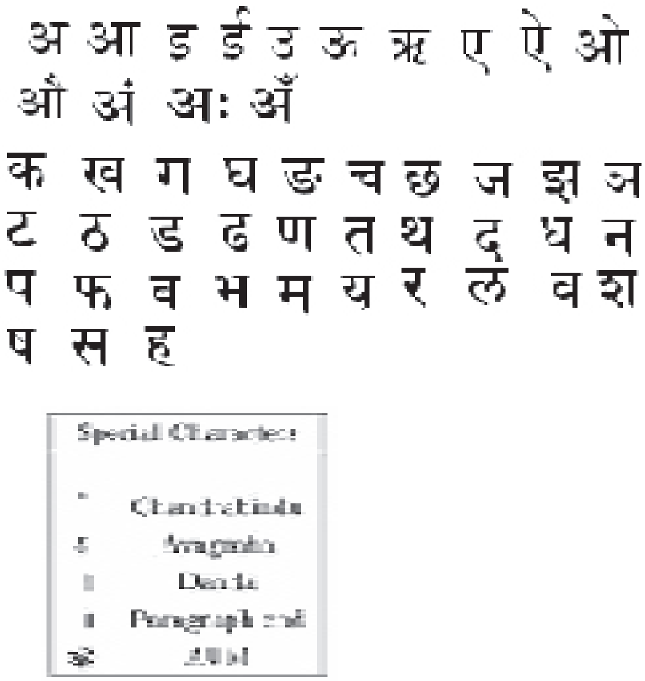
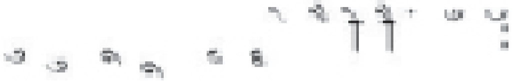


Figure 1 : Basic and special character set of Devnagari script.

The script has its own specified composition rules for combining vowels, consonants and modifiers. Vowels are used to produce their own sound or they are used to modify the sound of a consonant by attaching an appropriate modifier in an appropriate manner with them. Figure 2 below shows the modifiers in Devnagari script.

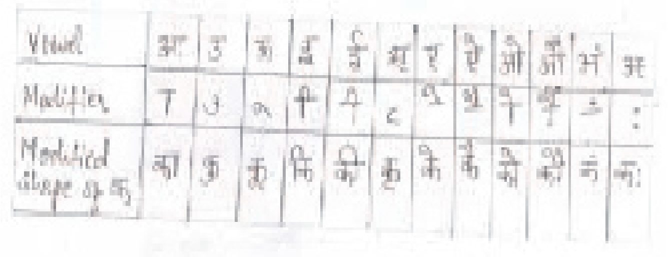
Modifier symbols are placed on top, bottom, left, right or on a combination of these. The consonants may also have a half form or shadow form. A half character is written touching the following character resulting in a composite character. In part, Devnagari owes its complexity to its rich



set of conjuncts.

Figure 2 : Modifiers

Modifier symbols are placed on top, bottom, left, right or on a combination of these. The consonants may also have a half form or shadow form. A half character is written touching the following character resulting in a composite character. In part,



Vowel	आ	उ	ऊ	इ	ई	ऋ	ए	ऐ	ओ	औ	अं	अः
Modifier	ा	ु	ू	ि	ी	ृ	े	ै	ो	ौ	ं	ः
Modified shape of क	का	कु	कू	कि	की	कृ	के	कै	को	कौ	कं	कः

Figure 3 : The modifier and the modified consonant.

3. PROPOSED TECHNIQUE

The paper attempts to present a method, which operates for identifying Devnagari characters in the database using following steps : data collection, preprocessing , feature extraction, recognition.

A. Data Collection

We performed our experiments over a database generated by collecting handwritten samples of Devnagari characters from 5 writers. The image files are stored in jpg format. Our database consists of over 150 samples. The test samples were also generated in the same manner.

B. Preprocessing

Preprocessing plays an important role in an OCR system. In this method preprocessing consists of : binarization, size normalization and thinning.

i) Binarization : In binarization , the preprocessor converts the image in the given file format into bitmap and converts the bitmap into array of bits having values 0 or 1, where 0 means a background pixel and 1 means a foreground pixel.

ii) Size normalization : It is necessary because the characters written by hand vary greatly in size and shape. To account for variety in shape, the character is normalized. By normalization, we mean that the character is made to fit into a standard size array. This size of array is chosen by trial and error method & the value that gives the best results is fixed. Characters of any size and shape can be processed and matched with the normalization technique.

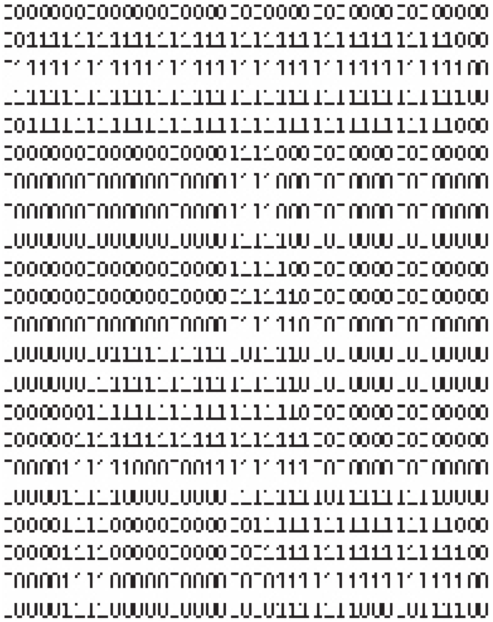


Figure 4 : Binarized image

Let the height & width of the character be denoted 'h' & 'w' respectively. Thus normalizing factor x normal and y normal in X & Y directions are given by -

o x normal = array size / w

o y normal = array size / h

Then if the pixel on the screen has the co-ordinates x' and y' relative to the top left corner of the bounding rectangle of the character, the X and Y co-ordinate of that pixel in the standard size array are -

o x = x' * x normal

o y = y' * y normal

Thus the final result is the 2-D integer array i.e. the standard approximation of the character. If the color of the pixel on the screen is not the same as background color then the corresponding array element is filled as '1' else it is filled as '0'. The normalization can be briefly represented by the following figure -

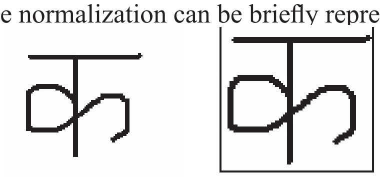


Figure 5: Size normalization character 'ka'.

iii) Thinning : Thinning of Devnagari characters is a very difficult task due to presence of loops and conjuncts. In our approach, we have developed a modified thinning which works as follows : to the normalized binarized image, we apply the following structuring elements (figure 6).

The procedure is repeated till no further changes occur. The output image, i.e the binarized thin image obtained after applying structuring elements still contains noise . We remove this noise by checking whether a black pixel can be safely converted into a white pixel. For this we search in the

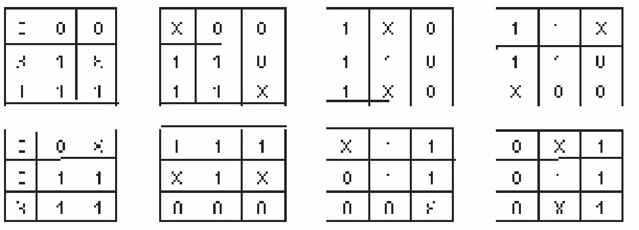


Figure 6: Structuring elements.

neighborhood of the black pixel i.e we find out the number of black pixels around the black pixel under
consideration. If this number is less than 2 then it is not eligible to be removed. But for more than 2 numbers, we count the number of white-black color combinations around the considered pixel. If this number is not equal to 1 then do nothing. But if that number is 1 then following procedure is followed : If (i,j) is considered as a black pixel (see Figure 8) then it is converted to white :
(i) if (i,j+1) or (i+1, j) or both (i-1,j) and (i,j-1) are white.
(ii) if (i-1,j) or (i,j-1) or both (i, j+1) and (i+1,j) are white. The output image is as shown in figure 7.

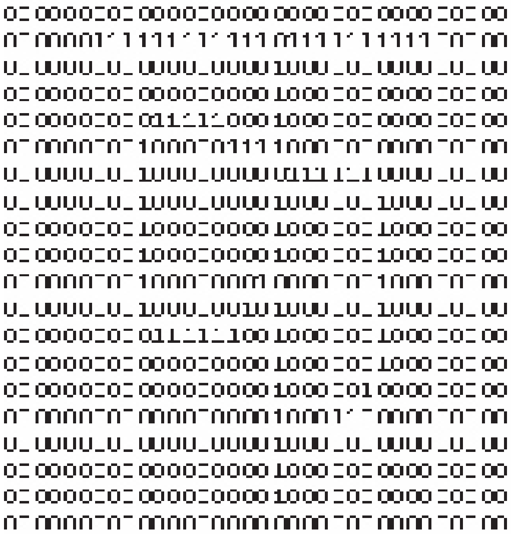


Figure 7 : Final thinned image

Figure 8 : 3X3 Window frame of considered pixel and its

i-1, j-1	i-1, j	i-1, j+1
i, j-1	i, j	i, j+1
i+1, j-1	i+1, j	i+1, j+1

surroundings 8 pixels

C. *Feature Extraction*

Following features are extracted in this step :

i) Endpoints : An endpoint is a point that is connected only from one side. We store the number of endpoints present in a character.

ii) Branch points : A branch point is a point that is connected by three sides. We store the number of branch points present in a character.

iii) Shirorekha detection : A unique feature of Devnagari Characters is presence of header line (shirorekha). We scan the character from left to right and a horizontal line that is above 1/3 part of the character is the shirorekha. It is assumed that the shirorekha is present in upper ½ part of the character.

iv) Vertical spine detection : A vertical spine is present in many Devnagari characters. A straight line to qualify as a spine should be at least ¾ of the height of the character. Type of spine (present in mid or at the end of the character or no spine present) is also detected.

v) Chain codes : We used Freeman chain coding for detecting loops and curves in a character. Chain codes are used to represent a boundary by a connected sequence of straight line segments of specified length and direction. The direction of each segment is coded by using a numbering scheme.

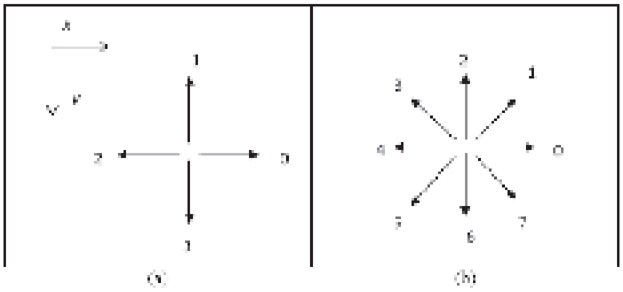
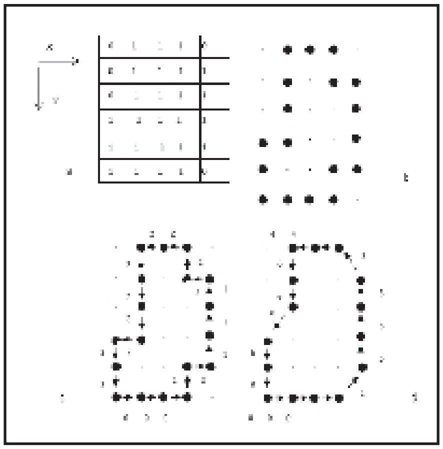


Figure 9 : Direction numbers for (a) 4-directional chain codes
(b) 8-directional chain code.

A chain code can be generated by following a boundary of an object in a clockwise direction and assigning a direction to the segments connecting every pair of pixels. First, we pick a starting pixel location anywhere on the object boundary. Our aim is to find the next pixel in the boundary. There must be an adjoining boundary pixel at one of the eight locations surrounding the current boundary pixel. By looking at each of the eight adjoining pixels, we will find at least one that is also a boundary pixel. Depending on which one it is, we assign a numeric code of between 0 and 7 as already shown in Figure 9.
For example, if the pixel found is located at the right of the current location or pixel, a code “0” is assigned. If the found is directly to the upper right, a code “1” is assigned. The process of locating the next boundary pixel and assigning a code is repeated until we came back to our first location or boundary pixel. The result is a list of chain codes showing the direction taken in moving from each boundary pixel to the next. The process of finding the boundary pixel and assigning a code is shown in Figure 10.

Figure 10 : a & b) A 4-connected object and its boundary; c & d) Obtaining



the chain code from the object in (a & b) with (c) for 4-connected and (d) for 8-connected pixel.

Given a scaled binary image, we first find the contour points of the character image. The chain code for the character contour will yield a smooth, unbroken curve as it grows along the perimeter of the character and completely encompasses the character. When there is multiple connectivity in the character, then there can be multiple chain codes to represent the contour of the character. We chose to move with minimum chain code number first. We divide the contour image in 5 × 5 blocks. In each of these blocks, the frequency of the direction code is computed and a histogram of chain code is prepared for each block. Thus for 5 × 5 blocks we get 5 × 5 × 8 = 200 features for recognition.

4. TESTS AND RESULTS

We store all features of each input character in a feature vector. Thus feature vectors of all characters in the database are constructed. A feature vector for the test sample is also constructed. We have used the minimum edit distance algorithm for final recognition. The algorithm calculates the similarity between two strings, say the source string s and the target string t. The distance is the number of deletions, insertions, or substitutions required to transform s into t.

We compare the feature vector of the test sample against the ones present in out database and find the minimum value amongst them. The minimum value is our matched string. The recognition rate of our method is nearly 87%.

5. CONCLUSIONS

A new approach of recognizing Devnagari characters is proposed here. The developed architecture is robust in the recognition of Devnagari characters. The algotithm worked well for all character images in our database.

REFERENCES

[1] Sandhya Arora, Latesh Malik and Debotosh Bhattachrajee, “ A Novel Approach For Handwritten Devnagari Recognition” in IEEE -International Conference on Signal And Image Processing, Hubli, Karnataka, Dec 7-9, 2006.

[2] M. Tellache, M. A. Sid-Ahmed, B. Abaza, “ Thinning algorithms for Arabic OCR” IEEE Pac Rim 1993. pp 248-251.

[3] RW Zhou, C Quek, GS Ng, “A novel single-pass thinning algorithm and an effective set of performance criteria” Pattern Recognition Letters 16 (1995) pp 1267-1275

[4] S. Ahmed, M. Sharmin and Chowdhury Mofizur Rahman , “A Generic Thinning Algorithm with Better Performance” Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT 2002), Bangladesh, Dec 2002, pp. 241-246.

[5] Louisa Lam, Seong-Whan Lee, “Thinning Methodologies -A Comprehensive Survey” Ieee Transactions On Pattern Analysis And Machine Intelligence, Vol. 14, No. 9, September 1992 , pp 869-885.

[6] B. B. Chowdhury and U.Pal, “A complete Printed Characterecognition Bangla OCR System”, vol. 31(5), 1998, pp. 531-549.

[7] D. Akhter and M. M. Ali, “A Fast Thinning Algorithm for Bangla Characters”, Proc.ICCIT, Dhala, Bangladesh, 1998, pp. 132-136

[8] Rafel C. Gonzalez and Richard Woods, “Digital Image Processing”, Second Edition, Pearson education, 2004.

[9] R. Bajaj, L. Dey, S. Chaudhury, “Devnagari numeral recognition by combining decision of multiple connectionist classifier” , Sadhana 27 (2002) 59-72

[10] E.R. Davies and A.P. Plummer, “Thinning Algorithms: A critique and new Methodology” ,Pattern Recognition 14, 1981,53-63

A Hybrid Evolutionary Optimization Approach for PI Controller Tuning Based on Gain and Phase Margin Specifications

Subhojit Ghosh[#], Prateek Puri and Padm Kant
Department of Electrical and Electronics Engineering
Birla Institute of Technology, Mesra, Rnachi
[#] *e mail: aceghosh@gmail.com.*

Abstract— **In this paper, a two-stage hybrid optimization algorithm has been developed for the tuning of PI controller based on frequency domain specifications. The proposed algorithm involves the combination of an evolutionary technique (Genetic algorithm) and a pattern search based (Hooke-Jeeves) method. The parameter tuning process has been framed as an optimization problem by considering an objective function based on gain and phase margin specifications. In the proposed algorithm, a global search is first carried out over the search space to determine an initial set of desired parameters using genetic algorithm (GA). The search is then refined in the second stage using Hooke-Jeeves method (initialized using the GA solution). The combination of a global and local technique offers the advantages of both the optimization techniques along with counteracting their disadvantages. In addition, the method does not rely on any numerical approximation. For a given transfer function, the controller parameters are able to meet the desired specifications with greater accuracy as compared to the existing techniques.**

Keywords: **Controller tuning, Gain margin, Phase margin, Hybrid Optimization, Genetic Algorithm.**

I. INTRODUCTION

Inspite, of the recent developments in intelligent control systems, Proportional-integral-derivative (PID) controllers are still widely used in industries owing to their simple structure and robust performance over a wide range of operating conditions. Tuning of PI and PID controllers have been widely reported in literature. The commonly used techniques include Ziegler-Nichols step response [1], Cohen Coon reaction curve [2], absolute/integral error minimization [3] and internal mode control (IMC) [4]. As against the more general approach of parameter tuning based on the dynamical behavior of the system, design based on frequency domain specifications (gain margin and phase margin) gives an additional measure about the robustness of the system. Also, it has a direct correspondence with the performance and stability of the closed loop system. Tuning of controller to satisfy frequency domain specifications involve solution of set of nonlinear equations corresponding to phase margin and gain margin. However, the complex nonlinearities as well as the coupling among the variables hinder obtaining the closed form solution of the equations. The solution is obtained by numerical methods [5] or approximation of the nonlinear functions [6]. The final solution of the numerical method based approaches is heavily dependent on the initial condition and the convergence is not always guaranteed. Whereas, the approximated techniques are not able to achieve high accuracy and at the same time are not suitable for use in adaptive control and auto tuning [7]. In this paper, the PI controller design has been formulated as an optimization problem, with the aim of minimizing the deviation between the desired specifications and the PI controlled closed loop system specifications. The two-stage algorithm merges a global search approach with a local search based method. This combination incorporates the advantages of both the algorithms while offsetting their individual limitations. The approach does not involve any approximation or numerical technique, allows faster convergence along with independency from initial parameterization.

The organization of the paper is as follows: In the next section, we briefly introduce the gain margin and phase margin related equations, Section 3 outlines the basics of the genetic algorithm and Hooke-Jeeves method along with the formulation of the implemented hybrid optimization problem. Simulation results are discussed in section 4 and finally section 5 provides conclusion.

II. GAIN MARGIN AND PHASE MARGIN

Considering the open loop transfer function to be given by $G_p(s)$ as

$$G_p(s) = \frac{K(1+\omega_{z_1}s)^{r_1} \quad (1+\omega_{z_m}s)^{r_m}}{(1+\omega_{p_1}s)^{p_1} \quad (1+\omega_{p_n}s)^{p_n}} e^{-Ls} \quad (m \geq n) \tag{1}$$

It is desired to achieve a gain margin and phase margin of A_m and Φ_m with the addition of a forward path PI controller given as

$$G_c(s) = K_p\left(1+\frac{1}{sT_I}\right) \tag{2}$$

From the definition of gain margin and phase margin

$$\arg[G_c(j\omega_p)G_c(j\omega_p)] = -\pi \tag{3}$$

$$A_m = \left|G_c(j\omega_p)G_p(j\omega_p)\right|^{-1} \tag{4}$$

$$\left|G_c(j\omega_g)G_p(j\omega_g)\right| = 1 \tag{5}$$

(6)

where ω_p and ω_g are phase and gain crossover frequency. Substituting for (1) and (2) in equations (3)-(6)

$$\arg[G_c(j\omega_g)G_p(j\omega_g)]+\pi = \phi_m$$

$$\frac{\pi}{2}+\tan^{-1}(\omega_p T_1)+z_1\tan^{-1}(\omega_p\omega_{z_1})+ \quad +z_k\tan^{-1}(\omega_p\omega_{z_m})$$
$$-\omega_p L - p_1\tan^{-1}(\omega_p\omega_{p_1}) - p_2\tan^{-1}(\omega_p\omega_{p_2}) \quad - p_k\tan^{-1}(\omega_p\omega_{p_m}) = 0 \tag{7}$$

$$A_m K_c K_p = \omega_p T_1 \frac{(1+\omega_p^2 T_1^2)^{0.5}}{\sqrt{(1+\omega_p^2\omega_{p_1}^2)^{p_1}}} \quad \frac{(1+\omega_p^2\omega_{z_m}^2)^{(0.5z_m)}}{(1+\omega_p^2\omega_{p_m}^2)^{(0.5p_m)}} \tag{8}$$

$$K_c K_p = \omega_g T_1 \frac{(1+\omega_g^2\omega_{p_1}^2)^{(0.5p_1)}}{\sqrt{(1+\omega_g^2 T_1^2)}} \quad \frac{(1+\omega_g^2\omega_{p_m}^2)^{(0.5p_m)}}{(1+\omega_g^2\omega_{z_m}^2)^{(0.5z_m)}} \tag{9}$$

$$\phi_m = \frac{\pi}{2}+\tan^{-1}(\omega_g T_1)+ \quad + Z_n\tan^{-1}(\omega_g\omega_{z_n}) \tag{10}$$

Tuning of PI controller involves solving the above nonlinear equations ((7)-(10)) to determine K_P and T_1 for given plant parameter $(K, \omega_{z_1}, \omega_{z_2}, \quad, \omega_{z_m}, \omega_{p_1}, \omega_{p_2}, \quad, \omega_{p_m}, L)$ and desired specifications (A_m and Φ_m). The presence of nonlinear functions hinders obtaining the closed form analytical solution. Numerical techniques based on the approximation of the *tan^{-1}* functions ([4], Ho et al. 1995) are able to meet the desired specifications but with a 10 % error margin. In this regard, a hybrid optimization approach for solving the above nonlinear equations has been proposed in the present work. The next section deals with the formulation of the proposed optimization algorithm.

III. HYBRID OPTIMIZATION FOR PI CONTROLLER TUNING

In the present problem, the solution of the nonlinear equations (5)-(8) has been framed as an optimization problem based on the minimization of an objective function obtained from the corresponding equations. For this purpose, a hybrid approach by coupling an evolutionary technique with a local search algorithm has been proposed. Traditional optimization techniques based on local search though providing faster convergence have the serious drawback of being trapped into the local optimum point. Whereas, evolutionary optimization methods like genetic algorithm [8] minimize the risk of getting converge to the local optimum because of the simultaneous processing of the entire search space. However, genetic algorithms require a large number of iterations for arriving at a solution with slower convergence, especially in the vicinity of the global optimum point. In this regard, a hybrid optimization algorithm has been proposed by incorporating a faster local optimization (Hooke-Jeeves [9]) technique into genetic algorithm. This allows faster convergence without getting trapped into the local optimum. In contrast to the existing approaches of combining local search with genetic algorithm [10], that incorporates local *search into the genetic operators, the proposed approach uses the solution obtained from genetic algorithm for the initialization of local search. This section gives a brief description on genetic algorithm and Hooke-Jeeves method followed by a description of the proposed approach.*

Hooke-Jeeves Algorithm

Hooke-Jeeves is a generalized pattern search based algorithm that searches along all the possible coordinate directions for a decrease in the objective function. The search space is iteratively adjusted by a combination of exploratory moves and heuristic pattern moves. An exploratory move is performed in the vicinity of the current base point (initially selected randomly) to find the best solution around it.

Genetic Algorithm

Genetic algorithm is an effective search method based on the Darwanian evolution theory of the survival of the fittest. Unlike conventional optimization algorithms, GA works on coded parameters and not on actual parameters. Since it does not include calculation of the derivative of the objective function, it can work even on non-smooth discontinuous functions. This advantage allows GA to solve complex multivariable problems involving multiple local optimum points, which is difficult or impossible to be solved by traditional methods. GA comprises of three major operations: reproduction, crossover and mutation. The tuning parameters are population size, selection rate, crossover and mutation probability. For an initially randomly selected set of points (population) in the search space, the selection operator chooses the best point (chromosome). New solutions are obtained by the crossover of selected chromosomes. Mutation operator maintain variability in the population by inducing random changes in the chromosomes.

Hybrid Optimization Algorithm

The flowchart of the proposed two-stage hybrid optimization algorithm is shown in Figure 1. In the first stage, a global search is carried out over the search space using genetic algorithm. The genetic algorithm operators are selected based on their effect in minimizing the fitness function and time of convergence of the algorithm. The fitness function is determined based on the phase and gain margin equations (7-10). Double point crossover is used conjunction with elitist strategy based tournament selection. The elitist strategy, which is able to preserve superior strings, is incorporated in the present work by replacing the worst string of a particular generation with the best string of the previous generation. Preserving more then one string to the future generations resulted in less diversity in the population without any appreciable improvement in the fitness function. The number of chromosomes in the initial population and the maximum number of generations is set at 20 and 100 in each run. The crossover and mutation probability was fixed at 0.8 and 0.1 respectively. The reproduction, crossover and mutation are repeated till termination criterion is satisfied.

In the second stage, the solution derived genetic algorithm is further refined by implementing a local search using Hooke-Jeeves method. The search directions are created iteratively such that they completely span the search space using a

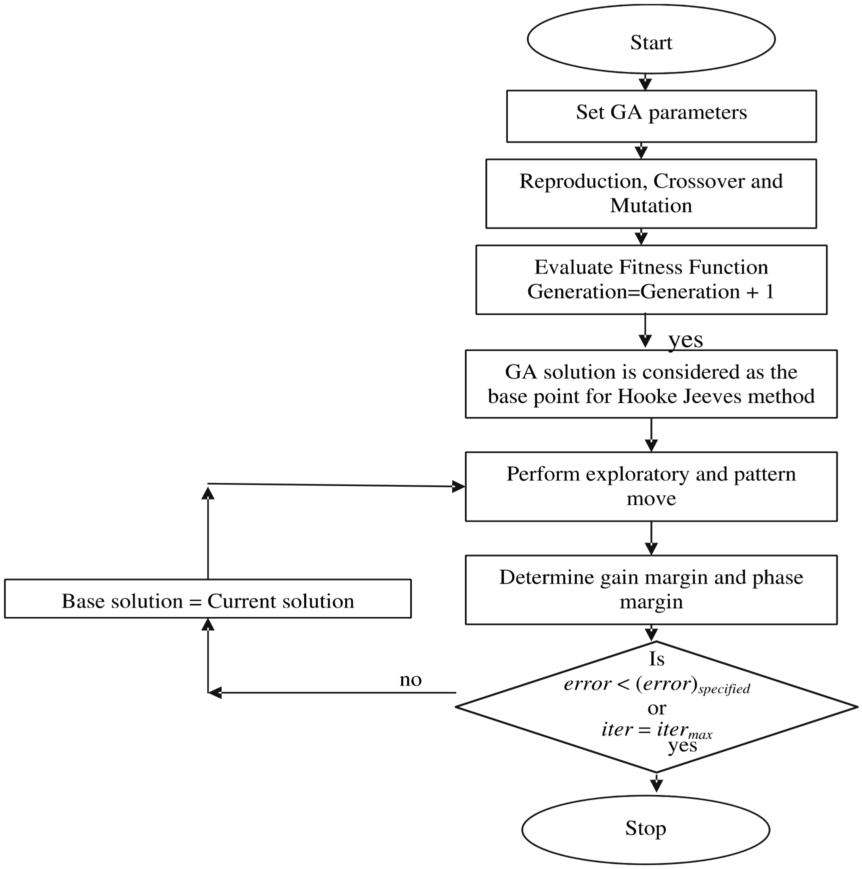

Start
Set GA parameters
Reproduction, Crossover and Mutation
Evaluate Fitness Function Generation=Generation + 1
yes
GA solution is considered as the base point for Hooke Jeeves method
Perform exploratory and pattern move
Base solution = Current solution
Determine gain margin and phase margin
no
Is
$error < (error)_{specified}$
or
$iter = iter_{max}$
yes
Stop

Figure 1: Hybrid Optimization Algorithm for PI Controller tuning.

combination of exploratory and heuristic pattern moves. The objective function considered for the Hooke-Jeeves method is the sum of absolute errors in gain and phase margin. During the exploratory move, perturbations are made around the current base point in the search space. If the search results in a better solution, then in the pattern move, a new point is found in the direction connecting the previous best point and the current solution. As compared to the standard genetic algorithm, the coupling with local search not only prevents the proposed algorithm from getting trapped into the local minima, but also results in faster convergence.

IV. SIMULATION RESULTS

In this section, the effectiveness of the proposed approach in meeting the desired specifications has been investigated for a fifth order system. Also a relative comparison in achieving the gain and phase margin has been made with GPM (Ho et al. 1995) and fuzzy neural network (FNN) [7] based methods. The plant is given as

$$G_p(s) = \frac{1}{(s+1)^5} \tag{11}$$

Different sets of specifications i.e. (5 dB, 75°), (3 dB, 60°) and (2.5 dB,70°) are considered for designing the PI controller. Table 1 reports the parameters and specifications achieved with the proposed method for a desired gain and phase margin of 5 dB and 75° respectively.

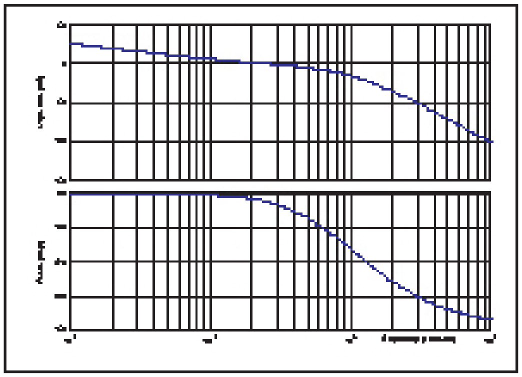


Figure 1: Bode plot for the PI controlled plant (11) with desired specification, $A_m = 2.5$ dB and $\Phi_m = 70^0$.

Table 2 compares its performance with the FNN and GPM based methods. As mentioned earlier the high error for GPM is attribute The bode plot of the PI controlled system for a sets of specifications of Table 2 is shown in Figure 1.

I. CONCLUSIONS

In this paper, a hybrid optimization algorithm has been proposed to determine the parameters of a PI controller for a higher order system. The proposed algorithm is a combination of an evolutionary algorithm and a pattern search method. The algorithm involves a local search using Hooke-Jeeves method on the defined parameter space, which is initialized through global search by genetic algorithm. The incorporation of local search along with genetic algorithm significantly improves the solution with faster convergence as compared to standard genetic algorithm. Simulation results show that the controller is able to meet the desired specifications quite accurately. Future work in this direction would involve incorporations of constraints related to the bandwidth and speed of the response of the system.

Table 1: Controller parameters and specifications obtained from the proposed hybrid optimization algorithm for the plant given by (11).

Desired Gain Margin (dB)	Desired Phase Margin (degree)	K_c	T_i	ω_g	ω_p	Achieved Gain Margin (dB)	Achieved Phase Margin (degree)
5	75	0.3528	2.9629	0.1220	0.5755	5	75.0932

Table 2: Comparison of the proposed hybrid optimization algorithm with GPM[6] and FNN[7]

TUNING METHOD	Desired Gain Marging (dB)	Desired Phase Marging (degree)	K_c	T_i	ω_g	ω_p	Achieved Gain Margin (dB)	Achieved Phase Margin (degree)
GPM [6]	3	60	0.4879	2.7300	0.1771	0.4828	3.8528	59.026
2.5	70	0.6350	3.7335	0.1823	0.5319	3.9115	65.728	
FNN [7]	3	60	0.5559	2.7281	0.2109	0.5621	3.0051	60.355
	2.5	70	0.8723	4.6649	0.2456	0.6297	2.5141	69.705
HYBRID	3	60	0.5510	2.6916	0.2111	0.5611	3.00	60
OPTIMIZATION	2.5	70	0.8733	4.7033	0.2464	0.6315	2.50	70

REFERENCES

[1] J.B. Ziegler, and N.B. Nichols (1942), Optimum settings for automatic controllers, Trans. ASME 64 , pp. 759-768.

[2] G.H. Cohen and G.A. Coon (1953), Theoretical consideration of retarded control, Trans. ASME 75, pp. 827-834.

[3] K.J. Astrom and T. Hagglund (1984), Automatic tuning of simple regulators with specifications on phase and amplitude margins, Automatica, Vol. 20, pp. 645–651.

[4] I. L. Chien and P.S. Fruehauf (1990), Consider IMC tuning to improve controller performance, Chemical Engineering Progress**.,** 86, pp. 33-41.

[5] Kaya, I. (2004), Tuning PI controllers for stable processes with specifications on gain margins and phase margins, ISA Transactions, 43, 297–304.

[6] W.K. Ho, C.C Hang and L.S. Cao (1995), Tuning of PID controllers based on gain and phase margin specification, Automatica , Vol. 31, pp. 497- 502.

[7] Chu, S.Y. and Teng, C.C. (1999), Tuning of PID controllers based on gain and phase margin specifications using fuzzy neural network, Fuzzy Sets and Systems, Vol. 101, pp. 21–30.

[8] D.E. Goldberg (1989), Genetic Algorithms in Search, Optimization, and Machine Learning, Reading, MA, Addison-Wesley.

[9] K. Deb(1988), Optimization for Engineering Design: Algorithms and Examples, India, Prentice-Hall.

[10] Y.G. Xu, G.R. Li and Z.P. Wu (2001), A Novel Hybrid Genetic Using Local Optimizer Based on Heuristic Pattern Move, Applied Artificial Intelligence, Vol.15, No. 7, pp. 601-63.

Particle Swarm Intelligence for Neural Network Classifier

Ms.Jayshri D.Dhande[1], Mr. D.R.Dandekar[2]
[1]*BDCE/Department of Electronics &Telecom. Engg. Wardha, India*
[2]*BDCE/Department of Electronics Engg. Wardha, India*
[1]Email: jayshridhande@rediffmail.com
[2]Email: d.dandekar@rediffmail.com

Abstract –**Machine learning has become one of the most popular classification methods worldwide in the field of decision support. *To* solve the classification task, the design of the most optimal automated classifier is a demanding problem, due to number of parameters to be set at the same time. This research was outlined the motivation for using an ANNs with the assistance of evolutionary algorithm such as particle swarm optimization to be efficient tool in finding the most optimal classifier. In this paper, we present MLP NN classifier & RBF NN classifier for radar data. This paper investigates and designs an optimal classifier using RBF NN with average accuracy 99.59%. And proposed another technique such as Particle Swarm Intelligence for improve the performance of various classifiers in terms of classification accuracy.**

Keywords: ANN, Classifier, MLPNN, RBFNN

I.INTRODUCTION

Classification is one of the important decision making tasks for many real world problems. Classification will be used when an object needs to be classified into a predefined class or group based on attributes of that object. There are many real world application that can be categorized as classification problems such as weather forecast, credit risk evaluation, medical diagnosis, bankruptcy prediction, speech recognition ,handwritten character recognition and quality control[1].

Generally there are two types of classification problems: binary problem and multiclass problem. While a binary problem is a situation in which an outcome of prediction has to be determined with a decision of yes or No, a multiple classification problem is a condition in which a predicted result is determined as multiple outcomes [2]. In order to solve the classification problems and prediction, many classification techniques have been proposed some of the successful techniques are Artificial Neural Networks (ANN), support vector machines (SVM) and classification trees. There are a number of other techniques that can also be applied to classification problems, for example linear regression , logistic regression discriminate analysis , genetic algorithms, fuzzy logic ,Bayesian networks and k-nearest neighbor techniques[3].Moreover, a number of hybrid techniques have also been implemented such as neuro-fuzzy based classification technique[5],.recursive partitioning of the majority class (REPMAC)algorithm[6],fuzzy probabilistic neural network[7].

In this paper, a design of optimal classifier for classification of radar returns from the ionosphere has been proposed using Neural Network with Particle swarm optimization. The radar data is obtained from Johns Hopkins university Ionosphere database[8].This radar data was collected by a system in Goose Bay ,Labrador .This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4kwatts. The targets were free electrons in the ionosphere “Good” radar returns are those showing evidence of some types of structure in the ionosphere “Bad” returns are those that do not pass the signal through ionosphere .Received signals were processed using an auto correlation function whose arguments are the time a pulse and pulse pulse number. There are 17 pulse numbers for the Goose Bay system. Thus, are 34 continuous valued attributes with respect to inputs and one additional attribute denoting class that is either “good” or “bad” according to the definition summarized above. This is a binary classification task. There are total 351 instances in this database.

II.PERVIOUS WORK

Classification problem is a decision making task where many research have been working on. There are number of techniques proposed to perform classification. Neural network is one of the artificial intelligent techniques that have many successful examples when applying to this problem.

In paper[9]the authors uses the different classification algorithms based on Artificial Neural Network such as the Perceptron, back propagation network & probabilistic neural network & concerned the performance of the networks.

p.Jeatrakul and K.W.Wong[10]the author presents a comparison of neural network techniques for binary classification problems. The classification performance obtained by five different types of neural networks. For comparison are back propagation Neural network(BPNN),Radial basis function Neural Network(RBFNN), General Regression Neural Network(GRNN), probabilistic Neural Network(PNN) & complementary Neural Network(CMTNN).The comparison done based on three benchmark datasets obtained from UCI machine learning repository.

In paper[11]author design optimal classifier using MLP NN trained with back propagation algorithm on ionosphere dataset. Using the first 200 instances for training which were carefully split almost 50% positive and 50% negative (equiprobable),they found that MLP NN trained standard back propagation algorithm attained an average of about 96% accuracy on the remaining 151 test instances, consisting of 124 "good" & only 27 "bad" instances. Accuracy on "good "instances were much higher than for "bad "instances. Back propagation algorithm was tested with several different numbers of hidden units (in [0 15]) and incremental results were also reported (corresponding to how well the different variants of back propagation did after a periodic number of epochs).

It appears from the literature review that for the multilayer preceptron (MLP) NN trained with back propagation reported average classification accuracy was about 96% on the test instances. This paper proposed RBF Neural network with particle swarm optimization technique for classification.

III.PROPOSED METHOD

The outline of the proposed approach is to design NN classifier model and achieved best classification accuracy. In the following sections each step will be discuss in detail.

A.MLP Neural Network Classifier

It is shown that from the literature review, a MLPNN having a single layer of neurons could classify a set of points perfectly if they were linearly separable.MLPNN having three layers of weight can generate arbitrary decision regions which may be non-convex and disjoint. However arbitrary decision boundaries cannot be generated with just two layers of weight. MLPNN is based on processing elements, which compute a nonlinear function of the scalar product of the the input vector and a weight vector. Its configuration is determined by the number of hidden layers, numbers of the neurons in teach of the hidden layers as well as the type of the activation functions used for the neurons. EBP algorithm is used for determining the connection weights (free parameters) from samples The network is generalized on test data.

the output layer for determination of output weights. It is impossible to suggest an appropriate number of Gaussians

B.RBF NN Classifier

RBF NN a nearest neighbor classifier. It uses Gaussian transfer functions having radial symmetry. The centers and widths of the Gaussian (radial basis functions) are set by unsupervised learning rules ,and supervised learning is applied to (cluster centers), because it is problem dependent. The behavior of any arbitrary function f(x) is described in a compact area s of the input space, by a combination of simpler functions Ôi(x)= y(!x-xi!) ,where y(.) is normally a Gaussian function. The appropriate to the function f(x) is given as f(x,w)= ? wi G(!x-xi!),where wi are real value entries of the coefficient vector w =[w1,w2,w3——,wn],f(x)being a real valued vector x=[x1,x2,x3—,xn]implements the input –output map of the RBFNN. Any arbitrary continuous function can be approximated with an RBFNN if localized Gaussian are placed to cover the space, the width of each Gaussian is controlled, the amplitude of each Gaussian is set.

C. Particle Swarm Optimization

Particle swarm optimization is a kind of evolution computation, which is an iterative optimization instrument similar to genetic algorithm PSO analogy prey behavior of birds. Such a scenario: a group of bird search food at random .In this area, there is only one food, however all birds don't know where the food is ,but know the distance to the food. Then what is optimal strategy to find food? Currently the simplest and most effective method is to search the food from this model to solve this kind problem. Each optimization is to search a bird in space which is called as 'particle'. All particles have fitness value determined by optimization function, every particle also have one velocity to determine direction and distance. Then particles follow the optimization particle to search PSO as are one random particle (random solution) iterative method is particles update themselves by tracking two "extreme" particles. The first is the optimization solution found by particles. This kind of solution is called as Pbest, the other is the optimization found by species. This kind of extremum is called as global extremum. When the two optimization are found, particle updates themselves by equation1 & 2 to find their own velocity & location.

v=v+c1*rand()*(pbest-present)+c2*rand()*(gbest-present)

Present= present + v

Where v is the particles velocity, present is the particles position currently rand() is random number among (0,1).c1,c2 as learning factor. Commonly c1=c2=2.The velocity in any dimension is limited in maximum velocity exceeds vmax, and then the velocity is vmax.

IV. EXPERIMENT RESULT & DISCUSSION

A three layer MLPNN is chosen as a classifier, variable parameter of MLPNN are as follows. Number of hidden layers, number of neurons in each hidden layer, transfer function of neurons in each hidden layers, transfer function of neurons in o/p layer, maximum number of epoch in supervised learning ,error threshold on training dataset. The MLPNN architecture (34-15-12-1) is design model with 0.01 mean square error threshold & 1000 epochs. The best performance obtained when transfer function of neurons in hidden layers as well as output layer shows be hyperbolic tangent (tanh) & the network showed be trained using conjugate gradient algorithm.

the RBFNN is designed using variable parameters, number of cluster centers (Gaussian basis functions),competitive learning rule,meric used in unsupervised competitive learning ,transfer function of neurons in o/p layer, learning rule used in the o/p layer ,maximum number of epoch used for supervised learning and error threshold for training dataset. The chosen optimal configuration of the RBFNN is trained three times with 100 epochs in unsupervised learning & 5000 epochs in supervised learning. Select 100 of no. of cluster centers, conjugate gradient supervised learning rule,tanh output layer transfer function and Euclidean competitive learning matrix.

Simulation result of MLPNN model gives Average classification accuracy is 94.44% and RBFNN classifier gives 99.59% accuracy.

V.CONCLUSION

We implemented a MLPNN Neural network and RBF Neural network classifiers for classification of radar data. For the classification of radar returns from the ionosphere, the decision boundaries form by the RBF NN classifier are seen to be more accurate than those formed by MLPNN classifier.

The main novelty of this paper is in the proposed PSO-based approach which aims at optimizing the performances of NN classifier in terms of classification accuracy. Also we explored various other implementations such as Support Vector Machine with PSO & empirically select this particular network since it achieved the best performance.

REFERENCES

[1] G.P.Zhange"*Neural networks for classification: a survey,"systems, Man,and cybernetics*, part C:Application and Reviews,IEEE Transaction on,vol.30.pp.451-462.2000.

[2] P.Kraipeerapun,"*Neural network classification based on quantification of uncertainty*",Murdoch University,2008.

[3] S.B.Kotsiantis,"*Supersed Machine learning: A review of classification techniques"*, informatics, vol.31.pp24-268,2007.

[4] Y.Sun, F.Karray,and Al-sharhan."*Hybrid soft computing techniques for heterogeneous data classification.*" In fuzzy systems, 2002, fuzz-IEEE02.

[5] h.Ahumada,G.L.Grinblat,L.C.Uzal,P.M.Granitto,and A.ceccatto,"REPMAC:Anew hybrid intelligent systems 2008.

[6] Y.Huang and C,Tian "*Research on credit risk assessment model of commercial banks based on fuzzy probabilitistic neural network* "in risk management & engineering management,2008

[7] Sigillito Vince(1989)Applied physics laboratory, john Hopkins university Ionosphere database.

[8] Xiao-Feng Gu & Lin Liu,Yuan-Yuan Huang; "*Data classification based on Artificial Neural Networks*" IEEE 2008.

[9] P.Jeatrakul and K.W.Wong; "*Comparing the performance of different Neural Networks for Binary classification problems*" Natural language processing 2009 SNLP-09,eight international symposium,111-115,IEEE 2009.

[10] Sigillito.V.G,Wing,S.P.Hutton,L.V..Baker,K.B(1989)" *Classification of radar returns from the ionosphere using neural networks*" John Hopkins APL technical digest ,vol 10 pp 262-266.

[11] Chanjun Zhu,Bin Wang;"PSO-*based RBF Neural Network Model for Teaching Quality Evalution*"2009 IEEE,41-50,CASE-2009 IITA .

To Improve the Rough Sets Based Classification Process Using Cut Sets and Rule Shortening Techniques

Naresh Kumar Nagwani
Assistant Professor, Department of CS&E, NIT Raipur
e-mail : nknagwani@gmail.com

Dr. Shrish Verma
Associate Professor, Department of IT, NIT Raipur
e-mail :shrishverma@nitrr.ac.in

***Abstract* - Classification is an important technique of data mining. Soft computing techniques are also used effectively for the classification. In this paper an improved rough set based model is proposed. Rough sets are popularly used for data pre-processing in data mining. In this paper cut sets and rule shortening techniques are used with rough sets to improve the classification process. The model is designed and experiment is done with the help of RSES (Rough Set Exploration System) software. The experiment is performed over the NASA software project repository and classification time is compared with the typical rough sets based classification algorithm, classification time is improved with the proposed model.**

I. INTRODUCTION

A. Rough Sets

Based on classical set theory, rough sets were introduced in 1982 by Pawlak [9]. Using the concept of equivalence relations, partitions of a set of objects can be formed, subsets of significant attributes identified, and decision rules extracted, all based on the attribute values of the objects. Rough set theory can be used to analyze the dependency relationship between the independent and the dependent variables. This relationship is used to determine whether the dependent attribute can be characterized by the values of the independent attributes. A main advantage of rough sets is that redundant attributes can be eliminated, and a subset of attributes with the same discrimination power as the original complete set of attributes emerges, called reducts. Once found, reducts can be used to generate decision rules to classify unseen objects. See [14] for a more detailed discussion on rough set theory

A reduct is a set of attributes that preserves partition. It means that a reduct is the minimal subset of attributes that enables the same classification of elements of the universe as the whole set of attributes. In other words, attributes that do not belong to a reduct are superfluous with regard to classification of elements of the universe.

Rough sets theory has been proposed by Professor Pawlak for knowledge discovery in databases and experimental data sets (Pawlak, 1982; 1991; Skowron, 1990). It is based on the concept of an *upper* and a *lower approximation* of a set, the *approximation space* and models of sets.

Certain attributes in an information system may be redundant and can be eliminated without losing essential classificatory information. One can consider feature (attribute) reduction as the process of finding a smaller (than the original one) set of attributes with the same or close classificatory power as the original set. Rough sets provide a method to determine for a given information system the most important attributes from a classificatory power point of view. The concept of the *reduct* is fundamental for rough sets theory. A reduct is the essential part of an information system (related to a subset of attributes) which can discern all objects discernible by the original set of attributes of an information system. Another important notion relates to a *core* as a common part of all reducts. The core and reduct are important concepts of rough sets theory that can be used for feature selection and data reduction.

Some attributes of an information system may be redundant (spurious) with respect to a specific classification A^* generated by attributes $A< Q$. By virtue of the dependency properties of attributes, one can find a *reduced* set of the attributes by removing *spurious* attributes, without loss of the classification power of the reduced information system.

B. The Rough Set Exploration System

The Rough Set Exploration System (RSES) is a set of software tools that are used for rough set computations in data mining [2]. RSES implements algorithms to manage and edit data structures that are used in user experiments and defined in the RSES library reduce data (objects and attributes) [17], quantify data [5], generate templates and decomposition trees [13], and classify objects [2]. The tool provides a graphical user interface which allows experiments to be constructed and executed with ease.

In our experiments we are interested in the discretization and the reduction algorithms. A discussion of background information for the discretization algorithm used by RSES is given in [5]. RSES also implements several reduction algorithms for reducing the number of irrelevant attributes [17]. These algorithms included an exhaustive search algorithm and several genetic algorithms. When the number of attributes is large (greater than 20), an exhaustive search for reducts is impractical. RSES uses genetic algorithms to find approximate and heuristic solutions to the attribute selection problem.

RSES 2.2 - Rough Set Exploration System 2.2 is a software tool that provides the means for analysis of tabular data sets with use of various methods, in particular those based on Rough Set Theory (see [6]). The RSES system was created by the research team supervised by Professor Andrzej Skowron.

The RSES system is freely available (for non commercial use) on the Internet. The software and information about it can be downloaded from: http://logic.mimuw.edu.pl/srses

The main aim of RSES is to provide a tool for performing experiments on tabular data sets. In general, the RSES system offers the following capabilities:

- import of data from text files,
- visualization and pre-processing of data including, among others, methods for discretization and missing value completion,
- construction and application of classifiers for both smaller and vast data sets, together with methods for classifier evaluation.

The RSES system is a software tool with an easy-to-use interface, at the same time featuring a bunch of method that make it possible to perform compound, non-trivial experiments in data exploration with use of Rough Set methods. The version 2.2 of RSES system was written in Java with some computational kernel written in GCC.

II. RELATED AND PREVIOUS WORK DONE

The combination of soft computing techniques like Rough sets; Genetic algorithms, neural networks etc. have been proven the best for number of complex problems. These techniques are also used for data mining and shown improvement over the complex data and data pre-processing. In this section some of the related and previous work done is discussed.

A tutorial is provided by Nguyen and Skowron [4]. The tutorial is a survey on rough set theory and some of its applications in Knowledge Discovery from Databases (KDD). The tutorial also covers the practice guide to analysis of different real life problems using rough set methods. The knowledge discovery from real-life databases is a multi-phase process consisting of numerous steps, including attribute selection, discretization of real valued attributes, and rule induction. A rule discovery process based on rough set theory is proposed by Zhong and Skowron [6]. The core of the process is a soft hybrid induction system called the Generalized Distribution Table and Rough Set System (GDT-RS) for discovering classification rules from databases with uncertain and incomplete data.

Data preprocessing is a step of the Knowledge discovery in databases (KDD) process that reduces the complexity of the data and offers better conditions to subsequent analysis. Rough sets theory is applied by Coaquira and Acuma [2] to three preprocessing steps: Discretization, Feature selection, and instance selection. A new algorithm of attribute reduction is proposed by Haung and Chen [11]. The algorithm uses the analogical matrix. The algorithm can reduce time complexity and spatial complexity of attribute reduction, and do not break the coherence of information contained in decision table. The basic concept and characteristics of the Rough Set theories are introduced by ZhuGe [5], and with an example the theories are used as form models for producing knowledge base and processing the decision result of each knowledge base, thereby enhancing the reliability and accuracy of the results in decision-making

The issues of Real World are Very large data sets, Mixed types of data, Uncertainty, Incompleteness (missing, incomplete data), Data change, Use of background knowledge etc. are discussed by Butalia et al [1], the use of rough sets are discussed to solve these problems. *The* case-based reasoning (CBR) becomes a novel paradigm that solves a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation. However, the acquisition of case knowledge is a bottleneck within case-based reasoning. The use of rough set and data mining to discover knowledge from traditional database and to construct case base is desired. An approach for case knowledge is discussed by Guo et al [3]. Rough set is used to preprocess the raw data that is noisy and redundant on the attribute. A Kohonen network is proposed to identify initial clusters within the data having been preprocessed. A knowledge discovery approach for software quality and resource allocation is proposed by Ramana et al. [7], which incorporates rough set theory, parameterized approximation spaces and rough neural computing. A software quality measure quantifies the extent to which some specific attribute is present in a system. Such measurements are considered in the context of rough sets. . It has been shown that rough sets work well in coping with the uncertainty in making decisions based on software engineering data.

The large number of Web-page documents comprises high dimensional huge text database with the development of Internet technology. The Web-page should be assigned to a category structure through the Web-page classification technology. Mining in high dimensional data is extraordinarily difficult because of the curse of dimensionality. To resolve these problems an algorithm is given by Yin et al. [8] to reduce the Web-page feature term and extract classification rule at last used attribute reduction on rough set theory.

III. PROPOSED MODEL

The major steps involved in the improvement of the classification task are shown in figure-1. The model is divided in the four stages. In the first stage cut sets are generated from the database. Then rules are generated from the cut sets in the second stage. In the next stage rule shortening method is

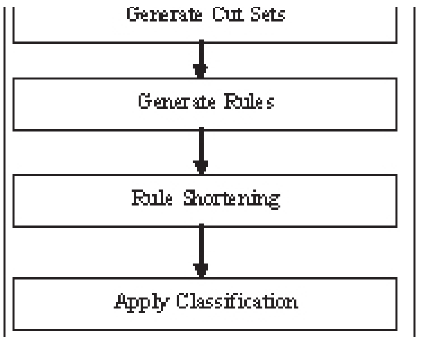


Fig.1. Steps in Classification Improvement Model

applied to reduce the generated rules. In the final stage classification method is applied to generate the classes.

The proposed model is designed using RSES system, the model is depicted in figure-2. The rule shortening RSES graphical user interface (GUI) is shown in figure-3.

IV. EXPERIMENTAL ANALYSIS

Nasa93 Dataset is used for the experiment of the classification. Nasa93 comes from a NASA-wide database recorded in the COCOMO 81 format. This data has been in the public domain for several years but few have been aware of it. It can now be found online in several places including the PROMISE (Predictor Models in Software Engineering) Web site. Nasa93 was originally collected to create a NASA tuned version of COCOMO, funded by the Space Station Freedom Program. Nasa93 contains data from six NASA centers, including the Jet Propulsion Laboratory. Hence, it covers a very wide range of software domains, development processes, languages, and complexity, as well as fundamental differences in culture and business practices between each center. All of these factors contribute to the large variances observed in this data set.

Fig.2. Overall Design of the System in RSES

The above dataset is taken over RSES system and process is set as per the fig. 2 and results are explored. The classification timing is compared against the typical classification algorithm. The proposed model works faster than typical classification algorithm. As dataset size increases the classification time also improves with scale of dataset size.

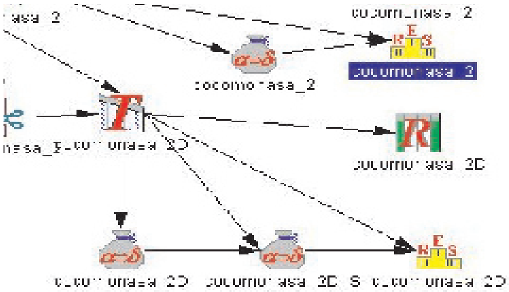
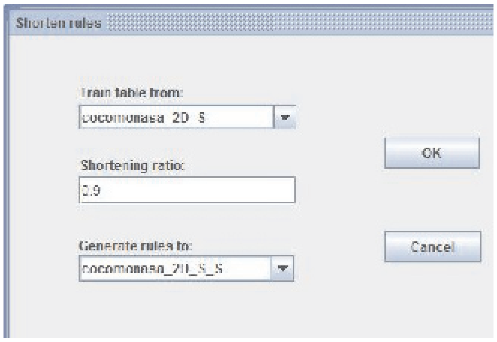


Fig.3. Rule Shortening GUI available in RSES.

V. CONCLUSION


Shorten rules

Train table from: cocomonasa 2D S

Shortening ratio: 0.9

Generate rules to: cocomonasa_2D_S_S

Rough set is a good option to data preprocessing tasks in the KDD process. Feature Selection using Rough sets theory is a way to identify relevant features. The cut sets and rule shortening techniques are applied in this paper to improve the classification timing for the classification task. The approach is useful for the time critical classification cases where the data set size is comparatively high.

REFERENCES

[1] Ayesha Butalia,, M.L Dhore, Ms. Geetika Tewani, "Application of Rough Sets in the field of Data mining", IEEE First International Conference on Emerging Trends in Engineering and Technology, 2008.

[2] Frida Coaquira and Edgar Acuña, "Applications of Rough Sets Theory in Data Preprocessing for Knowledge Discovery", Proceedings of the World Congress on Engineering and Computer Science 2007, WCECS 2007, October 24-26, 2007.

[3] Haifeng Guo, Xiaoming Zhou. Yulong Zhu, "Knowledge Acquisition based on Rough Set and Data Mining", IEEE International Conference on Future BioMedical Information Engineering, 2009.

[4] Hung Son Nguyen, Andrzej Skowron, "Rough Set Approach to KDD".

[5] Jianping ZhuGe , "Application of Rough Set Theory in Knowledge Discovery from Multiple Knowledge Base", IEEE 2009 International Symposium on Intelligent Ubiquitous Computing and Education, 2009.

[6] Ning Zhong, Andrzej Skowron, "A ROUGH SET-BASED KNOWLEDGE DISCOVERY PROCESS", Int. J. Appl. Math. Comput. Sci., Vol.11, No.3, pp. 603-619, 2001.

[7] Sheela Ramanna, James F. Peters, Taechon Ahn, "Software Quality Knowledge Discovery: A Rough Set Approach", Proceedings of the 26th IEEE Annual International Computer Software and Applications Conference (COMPSAC'02) 2002 .

[8] Shiqun Yin, Fang Wang, Zhong Xie, Yuhui Qiu, "Study on Web-page Classification Algorithm Based on Rough Set Theory", IEEE International Symposiums on Information Processing 2008.

[9] The Promise data Repository: http://promisedata.org

[10] The Rough Set Exploration System: http://logic.mimuw.edu.pl/~rses/

[11] Yan Huang, Shulin Chen , "An Algorithm of Attribute Reduction Based on Rough Sets", IEEE International Conference on Computer Science and Software Engineering, 2008.

Data Mining and Soft Computing Techniques for Prediction of Time Series Data: An Overview

Ms. Jyoti bala gupta
Department of IT
DR. C.V.RAMAN University, Bilaspur (C.G.)
E-mail: jyoti_jbg@yahoo.co.in

***Abstract*— This paper represents the prediction of time series data, and gives an overview of the different representation techniques of time series data. The purpose of this paper is to provide a consolidation work to mining time series, and to serve as a starting point for providing future research on time series data. To analyze a large number of highly periodic time series data we have to take the help of time series modeling procedure.**

There are various data mining and soft computing techniques given or suggested by different authors which can be applied on time series data for prediction purpose. The time series data mining framework adapts and innovates data mining and soft computing concepts to analyzing time series data. Time series analysis is often associated with the discovery and use of patterns and prediction of future values. Time series analysis creates a set of methods that reveal hidden temporal patterns that are characteristic and predictive of time series events.

Keywords— *Time series data mining, clustering, classification, rule generation.*

I. INTRODUCTION

The approach presented in this paper is a general one and can be applied to any time series data sequence for mining for similarities in the data. An improvement of technological process control level can be achieved by time series analysis in order to prediction of their future behavior. The paper deals with the utilization of soft computing and data mining to fix best the prediction of time series. We can find an application of this prediction by the control in production of energy, heat, etc.The modern economy has become more and more information-based. The widespread uses of information technology, a large number of data are collected in on-line, real-time environments, which results in massive amounts of data. Such ***time-ordered*** data typically can be aggregated with an appropriate time interval, yielding a large volume of equally spaced ***time series*** data. Such data can be explored and analyzed using many useful tools and methodologies developed in modern time series analysis. In many domains the experts want to know why a decision was made, otherwise they are unlikely to trust the advice generated by automated data analysis methods. Data mining is the exploration and analysis of data in order to discover meaningful patterns. The term data mining refers to information elicitation. On the other hand, soft computing deals with information processing. Both are an interdisciplinary field that combines artificial intelligence, computer science, machine learning, data-base management, data visualization, mathematics algorithms and statistics. This technology provides different methodologies for decision-making, problem solving, analysis, planning, diagnosis, detection, integration, prevention, learning and innovation. The two primary goals of data mining tend to be prediction and description. Prediction involves using some variables or fields in the database to predict unknown or future values of other variables of interest. The second goal which leads to descriptive model, describes patterns in existing data which may be used to guide decisions as opposed to making explicit predictions.

II. DATA MINING AND SOFT COMPUTING

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified.

Data mining is the process of extracting patterns from data and the process of finding hidden information or structure in a data collection. This includes extraction, selection, pre-processing, and transformation of features describing different aspects of the data it is seen as an increasingly important tool by modern business to transform data into business intelligence giving an informational advantage. It is currently used in a wide range of profiling practices, such as marketing, surveillance, fraud detection, and scientific discovery.

The principal components of data mining commonly involve four classes of tasks:

1. **Clustering -** is the task of discovering groups and structures in the data that are in some way or another “similar”, without using known structures in the data.
2. **Classification** - is the task of generalizing known structure to apply to new data. For example, an email program might attempt to classify an email as legitimate or spam. Common algorithms include decision tree learning, nearest neighbor, naive Bayesian classification, neural networks and support vector machines.
3. **Regression** - Attempts to find a function which models the data with the least error.

4. **Association rule learning** - Searches for relationships between variables.

Soft computing is a set of methodologies (like fuzzy logic) that its aim is to exploit the tolerance for imprecision, uncertainty, approximate reasoning and partial truth in order to achieve robustness, low solution cost and close resemblance with human like decision-making. Soft Computing Model always composed of Fuzzy logic, Neural Network, Genetic algorithm etc. Most of the time, these three components are combine in different ways to form model. Soft computing is essentially used for information processing by employing methods, which are capable to deal with imprecision and uncertainty especially needed in ill-defined problem areas.

The principal components of soft computing are as follows:-

1) Neuro Computing + Fuzzy Logic (Neurofuzzy: NP)
(2) Fuzzy Logic + Genetic Algorithm (Fuzzy genetic: FG)
(3) Fuzzy logic + Chaos theory (fuzzy chaos: FCh)
(4) Neural Networks + Genetic Algorithm (Neurogenetic: NG)
(5) Neural Networks + Chaos theory (Neurochaos: NCh)
(6) Fuzzy Logic + Neural Networks + Genetic Algorithm (Fuzzyneurogenetic: FNG)
(7) Neural Networks + Fuzzy Logic + Genetic Algorithm (Neurofuzzygenetic: NFG)
(8) Fuzzy Logic + Probabilistic reasoning (Fuzzy probabilistic: FP)

The term data mining refers to information elicitation. On the other hand, soft computing deals with information processing. Referring to this synergetic combination, the basic merits of data mining and soft computing paradigms are pointed out and novel data mining implementation coupled to a soft computing approach for knowledge discovery is presented.

III. TIME SERIES DATA MINING

Time series data accounts for a large fraction of the data stored in financial, medical and scientific databases. Recently there has been an explosion of interest in data mining time series, with researchers attempting to index, cluster, classify and mine association rules from increasing massive sources of data. Time series data often arise when monitoring industrial processes or tracking corporate business metrics.

Time series analysis accounts for the fact that data points taken over time may have an internal structure (such as autocorrelation, trend or seasonal variation) that should be accounted for.

This section will give a brief overview of some of the more widely used techniques.

A **time series** is a sequence of data points, measured typically at successive times spaced at uniform time intervals **Time series *analysis*** comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. **Time series *forecasting*** is the use of a model to forecast future events based on known past events: to predict data points before they are measured. An example of time series forecasting in econometrics is predicting the opening price of a stock based on its past performance. Time series are very frequently plotted via line charts.

Time series data have a natural temporal ordering. This makes time series analysis distinct from other common data analysis problems, in which there is no natural ordering of the observations._

The usage of time series models is twofold:

- Obtain an understanding of the underlying forces and structure that produced the observed data.
- Fit a model and proceed to forecasting, monitoring or even feedback and feed forward control.

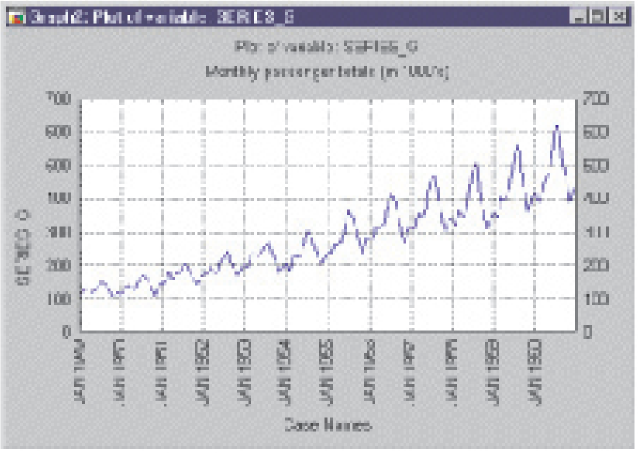
Time Series Analysis is used for many applications such as:

- Economic Forecasting
- Sales Forecasting
- Budgetary Analysis
- Stock Market Analysis
- Yield Projections
- Process and Quality Control
- Inventory Studies
- Workload Projections
- Utility Studies

There are two main goals of **time series analysis**:

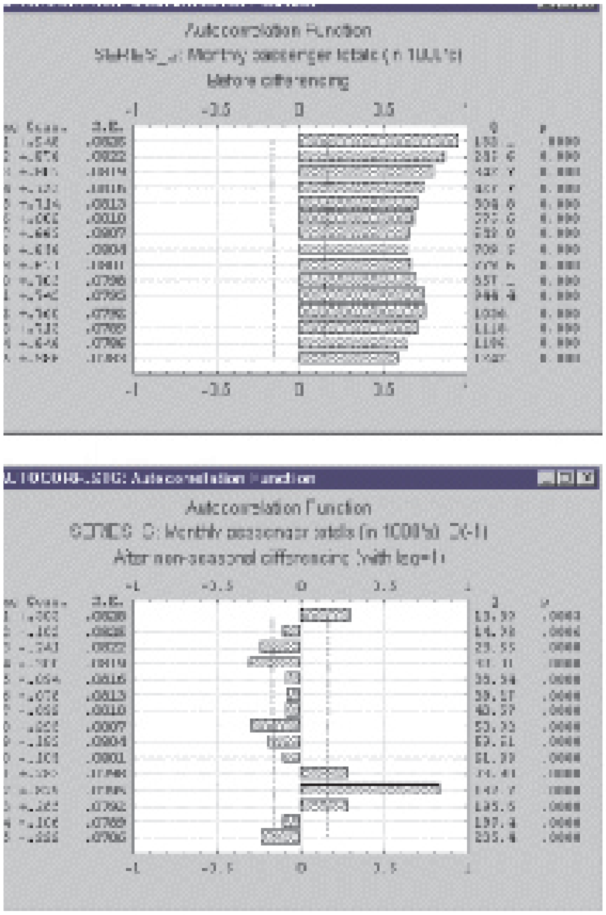
(a) Identifying the nature of the phenomenon represented by the sequence of observations, and
(b) Forecasting (predicting future values of the time series variable).

Both of these goals require that the pattern of observed time series data is identified and more or less formally described. Once the pattern is established, we can interpret and integrate it with other data (i.e., use it in our theory of the investigated phenomenon, e.g., seasonal commodity prices). Regardless of the depth of our understanding and the validity of our interpretation (theory) of the phenomenon, we can extrapolate the identified pattern to predict future events. Most time series patterns can be described in terms of two basic classes of components: trend and seasonality. Those two general classes of time series components may coexist in real-life data. For example, sales of a company can rapidly grow over years but



they still follow consistent seasonal patterns.

This general pattern is well illustrated in a “classic” *Series G* data set representing monthly international airline passenger totals (measured in thousands) in twelve consecutive years from 1949 to 1960. If you plot the successive observations (months) of airline passenger totals, a clear, almost linear trend emerges, indicating that the airline industry enjoyed a steady growth over the years (approximately 4 times more passengers traveled in 1960 than in 1949). At the same time, the monthly figures will follow an almost identical pattern each year (e.g., more people travel during holidays than during any other time of the year). This example data file also illustrates a very common general type of pattern in time series data, where the amplitude of the seasonal changes increases with the overall trend. This pattern which is called *multiplicative seasonality* indicates that the relative amplitude of seasonal changes is constant over time, thus it is related to the trend.



Time series data pattern with the help of graph.

IV. RELATED WORK FOR TIME SERIES DATA

To understand the piece of different techniques of data mining & soft Computing for predict time series data the contribution of different author in this field are described here.

Satyendra Nath Mandal explained the Soft Computing Model always composed of Fuzzy logic, Neural Network, Genetic algorithm etc. All this combination is widely used in prediction of time series data. If in a time series data, initial change are observed during some time interval, the final value of this data must be predicted. In his paper, an effort has been made to use soft computing approaches for predicting a final product of a time series data. His paper presents experimental results of a parallel implementation of a soft-computing algorithm for model discovery in multivariate time series, possibly with missing values. Most of the time, these three components are combine in different ways to form model. The objective of his survey is to find out the production of a particular type plant using certain initial parameters.

The data used in his paper from a statistical survey on the mustard plant. The same types of data are taken from different mustard plant. The entire component which are observed during data collection such as shoot length, number of leaf, root length etc are changes on time. But among all these parameters, the growth of shoot length can be observed during its life time until yield will produce. The shoot length is growing continuously and finally produce yield. So, soft computing approach to observe the growth of shoot length and can be predicted the yield. In his paper, a effort can be made to predict the yield applying fuzzy logic, neural network and genetic algorithm on the same data and finally based on the error analysis one method has been selected to predict the yields.

His paper presented a systematic approach to design neural networks for optimizing applications. They illustrated the methodology of neuro genetic system is used only cross over of all forecasted value from artificial neural network with fuzzy input.

This approach will be applied in different types of time series data like stoke market, financial data, and market demand and supply prediction.

Lon-Mu Liu, Siddhartha Bhattacharyy proposed the widespread use of modern information technology, given a large number of time series may be collected during normal business operations. They use a fast-food restaurant franchise as a case to illustrate how data mining can be applied to such time series, and help the franchise reap the benefits of such an effort. Time series data mining at both the store level and corporate level are discussed on their paper. In their paper, they employ a real-life business case to show the need for and the benefits of data mining on time series, and discuss some automatic procedures that may be used in such an application. To have a better focus, they shall employ one particular example to illustrate the application of data mining on time series. The concepts and methodologies can be readily applied to other similar business operations.

In their paper the adaptation of data mining on time series promises to assist the restaurant industry in several ways.

Data mining (1) provides a method to process large amounts of data in an automated fashion, (2) provides a method of distilling vast amounts of data into information that is useful for inventory planning, labor scheduling, and food preparation planning, and (3) offers a consistent, reliable and accurate method of forecasting inventory and product depletion rates over set temporal periods (e.g., hourly, daily, weekly, monthly, etc.) commonly used in business planning.

The time series data mining procedures discussed in their paper have been implemented in a fast-food restaurant franchise. It is easy to see their similar approach can be applied to other business operations and reap the benefits of time series data mining. More generally, an interesting review article on the current and potential role of statistics and statistical thinking to improve corporate and organizational performance can be found in Dransfield, Fisher and Vogel.

Hehua Chi a, Juebo Wu b, *, Shuliang Wang a, b, Lianhua Chi c, Meng Fang a, presented a novel approach of mining time-series data is proposed based on cloud model, which described by numerical characteristics.

In their paper firstly, the cloud model theory is introduced into the time series data mining. Time-series data can be described by the three numerical characteristics as their features: expectation, entropy and hyper-entropy. Secondly, the features of time-series data can be generated through the backward cloud generator and regarded as time-series numerical characteristics based on cloud model. In accordance with such numerical characteristics as sample sets, the prediction rules are obtained by curve fitting. Thirdly, the model of mining time-series data is presented, mainly including the numerical characteristics and prediction rule mining.

Their paper presented a method of time series data rules mining, which played an important significance for getting time series space association rules and doing the practical application by time series space association rules. Finally, the time series data were predicted by the forecasting rules. Their experimental results showed that the method is feasible and effective.

V. CONCLUSIONS

We employ the need for and the benefits of data mining & soft computing on time series. We also present the related work for data mining & soft computing in time series and an approach on time series data mining. We conducted an extensive consolidation on representation methods for time series data. Time series description can sometimes be done directly based on the representation, using feature based or model based representations. There is still the need for more comparative studies that evaluate different methods on several data sets with different characteristics. We have gone beyond simply combining conventional methods with time series feature extraction and start to integrate the temporal characteristics of the data more tightly into the mining algorithms.

VI. REFERENCES

[1] Jiawei Han and Micheline Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, CA, 2005.

[2] Lon-Mu Liu, Siddhartha Bhattacharyya, Stanley L. Sclove, Rong Chen(*) etal has paper DATA MINING ON TIME SERIES: AN ILLUSTRATION USING FAST-FOOD RESTAURANT FRANCHISE DATA(1-28).

[3] G.E.P., Jenkins, G.M. and Reinsel, G.C. (1994). Time Series Analysis: Forecasting and Control. Third Edition. Prentice Hall.

[4] Chaudhuri, S. and Dayal, U. (1997). “An Overview of Data Warehousing and OLAP Technology” ACM SIGMOD Record 26(1), March 1997.

[5] Fayyad, U. M. (1997). “Editorial.” Data Mining and Knowledge Discovery 1: 5-10.

[6] Friedman, J. H. (1997). “Data Mining and Statistics: What’s the Connection?” Proceedings of Computer Science and Statistics: the 29th Symposium on the Interface.

[7] E. J. Keogh. A Decade of Progress in Indexing and Mining Large Time Series Databases. In VLDB, 2006.

[8] E. J. Keogh and S. Kasetty. On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. Data Min. Knowl. Discov., 7(4), 2003.

[9] Weiss, S. M. and Indurkhya, N. (1998). Predictive Data Mining. San Francisco: Morgan Kaufmann Publishers. Widom, J. (1995). “Research Problems in Data Warehousing”. Proceedings of 4th International Conference on Information and Knowledge Management (CIKM), November 1995

[10] http://www.jatit.org/volumes/research-papers/Vol4No12/1Vol4No12.pdf

[11] http://www.springer.com/engineering/mathematical/book/978-3-540-72431-5

A Comparative Study of Decision Tree Based Data Mining Algorithms and its Ensemble Model for Classification of Data

Dr. H.S.Hota
Lecturer, Dept. C.S.I.T, G.G.V, Bilaspur, C.G
Email- hota_hari@rediffmail.com

Mrs. Pushpalata Pujari
Lecturer, Dept. C.S.I.T, G.G.V, Bilaspur, C.G
Email- pujari.lata@rediffmail.com

Abstract-**This paper presents a comparative study of decision tree data mining algorithms CART, QUEST, CHAID and ensemble model for prediction of radar returns on ionosphere data. The ionosphere dataset investigated in this study is taken from UCI machine learning repository. A comparative study is carried out among CART, CHAID, QUEST algorithms and ensemble model. The proposed ensemble model combines all the above models by using confidential-weighted voting scheme. The classification performance of all algorithms and ensemble model are presented by using statistical performance measures like accuracy, specificity and sensitivity. Experimental study has shown that the ensemble model may be competent techniques for prediction of radar returns than the individual models**

Index Terms-**Data mining, Ionosphere data, prediction, CART, QUEST, CHAID, decision tree, decision rule, Information gain, Gini index, quadratic discriminate analysis, Gain chart, ROC.**

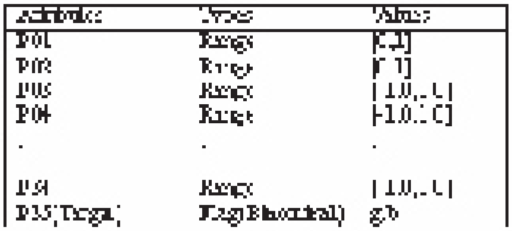
[I] INTRODUCTION

Classification [1] model is an analysis techniques used to describe data classes. In classification a model or classifier is constructed to predict categorical labels. Data classification is a two step process. In the first step a model or classifier is built describing a predetermined set of data classes or concepts. In the second step the model is used for classification. A test set is used in the second to test tuples and their associated class labels. These tuples are randomly selected from the general data set. The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier. In this paper different classification technique of data mining such as CART, QUEST, CHAID and ensemble model is analyzed on ionosphere dataset. The idea of the ensemble model is to employ multiple models to do better than a single individual model. The proposed system uses an ensemble of three decision tree data mining algorithms CART, CHAID and QUEST. A comparative study is carried out among the classification algorithms and its ensemble model for the prediction of radar condition on ionosphere dataset.

[II] DATA SET DESCRIPTION

The ionosphere dataset used in this study is taken from UCI machine learning dataset [7]. The targets were free electrons in the ionosphere. “Good” radar returns are those showing evidence of some type of structure in the ionosphere. “Bad” returns are those that do not; their signals pass through the ionosphere. In this dataset there are 34 numbers of instances. Each instance is having 17 pulse numbers for the system, and is described by 2 attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal. All 34 predictor’s attributes are continuous. The 35th attribute which is target attribute is either “good” or “bad” according to the definition summarized above. Table I shows the structure ofionosphere dataset which contains 35 attributes out of which 34 are input attributes and one output (target) attribute. Models [2] are developed in two phases: training and testing. The training dataset is used to train or build a model. Once a model is built on training data, the accuracy of the model on unseen data (testing) can be found. Two mutually exclusive datasets, a training dataset comprising 60% of the total ionosphere dataset, and a testing dataset of 40% is created by using partitioning node and balanced node portioning techniques. Classification techniques are applied on this data set. In all there are 126 numbers of instances for “bad” class and 225 numbers of instances for “good” class.

Table I: Attributes of ionosphere dataset



Attributes	Types	Values
P01	Range	[0,1]
P02	Range	[0,1]
P03	Range	[-1.0,1.0]
P04	Range	[-1.0,1.0]
.	.	.
P34	Range	[-1.0,1.0]
P35(Target)	Flag(Nominal)	g,b

[III] METHODOLOGY

Mainly decision tree based classification algorithm is considered to meet the objective of this piece of research work. Decision tree models allow developing classification systems that predict or classify future observations based on a set of decision rules. A decision tree [3] based classifier splits a dataset on the basis of discrete decision using certain threshold on the attribute values.. Decision tree can be converted into a collection of if-then rules (a rule set), which shows the information in a more comprehensible form.

A. CART (Classification and Regression Tree) classifier

The Classification and Regression (CART) [6] tree method uses recursive partitioning to split the training records into segments with similar output field values. The CART Tree node starts by examining the input fields to find the best split, measured by the reduction in an impurity index that results from the split. CART uses Gini index [12] splitting records measures in selecting the splitting attribute. Pruning is done in CART by using a training data set. The split defines two subgroups, each of which is subsequently split into two more subgroups, and so on, until one of the stopping criteria is triggered. All splits are binary (only two subgroups).

B. CHAID (Chi-squared Automatic Interaction Detection) classifier

CHAID [6], or Chi-squared Automatic Interaction Detection, is a classification method for building decision trees by using chi-square [8] statistics to identify optimal splits. CHAID first examines the cross tabulations between each of the predictor variables and the outcome and tests for significance using a chi-square independence test. If more than one of these relations is statistically significant, CHAID selects the predictor that is the most significant (smallest p value). If a predictor has more than two categories, these are compared, and categories that show no differences in the outcome are collapsed together. This is done by successively joining the pair of categories showing the least significant difference. This category-merging process stops when all remaining categories differ at the specified testing level.

C. QUEST (Quick, Unbiased, Efficient Statistical Tree) decision tree classifier

QUEST [6] is a binary classification method for building decision trees uses a sequence of rules, based on significance tests, to evaluate the predictor variables at a node. For selection purposes, as little as a single test may need to be performed on each predictor at a node.spliting predicate in QUEST. Splits are determined by running quadratic discriminate analysis using the selected predictor on groups formed by the target categories. It separates splitting predicate selection into variable selection and split point selection. It uses statistical significance tests instead of impurity function.

D. Performance measurement

The performance of individual models and ensemble model are evaluated by using different statistical measures [4] such as classification accuracy which measures the proportion of correct predictions considering the positive (P) and negative (N) inputs, sensitivity which measures the proportion of the true positives (TP) and specificity which measures the proportion of the true negatives (TN), The three statistical measures are evaluated as follows

i) Classification accuracy= (TP + TN) / (P + N) (1)

ii) Sensitivity = TP/ (TP+FN)
(2)

iii) Specificity =TN/ (TN +FP)
(3)

[IV] EXPERIMENTAL WORK

Experimental work is carried out by applying the data set to each algorithm and ensemble model. The decision tree generated by each classifier explained in methodology part (CART, QUEST, and CHAID) is shown in Fig 1,2, and 3 respectively.

A. CART: Fig 1 shows the tree generated by CART classifier according to the algorithm discussed in previous section. The corresponding rule base just below it represents this tree.

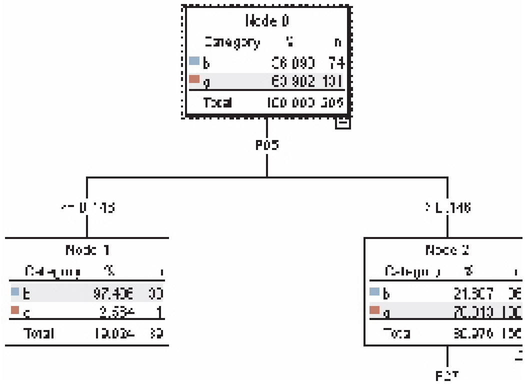
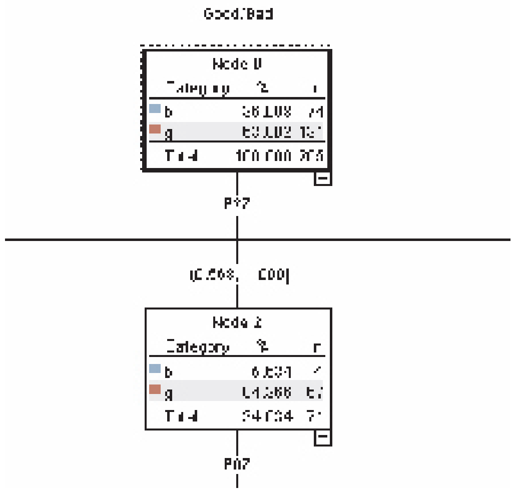


Fig.1. A part of decision tree generated by CART classifier.

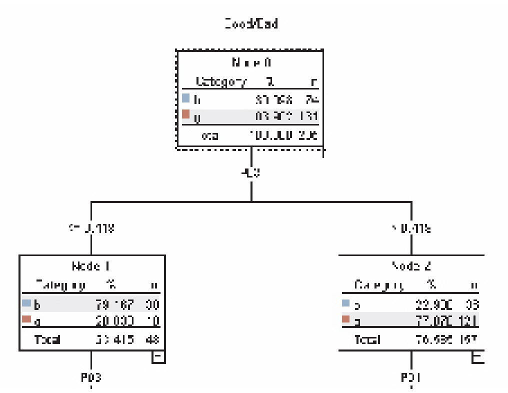
B. CHAID: Similarly training set is applied to CHAID classifier and tree generated by this classifier is depicted in fig-2.

Fig.2. A part of Decision tree generated by CHAID classifier



C. QUEST: In similar manner training set is applied to QUEST classifier. The tree generated by QUEST classifier is depicted in fig 3

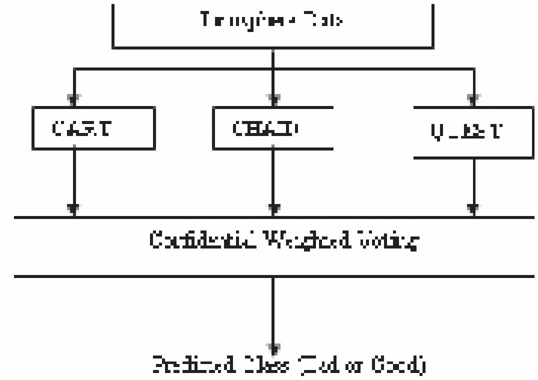
Fig.3. A part of decision tree generated by QUEST classifier



[V] ENSEMBLE OF CART CHAID AND QUEST MODEL

An ensemble model is proposed in this research paper using all the above classifier (CART, CHAID and QUEST). Ensemble combines the output of several classifier produced by weak learner into a single composite classification [13]. The block diagram of the ensemble model is shown in fig-4. The three models are combined by using confidential weighted voting scheme [5] where weights are weighted based on the confidence value of each prediction. Then the weights are summed and the value with highest total is again selected. The confidence for the final selection is the sum of the weights for the winning values divided by the number of models included in the ensemble model. If one model predicts no with a higher confidence than the two yes predictions combined, then no wins.

Fig.4. Ensemble model of three classifier (CART, CHAID, and QUEST)

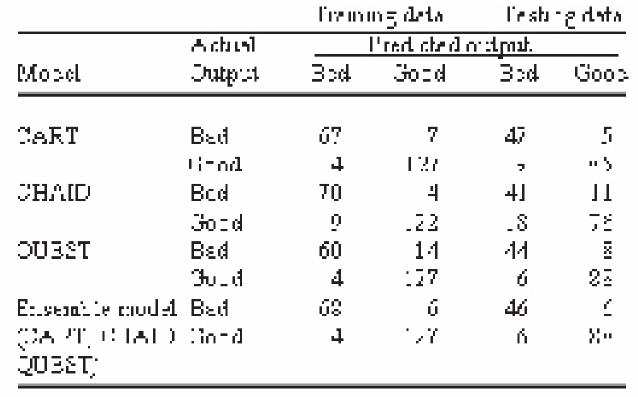


[VI] RESULTS AND DISCUSSION

After applying training data and training data set to each classifier along with ensemble model a confusion matrix is obtained to identify true positive, true negative, false positive, and false negative values as shown in table II.

Table II: Confusion matrices of different model for training and testing data.

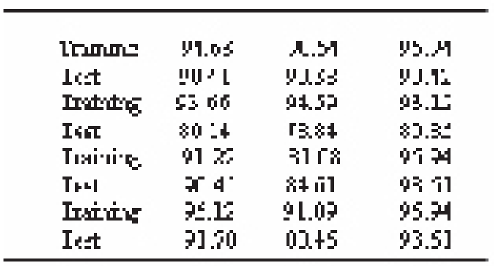
Each cell of the above table contains the row number of samples classified for the corresponding combination of desired and actual model output. The prediction are compared with original classes to identify true positive, true negative, false positive and false negative. Table III represents the value of three statistical measures classification accuracy, sensitivity and specificity of the three predictive models and their ensemble model. The table shows that CART has achieved 94.63% of accuracy on training dataset and 90.41% of accuracy on testing dataset. These results show that the accuracy of CART is better than the other two models. Ensemble model shows accuracy of 95.12 on training dataset and 91.78 on testing dataset. The ensemble method has achieved a better result for both training and testing sample than any one of its individual model.



		Training data		Testing data	
	Actual	Predicted output			
Model	Output	Bad	Good	Bad	Good
CART	Bad	67	7	47	5
	Good	4	127	[illegible]	[illegible]
CHAID	Bad	70	4	41	11
	Good	9	122	18	72
QUEST	Bad	60	14	44	8
	Good	4	127	6	83
Ensemble model (CART, CHAID, QUEST)	Bad	68	6	46	6
	Good	4	127	6	84

Table III: Comparative statistical measures for different models

Another way to compare the performance of different classifier is gain chart and ROC (Receiver Operating Characteristics) [14]. The gains chart [6] plots the values in the Gains % column from the table. Gains are defined as the proportion of hits in each increment relative to the total number of hits in the tree, using the equation:



Training	94.63	[illegible]	95.44
Test	90.41	91.58	[illegible]
Training	93.66	94.52	93.12
Test	80.14	73.84	83.32
Training	91.35	81.78	97.34
Test	90.41	84.61	93.11
Training	95.12	91.09	95.34
Test	91.78	03.45	93.51

(Hits in increment / total number of hits) x 100% (4)

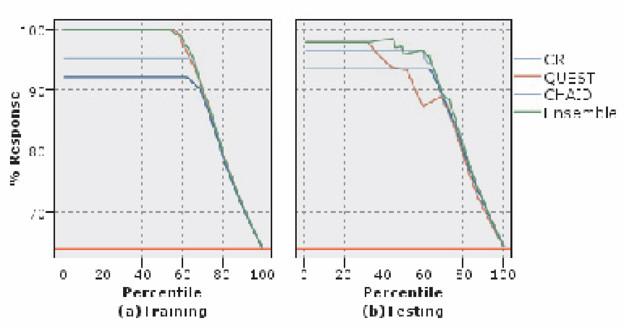
Cumulative gains charts always start at 0% and end at 100% as we go from left to right. For a good model, the gains chart will rise steeply toward 100% and then level off. A model that provides no information will follow the diagonal from lower left to upper right the steeper the curve, the higher the gain. Fig-4(a) & (b) shows the cumulative gain chart of three models and its ensemble model for training and testing dataset respectively. The higher curves are of the ensemble model and CART model among the individual models.

Fig-4 .Gain chart for three models and its ensemble model

R.O.C [6] chart plots the values in the Response (%) column of the table. The response is a percentage of records in the increment that are hits, using the equation:

(Responses in increment / records in increment) x 100% (5)

ROC chart is based on the conditional probabilities sensitivity and specificity [11] .It is a plot of sensitivity on the vertical axis and one minus the specificity on horizontal axis for different values of the thresholds. Response charts usually start near 100% and gradually descend until they reach the overall response rate (total hits / total records) on the right edge of the chart. For a good model, the line will start near or at 100% on the left, remain on a high plateau as you move to the right, and then trail off sharply toward the overall response rate on the right side of the chart. Fig-5 shows the ROC chart of three models and its ensemble model for training and testing dataset. The overall response rate of ensemble model is found to be higher than the individual models.



[VII] CONCLUSION

Three different decision tree algorithms CART, CHAID & QUEST have been applied on ionosphere data. In CART the attribute that maximizes the reduction in impurity (or, equivalently, has the minimum Gini index) is selected as the splitting attribute. In QUEST Splits are determined by running quadratic discriminate analysis using the selected predictor on groups formed by the target categories. CHAID builds decision trees by using chi-square statistics to identify optimal splits. By combining scores of these models more precise model called ensemble model is obtained. They are all combined by using confidential weighted voting scheme. The performance of this model and their ensemble model were investigated and it is found that accuracy of ensemble model is better (95.12 for training data and 91.78 for testing data) than all other individual model, say for CART accuracy is 94.63 and 90.41 respectively for training and testing data. Performance of algorithm has also investigated with the help of gain chart and ROC chart for both training and testing data. Chart clearly shows that accuracy of ensemble model is higher than that of any individual model for classification of ionosphere data.

REFERENCES

[1] Jiwaei Han, Kamber Micheline, Jian Pei Data mining: Concepts and Techniques, Morgam Kaufmann Publishers (Mar 2006).

[2] Cabena, Hadjinian, Atadler, Verhees, Zansi "Discovering data mining from concept to implementation" International Technical Support Organization, Copyright IBM corporation 1998.

[3] S.Mitra, T. Acharya "Data Mining Multimedia, Soft computing and Bioinformatics, A john Willy & Sons, INC, Publication, 2004.

[4] Alaa M. Elsayad "Predicting the severity of breast masses with ensemble of Bayesian classifiers" journal of computer science 6 (5): 576-584, 2010, ISSN 1549-3636

[5] Alaa M. Elsayad " Diagnosis of Erythemato-Squamous diseases using ensemble of data mining methods" ICGST-BIME Journal Volume 10, Issue 1, December 2010

[6] SPSS Clementine help file. http//www.spss.com

[7] UCI Machine Learning Repository of machine learning databases. University of California, school of Information and Computer Science, Irvine. C.A. http://www.ics.uci.edu/~mlram,?ML.Repositary.html w

[8] Michael J. A. Berry Gordon Linoff, "Data Mining Techniques ", John Wiley and Sons, Inc.

[10] Jozef Zurada and Subash Lonial "Comparison of The Performance of Several Data Mining methods for Bed Debt Recovery in The Health Care Industry".

[11] Matthew N Anyanwu &Sajjan G Shiva " Comparative Analysis of serial Decision Trees Classification Algorithms",(IJCSS), Volume (3) : Issue (3)

[12] Mahesh Pal "Ensemble Learning With Decision Tree for Remote Sensing Classification", World Academy of Science, Engineering and Technology 36 2007.

[13] Kelly H. Zou, PhD; A. James O'Malley, PhD; Laura Mauri, MD, MSc "ROC Analysis for Evaluating Diagnostic Test and Predictive Models"

Comparison between Calculated Crack Growth Rate and Target Crack Initiation Angle by using Artificial Neural Network

ARUNA THAKUR,
Department of Industrial & Production, I.T.G.G.U. Bilaspur
Email : ar_aruna_tk@yahoo.co.in

A.R.CHAUDHARY, V. MAHOBIYA, J.P.EKKA
Department Of Industrial & Production, I.T.G.G.U. Bilaspur

During recent years, the fracture mechanics has obtained a considerable importance for studying the crack growth behavior under static and fatigue loading. Several catastrophic failures, over the years, have resulted in a sharp awareness of the effect of the cracks and stress raiser in the manufactured parts on their failure strength.

Artificial neural network (ANN) is an intelligent tool with parallel computational capability. It can perform nonlinear mapping in short duration. Once neural network is trained, it provides acceptable recommendations in a short time. The concept of an artificial neural network is inspired from the working of the human brain. The brain acts as a highly complex nonlinear parallel computer. Similarly an artificial neural network is a massively parallel distributed processor with neurons as processing units. Here we can compare the experimental results with ANN. In this we are using "MATLAB" programming software for running ANN training and testing data.

INTRODUCTION

The chapter deals with the experimental and simulation details of the crack extension angle, crack propagation and fatigue life in single inclined crack, multiple edge and inclined cracks in commercially available aluminum alloy.The crack propagation is monitored using the traveling microscope. The crack initiation direction under different crack position have been simulated by artificial neural network and compared with the experimental results. Prediction of the fatigue crack growth rate in multiple crack problems is the subject of this chapter.

RESULT AND DISCUSSION

In this section we are going to discuss crack growth rate by experimentally and ANN.

Crack Growth Rate

The effective crack length is determined from the crack increment length Δa. The crack increment length Δa is measured by traveling microscope after each specified number of cycles. The effective crack length is calculated using following equation (Sih and Barthelemy. 1980.). a=a$_{new}$

(1.1)

$$a_{new} = a_{old} + \frac{\Delta a + a_{old}\sin\theta}{a_{old} + \Delta a\cos\theta}$$

where aold is the previous crack length

The effective crack length, number of cycles and other loading parameters are given in Table 1.1. The result of effective crack length and corresponding number of stress cycles N is shown in Figure 1.1 (a) for different H and S values. Figure 1.1 (a) shows that when the crack tips distance S is less; i.e. for lower value of S/H, crack acceleration occurs. Figure 1.1 (a) also shows the crack extension with number of stress cycle for inclined crack positions with different S/H ratio.

The crack growth rate $\Delta a / \Delta N$ obtained from the experimental observation of crack extension and stress cycle for two center cracks for different crack tip distance S and crack offset distance H are shown in figures 1.1 (b). The crack growth rate was calculated from the estimated crack length and corresponding load cycle as follows.

$$(\Delta a)_{i+1} = a_{(i+1)} - a_i$$

$$(\Delta N)_{i+1} = N_{(i+1)} - N_i$$

$$\left(\frac{\Delta a}{\Delta N}\right)_{i+1} = \left(\frac{da}{dN}\right)_{i+1} = \frac{a_{(i+1)} - a_i}{N_{(i+1)} - N_i} \tag{1.2}$$

The mode I stress intensity factor corresponding to (i+1)th cycle is obtained from the relation

$$(\Delta K_I)_{i+1} = \Delta\sigma\sqrt{\pi \times a_{(i+1)}} \tag{1.3}$$

Where $\Delta\sigma = \sigma\text{max} - \sigma\text{min}$

σmax and σmin are the maximum and minimum stress in the load cycle.

It is well known that the interaction between multiple cracks have major influence on the crack growth behaviour. Generally the crack growth behaviour can be studied either analytically or experimentally. In the experimental approach, the effect of interaction on the crack growth behaviour can be studied directly through the observations of the crack growth behaviour.

Prediction Of Crack Growth Rate By Ann

Figure 1.2 illustrated the ANN structure with seven input layer, one hidden layer and one out put layer. Each layer has different number of neurons. In the present case, seven, four and one neurons are taken for input, hidden and output layers, respectively. Figure 1.3 and figure 1.4 shows the variation of RMSE and r with selected neurons at 4000 iterations. From the figure 1.3 and figure 1.4, it can be seen that minimum error occurs corresponding to eight neurons at the hidden layer. Figure 1.5 shows that the variation between experimental and ANN crack growth rate. The seven input parameters are right crack length (a1), left crack length (a2), right crack angle (α1), left crack angle (α2), vertical offset distance (H), crack tip distance (S), stress intensity factor (K) and one crack growth rate (da/dN). The optimum ANN structure is found on the basis of RMSE and correlation coefficient is 7-[4]1-1. Figure 1.6 and 1.7 shows the experimental and predicted growth rate for training and testing of the data set. The predicted values are found to be very close to the experimental values. From the above results it can be concluded that artificial neural network with one hidden layer and 4-9 neurons in hidden layer can be effectively used to predict the crack initiation direction, growth rate in multiple crack problem.

Table 1.1 Crack increment, load cycles and crack growth rate details

Speci- men id	Δa (mm)	Cycle (N)	a (m)	ΔK M Pa √ m	$\frac{da}{dN}$ m/Cycle
C321	3.0	2.33×10^{4}		1.7	9.21×10^{-8}
C321	6.0	2.47×10^{4}		2.09	3.18×10^{-6}
C321	9.0	2.5×10^{4}		2.62	2.86×10^{-5}
C321	13.0	2.51×10^{4}		3.17	5.45×10^{-5}
C322	4.0	1.18×10^{4}		1.6	2.46×10^{-7}
C322	8.0	1.21×10^{4}		2.23	1.94×10^{-5}
C322	12.0	1.23×10^{4}		2.94	3.37×10^{-5}
C322	16.5	1.25×10^{4}		3.63	5.84×10^{-5}
C323	3.0	1.12×10^{4}		2.93	1.7×10^{-7}
C323	6.0	1.13×10^{4}		3.83	6.95×10^{-5}
C323	9.0	1.15×10^{4}		4.84	9.15×10^{-5}
C323	13.5	1.135×10^{4}		6.029	5.12×10^{-5}
C324	2.0	1.3×10^{4}		2.4	7.75×10^{-8}
C324	7.0	1.31×10^{4}		3.587	1.5×10^{-4}
C324	11.0	1.32×10^{4}		4.79	8.95×10^{-5}
C324	15.0	1.34×10^{4}		5.95	2.89×10^{-5}
C324	18.0	1.35×10^{4}		7.00	1.58×10^{-4}

Where C denotes central inclined cracks with different -2 crack tip distance(S) and crack offset distance (H). In this C321, C322, C323 and C324 have this value of S and H. C321 (S=18, H=14), C322 (S=24, H=10), C323 (S=24, H=8) and C324 (S=12, H=20)

Figure 1.1 (a) Crack length vs. number of cycle for central inclined crack

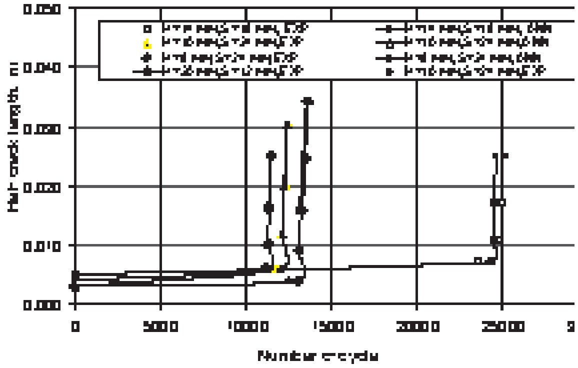


Figure 1.1 (b) Stress intensity factor vs. crack growth rate for central inclined crack

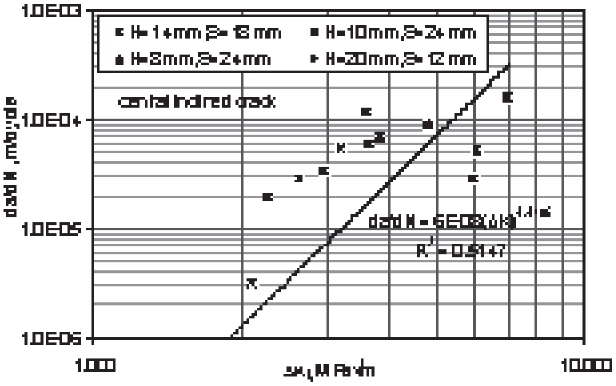


Figure 1.2 ANN architecture

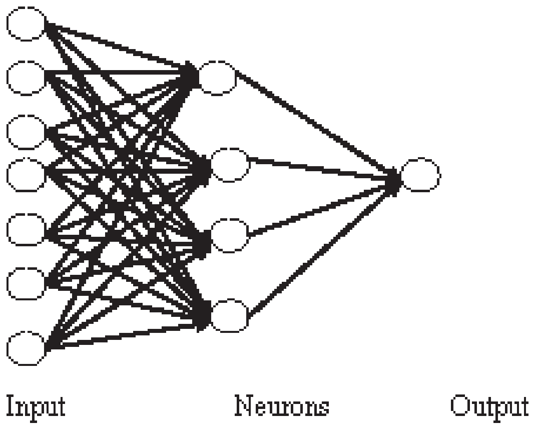


Figure 1.3 Dependence of training da/dN RMSE value and correlation coefficient on Numbers of Iterations at four numbers of neurons in a hidden layer and 0.8 learning rate

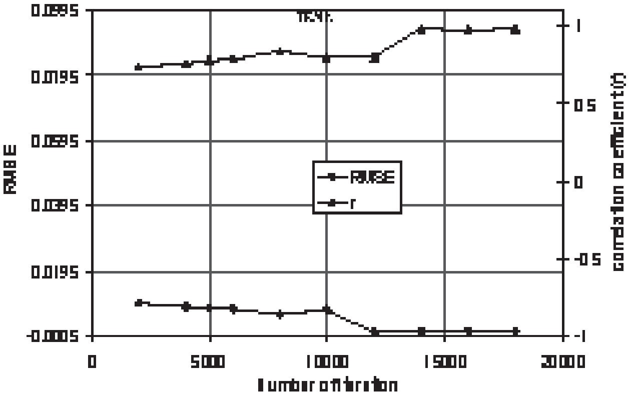


Figure 1.4 Variation of training RMSE value with learning rate at four numbers of neurons in a hidden layer and at 16000 number of iteration

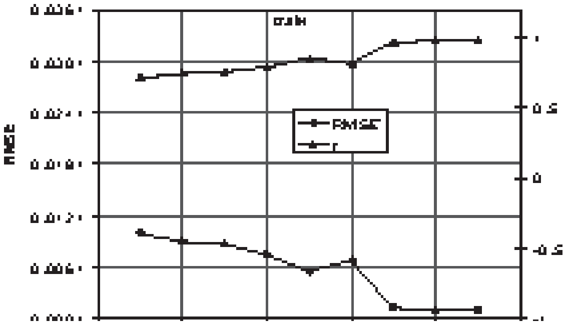


Figure 1.5 Comparison of experimental and ANN predicted crackgrowth for multiple inclined cracks

Figure 1.6 Comparison between Calculated da/dN and target crack initiation angle for training data

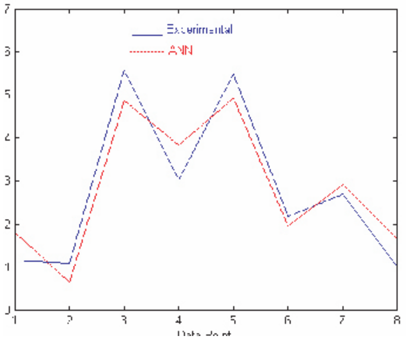
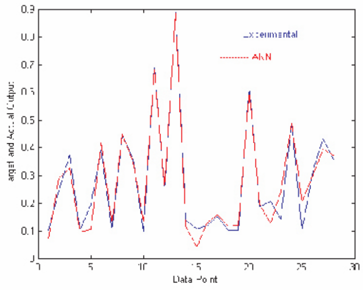
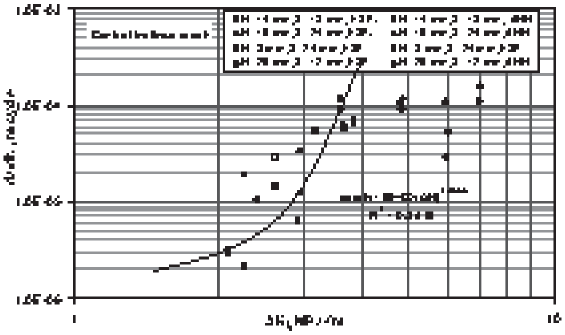


Figure 1.7Comparison between Calculated da/dN and target crack initiation angle for testing data





SUMMARY & CONCLUSIONS

From the present investigations, following conclusions are drawn. Artificial neural network can be applied to predict the crack growth rate for different crack position with reasonable accuracies.The crack growth rate can also be predicted with reasonable accuracies by ANN architecture [7] - [4]1- [1]. The optimum learning rate parameter at 16000 iteration is ? =0.8. The minimum root mean square error is found to be 0.0025 with 0.947 correlation coefficient.

References

Effis, J.; Subramonian, N. and Leibowitz, H. (1977). Crack border stress and displacement equations revisited, Engg. Fracture Mech., 9: 189-210.

Erdogan, F. and Sih, G.C. (1963). On crack extension in plates under plane loading and transverse shear, Trans. ASME, J. Basic Engg. 85: 519-527.

Finnie, I. and Weiss, H.D. (1974). Some observations on Sih's strain energy density approach for fracture prediction, Int. J. Fracture Mechanics. 10: 136.

Forman R. G.; Kearney V. E. and Engle R. M. (1967). Journal of Basic Engg, Trans. of ASME, 89: 459-464.

Griffith, A.A. (1921). The phenomena of rupture and flow in solids, Phil. Trans. Royal Soc. London, 221: 163-198.

J.P. Laures;M. Kachanov. (1991). Int. J. Fract. 48: 255.

K.Y. Lam; S.P. Phua. (1991). Engng. Fract. Mech. 40: 585.

Kamaya, M. A. (2005). Influence of the interaction on stress intensity factor of scraemi elliptical surface crack. ASME 71352.

Insight into Utility of Soft Computing in Datamining

[1]Vaishali.P.Khobragade, [2]Dr.A.Vinayababu, [3]Srinivas Kathuroju

[1]Department of Computer Science & Information Technology, Jyothishmathi Institute of Tech & Sciences, JNTU, Hyderabad, AP, INDIA.

[2]Professor of CSE, Director of Admissions JNTUH, Kukatpally, Hyderabad, AP, INDIA.

[3]Department of Computer Science & Information Technology, Jyothishmathi Institute of Tech & Sciences, JNTU, Hyderabad, AP, INDIA.

e-mail: [1]vaishali5599@gmail.com, [2]dravinayababu@yahoo.com, [3]srinivaskathuroju@gmail.com

Abstract— **Datamining now a days has become extremely important because it enables knowledge extraction from abundant data availability. Soft computing is a set of methodologies which work synergistically and provide in one form or another flexible information processing capabilities extending the sachet of problems that data mining can solve efficiently. Recently soft computing techniques i.e fuzzy logic, genetic algorithm, neuro computing etc. are gaining growing popularity for their remarkable ability of handling real life data in an environment of vagueness, and contained knowledge. The present paper provides a brief survey of the available literature on use of soft computing in data mining. An outline of different soft computing methods and the utility of the different soft computing methodologies in datamining is highlighted. Further we list out some key applications of soft computing with datamining.**

Keywords : *Softcomputing,Datamining,Fuzzylogic, Neurocomputing,Genetic algorithm , Rough sets*

I. INTRODUCTION

With the invention of sophisticated tool and the innovations in science and technology has resulted in explosive growth in stored and trainsient data, hence to transfer the vast amount of data into useful information and knowledge. Data mining refers to the extraction of uesful information from a large set of data. It is a technique for the discovery of patterns hidden in large data sets, focusing on issues relating to their feasibility, use fulness, effectiveness, ands calability [2]. The term data mining refers to information elicitation. On the other hand, soft computing deals with information processing. If these two key properties can be combined in a constructive way, then this formation can effectively be used for knowledge discovery in large databases In this context, in the following two sections the properties of data mining and machine learning paradigms are pointed out. The present article provides an overview of the available literature on data mining, and its aspects. This is followed by Section -2 which discusses the state of art of soft computing and we discuss each of the soft computing methods in brief .the following by section-4 which brings out the relevance of different soft computing methods in data mining .finally we conclude with section -4, where the significane of soft computing in data mining is highlighted.

II. DATA MINING OVERVIEW

Simply stated, datamining refers to extracting or mining knowledge from large amounts of data. Knowledge discovery as a process is depicted in Figure 1 and consists of an iterative sequence of the following steps:

1) Data cleaning is to remove noise or irrelevant data
2) Data integration is where multiple data sources may be combined
3) Data selection where data relevant to the analysis task are retrieved from the database
4) Data transformation where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance, data mining is an essential process where intelligent methods are applied in order to extract data patterns
5) Pattern evaluation is to identify the truly interesting patterns representing knowledge based on some interestingness measures
6) Knowledge presentation where visualization and knowledge representation techniques are used to present the mined knowledge to the user.

In order to understand how and why data mining works, it’s necessary to understand the methods and tools of data mining which can be categorized as follows:

Characterization: Data characterization is a traditional summarization of general features of objects in a target class, and produces what is called characteristic rules. The data relevant to a user-specified class are normally retrieved by a database query and run through a summarization

Discrimination: Data discrimination produces what are called discriminant rules and is basically the comparison of the general features of objects between two classes referred to as the target class and the contrasting class.

Association rules: It studies the frequency of items occurring together in transactional databases, and based on a threshold called support, identifies the frequent item sets.

Classification[15][16]: Classification approaches normally use a training set where all objects are already associated with known class labels.

Prediction: Prediction is referred to the forecast of missing numerical values, or increase/ decrease trends in time related data. The major idea is to use a large number of past values to consider probable future values.

Clustering: Clustering is the process of grouping a set of

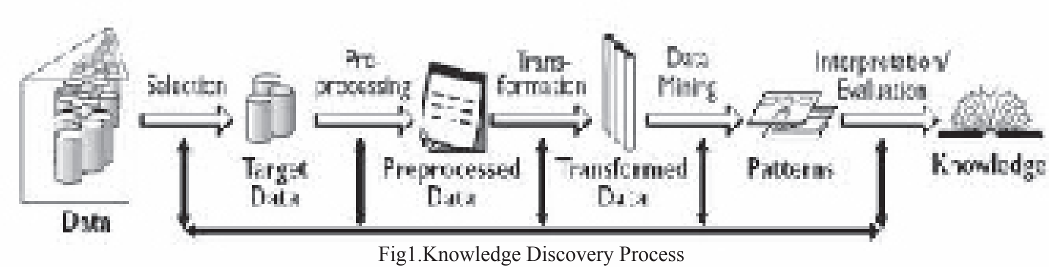


Fig1.Knowledge Discovery Process

physical or abstract objects into classes of similar objects. Clustering is also called unsupervised classification, because the classification is not dictated by given class labels.

Outlier analysis: The analysis is to identify and explain exceptions. Outliers are data elements that cannot be grouped in a given class or cluster. While outliers can be considered noise and discarded in some applications, they can reveal important knowledge in other domains, and thus can be very significant and their analysis is valuable.

Visualization: Visualization uses interactive graphs to demonstrate mathematically induced rules and scores, and is far more sophisticated than pie or bar charts. Visualization is used primarily to depict three dimensional geographic locations of mathematical coordinates[3].

PROCESS OF DATA MINING

For all these data mining algorithms and methods mentioned in the above, methodologies for data selection, cleaning, and transformation play a necessary and critical role. For data selection, data needs to extracted from different databases and joined, and perhaps sampled. Once selected, the data may need to be cleaned. If the data is not derived from a warehouse but from disparate databases, values may be represented using different notations in the different databases. Also, certain values may require special handling since they may imply missing or unknown information. After the selection and cleaning process, certain transformations may be necessary. Once the mining is done, visualization plays an important role in providing adequate bandwidth between the results of the data mining and the end user Fig.1.

III. AN OVERVIEW OF SOFT COMPUTING

Soft computing is a set of methodologies that works hand in hand and provides, some form of flexible information processing capability [10]. Its aim is to exploit the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieve tractability, robustness, and low-cost solutions[11]. Methodologies like fuzzy sets, neural networks, genetic algorithms, and rough sets are most widely applied in the data. An excellent survey demonstrating the significance of soft computing tools in data mining problem is provided by Mitra et al. [5].

SOFT COMPUTING METHODS

FUZZY LOGIC

The concept of fuzzy logic was conceived by Lotfi Zadeh. Fuzzy logic is an organized method for dealing with imprecise data. This data is considered to be as fuzzy sets[2]. Fig.2 shows how values for the continuous attribute income are mapped into the discrete categories flow, medium, high, as well as how the fuzzy membership or truth values are calculated. Fuzzy logic systems typically provide graphical tools to assist users in this step.

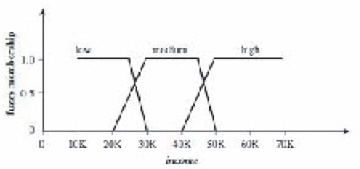


Fig.2 Fuzzy Logic

Rather than having a precise cutoffs between categories or sets, fuzzy logic uses truth values between 0:0 and 1:0 to represent the degree of membership that a certain value has in a given category[2].

NEURAL NETWORKS

An artificial neural network (ANN), [2], [4] often just called a “neural network” (NN), is a mathematical model or computational model based on biological neural networks, in other words, is an emulation of biological neural system. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. This behaviour of the neuron can be captured by a simple model which can be shown as:

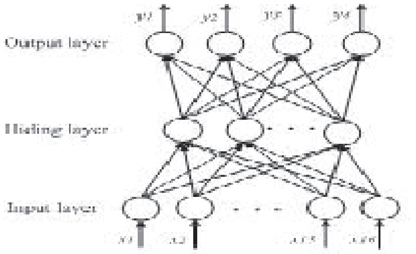


Fig.3 Neural Network

Every component of the model bears a direct analogy to the actual constituents of a biological neuron and hence is termed as ANN.

ROUGH SET APPROACH

Rough set theory was developed by Pawlak[10] for classification analysis of data bases. Rough set theory[2], can be used for classification to discover structural relationships within imprecise or noisy data. It applies to discrete-valued attributes. Rough sets can be used to approximately or roughly define such classes. A rough set definition for a given class F is approximated by two sets - a lower approximation of F and an upper approximation of F. The lower approximation of F consists of all of the data samples which, based on the knowledge of the attributes, are certain to belong to F without ambiguity. The upper approximation of F consists of all of the samples which, based on the knowledge of the attributes, cannot be described as not belonging to F. The lower and upper approximations for a class F are shown in Fig.4, where each rectangular region represents an equivalence class. Decision rules can be generated for each class. Rough sets can also be used for feature reduction. The problem of finding the minimal subsets of attributes that can describe all of the concepts in the given data set is NP-hard[10][14].

GENETIC ALGORITHM

Genetic algorithm **(GA),** belonging to a class of randomized heuristic and adaptive search techniques based on the principal of natural selection, is an attractive tool to find near optimal solutions for optimization problems. Genetic algorithms [2]attempt to incorporate ideas of natural evolution. In general, genetic learning starts as follows. An initial population is created consisting of randomly generated rules. Each rule can be represented by a string of bits.

IV.RELEVANCE OF SOFTCOMPUTING METHODS IN DATAMINING

Each of the soft computing methods have their own characteristic ,based upon which they can be suitably used in data mining process .Encapsulation of each of these methods in the data mining process has brought about a significant difference in the approach of information extraction and processing.

NEURAL NETWORKS IN DATAMINING

Neural Network based data mining approach consists of three major phases[3]

Network construction and training: This phase constructs and trains a three layer neural network based on the number of attributes and number of classes and chosen input coding method.

Network pruning: The pruning phase aims at removing redundant links and units without increasing the classification error rate of the network.

Rule extraction: This phase extracts the classification rules from the pruned network. The rules generated are in the form of "if *(a, Bv,) and* (x, *Bv,) and ... and* ***(x,*** *Bv,)* then *Cy* where *a,s* are the attributes of an input tuple, *v* ,are constants,& are relational operators (=, <, *2,* <>), and ***Ci*** is one of the class labels.

Neural networks have found wide application in areas such as pattern recognition, image processing, optimization, fore casting.

FUZZY LOGIC IN DATAMINING

Since fuzzy sets allow partial membership, Fuzzy logic is basically multivalued logic that allows intermediate values to be defined between conventional evaluation such as yes/no, true/false etc[4]. The use of fuzzy techniques has been considered to be one of the key components of data mining systems because of the affinity with human knowledge representation [10]. Wei and Chen [9]have mined generalized association rules with fuzzy taxonomic

structures. Fuzzy logic systems have been used in numerous areas for classification, including health care and finance. Fuzzy logic is useful for data mining systems performing classification. It provides the advantage of working at a high level of abstraction. Decision making is very important in data mining which involves social, economical and scientific applications. At this junction fuzzy data mining comes as great help to data miners.

ROUGH SETS IN DATAMINING

The main goal of rough sets is induction of approximation of

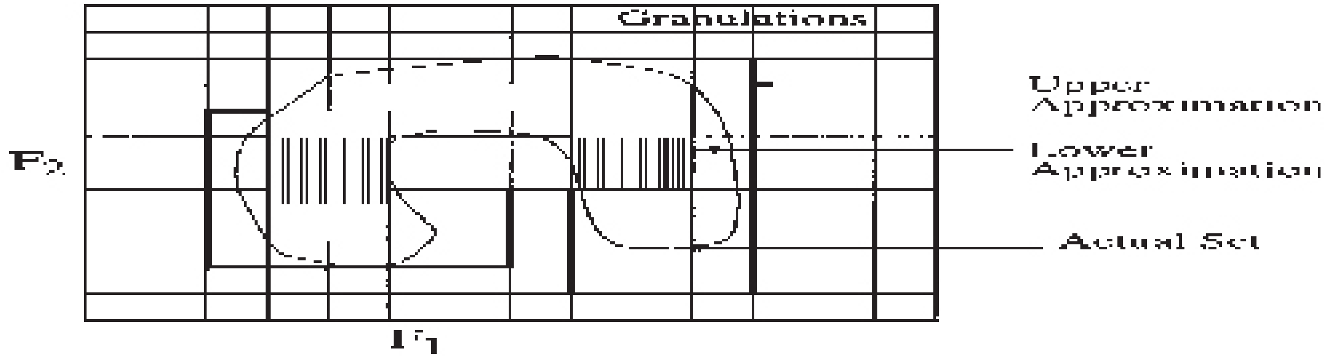


Fig.4. Rough set approach

concepts[6].Rough sets constitutes a sound basis for KDD. It offers mathematical tools to discover patterns hidden in data and hence used in the field of mining. Rough sets can be used as a framework for data mining specially in the areas of soft computing where exact data is not required and in some areas where approximation data can be of great help. Rough set theory can be used in different steps in data processing such as computing lower and upper approximation. Analyzing knowledge, Computing accuracy and quality of approximations, Calculating minimal number of attributes describing the concepts and deriving a desicion algorithm as asset of attributes. analyzing conflicts in data[17]. An excellent survey demonstrating the significance of soft computing tools in data mining problem is provided by Mitra et al. [5].

V. CONCLUSION

Statistic plays a vital role in the methods of data mining. The methods of data mining are inherited from strict sciences. Soft computing methods are fundamentally used for information processing by employing methods, which are skilled to deal with imprecision, vagueness and uncertainty ,needed in application areas where the problems are not hazy. The outcomes are exact within the error bounds estimated where as in the case of soft computing they are approximate and in some instances they may be interpreted as outcomes from an intelligent behavior. From these basic properties, it may be concluded that, both paradigms have their own merits and by observing these merits synergistically these paradigms can be used in a complementary way for knowledge discovery in databases. In this paper, we have discussed the different soft computing methods and further we have put forward the relevance of each of these methods in data mining.

REFERENCES

[1] D. E. Goldberg, Genetic Algorithms in Search, Optimization,and Machine Learning, Addison-Wesley, 1989.

[2] Dataminig Concepts and Techniques- Jiawei Han, Micheline Kamber

[3] Effective Data MiningUsing Neural Networks Hongjun Lu, Rudy Setiono, and Huan Liu, IEEE Transactions on knowledge and data engineering,vol.8,No.6 December 1996.

[4] Principles of Soft Computing-S.N.Sivanandam and S.N.Deepa

[5] Sushmita Mitra , Sankar K. Pal, Pabitra Mitra ,"Data Mining in Soft Computing FrameworkA Survey", IEEE Transactions on Neural Networks, Vol. 13, No. 1, January 2002

[6] J.Grzymala-Busse,R.Swiniarski,N.Zhong and Z.Zizrko International Journel of Apploied mathematics and Computer science,Special Issue on Rough Sets, and Its Applications.

[7] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data*. Dordrecht, : Kluwer, 1991.

[8] A. Skowron and C. Rauszer, "The Discernibility Matrices and functions in Information Systems," Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory, R. Slowi_nski, ed. pp. 331-362, Dordrecht: Kluwer Academic, 1992..

[9] Q. Wei and G. Chen, "Mining generalized association rules with fuzzy taxonomic structures," in *Proc. NAFIPS 99*, New York, June 1999, pp.477–481

[10] A. Maeda, H. Ashida, Y. Taniguchi, and Y. Takahashi, "Data mining system using fuzzy rule induction," *Proc. IEEE Int. Conf. Fuzzy Syst.* Fuzz IEEE *95*, pp. 45–46, Mar. 1995.

[11] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data*. Dordrecht, : Kluwer, 1991.

[12] I.W. Flockhart and N. J. Radcliffe, "A genetic algorithm-based approach to data mining," in *Proc. 2nd Int. Conf. Knowledge Discovery DataMining (KDD-96)*. Portland, OR, Aug. 2–4, 1996, p. 299.

[13] M. L. Raymer, W. F. Punch, E. D. Goodman, and L. A. Kuhn "Genetic programming for improved data mining: An application to the biochemistry of protein interactions," in *Proc. 1st Annu. Conf. Genetic Programming 1996*, Stanford Univ., CA, July 28–31, 1996, pp. 375–380.

[14] A. Teller and M. Veloso, "Program evolution for data mining," *Int. J. Expert Syst.*, vol. 8, pp. 216–236, 1995

[15] T. M. Mitchell, "Machine learning and data mining," *Commun. ACM*, vol. 42, no. 11, 1999.

[16] U. Fayyad, G. P. Shapiro, and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Commun. ACM*, vol. 39, pp. 27–34, 1996.

[17]H.S.Nguyen and S.H.Nguyen,"Discretization Methods in Data Mining",Vol.1,451-482,Physica-Verlag(1998).

.

Implementation of Mango Expert Advisory System using Parallel ABC algorithm with SMA technique

Ms. P.T.Swati, M.Tech (CST)
Dept. of CSE, VIIT, Visakhapatnam

Mr. DVPR Sridhar
Assoc. Professor, Dept. of CSE, VIIT, Visakhapatnam

Prof. M. S. Prasad Babu
Dept. of CS & SE, Andhra University, Visakhapatnam

Abstract: The present developed paper deals with the development of expert systems using machine learning algorithm techniques to advice the farmers through online in villages.An expert system can be defined as an application program that makes decisions or solves problems by using knowledge and analytical rules defined by a subject expertise in the field. Machine Learning is a mechanism ,used in the development of Expert systems, concerned with writing a computer program that automatically improves with experience. ABC Algorithm was considered as base and designed a new algorithm known as Parallel Artificial Bee Colony (ABC) Optimized Algorithm. Using this Parallel ABC Optimized Algorithm, we developed a new 'Mango Expert Advisory System'. This system is mainly aimed for identifying the diseases and disease management in mango fruits and mango plants to advise the farmers through online in the villages to obtain standardized yields. The present advisory system is designed by using JSP as front end and MYSQL as backend.

Key words: Expert Systems, Machine Learning, ABC Algorithm, Parallel ABC, Optimization, Mango Crop, JSP & MYSQL.

INTRODUCTION

Expert Systems:

An expert system is a computer programme where data is stored and manipulated by the programme to come up with advises, hints, directions in reaction of input by users of data acquisition devices. Expert systems are used in professional areas like Diagnostics (Medicines), Construction, and Simulation.

Since the growth rate in this specific field is 100 percent it is to be expected that expert systems will be applied in other areas. In principle is an expert system a further development of the third generation programming languages.The key idea in expert systems technology is that it is not explicitly programmed when the system is built and problem solving is accomplished by applying specific knowledge rather than specific technique to the developing system.

Mango:

Mango is one of the most commonly used fruits in India. Its scientific name is Mangiferia indica. It is the leading fruit crop in India and it is considered to be the king of fruits. Mango occupies around 22% of the total crop production in India.

Even though mango is cultivated in all over India but Karnataka, TamilNadu, Andhra Pradesh and Bihar are the premium producers of Mango in India. There are about five popular varieties in mango. Mangoes contain phenols which is having powerful antioxidant and anticancer abilities. Another benefit of mango is it is having iron in very high amounts which can help to pregnant women and people with anemia are advised to eat this fruit regularly. In addition to all these health benefits, mango is packed with vitamins and nutrients. Some of these include Vitamins A, E and Selenium and many others.

Machine Learning:

Machine Learning is a scientific discipline which is concerned with the design and development of computer programs that automatically improves with experience. It is a very young scientific discipline whose birth can be placed in the mid-seventies. Machine learning usually refers to the changes in systems that perform tasks associated with artificial intelligence (AI). Such tasks involve recognition, diagnosis, planning, robot control, prediction, etc. The changes might be either enhancements to already performing systems or synthesis of new systems**.**

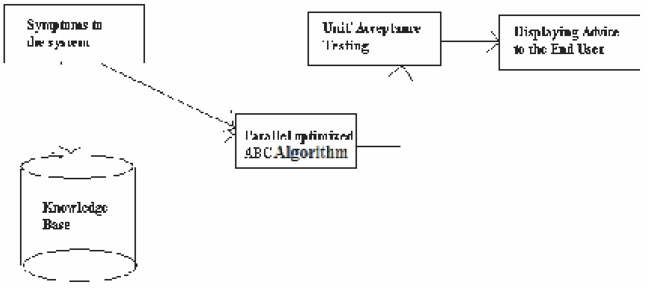
ABC Algorithm:

The Artificial Bee Colony (ABC) Algorithm [1] is a meta-heuristic algorithm for numerical optimization. Many meta-heuristic algorithms, inspired from nature, are efficient in solving numerical optimization problems. ABC algorithm is motivated by the intelligent foraging behavior of honey bees. The ABC algorithm [2,3] was first proposed by Karaboga in 2005 for unconstrained optimization problems. Subsequently, the algorithm has been developed by Karaboga and Basturk and extended to constrained Optimization problems. Improvements to the performance of the algorithm and a hybrid version of the algorithm have been also been proposed. The ABC algorithm is a swarm-based algorithm. It is very simple and flexible when compared to the other Swarm Based algorithms such as Particle Swarm Optimization (PSO) and Ant

Colony Optimization Algorithms. Researchers have come up with several real-world applications for the ABC algorithm.

Proposed System:

A Web Application of Expert Advisory System for Mango Fruits and Mango Plants is developed by using ABC Algorithm as base and modified this ABC Algorithm as parallel Artificial Bee Colony Optimization Algorithm. The proposed Architecture is as follows:


Symptoms to the system
Unit/ Acceptance Testing
Displaying Advice to the End User
Parallel optimized ABC Algorithm
Knowledge Base

Proposed Algorithm:

In general, parallel architectures may use either a shared memory or a message passing mechanism to communicate in between the multiple processing elements. Parallel meta-heuristic algorithms have been developed for both these kinds of architectures. A parallel implementation of the algorithm is designed for an optimized architecture, which overcomes these dependencies. The entire colony of bees is divided equally among the available processors. Each processor has a set of solutions in a local memory. A copy of each solution is also maintained in a global shared memory. During each cycle the set of bees at a processor improves the solutions in the local memory. The output optimization can be taken as the total number of symptoms matching divided by the total number of symptoms in the system. At the end of the cycle, the solutions are copied into the corresponding slots in the shared memory by overwriting the previous copies. The solutions are thus made available to all the processors.

Step.1. Generate SN initial solutions randomly and evaluate them. Place them in the shared memory S.

Step.2. Divide the solutions equally among p processors by copying SNp solutions to the local memory of each processor.

Step.3. Steps 4 to 10 are carried out in parallel at each processor Pr.

Step.4. For each solution in the local memory Mr of the range processor Pr, determine a neighbor.

Step.5. Calculate the optimization for the solutions in Mr.

Step.6. Place the onlookers on the food sources in Mr and improve the corresponding solutions (as in step 4).

Step.7. Determine the abandoned solution (if any) in Mr and replace it with a new randomly produced solution.

Step.8. Record the best local solution obtained till now at Pr.

Step.9. Copy the solutions in Mr to the corresponding slots in S.

Step.10. Determine the global best solution among the best local solutions recorded at each processor.

The parallelism of an algorithm can be implemented in two ways. The first is the Parallelism can be implemented by using different processors over different systems, and the second process is the implementing the concept of multithreading for achieving the parallelization of an algorithm. In the proposed algorithm, the parallelism is implemented by the second process of multithreading. By using the Shared Memory Architecture, we split the knowledge base in to different processes and each process is implemented parallel by multithreading.

Database Generation:

In this section, the setup for production rules in the knowledge base is presented. Generally the rules are of the form,
R1: S1=1, S2= 0, S3= 0,S4= 0, S5=0,S6= 1,S7= 0,S8=1, S9= 0,S10= 0,S11= 0,S12= 0
Resultant disease may be D1

R2: S1= 1, S2=1, S3= 0,S4= 0, S5= 0,S6= 0 ,S7=1,S8= 0 ,S9= 0 ,S10= 0 ,S11=0,S12= 1
Resultant disease may be D3

R3: S1= 0,S2= 1 ,S3= 0 ,S4= 0 , S5= 1,S6= 1 ,S7= 0,S8= 0 ,S9= 0 ,S10=1 ,S11=0 ,S12= 0
Resultant disease may be D5.

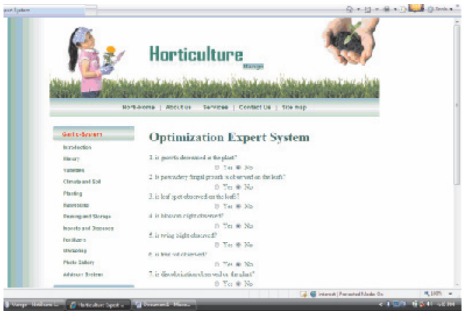
Table:1 Database format

Disease	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	Cure	Rules
D1	1	1	0	0	0	0	0	0	0	0	0	0	0	C1	R1
D2	0	0	1	0	1	1	0	0	0	0	0	0	0	C2	R2
D3	0	0	0	1	0	0	1	1	0	0	0	0	0	C3	R3
D4	0	0	1	0	0	0	1	1	0	0	1	0	0	C4	R4
D5	0	0	0	0	0	0	0	0	1	1	0	1	0	C5	R5
D6	0	0	0	0	0	0	0	0	0	0	1	1	1	C6	R6

A rule is created by using the different combinations of symptoms for identifying the disease. Here, in our system the rules are created by using the different possible combinations of symptoms to identify the affected diseases in mango crop or fruits. A rule is in the form of combination of symptoms in terms of binary values where '1' stands for presence of disease and '0' stands for absence of disease. Using this methodology, we had created the rules in the knowledge base in above form of representation.

Test Results:

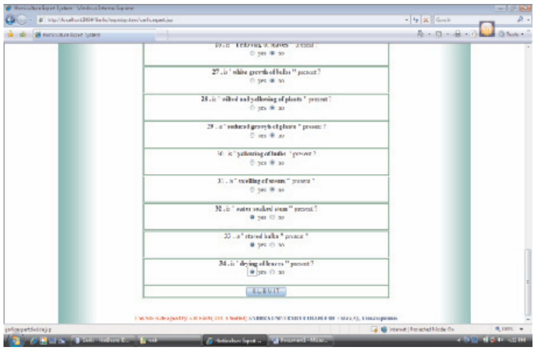
Fig:1. Selection of Symptoms
Fig:2. Selection of Symptoms



Optimization Expert System

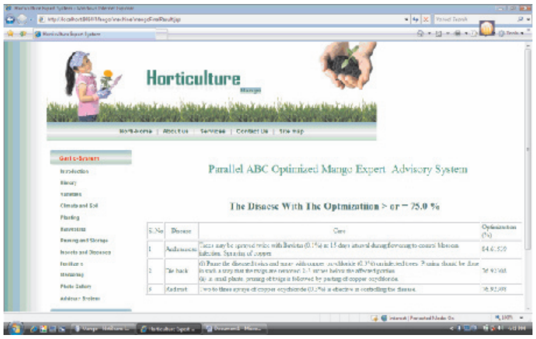
1. is growth decreased in the plant?
○ Yes ● No
2. is powdery fungal growth is observed on the leaf?
○ Yes ● No
3. is leaf spot observed on the leaf?
○ Yes ● No
4. is blossom blight observed?
○ Yes ● No
5. is twig blight observed?
○ Yes ● No
6. is tree rot observed?
○ Yes ● No
7. is discoloration observed on the plant?
○ Yes ● No

Fig: 3. Displaying advice to the end user



[illegible] "yellowing of leaves" present?
○ yes ● no
27. is "white growth of bulbs" present?
○ yes ● no
28. is "wilted and yellowing of plants" present?
○ yes ● no
29. is "reduced growth of plants" present?
○ yes ● no
30. is "yellowing of bulbs" present?
○ yes ● no
31. is "swelling of stems" present?
○ yes ● no
32. is "water soaked stem" present?
● yes ○ no
33. is "stored bulbs" present?
● yes ○ no
34. is "drying of leaves" present?
● yes ○ no

CONCLUSIONS:



Parallel ABC Optimized Mango Expert Advisory System

The Disease With The Optmization > or = 75.0 %

S.No	Disease	Cure	Optimization (%)
1	Anthracnose	Trees may be sprayed twice with Benlate (0.1%) at 15 days interval during flowering to control blossom infection. Spraying of copper	84.61538
2	Die back	(i) Prune the diseased twigs and spray with copper oxychloride (0.3%) on infected trees. Pruning should be done in such a way that the twigs are removed 2-3 inches below the affected portion (ii) in small plants, pruning of twigs is followed by pasting of copper oxychloride.	76.92308
3	Red rust	Two to three sprays of copper oxychloride (0.3%) is effective in controlling the disease.	76.92308

In the proposed system, A Parallel Implementation of Optimized Artificial Bee Colony (ABC) Algorithm was developed which gives better results compared to implementation of general ABC Algorithm. In the present investigation it was found that, the Parallel Optimized ABC gives a better optimization compared with general ABC Algorithm. The algorithm used in the present system can be treated as quite effective; in most of the cases it finds a solution which represents a good approximation to the optimal one. Its main emphasis is to have a well designed interface for giving mango plant related advices and suggestions sin the area to farmers by providing facilities like online interaction between expert system and the user without the need of expert all times. By the thorough interaction with the users and beneficiaries the functionality of the System can be extended further to many more areas in and around the world.

REFERENCES:

1. B.Basturk, Dervis Karaboga, An Artificial Bee Colony (ABC) Algorithm for Numeric function Optimization, IEEE Swarm Intelligence Symposium 2006, May 12-14, 2006, Indianapolis, Indiana, USA.
2. D. Karaboga, B. Basturk, A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm, Journal of Global Optimization, Volume: 39, Issue: 3, pp: 459-471, Springer Netherlands, 2007. doi: 10.1007/s10898-007-9149.
3. D. Karaboga, B. Basturk, On The Performance Of Artificial Bee Colony (ABC) Algorithm, Applied Soft Computing, Volume 8, Issue 1, January 2008, Pages 687-697. doi:10.1016/j.asoc.2007.05.007
4. D. Karaboga, B. Basturk, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, Vol: 4529/2007, pp: 789-798, Springer- Verlag, 2007, IFSA 2007. doi: 10.1007/978-3-540-72950-1_77
5. D. Karaboga, B. Basturk Akay, Artificial Bee Colony Algorithm on Training Artificial Neural Networks, Signal Processing and Communications Applications, 2007. SIU 2007, IEEE 15th. 11-13 June 2007, Page(s):1 - 4, doi: 10.1109/SIU.2007.4298679
6. D. Karaboga, B. Basturk Akay, C. Ozturk, Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks, LNCS: Modeling Decisions for Artificial Intelligence, Vol: 4617/2007, pp:318-319, Springer-Verlag, 2007, MDAI 2007. doi: 10.1007/978-3-540-73729-2_30

Optical Character Recognition Using Multilayer Perceptron

Sushma Malviya
Asstt. Preof., MCA Department
People's Instt. of Management & Research, Bhopal (MP)
e-mail : sush1109@rediffmail.com

***Abstract* - This paper describes creating the Character Recognition System, in which Creating a Character Matrix and a corresponding Suitable Network Structure is key. In addition, knowledge of how one is Deriving the Input from a Character Matrix must first be obtained before one may proceed. Afterwards, the Feed Forward Supervised Learning Algorithm gives insight into the workings of a neural network; followed by the Multi Layer Perceptron Algorithm which compromises training, Calculating Error, and Modifying Weights.**

Key words: MLP, Document digitization, OCR,, pattern recognition, supervised learning.

1. INTRODUCTION

Optical character recognition, usually abbreviated to OCR, is the mechanical or electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editable text. OCR is a field of research in pattern recognition, artificial intelligence and machine vision. Though academic research in the field continues, the focus on OCR has shifted to implementation of proven techniques. Optical character recognition (using optical techniques such as mirrors and lenses) and digital character recognition (using scanners and computer algorithms) were originally considered separate fields. Because very few applications survive that use true optical techniques, the OCR term[2] has now been broadened to include digital image processing as well. A Neural Network (NN)[12] is a wonderful tool that can help to resolve OCR type problems. Of course, the selection of appropriate classifiers is essential. The concept of Neural Networks[13] is highly inspired by the recognition mechanism of the human brain. There is no universally accepted definition of neural network[4], but there are some architectural and fundamental elements that are the same for all neural networks The use of Artificial Neural Network implementations with networks employing specific guides (learning rules) to update the links (weights) between their nodes. Such networks can be fed the data from the graphic analysis of the input picture and trained to output characters in one or another form.

Specifically some network models use a set of desired outputs to compare with the output and compute an error to make use of in adjusting their weights. Such learning rules are termed as Supervised Learning. One such network with supervised learning rule is the Multi-Layer Perception (MLP) model. It uses the Generalized Delta Learning Rule for adjusting its weights and can be trained for a set of input/desired output values in a number of iterations. We employed the MLP technique mentioned and excellent results were obtained for a number of widely used font types. The applications of this technique range from document digitizing and preservation to handwritten text recognition in handheld devices

2. CREATING THE CHARACTER RECOGNITION SYSTEM

The matrixes of each letter of the alphabet must be created along with the network structure. In addition, one must understand how to pull the Binary Input Code from the matrix, and how to interpret the Binary Output Code, which the computer ultimately produces.

Character Matrixes

A character matrix is an array of black and white pixels; the vector of 1 represented by black, and 0 by white. They are created manually by the user, in whatever size or font imaginable; in addition, multiple fonts of the same alphabet may even be used under separate training sessions.

Creating a Character Matrix

First, in order to endow a computer with the ability to recognize characters, we must first create those characters. The first thing to think about when creating a matrix is the size that will be used. Too small and all the letters may not be able to be created, especially if you want to use two different fonts. On the other hand, if the size of the matrix is very big, their may be a few problems: Despite the fact that the speed of computers double every third year, their may not be enough processing power currently available to run in real time. Training may take days, and results may take hours. In addition, the computer's memory may not be able to handle enough neurons in the hidden layer needed to efficient and accurately process the information.

However, the number of neurons may just simply be reduced, but this in turn may greatly increase the chance for error. A large matrix size of 10 x 15 was created, through the steps as explained above.

First Font	Second Font
0000000000	0000000000
0000110000	1111111111
0001111000	1111111111
0011111100	1100000011

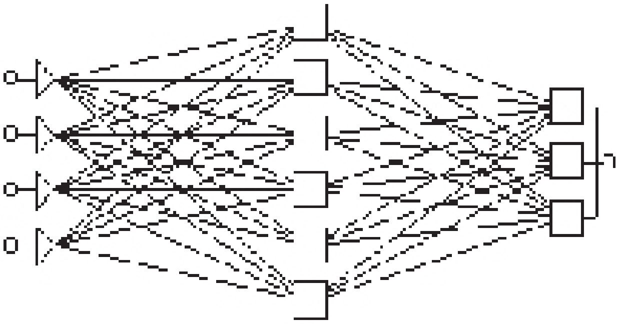
```
0110000110          1100000011
1100000011          1100000011
1100000011          1111111111
1100000011          1111111111
1111111111          1100000011
1111111111          1100000011
1100000011          1100000011
1100000011          1100000011
1100000011          1100000011
1100000011          1100000011
1100000011          1100000011
```

Figure 1 Character Matrix A of Different Fonts

3. MULTI LAYER PERCEPTRON

In Neural Networks, there are two different overall Learning paradigms [7]. The first one is supervised learning, also known as learning with a teacher. The second one is called unsupervised learning also referred to as the learning without a teacher A multilayer perceptron is a supervised feed forward artificial neural network model that maps sets of input data onto a set of appropriate output. It is a modification of the standard linear perceptron in that it uses three or more layers of neurons (nodes) with nonlinear activation functions, and is more powerful than the perceptron in that it can distinguish data that is not linearly separable, or separable by a hyper plane.

Figure 2 Typical Feedforward Network



A typical feed forward network has neurons arranged in a distinct layered topology. The input layer is not really neural at all: these units simply serve to introduce the values of the input variables. The hidden and output layer neurons are each connected to all of the units in the preceding layer. Again, it is possible to define networks that are partially-connected to only some units in the preceding layer; however, for most applications fully-connected networks are better.

4. ARCHITECTURE

The Multi-Layer Perceptron Neural Network is the most popular network architecture in use today. The units each perform a biased weighted sum of their inputs and pass this activation level through an activation function to produce their output, and the units are arranged in a layered feed forward topology. The network thus has a simple interpretation as a form of input-output model, with the weights and thresholds (biases) the free parameters of the model. Such networks can model functions of almost arbitrary complexity, with the number of layers, and the number of units in each layer, determining the function complexity. Important issues in Multilayer Perceptrons MLP) design include specification of the number of hidden layers and the number of units in each layer.

5. ACTIVATION FUNCTION

Most common activation functions[14] are the logistic and hyperbolic tangent sigmoid functions. The activation function used here are the **hyperbolic tangent function:**

$$f(x) = \frac{2}{(1+e^{-2x})} - 1$$ **and derivative**

$$f(x) = f(x)(1-f(x))$$

6. NETWORK FORMATION

The MLP Network implemented for the purpose is composed of 3 layers, one input, one hidden and one output.

The input layer constitutes of 150 neurons which receive pixel binary data from a 10 x 15 symbol pixel matrix. The size of this matrix was decided taking into consideration the average height and width of character image that can be mapped without introducing any significant pixel noise[8]. The hidden layer constitutes of 250 neurons whose number is decided on the basis of optimal results on a trial and error basis. The output layer is composed of 16 neurons corresponding to the 16-bits of Unicode encoding.

To initialize the weights[9] a random function was used to assign an initial random number which lies between two preset integers named *weight_bias*. The weight bias is selected from trial and error observation to correspond to average weights for quick convergence.

Here the parameters used are:

Learning rate = 150

Sigmoid Slope = 0.014

Weight bias = 30 (determined by trial and error)

Number of Epochs = 300-600 (depending on the complexity of the font types)

Mean error threshold value = 0.0002 (determined by trial and error)

7. TESTING

The testing phase of the implementation is simple and straightforward. Since the program is coded into modular parts the same routines that were used to load, analyze and compute network parameters[10] of input vectors in the training phase can be reused in the testing phase as well

The network has been trained and tested for a number of widely used font type in the Latin alphabet. Since the implementation of the software is open and the program code is scalable, the inclusion of more number of fonts from any typed language alphabet is straight forward.

The necessary steps are preparing the sequence of input symbol images[7]-[15] in a single image file (*.bmp [bitmap] extension), typing the corresponding characters in a text file (*.cts [character trainer set] extension) and saving the two in the same folder (both must have the same file name except for their extensions)

8. SYSTEM RESULT & PERFORMANCE

The reliability of the pattern recognition system is measured by testing the network with hundreds of input vectors with varying quantities of noise. At each noise level, 100 presentations of different noisy versions of each letter are made and the network's output is calculated. The output is then passed through the competitive transfer function so that only one of the 26 outputs (representing the letters of the alphabet), has a value of 1.

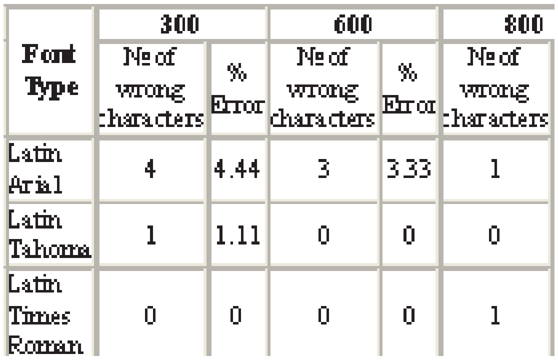
Due to the random valued initialization of weight values results listed represent only typical network performance and exact reproduction might not be obtained with other trials.

A. Results for variation in number of Epochs

Number of characters=90, Learning rate=150, Sigmoid slope=0.014

Table 1 Result for variation in number of Epochs

By increasing the number of iterations has generally a positive proportionality relation to the performance of the network.



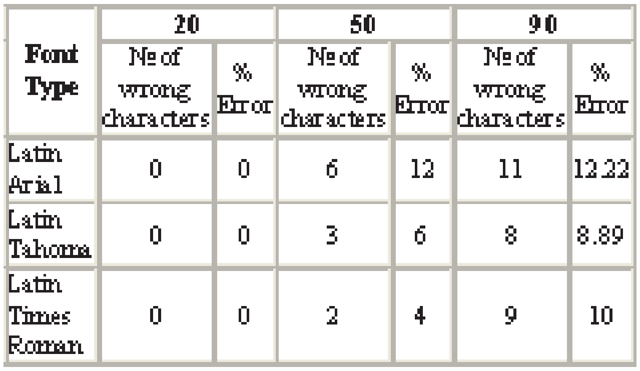
Font Type	300		600		800
	№ of wrong characters	% Error	№ of wrong characters	% Error	№ of wrong characters
Latin Arial	4	4.44	3	3.33	1
Latin Tahoma	1	1.11	0	0	0
Latin Times Roman	0	0	0	0	1

However in certain cases further increasing the number of epochs has an adverse effect of introducing more number of wrong recognitions

B. Results for variation in number of Input characters

Number of Epochs=100, Learning rate=150, Sigmoid slope=0.014

Table 2 Result for variation in number of Input characters



Font Type	20		50		90	
	№ of wrong characters	% Error	№ of wrong characters	% Error	№ of wrong characters	% Error
Latin Arial	0	0	6	12	11	12.22
Latin Tahoma	0	0	3	6	8	8.89
Latin Times Roman	0	0	2	4	9	10

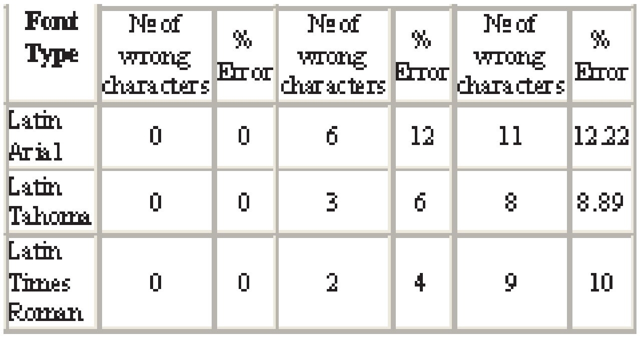
The size of the input states is also another direct factor influencing the performance. It is natural that the more number of input symbol set the network is required to be trained for the more it is susceptible for error. Usually the complex and large sized input sets require a large topology network with more number of iterations

C. Results for variation in Learning rate parameter

Number of characters=90, Number of Epochs=600, Sigmoid slope=0.014

Table 3 Result for variation in Learning rate parameter

Learning rate parameter variation also affects the network performance for a given limit of iterations. The less the value of this parameter, the lower the value with which the network updates its weights. This intuitively implies that it will be less likely to face the over learning difficulty discussed above since it will be updating its links slowly and in a more refined manner. But unfortunately this would also imply more number of iterations is required to reach its optimal state.



Font Type	№ of wrong characters	% Error	№ of wrong characters	% Error	№ of wrong characters	% Error
Latin Arial	0	0	6	12	11	12.22
Latin Tahoma	0	0	3	6	8	8.89
Latin Times Roman	0	0	2	4	9	10

9. CONCLUSION & FUTURE WORK

In this paper, in order to recognize the character efficiently which is in mass document constructed the Optical Character Recognition System, and the section that needs for recognizing standardized document style and characters are pre-designated. Optical Character Recognition System that is constructed by using artificial neural network algorithm expects to be efficient in character recognition of mass standardized document. The basic idea of using extracted features to train an ANN seems to work although the success rate is not impressive, it could have been worse. There are several possible changes that the current bottleneck for speed is the feature extraction stage. With some work, it should be possible to speed this up considerably by re-implementing it in VB.NET. The other obvious step is to increase the training data set. This requires some effort, but clearly more training data will lead to a more robust and accurate ANN. Some other fairly trivial features are still missing. For example, characters like the apostrophe (‘) and comma (,) look very similar and can be distinguished only be vertical location. The same holds for the dot on the i and a period.

Looking at the actual results, it is easy to see certain patterns. There are certain groups of characters which are often confused. Examples include the set (fi, t, l, f, Ig). This fact can be used in a manner suggested by Sural and Das (1999), namely, we can use multiple ANN's for classification. In the burst stage, we would have some ‘super classes’ which consist of more than one glyph that look very similar. For glyphs classified to such ‘super classes’, another ANN tailored for that particular group can be used to resolve the ambiguity. However, this is not a trivial task, and would have to be carefully designed.

10. REFERENCES

1 Bigus, Joseph P, “Data Mining with Neural Networks”, McGraw Hill, 1996. 24. Wenke Lee, Sal Stolfo and Kui Mok. ‘CA, May 1999.

2 Chaudhuri, B.B., Pal, U., “A complete OCR system”,

3. Hastie, T., Tibshirani, R., and Friedman, J. The Elements of Statistical Learning, Springer 2001

4. Neural Network, A. CS229 Lecture Notes, Fall 2008

5. Suzuki et al. INFTY - An Integrated OCR System for Mathematical Documents, DocEng ’03, Grenoble,France. 2003

6. Muller et al. An Introduction to Kernel-Based Learning Algorithms, IEEE Transactions on Neural Networks, Vol. 12, No. 2, March 2001

7. Barber, D. Learning from Data: Dimensionality Reduction, 2004

8. Shlens, J. A Tutorial on Principle Component Analysis, Institute for Nonlinear Science, UCSD, 2005

9. Dailey, M. Principal Component Analysis Tutorial, Asian Institute of Technology, 2008

10. Le Cun, Y., Bottou, L., and Ha_ner, P. (1998). Gradient-based learning applied to document regognition, Proceedings of the IEEE,86(11), 2278-2324

11. Mardia, Kanti V. “Statistics and Images”, Vol 1, Vol 2Cartax Publishing Company, Oxford, UK, 1995.

12. T. Matsuoka, H. Hamada and R. Nakatsu, “Syllable Recognition Using Integrated Neural Networks,” in Proceedings of the IEEE IJCNN, Vol. I, pp. 251-258, 1989. 13. A. Waibel, Connectionist Glue: Modular Design of Neural Speech Systems,” in Proceedings of the 1988 Connectionist Models Summer School, pp. 417-425, 1988.

14. J. L. Mcclelland and D.E. Rumelhart, “Learning Internal Prepresentation by Error Propagation,”, Parrallel Distributed Processing, Vol. 1, 1986.

15. Chaudhuri, B.B., Pal, U., “A complete Bangla OCR system”, Pattern Recognition, Vol. 31, pp. 531–549, 1998.

16. Cash, G. and Hatamian, M. “Optical character recognition by the method of moments”. Computer Vision, Graphics, and Image Processing, Vol. 39, pp. 291-310, 1987.

17. De Luca, P. and Gisotti, A. “Printed character preclassification based on word structure”. Pattern Recognition, Vol. 24, pp. 609- 615, 1991.

18. R.M.K. Sinha, et al., “Hybrid contextual text recognition with string matching”, IEEE PAMI, Vol. 15, pp. 915–923, 1993.

Application of Decision Tree for Fuzzy Classifying in Educational Organization

Purnendu Ruj[1] and Pabitra Kumar Dey[2]

Student, M.Tech. (CST), Dept. of Computer Science & Engineering[1], Sr.Lecturer, Department of Computer Application[2], Dr. B.C Roy Engineering College, Durgapur-713206,West Bengal, India

Abstract: **Decision tree induction and Classification are two of the most prevalent data mining techniques used separately or together in many business applications. Researchers from various disciplines such as statistics, machine learning, pattern recognition, and data mining considered the issue of building a decision tree from the available data. The ability to analyze large amounts of data for the extraction of valuable information presents a competitive advantage for any educational organization. The technologies of data mining support that ability. Generating generalize function, which depends on the data set, and depending on the generalize function, construct decision tree. This article proposed the application of decision tree to improve decision support information for educational organization. It will help to gather student performance information by their overall grading system, which will be not only depends on marks but also depends on some extra qualities like attendance or behavior.**

Keywords: Data Mining, Decision Tree Induction, Fuzzy Logic and Classification.

I. INTRODUCTION

Data mining is most useful in an exploratory analysis scenario in which there are no predetermined notions about what will constitute an "interesting" outcome. Data classification, an important task of data mining, is the process of finding the common properties among a set of objects in a database and classifies them into different classes. Decision Trees are widely used in classification [1]. It assigns class labels to data objects based on prior knowledge of class which the data records belong. However, integration of an that deals with knowledge extraction from database records and prediction of class label from unknown data set of records (Tan et al, 2006)[2]. In classification a given set of data records is divided into training and test data sets. The training data set is used in building the classification model, while the test data record is used in validating the model. The model is then used to classify and predict new set of data records that is different from both the training and test data sets (Garofalakis et al, 2000 and Gehrke et al, 1998)[3,4]. Supervised learning algorithm (like classification) is preferred to unsupervised learning algorithm (like clustering) because its prior knowledge of the class labels of data records makes feature/attribute selection easy and this leads to good prediction/classification accuracy. Some of the common classification algorithms used in data mining and decision support systems are: neural networks (Lippmann, 1987)[5], logistic regression (Khoshgoftaar et al, 1999)[6], Decision trees (Quinlan, 1993)[7] etc. Among these classification algorithms decision tree algorithms is the most commonly used because of it is easy to understand and cheap to implement. It provides a modeling technique that is easy for human to comprehend and simplifies the classification process (Utgoff and Brodley, 1990)[8].

Accurate information about an educational organization's state is necessary in order to make strategic decisions. An increasing number of heterogeneous information systems and an expanding data volume make retrieving meaningful information more difficult. Educational organizations use information system. In the same way, they struggle with a growing data volume and a difficulty in making use of the stored data. However, decision support information for education is usually less centered on increase of financial benefit. What are more important are the reduction of effort and the improvement of quality. Student grading is a relatively new term in educational information systems. It is often used to describe the performance-based on overall student activity, including their marks, attendance, behavior etc. or generally the use of information technology for the improvement of teaching quality. A decision tree in an educational environment can enable educational organizations to make information accessible and analyzable in order to provide a basis for their decision support.

This article proposed the application of decision tree to improve decision support information for educational organization. It will help to gather student performance information by their overall grading system, which will be not only depends on marks but also depends on some extra qualities like attendance or behavior.

The paper is organized as follows: Section 2 discuss about the Data Classification with Decision Tree Induction. Section 3 focuses about the basic concepts of data mining. Section 4 represents the design of fuzzy database taking student gradation dataset into account. Experiment and results are carried out on section 5. Finally, section 6 concludes the paper.

II. DATA CLASSIFICATION WITH DECISION TREE INDUCTION

The concept of classification as it is understood in this article stems from the field of data analysis. Generally, data elements are classified by calculating a membership function for given classes. The fuzzy classification is an extension of the traditional classification in which each object is assigned to exactly one class; meaning that the membership degree of the object is 1 in this class and 0 in all the others and the objects in the classes is therefore mutually exclusive. In contrast, a fuzzy classification allows the objects to belong to several classes at the same time; furthermore, each object has membership degrees which express to what extent this object belongs to the different classes. Fuzzy classification is described as the process of assigning data elements to fuzzy sets by using fuzzy dimension value categories as classification features and may be defined as follows: Let an Object O be given. This object is characterized by a t-dimensional feature vector x_O of a universe of discourse U. A set { $c_1, \ldots, c_n$ } of classes is given. The task is to calculate a membership vector ($m_1, \ldots, m_n$) for the object O, where m_i is the degree of membership of O to class c_i.

A decision tree is a classification scheme, in which the set of records available for developing classification methods is generally divided into two disjoint subset - a training set and a test set. The former is used for deriving the classifier, while the latter is used to measure the accuracy of the classifier. The accuracy of the classifier is determined by the percentage of the test examples that are correctly classified. A decision tree is a predictive modeling technique used in classification, clustering and predictive tasks. Decision trees use a "divide and conquer" technique to split the problem search space into subsets.

Decision Tree (DT):

A ***Decision Tree Model*** is a computational model consisting of three parts:

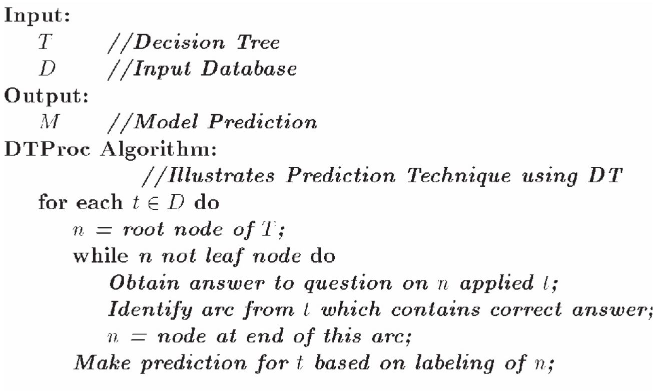
Decision Tree, where the root and each internal node is labeled with a question, the arcs represent each possible answer to the associated question and each leaf node represents a prediction of a solution to the problem.

Algorithm to create the tree

Algorithm that applies the tree to data

Advantages:

- Easy to understand.
- Easy to generate rules



```
Input:
  T    //Decision Tree
  D    //Input Database
Output:
  M    //Model Prediction
DTProc Algorithm:
          //Illustrates Prediction Technique using DT
  for each $t \in D$ do
    $n$ = root node of $T$;
    while $n$ not leaf node do
      Obtain answer to question on $n$ applied $t$;
      Identify arc from $t$ which contains correct answer;
      $n$ = node at end of this arc;
    Make prediction for $t$ based on labeling of $n$;
```

- DT relatively faster learning speed (compared to other classification methods)
- DT is ultimately convertible to a set of simple and easy to understand classification rules
- Due to intuitive graphical representation of DT, they are easy to assimilate by humans
- Accuracy of decision tree classifiers is comparable or superior to other models
- DT work well when there are logical connections between symptoms and solutions

III. DATA MINING

Data mining is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. Data mining, the extraction of the hidden predictive information from large databases, is a powerful new technology with great potential to analyze important information in the data warehouse. Data mining is the search for the relationships and global patterns that exist in large databases but are hidden among vast amount of data, such as the relationship between patient data and their medical diagnosis. It is the process of discovering meaningful, new correlation patterns and trends by sifting through large amount of data stored in repositories, using pattern recognition techniques as well as statistical and mathematical techniques. Data mining is a part of a process called KDD-knowledge discovery in databases which consists basically the steps such as data selection, data cleaning, pre-processing, and data transformation. Association rule techniques are used for data mining if the goal is to detect relationships or associations between specific values of categorical variables in large data sets. According [9]: "Data mining is the process of discovering meaningful patterns and relationships that lie hidden within very large databases".

IV. STUDENT GRADATION DATASET

We have considered the student dataset of MCA 3rd Semester of Dr.B.C.Roy Engineering College, Durgapur. Taking the consideration of the class test marks and the attendance of the students, this article proposed student gradation system for their internal marks as well as the overall student performance. The fig.1. represents the snapshot of the student attendance with their marks in different subject.

V. Experiment and Results

The membership function is generated from the fuzzy database corresponding to each record into values which lies in the range of 0 to 1 taking marks/(maximum of marks) and attendance/(maximum of attendance) as parameter. The membership value of each student for marks and attendance are shown in the fig.2 & the decision tree is shown in the fig.3

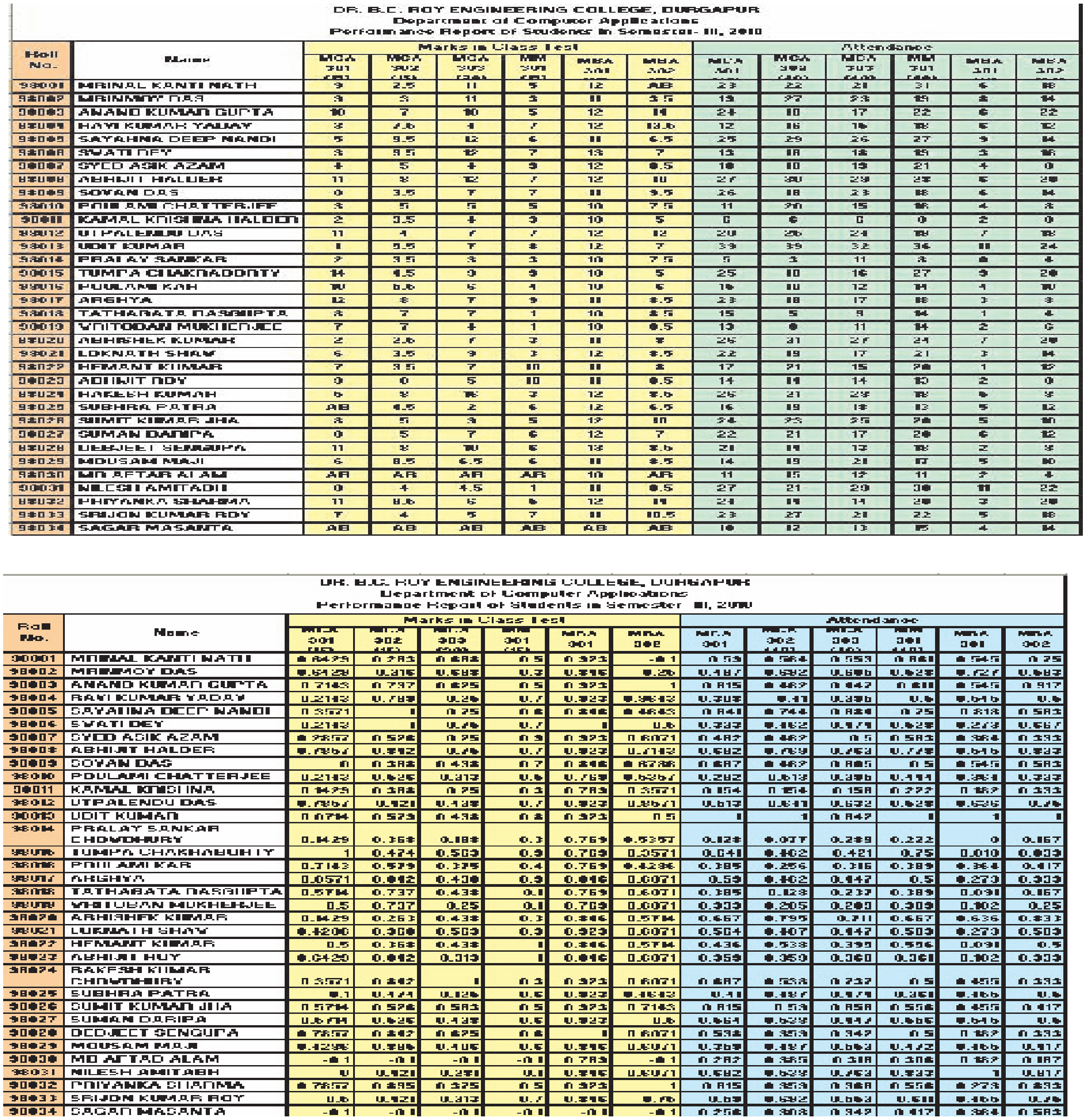
Membership Function for Marks & Attendance:-

$$\text{ì}_{marks}(x) = x / (\text{maximum of marks}) ,$$
$$0 = \text{ì}_{marks}(x) = 1 \qquad (1)$$

$$\text{ì}_{attendance}(y) = y / (\text{maximum of attendance}) ,$$
$$0 = \text{ì}_{attendance}(y) = 1 \qquad (2)$$

Generating Rules from the Decision Tree:

- If Marks is High & Attendance is High Then Class is Very Good.
- If Marks is High & Attendance is Medium Then Class



DR. B.C. ROY ENGINEERING COLLEGE, DURGAPUR
Department of Computer Applications
Performance Report of Students in Semester- III, 2010

Roll No.	Name	Marks in Class Test						Attendance					
		MCA 301	MCA 302	MCA 303	MM 301	MBA 301	MBA 302	MCA 301	MCA 302	MCA 303	MM 301	MBA 301	MBA 302
98001	MRINAL KANTI NATH	9	2.5	11	5	12	AB	23	22	21	31	6	18
98002	MRINMOY DAS	9	3	11	3	11	3.5	19	27	23	19	8	14
98003	ANAND KUMAR GUPTA	10	7	10	5	12	14	24	18	17	22	6	22
98004	RAVI KUMAR YADAV	3	7.5	4	7	12	13.5	12	16	15	18	6	12
98005	SAYAHNA DEEP NANDI	5	9.5	12	6	11	6.5	25	29	26	27	9	14
98006	SWATI DEY	3	9.5	12	7	13	7	13	18	18	19	3	16
98007	SYED ASIK AZAM	4	5	4	9	12	8.5	19	18	19	21	4	8
98008	ABHIJIT HALDER	11	8	12	7	12	10	27	30	29	28	6	20
98009	SOVAN DAS	0	3.5	7	7	11	9.5	26	18	23	18	6	14
98010	POULAMI CHATTERJEE	3	5	5	6	10	7.5	11	20	15	16	4	8
98011	KAMAL KRISHNA HALDER	2	3.5	4	3	10	5	6	6	6	8	2	8
98012	UTPALENDU DAS	11	4	7	7	12	12	20	26	24	19	7	18
98013	UDIT KUMAR	1	5.5	7	8	12	7	33	39	32	36	11	24
98014	PRALAY SANKAR	2	3.5	3	3	10	7.5	5	3	11	8	0	4
98015	TUMPA CHAKRABORTY	14	4.5	9	9	10	5	25	10	16	27	9	20
98016	POULAMI KAR	10	5.5	6	4	10	6	15	10	12	14	4	10
98017	ARGHYA	12	8	7	9	11	8.5	23	18	17	18	3	8
98018	TATHAGATA DASGUPTA	8	7	7	1	10	8.5	15	5	9	14	1	4
98019	VRITODAN MUKHERJEE	7	7	4	1	10	8.5	13	8	11	14	2	6
98020	ABHISHEK KUMAR	2	2.5	7	3	11	8	26	31	27	24	7	20
98021	LOKNATH SHAW	6	3.5	9	3	12	8.5	22	19	17	21	3	14
98022	HEMANT KUMAR	7	3.5	7	10	11	8	17	21	15	20	1	12
98023	ABHIJIT ROY	9	8	5	10	11	8.5	14	14	14	13	2	8
98024	RAKESH KUMAR	5	8	16	3	12	8.5	26	21	28	18	5	8
98025	SUBHRA PATRA	AB	4.5	2	6	12	6.5	16	19	18	13	5	12
98026	SUMIT KUMAR JHA	8	5	9	5	12	10	24	23	25	20	5	10
98027	SUMAN DARIPA	0	5	7	6	12	7	22	21	17	20	6	12
98028	DEBJEET SENGUPTA	11	8	10	6	13	8.5	21	14	13	18	2	8
98029	MOUSAM MAJI	6	8.5	6.5	6	11	8.5	14	19	21	17	5	10
98030	MD AFTAB ALAM	AB	AB	AB	AB	10	AB	11	15	12	11	2	4
98031	NILESH AMITABH	0	4	4.5	1	11	8.5	27	21	29	30	11	22
98032	PRIYANKA SHARMA	11	8.5	6	5	12	14	24	14	14	20	3	20
98033	SRIJON KUMAR ROY	7	4	5	7	11	10.5	23	27	21	22	5	18
98034	SAGAR MASANTA	AB	AB	AB	AB	AB	AB	10	12	13	15	4	14

DR. B.C. ROY ENGINEERING COLLEGE, DURGAPUR
Department of Computer Applications
Performance Report of Students in Semester III, 2010

Roll No.	Name	Marks in Class Test						Attendance					
		MCA 301	MCA 302	MCA 303	MM 301	MBA 301	MBA 302	MCA 301	MCA 302	MCA 303	MM 301	MBA 301	MBA 302
98001	MRINAL KANTI NATH	0.6429	0.263	0.688	0.5	0.923	-0.1	0.59	0.564	0.553	0.861	0.545	0.75
98002	MRINMOY DAS	0.6429	0.316	0.688	0.3	0.846	0.25	0.487	0.692	0.605	0.528	0.727	0.583
98003	ANAND KUMAR GUPTA	0.7143	0.737	0.625	0.5	0.923	1	0.615	0.462	0.447	0.611	0.545	0.917
98004	RAVI KUMAR YADAV	0.2143	0.789	0.25	0.7	0.923	0.9643	0.308	0.41	0.395	0.5	0.545	0.5
98005	SAYAHNA DEEP NANDI	0.3571	1	0.75	0.6	0.846	0.4643	0.641	0.744	0.684	0.75	0.818	0.583
98006	SWATI DEY	0.2143	1	0.75	0.7	1	0.5	0.333	0.462	0.474	0.528	0.273	0.667
98007	SYED ASIK AZAM	0.2857	0.526	0.25	0.9	0.923	0.6071	0.487	0.462	0.5	0.583	0.364	0.333
98008	ABHIJIT HALDER	0.7857	0.842	0.75	0.7	0.923	0.7143	0.692	0.769	0.763	0.778	0.545	0.833
98009	SOVAN DAS	0	0.368	0.438	0.7	0.846	0.6786	0.667	0.462	0.605	0.5	0.545	0.583
98010	POULAMI CHATTERJEE	0.2143	0.526	0.313	0.6	0.769	0.5357	0.282	0.513	0.395	0.444	0.364	0.333
98011	KAMAL KRISHNA	0.1429	0.368	0.25	0.3	0.769	0.3571	0.154	0.154	0.158	0.222	0.182	0.333
98012	UTPALENDU DAS	0.7857	0.421	0.438	0.7	0.923	0.8571	0.615	0.641	0.632	0.528	0.636	0.75
98013	UDIT KUMAR	0.0714	0.579	0.438	0.8	0.923	0.5	1	1	0.842	1	1	1
98014	PRALAY SANKAR CHOWDHURY	0.1429	0.368	0.188	0.3	0.769	0.5357	0.128	0.077	0.289	0.222	0	0.167
98015	TUMPA CHAKRABORTY	1	0.474	0.563	0.9	0.769	0.3571	0.641	0.102	0.421	0.75	0.818	0.833
98016	POULAMI KAR	0.7143	0.579	0.375	0.4	0.769	0.4286	0.385	0.256	0.316	0.389	0.364	0.417
98017	ARGHYA	0.8571	0.842	0.438	0.9	0.846	0.6071	0.59	0.462	0.447	0.5	0.273	0.333
98018	TATHAGATA DASGUPTA	0.5714	0.737	0.438	0.1	0.769	0.6071	0.385	0.128	0.237	0.389	0.091	0.167
98019	VRITODAN MUKHERJEE	0.5	0.737	0.25	0.1	0.769	0.6071	0.333	0.205	0.289	0.389	0.182	0.25
98020	ABHISHEK KUMAR	0.1429	0.263	0.438	0.3	0.846	0.5714	0.667	0.795	0.711	0.667	0.636	0.833
98021	LOKNATH SHAW	0.4286	0.368	0.563	0.3	0.923	0.6071	0.564	0.487	0.447	0.583	0.273	0.583
98022	HEMANT KUMAR	0.5	0.368	0.438	1	0.846	0.5714	0.436	0.538	0.395	0.556	0.091	0.5
98023	ABHIJIT ROY	0.6429	0.842	0.313	1	0.846	0.6071	0.359	0.359	0.368	0.361	0.182	0.333
98024	RAKESH KUMAR CHOWDHURY	0.3571	0.842	1	0.3	0.923	0.6071	0.667	0.538	0.737	0.5	0.455	0.333
98025	SUBHRA PATRA	-0.1	0.474	0.125	0.6	0.923	0.4643	0.41	0.487	0.474	0.361	0.455	0.5
98026	SUMIT KUMAR JHA	0.5714	0.526	0.563	0.5	0.923	0.7143	0.615	0.59	0.658	0.556	0.455	0.417
98027	SUMAN DARIPA	0.0714	0.526	0.438	0.6	0.923	0.5	0.564	0.538	0.447	0.556	0.545	0.5
98028	DEBJEET SENGUPTA	0.7857	0.842	0.625	0.6	1	0.6071	0.538	0.359	0.342	0.5	0.182	0.333
98029	MOUSAM MAJI	0.4286	0.895	0.406	0.6	0.846	0.6071	0.359	0.487	0.553	0.472	0.455	0.417
98030	MD AFTAB ALAM	-0.1	-0.1	-0.1	-0.1	0.769	-0.1	0.282	0.385	0.316	0.306	0.182	0.167
98031	NILESH AMITABH	0	0.421	0.281	0.1	0.846	0.6071	0.692	0.538	0.763	0.833	1	0.917
98032	PRIYANKA SHARMA	0.7857	0.895	0.375	0.5	0.923	1	0.615	0.359	0.368	0.556	0.273	0.833
98033	SRIJON KUMAR ROY	0.5	0.421	0.313	0.7	0.846	0.75	0.59	0.692	0.553	0.611	0.455	0.75
98034	SAGAR MASANTA	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1	0.256	0.308	0.342	0.417	0.364	0.583

Figure 2: Membership Value of Students for marks and attendance

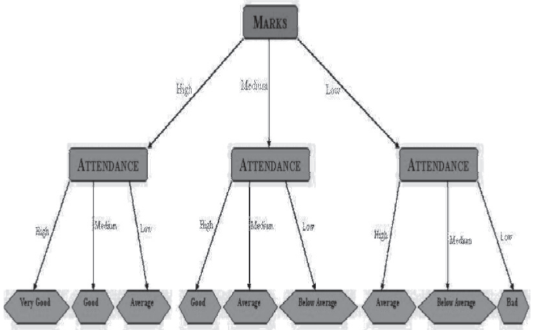
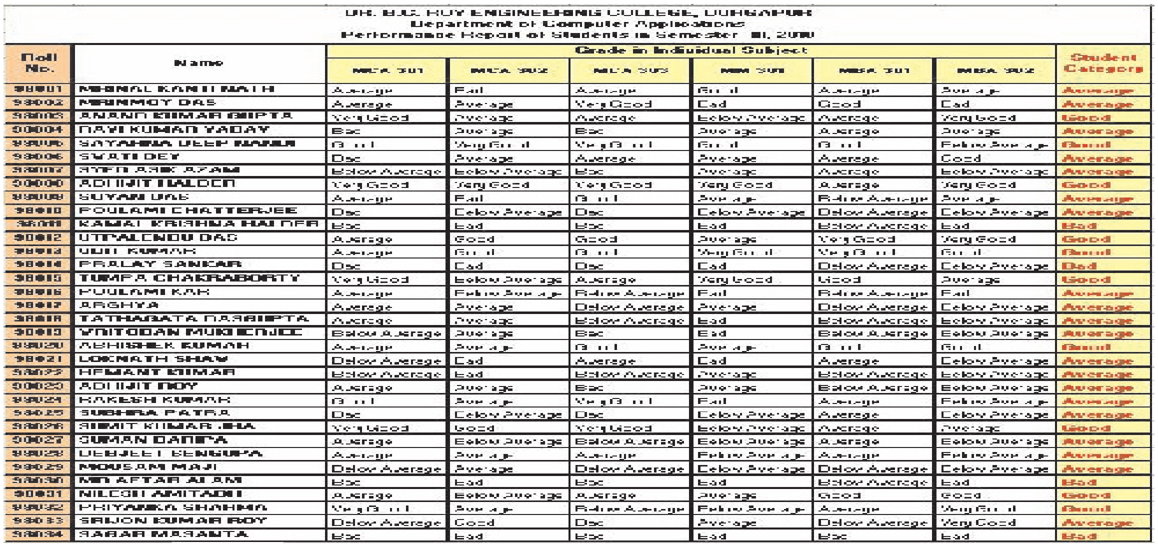


Figure 3: Decision Tree

is Good.

- If Marks is High & Attendance is Low Then Class is Average.
- If Marks is Medium & Attendance is High Then Class is Good.
- If Marks is Medium & Attendance is Medium Then Class is Average.
- If Marks is Medium & Attendance is Low Then Class is Below Average.
- If Marks is Low & Attendance is High Then Class is Average.
- If Marks is Low & Attendance is Medium Then Class is Below Average
- If Marks is Low & Attendance is Low Then Class is Bad.

From the Decision Tree and the Given Inference rule the Gradation of students are shown in the Fig.4.



DR. B.C. ROY ENGINEERING COLLEGE, DURGAPUR
Department of Computer Applications
Performance Report of Students in Semester III, 2000

Roll No.	Name	Grade in Individual Subject						Student Category
		MCA 301	MCA 302	MCA 303	MM 301	MBA 301	MBA 302	
98001	MRINAL KANTI NATH	Average	Fail	Average	Good	Average	Average	Average
98002	MRINMOY DAS	Average	Average	Very Good	Bad	Good	Bad	Average
98003	ANAND KUMAR GUPTA	Very Good	Average	Average	Below Average	Average	Very Good	Good
98004	RAVI KUMAR YADAV	Exc	Average	Exc	Average	Average	Average	Average
98005	SATYAHRA DEEP NANDA	Good	Very Good	Very Good	Good	Good	Below Average	Good
98006	SWATI DEY	Exc	Average	Average	Average	Average	Good	Average
98007	SYED ASIK AZAM	Below Average	Below Average	Exc	Average	Average	Below Average	Average
98008	ABHIJIT HALDER	Very Good	Very Good	Very Good	Very Good	Average	Very Good	Good
98009	SUYAN DAS	Average	Fail	Good	Average	Below Average	Average	Average
98010	POULAMI CHATTERJEE	Exc	Below Average	Exc	Below Average	Below Average	Below Average	Average
98011	KAMAL KRISHNA HALDER	Exc	Bad	Exc	Bad	Below Average	Bad	Bad
98012	UTPALENDU DAS	Average	Good	Good	Average	Very Good	Very Good	Good
98013	UDIT KUMAR	Average	Good	Good	Very Good	Very Good	Good	Good
98014	PRALAY SANKAR	Exc	Bad	Exc	Bad	Below Average	Below Average	Bad
98015	TUMPA CHAKRABORTY	Very Good	Below Average	Average	Very Good	Good	Average	Good
98016	POULAMI KAR	Average	Below Average	Below Average	Fail	Below Average	Fail	Average
98017	ARGHYA	Average	Average	Below Average	Average	Below Average	Below Average	Average
98018	TATHAGATA DASGUPTA	Average	Average	Below Average	Bad	Below Average	Below Average	Average
98019	VINITODAN MUKHERJEE	Below Average	Average	Exc	Bad	Below Average	Below Average	Average
98020	ABHISHEK KUMAR	Average	Average	Good	Average	Good	Good	Good
98021	LOKNATH SHAW	Below Average	Bad	Average	Bad	Average	Below Average	Average
98022	HEMANT KUMAR	Below Average	Bad	Below Average	Average	Below Average	Below Average	Average
98023	ABHIJIT ROY	Average	Average	Exc	Average	Below Average	Below Average	Average
98024	RAKESH KUMAR	Good	Average	Very Good	Fail	Average	Below Average	Average
98025	SUBHRA PATRA	Exc	Below Average	Exc	Below Average	Average	Below Average	Average
98026	SUMIT KUMAR JHA	Very Good	Good	Very Good	Below Average	Average	Average	Good
98027	SUMAN DANDA	Average	Below Average	Below Average	Below Average	Average	Below Average	Average
98028	DEBJEET SENGUPTA	Average	Average	Average	Below Average	Average	Below Average	Average
98029	MOUSAM MAJI	Below Average	Average	Below Average	Below Average	Below Average	Below Average	Average
98030	MD AFTAR ALAM	Exc	Bad	Exc	Bad	Below Average	Bad	Bad
98031	NILESH AMITABH	Average	Below Average	Average	Average	Good	Good	Good
98032	PRIYANKA SHARMA	Very Good	Average	Below Average	Below Average	Average	Very Good	Good
98033	SRIJON KUMAR ROY	Below Average	Good	Exc	Average	Below Average	Very Good	Average
98034	SAGAR MASANTA	Exc	Bad	Exc	Bad	Exc	Bad	Bad

Figure 4: Gradation of Students in each Subject and Overall Student Category

VI.CONCLUSION

In this article we define the membership function, calculate the decision tree and the fuzzy inference rules in the MATLAB environment and this paper gives some idea of overall performance of the students and from that we conclude the three student category as Good, Average and Poor from the student database. The development of the mathematical framework and the implementation of the prototype application present a feasibility study for fuzzy classification in OLAP. The main approach is to turn the abstract idea into a specific mathematical framework and a corresponding prototype computer program. The advantage of this type of classification is a more accurate allocation of tuples to classes in a predefined manner.

VII. REFERENCES

[1] J. R. Quinlan. Induction of decision trees. Machine Learning, 1:81–106, 1986.

[2] Tan, P., Steinbach, M. and Kumar, V. (2006). Introduction to Data.

[3] Garofalakis, M., Hyun, D., Rastogi, R. and Shim, K. (2000). Efficient algorithms for constructing decision trees with constraints. Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 335 – 339.

[4] Gehrke, J., Ramakrishnan, R., Ganti, V. (1998). RainForest - a Framework for Fast Decision Tree Construction of Large Datasets.Proceedings of the 24th VLDB conference, New York, USA. pp.416-427.

[4] Gehrke, J., Ramakrishnan, R., Ganti, V. (1998). RainForest - a Framework for Fast Decision Tree Construction of Large Datasets.Proceedings of the 24th VLDB conference, New York, USA. pp.416-427.

[5] Lippmann, R. (1987). An Introduction to computing with neural nets. IEEE ASSP Magazine, vol. (22).

[6] Khoshgoftaar, T.M and Allen, E.B. (1999). Logistic regression modeling of software quality. International Journal of Reliability, Quality and Safety Engineering, vol. 6(4, pp. 303-317.

[7] Quinlan, J. R. (1993). C45: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA.

[8] Utgoff, P and Brodley, C. (1990). An Incremental Method for Finding Multivariate Splits for Decision Trees, Machine Learning: Proceedings of the Seventh International Conference pp.58.

[9] Claude Seidman. "Data Mining with Microsoft SQL Server 2000 Technical Reference".

Fuzzy B-algebras with Bipolar-valued membership function

Arsham Borumand Saeid[#1], Marjan Kuchaki Rafsanajani[*2]

[#] *Department of Math,Shahid Bahonar University of Kerman, Kerman, Irany*
[1]arsham@uk.ac.ir

[*] *Department of Computer Science,Shahid Bahonar University of Kerman, Kerman, Iran*
[3]kuchaki@uk.ac.ir

Abstract-In this note, by using the concept of Bipolar-valued fuzzy set, the notion of bipolar-valued fuzzy -algebra is introduced. Moreover, the notions of (strong) negative s-cut (strong) positive t-cut are introduced and the relationship between these notions and crisp subalgebras are studied

Keywords- Bipolar-valued fuzzy sets, Bipolar-valued fuzzy -algebra, (strong) negative s-cut, (strong) positive t-cut.

I. INTRODUCTION

T Y. Imai and K. Ise'ki introduced two classes of abstract algebras: BCK-algebras and BCI-algebras [4,5]. It is known that the class of BCK-algebras is a proper subclass of the class of BCI-algebras. In [2, 3] Q. P. Hu and X. Li introduced a wide class of abstract algebras: BCH-algebras. They have shown that the class of BCI-algebras is a proper subclass of the class of BCH-algebras.

J. Neggers and H. S. Kim [10] introduced the notion of d–algebras which is another generalization of BCK-algebras, and also they introduced the notion of B-algebras [11, 12]. Moreover, Y. B. Jun, E. H. Roh and H. S. Kim [8] introduced a new notion, called a BH-algebra, which is a generalization of BCH/BCI/BCK-algebras. Walendziak obtained the another equivalent axioms for B-algebra [14]. H. S. Kim, Y. H. Kim and J. Neggers [7] introduced the notion a (pre-) Coxeter algebra and showed that a Coxeter algebra is equivalent to an abelian group all of whose elements have order 2, i.e., a Boolean group.

In 1965, Zadeh [13] introduced the notion of a fuzzy subset of a set; fuzzy sets are a kind of useful mathematical structure to represent a collection of objects whose boundary is vague. Since then it has become a vigorous area of research in different domains, There have been a number of generalizations of this fundamental concept such as intuitionistic fuzzy sets, interval-valued fuzzy sets, vague sets, soft sets etc [2].

Lee [10] introduced the notion of bipolar-valued fuzzy sets. Bipolar-valued fuzzy sets are an extension of fuzzy sets whose membership degree range is enlarged from the interval [0, 1] to [-1, 1]

In a bipolar-valued fuzzy set, the membership degree 0 means that elements are irrelevant to the corresponding property, the membership degree (0,1] indicates that elements somewhat satisfy the property, and the membership degree [-1,0) indicates that elements somewhat satisfy the implicit counter-property. Bipolar-valued fuzzy sets and intuitionistic fuzzy sets look similar each other. However, they are different each other (see [10,11]).

Now, in this note we use the notion of Bipolar-valued fuzzy set to establish the notion of bipolar-valued fuzzy *BM* -algebras; then we obtain some-related which have been mentioned in the abstract.

PRELIMINARIES

In this section, we present now some preliminaries on the theory of bipolar-valued fuzzy set. In his pioneer work [13], Zadeh proposed the theory of fuzzy sets. Since then it has been applied in wide varieties of fields like Computer Science, Management Science, Medical Sciences, Engineering problems etc. to list a few only.

Definition 2.1. [10] Let G be a nonempty set. A *bipolar-valued fuzzy set B in G* is an object having the form

$$B = \{(x, \mu^+(x), \nu^-(x)) | x \in G\}$$

Where $\mu^+ : G \to [0,1]$ and $\nu^- : G \to [-1,0]$ are mappings.

The positive membership degree $\mu^+(x$ denotes the satisfaction degree of an element x to the property corresponding to a bipolar-valued fuzzy set, $B = \{(x, \mu^+(x), \nu^-(x)) | x \in G$

and the negative membership degree $V^-(x)$ denotes the satisfaction degree of an element x to some implicit counter-property corresponding to a bipolar-valued fuzzy set .

$B = \{(x, \mu^+(x), \nu^-(x)) | x \in \mu^+(x) \neq ($ If and, $V^-(x)=$

$B = \{(x, \mu^+(x), \nu^-(x)) | x \in G\}$ $\mu^-(x) = 0$ 0,

it is the situation that x is regarded as having only positive satisfaction for $\nu^-(x) \neq 0$. If

and, $B = \{(x, \mu^+(x), \nu^-(x)) | x \in G\}$

it is the situation that x does not satisfy the property of $B = \{(x, \mu^+(x), \nu^-(x)) | x \in G\}$ but somewhat satisfies the counter property of $\mu^+(x) \neq 0$.

It is possible for an element x to be such that $\nu^-(x) \neq 0$ and $B = (\mu^+, \nu^-)$ when the membership function of the property overlaps that of its counter property over some portion of G . For the sake of simplicity, we shall use the symbol for the bipolar-valued fuzzy set $B = \{(x, \mu^+(x), \nu^-(x)) \mid x \in G\}$

Definition 2.2. **[6].** Let be a non-empty set with a binary operation "*" and a constant " 0". Then *(X, *, 0)* is called a *BM*-algebra if it satisfies the following conditions:
(i) $x * 0 = x$
(ii) $x * x = 0$
*(iii) $(x*y)* z = x* (z*(0*y))$*

for all $x, y, z \in X$

We can define a partial ordering $\leq$ by $x \leq y$ if and only if $x * y = 0$

A nonempty subset S of X is called a subalgebra of X if $x * y \in S$, for all $x, y \in S$

Definition 2.3. [12] Let μ be a fuzzy set in a *BM*-algebra Then μ is called a fuzzy B -algebra of X if

$$\mu(x * y) \geq \min\{\mu(x), \mu(y)\}$$

for all $x, y \in X$.

BIPOLAR-VALUED FUZZY SUBALGEBRAS OF B-ALGEBRAS

From now on $(X, *, 0)$ is a B-algebra, unless otherwise is stated.

Definition 3.1. A bipolar-valued fuzzy set $B = (\mu^+, \nu^-)$ is said to be a bipolar-valued fuzzy subalgebra a B-algebra X if it satisfies the following conditions:

(BF1) $\mu^+(x * y) \geq \min\{\mu^+(x), \mu^+(y$

(BF3) $\nu^-(x * y) \leq \max\{\nu^-(x), \nu^-(y$

for all $x, y \in X$.

Example 3.2. Consider a *BCI*-algebra $X = \{0, a, b, c\}$ with the following Cayley table:

*	0	*a*	*b*	*c*
0	0	*a*	*b*	*c*
a	*a*	0	*c*	*b*
b	*b*	*c*	0	*a*
c	*c*	*b*	*a*	0

Let $B = (\mu^+, \nu^-)$ be a bipolar-valued fuzzy set in X with the mappings μ^+ and ν^- defined by:

$$\mu^+(x) = \begin{cases} 0.7 & \text{if } x = 0 \\ 0.3 & \text{if } x \neq 0 \end{cases}$$

and

$$\nu^-(x) = \begin{cases} -0.4 & \text{if } x = 0 \\ -0.2 & \text{if } x \neq 0 \end{cases}$$

It is routine to verify that B is a bipolar-valued fuzzy subalgebra of X.

Lemma 3.3. If *B is a bipolar-valued fuzzy subalgebra of X,* then $\mu^+(0) \geq \mu^+(x)$ $\nu^-(0) \leq \nu^-(x)$, for all .

Proposition 3.4. Let *B be a bipolar-valued fuzzy subalgebra of X and let $n \in N$. Then*

$$(i) \quad \mu^+(\prod^n x * x) \geq \mu^+(x)$$

$$\textit{and } \nu^-(\prod^n x * x) \leq \nu^-(x),$$

for any odd number n,

Theorem 3.5. *Let B be a bipolar-valued fuzzy subalgebra of X. If there exists a sequence $\{x_n\}$ in X , such that*

$$\lim_{n \to \infty} \mu^+(x_n) = 1 \quad \textit{and} \quad \lim_{n \to \infty} \nu^-(x_n) = -1$$

Then $\mu^+(0) = 1 \textit{ and } \nu^-(0) = -1$

Theorem 3.6. *The family of bipolar-valued fuzzy subalgebras of X forms a complete distributive lattice under the ordering of bipolar-valued fuzzy set inclusion $\subset$.*

A fuzzy set μ of X is called anti fuzzy subalgebra of X, if $\mu(x * y) \leq \max\{\mu(x), \mu(y)\}$ for all $x, y \in X$.

Proposition 3.7. *A bipolar-valued fuzzy set B of X is a bipolar-valued fuzzy subalgebra of X if and only if μ^+ is a fuzzy subalgebras and ν^- is an anti fuzzy subalgebras of X.*

Definition 3.8. *Let $B = (\mu^+, \nu^-)$ be a bipolar-valued fuzzy set and* $(s, t) \in [-1, 0] \times [0, 1$

1) The sets $B_t^+ = \{x \in X \mid \mu^+(x)$ and $B_s^- = \{x \in G \mid \nu^-(x) \leq$ which are called positive ${}^{>}B_t^+ = \{x \in X \mid \mu^+(x) >$ *t-* cut of $B = (\mu^+, \nu^-)$ and negative *s-cut* of $B = (\mu^+, \nu)$, respectively,

2) The sets ${}^{<}B_s^- = \{x \in G \mid \nu^-(x) <$ and,

$X_B^{(t,s)} = \{x \in X \mid \mu^+(x) \geq t,\ \nu^-(x)$ which are called strong positive t- cut of *B = (μ⁺, ν⁻)* and the strong negative s-cut of *B = (μ⁺, ν⁻)*, respectively,

3) The set $^S X_B^{(t,s)} = \{x \in X \mid \mu^+(x) > t,\ \nu^-($ is called an *(s, t)* -level subset of *B*,

4) The set $(s,t) \in \operatorname{Im}(\mu^+) \times \operatorname{Im}(\nu^-)$ is called a strong *(s, t)*-level subset of *B*,

5) The set of all $(s,t) \in \operatorname{Im}(\nu^-) \times \operatorname{Im}(\mu^+)$ is called the image of *B = (μ⁺, ν⁻)* .

Theorem 3.9. *Let B be a bipolar-valued fuzzy subset of X such that the least upper bound to of* $\operatorname{Im}(\mu^+)$ *and the greatest lower bound* s_0 *of* $\operatorname{Im}(\nu^-)$*exist. Then the following conditions are equivalent:*

(i) B is a bipolar-valued fuzzy subalgebra of X,

(ii) For all , the nonempty strong level subset $X_B^{(t,s)}$ *of B is a (crisp) subalgeba of X .*

(iii) For all $(s,t) \in \operatorname{Im}(\nu^-) \times \operatorname{Im}(\mu^+) \setminus (s_0, t_0)$ *the nonempty strong level subset* $^S X_B^{(t,s)}$ *of B is a (crisp) subalgeba of X.*

(iv) For all $(s,t) \in [-1,0] \times [0,1]$*, the nonempty strong level subset* $^S X_B^{(t,s)}$ *of is a (crisp) subalgeba of X.*

(v) For all $(s,t) \in [-1,0] \times [0,1]$*, the nonemptyg level subset* $X_B^{(t,s)}$ *of B is a (crisp) subalgeba of X.*

Theorem 3.10. *Each subalgebra of X is a level subalgebra of a bipolar-valued fuzzy subalgebra of X.*

Theorem 3.11. *Let S be a subset of X and B be a bipolar-valued subset of X which is given in the proof of Theorem 3. 10. If B is a bipolar-valued fuzzy subalgebra of X then S is a subalgebra of X.*

$$X_B = \{x \in X \mid \mu^+(x) = \mu^+(0), \nu^-(0) = \nu^-(x)\}$$

fuzzy subalgebra of X, then the set is a subalgebra of X .

$$\mu_N^+(x) = \begin{cases} \alpha & \text{if } \alpha \in M \\ \beta & \text{otherwise} \end{cases}$$

$$\nu_N^-(x) = \begin{cases} \gamma & \text{if } x \in M \\ \delta & \text{otherwise} \end{cases}$$

Theorem 3.14. *Let M be a su*[illegible]*s a bipolar-valued fuzzy set of X defined by:*

and $\alpha, \beta \in [0,1]$ $\gamma, \delta \in [-1,0]$ $\alpha \geq \beta$ $\gamma \leq \delta$

For all and with

$$X_N = M$$

$$X_N := \{x \in X \mid \mu_N^+(x) = \mu_N^+(0), \nu_N^-(x) = \mu_N^+(0)\}$$

$$= \{x \in X \mid \mu_N^+(x) = \alpha, \nu_N^-(x) = \gamma\} = M$$

and . Then N is a bipolar-valued fuzzy subalgebra if and only if M is a subalgebra of X. Moreover, in this case .

CONCLUSION

Bipolar-valued fuzzy set is a generalization of fuzzy sets. In the present paper, we have introduced the concept of bipolar-valued fuzzy subalgebras of B-algebras and investigated some of their useful properties. In our opinion, these definitions and main results can be similarly extended to some other algebraic systems such as groups, semigroups, rings, nearrings, semirings (hemirings), lattices and Lie algebras. It is our hops that this work would other foundations for further study of the theory of -algebras. Our obtained results can be perhaps applied in engineering, soft computing or even in medical diagnosis.

In our future study of fuzzy structure of B-algebras may be the following topics should be considered:

REFERENCES

(1) W. L. Gau and D. J. Buehrer, 1993. Vague sets, IEEE Transactions on Systems, Man and Cybernetics 23: 610-614.

(2) Q. P. Hu and X. Li, On BCH-algebras, Math. Seminar Notes 11 (1983),313-320.

(3) Q. P. Hu and X. Li, On proper BCH-algebras, Math. Japonica 30 (1985),659-661.

(4) K. Iseki and S. Tanaka, An introduction to theory of BCK-algebras, Math.Japonica 23 (1978), 1-26.

(5) K. Iseki, On BCI-algebras, Math. Seminar Notes 8 (1980), 125-130.

(6) Y. B. Jun, E. H. Roh and H. S. Kim, On BH-algebras, Sci. Math. JaponicaOnline 1 (1998), 347-354.

(7) C. B. Kim and H. S. Kim, On BM-algebras, Sci. Math. Japo. Onlinee-2006 (2006), 215-221.

(8) H. S. Kim, Y. H. Kim and J. Neggers, Coxeters and pre-Coxeter algebrasin Smarandache setting, Honam Math. J. 26(4) (2004) 471-481.

(9) K. M. Lee, 2000. Bipolar-valued fuzzy sets and their operations, Proc. Int. Conf. on Intelligent Technologies, Bangkok, Thailand. 307-312.

(10) K. M. Lee, 2004. Comparison of interval-valued fuzzy sets, intuitionistic fuzzy sets, and bipolar-valued fuzzy sets, J. Fuzzy Logic Intelligent Systems 14, No. 2: 125-129.

(11) J. Meng and Y. B. Jun, BCK-algebras, Kyung Moon Sa, Co., Seoul (1994).

(12) J. Neggers and H. S. Kim, On d-algebras, Math. Slovaca 49 (1999), 19-26.

(13) J. Neggers and H. S. Kim, On B-algebras, Mate. Vesnik 54 (2002), 21-29.

(14) J. Neggers and H. S. Kim, A fundamental theorem of B-homomorphism for B-algebras, Int. Math. J. 2 (2002), 215-219.

(15) A. Rosenfeld, Fuzzy Groups, J. Math. Anal. Appl., 35 (1971), 512-517.

(16) A. Walendziak, Some axiomatizations of B-algebras, Math. Slovaca 56,No. 3 (2006), 301-306.

(17) A. Walendziak, A note on normal subalgebras in B-algebras, Sci. Math.Japo. Online e-2005 (2005), 49-53.

(18) L. A. Zadeh, Fuzzy Sets, Inform. Control, 8 (1965),

Study and Analysis of Temporal Data Mining Using Cluster Graphs To Find Useful Patterns for Diabetes Management

Snehlata Mandal[#1], Vivek Dubey[*2]

[#]*ME-CTA, CS Department,*
Shri Shankaracharya college of Engg. and Technology, CSVTU, Bhilai, Chattisgarh, India
[1]mandal.sneha@gmail.com

[*]*CS Department,*
Shri Shankaracharya college of Engg. and Technology, CSVTU, Bhilai, Chattisgarh, India
[1]vivekdubey22@gmail.com

Abstract— **Organizations and firms are capturing increasingly more data about their customers, suppliers, competitors, and business environment. Most of this data is temporal in nature. Data mining and business intelligence techniques are often used to discover patterns in such data; however, mining temporal relationships typically is a complex task. I propose a data analysis and visualization technique for representing trends in temporal data using a clustering based approach. I am using a system that implements the temporal cluster graph construct, which maps temporal data to a two-dimensional directed graph that identifies trends in dominant data types over time.**

In this paper, I present temporal clustering-based technique, to visualize temporal data to identifying trends for controlling diabetes mellitus. Given the complexity of chronic disease prevention, diabetes risk prevention and assessment may be critical area for improving clinical decision support. Information visualization utilizes high processing capabilities of the human visual system to reveal patterns in data that are not so clear in non-visual data analysis.

Keywords— **Data Mining, Clustering, temporal data mining, temporal database, blood glucose, diabetes mellitus**

I. INTRODUCTION

This Diabetes mellitus often simply referred to as diabetes—is a group of metabolic diseases in which a person has high blood sugar, either because the body does not produce enough insulin or because cells do not respond to the insulin that is produced.

This high blood sugar produces the classical symptoms of 1. polyurea (frequent urination)2. polydypsia(increased thirst) and 3. polyphagia(increased hunger).

There are three main types of diabetes:

Type 1 diabetes: results from the body's failure to produce insulin, and presently requires the person to inject insulin. (Also referred to as *insulin-dependent* diabetes mellitus, *IDDM* for short, and *juvenile* diabetes.)

<u>Type 2 diabetes</u>: results from insulin resistance, a condition in which cells fail to use insulin properly, sometimes combined with an absolute insulin deficiency.

<u>Gestational diabetes</u>: is when pregnant women, who have never had diabetes before, have a high blood glucose level during pregnancy. It may precede development of type 2 DM.

II.NEED

Diabetes mellitus is a costly chronic disease. Much of the burden of preventing, diagnosing and managing diabetes falls on primary care physician who often have insufficient resources to effectively prevent and manage disease. At patient level monitoring and responding to changes in risk are important due to rise of pay-for-performance initiatives. Given the complexity of chronic disease prevention, diabetes risk prevention and assessment may be critical area for improving clinical decision support. Information visualization utilizes high bandwidth processing capabilities of the human visual system to reveal patterns in data that are not so clear in non-visual data analysis.

III. PROBLEM DESCRIPTION

A person suffering from diabetes mellitus cannot be cured that is diabetes mellitus can be controlled but there are no permanent solutions to cure the disease. Thus a person suffering from diabetes will possess a history of data related to his blood glucose level. The earlier age the larger the history of database. Thus to control diabetes an individual must always check on its glucose level, to see whether he has controlled glucose level. The blood glucose level is the measure of the severity of diabetes in an individual. This will help him to take proper medicines, diet and exercise so that he has normal glucose level. If required he will have to take insulin doses. The blood glucose value is not constant that is it changes with time. For a same person in a day we can have different

BG values. Thus BG values are temporal in nature. Thus the changing value of BG level can be analysed. The analysis can be done based on existing database on daily basis, weekdays, weekends, both, two weeks, one month, all dates.

IV. IMPLEMENTATION

A. Overview

The Implementation step consists of two main phases: 1) Offline pre-processing of the data and 2) Online interactive analysis and graph rendering. In the pre-processing phase, the data set is partitioned based on time periods, and each partition is clustered using one of many traditional clustering techniques such as a hierarchical approach. The results of the clustering for each partition are used to generate two data structures: the node list and the edge list. Creating these lists in the pre-processing phase allows for more effective (real-time) Visualization updates of the output graphs. Based on these data structures, graph entities (nodes and edges) are generated and rendered as a temporal cluster graph in the system output window.

B. Process

Data Clustering - Cluster analysis or clustering is the assignment of a set of observations into subsets (called *clusters*) so that observations in the same cluster are similar in some sense. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics.

C. Hierarchical Clustering

Hierarchical clustering creates a hierarchy of clusters which may be represented in a tree structure called a dendrogram. The root of the tree consists of a single cluster containing all observations, and the leaves correspond to individual observations. Algorithms for hierarchical clustering are generally either agglomerative, in which one starts at the leaves and successively merges clusters together; or divisive, in which one starts at the root and recursively splits the clusters. Any valid metric may be used as a measure of similarity between pairs of observations. The choice of which clusters to merge or split is determined by a linkage criterion, which is a function of the pair wise distances between observations.

D. Algorithm for Hierarchical Clustering

Given a set of N items to be clustered, and an NxN distance (or similarity) matrix, the basic process hierarchical clustering is this:

1) Start by assigning each item to its own cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters equal the distances (similarities) between the items they contain.

2) Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one less cluster.

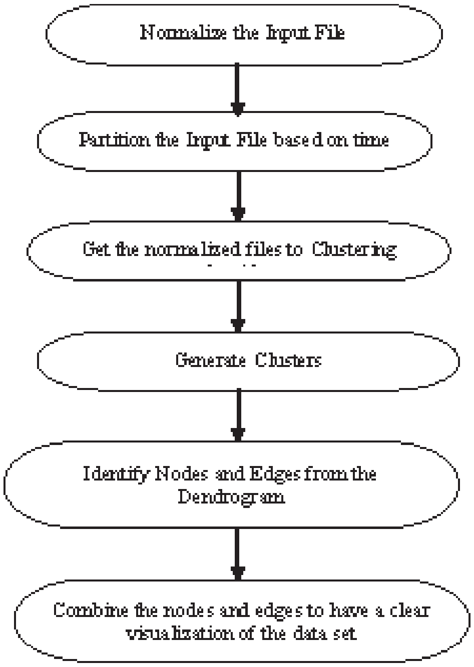

Normalize the Input File
Partition the Input File based on time
Get the normalized files to Clustering
Generate Clusters
Identify Nodes and Edges from the Dendrogram
Combine the nodes and edges to have a clear visualization of the data set

Fig. 1 Implementation

3) Compute distances (similarities) between the new cluster and each of the old clusters.

4) Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

V. CONCLUSION

The paper introduces cluster analysis and hierarchical algorithm which is one of the clustering methods. Hierarchical clustering algorithm is implemented on diabetes management to find tends in patient history to find out useful patterns.

REFERENCES

[1] Gediminas Adomavicius and Jesse Bockstedt*, "C-TREND: Temporal cluster graph for identifying and visualizing trends in multi-attribute transactional data"*, IEEE Transaction on knowledge and data engineering, Vol.20, No.6. June 2008.

[2] J. Roddick and M. spiliopoulou, *"A survey of temporal knowledge discovery paradigms and methods",* IEEE trans knowledge and data engg., vol. 14, no. 4, pp. 750-767, July/Aug 2002.

[3] Margaret H. Dunham, *"Data-mining Introductory and Advanced Topic"*, vol. 6, pp. 245-275, 2009.

[4] Gajendra Sharma, "*Data-mining Data warehousing and OLAP",* vol. 2, pp. 337-349, 2010.

[5] Paul R. Cohen and Carole R. Beal, *"Temporal data mining for educational applications",* Int J Software Informatics, vol. 3, no. 1, pp.31-46, March 2009.

[6] J. Abello and J. Korn, *"MGV: A System of Visualizing Massive Multi-Digraphs,"* IEEE Trans. Visualization and Computer Graphics,vol. 8, no. 1, pp. 21-38, Jan.-Mar. 2001.

[7] Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[8] Akash Rajak and Kanak Saxena, *"Modeling Clinical database using time series based Temporal Mining,"* International Journal of computer Theory and Engineering, Vol. 2,No.2, April, 2010.

[9] C.M. Antunes and A.L. Oliveira, *"Temporal Data Mining: An Overview,"* Proc. ACM SIGKDD Workshop Data Mining, pp. 1-13, Aug. 2001.

[10] C. Apte, B. Liu, E. Pednault, and P. Smyth, *"Business Applications of Data Mining,"* Comm. ACM, vol. 45, no. 8, pp. 49-53, 2002.

[11] G.C. Battista, P. Eades, R. Tamassia, and I.G. Tollis, Graph Drawing. Prentice Hall, 1999.

[12] B. Becker, R. Kohavi, and D. Sommerfield, *"Visualizing the Simple Bayesian Classifier,"* Proc. ACM SIGKDD Workshop Issues on the Integration of Data Mining and Data Visualization, 1997.

[13] B. Bederson, "*Pad++: Advances in Multiscale Interfaces*," Proc. Conf. Human Factors in Computing Systems (CHI '94), p. 315, 1994.

[14] D.J. Berndt and J. Clifford, *"Finding Patterns in Time Series: A Dynamic Programming Approach,"* Advances in Knowledge Discoveryand Data Mining, pp. 229-248, 1995.

Feature Selection Using Polynomial Neural Network

Amit Saxena[#1], Dovendra Patre[#2], Abhishek Dubey[#3]
[#1,#2] *Dept. of CSIT*
[#3] *Dept. of IT*
[#1] *amitsaxena65@rediffmail.com*
[#2] *dovendra_patre@gmail.com*
[#3] *abhishek_dubey003@hotmail.com*
[#1,#2] *G.G.Vishwavidyalaya, Bilaspur (C.G.),India*
[#3] *Salalah College of Technology, Salalah, Oman*

Abstract—**From the previous studies of Polynomial Neural Network (PNN) we found that the PNN model becomes more complex and expensive as the number of layers increases. The number of layers depends on the dimensions of the datasets. In the conventional PNN approaches due to the expansion of the whole network at different levels the complexity is increased, therefore it needs more computation time in solving classification task. In this context we propose a novel approach for feature subset selection by the PNN using Genetic Algorithm (GA). A randomly selected subset of features of a dataset is passed to the PNN as input. The classification accuracy of PNN is taken as the fitness function of GA. In the proposed scheme, less number of features selected by the GA prevents PNN to grow at very early stages which reduces the computation cost as well as time. The proposed scheme is simulated on three benchmark datasets. It is observed that the number of PD's become very less and hence the complexity is decreased. The proposed scheme therefore takes much less time but still produces high classification accuracies.**

***Keywords*- Polynomial Neural Net, Genetic Algorithm, Feature Selection, Pattern Classification.**

I. INTRODUCTION

It is known that the amount of information is increasing day by day due to development of technologies such as metrological data, traffic or share market information, documents and news articles, medical data etc. All of this information is mined so that the relations among components of the underlying systems are better understood and their models can be built. Due to the storage of Microarray, Mass Spectrometry (MS) Technologies, Remote Sensing etc., this produces large quantities of data with higher dimension [6]. This high dimensional nature of the data demands the development of special data analysis procedures that are able to adequately handle such data. The dimensions of a dataset are usually characterized by its features. In a typical classification problem, there can be p patterns, n features and c classes. Each pattern belongs to a class. Feature selection is a process to identify a small subset of features m<=n, which can classify a pattern reasonably accurately. Feature selection techniques aim to discard the bad and irrelevant features from the available set of features. This reduction of features may improve the performance of classification, function approximation, and other pattern recognition systems in terms of speed, accuracy, and simplicity [3]. Another importance of feature selection is in the task of mining large databases which is also known as dimensionality reduction [1]. With an increase in dimensionality, the hyper-volume increases exponentially and thus large dimensionality of data demands a large number of training samples for an adequate representation of the input space. Dimensionality reduction can be done by selecting a small but important subset of features and generating (extracting) a lower dimensional data preserving the distinguishing characteristics of the original higher dimensional data [10]. Feature Selection leads to savings measurement cost and time because some of the features (redundant) get discarded. This concept can be utilized in pruning those networks which otherwise would have taken a large time to compute in presence of all features. In practice it is often found that additional features actually degrade the performance of a classifier designed using class-conditional density estimates when the training set is small with respect to the dimensionality [14]. The performance of the classifier, constructed from a fixed number of training instances, may degrade with the increase in dimensionality as was illustrated by Trunk [15]. When feature selection methods use class information, we call it supervised feature selection otherwise it is an unsupervised feature selection [9]. Feature selection is also important in other areas such as finding cluster structures in data or in other exploratory data analysis. Polynomial Neural Networks (PNN) have emerged recently as an extension of Artificial Neural Networks (ANN) [5]. Some of the limitations of ANN have been claimed to be countered in PNN. The PNN have also been used as classifiers like ANN. The classification time depends on the number of features and the size of PNN, i.e. its architecture [7]. In this paper we investigate a scheme to evaluate the performance of PNN classifier using reduced number of features.

II. POLYNOMIAL NEURAL NETWORKS (PNN)

PNN is a flexible neural architecture whose topology is not predetermined or fixed like a conventional ANN but grown through learning layer by layer. The design is based on Group Method of Data Handling (GMDH) which was invented by Prof. A. G. Ivankhnenko in the late 1960s [16]. As described in [11], the GMDH generates successive layers with complex links that are individual terms of a polynomial equation. The individual terms generated in the layers are partial descriptions (PDs) of data being the quadratic regression polynomials with two inputs. The first layer is created by computing regressions of the input variables and choosing the best ones for survival. The second layer is created by computing regressions of the values in the previous layer along with the input variables and retaining the best candidates. More layers are built until the network stops getting better based on termination criteria.

In a feed-forward neural network (FNN) [12], to achieve high classification accuracy, one has to provide in advance, a well defined structure of FNN, such as, the number of input nodes, hidden and output neurons, and assume a proper set of relevant features. To alleviate this drawback of ANN [13]; PNN can be used for classification purposes. Evolutionary approach based PNN generates populations or layers of neurons/simulated units/partial descriptions (PDs) and then trains and selects those neurons, which provide the best classification. Using this approach during learning, the PNN model generates the new population of neurons and the number of layers and the complexity of the network increases [8] until a predefined criterion is met. Such models can be comprehensively described by a set of short-term polynomials thereby developing a PNN classifier. Coefficients of PNN can be estimated by least square fitting. The network architecture grows depending on the number of input features, PNN model selected, number of layer required, and the number of PD's preserved in each layer. Fig. 1 shows a basic PNN model with all inputs. Fig. 2 describes how a PD is computed at a node of a PNN's layer with reduced features using proposed scheme.

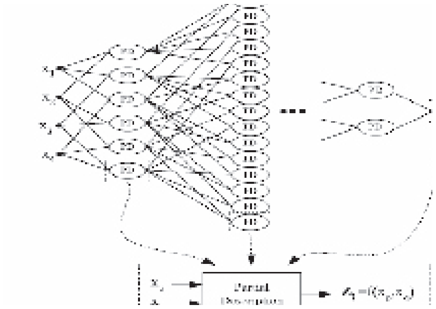


Figure-1 Basic PNN Model

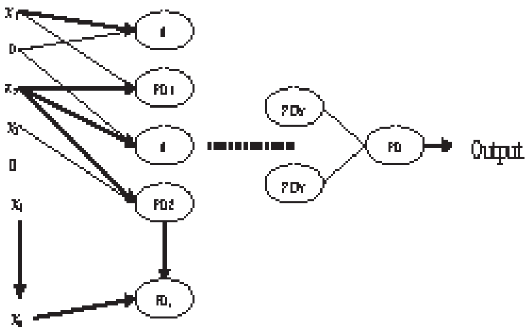


Figure-2 Computation of PD at a node of a PNN's layer with some inputs treated absent and indicated by 0 using proposed scheme.

III. GENETIC ALGORITHM (GA)

First pioneered by John Holland in the 1960s, GA have been widely studied with interest, experimented and applied in many fields in science and engineering worlds. GA is an evolutionary algorithm, which optimizes a fitness function to find the solution of a problem. Different evolutionary algorithms have been used for feature selection. In a typical GA, each chromosome represents a prospective solution of the problem. The problem is associated with a fitness function – higher fitness refers to a better solution. The set of chromosomes is called a population. The population goes through a repeated set of iterations (or generations) with crossover and mutation operations to find better solutions. At a certain fitness level or after a certain number of generations, the procedure is stopped and the chromosome giving the best solution is preserved as the best solution of the problem. A detailed description of GA can be found in [4].

IV. PROPOSED SCHEME

Proposed scheme is depicted in Figure-3. Feature set represents the set of all features in the dataset. A subset of this feature set is selected randomly which becomes a part of population (one chromosome) to be used in GA. The existence of a feature in the subset is represented by a 1 and its absence by a 0 in every chromosome of the population. This subset of features is given as input to PNN. The classification accuracy of PNN is calculated using training and testing patterns from the dataset. This accuracy serves as the fitness value of the GA. Similarly the fitness of other subsets of features in the initial population is also calculated. The simple one point crossover and mutation operations are applied on initial population to produce a modified population. The fitness' of the chromosomes of the modified population are compared with those of the initial population. The better chromosomes (Subsets) are retained in the next population. This completes one generation and the population with chromosomes of higher fitness values becomes the initial population for the next generation. This process continuous for a number of generations and at a satisfactory level the process is stopped. The feature subset with best classification accuracy of PNN is noted for comparison.

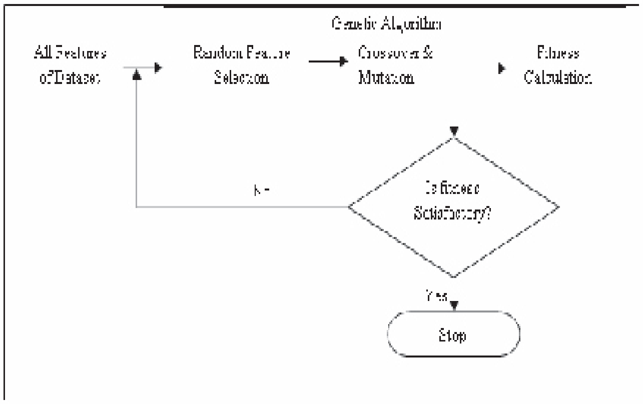


Figure-3 Proposed Technique

5. EXPERIMENTAL STUDIES

A. Summary of the datasets used for classification

The performance of different models is evaluated using the benchmark classification databases. Out of these the most frequently used databases in the area of neural networks are IRIS, PIMA and BUPA liver disorders. These datasets are taken from the UCI machine repository [2]. Table-I presents the summary of the main features of the datasets used for experimental studies.

B. Simulations and results

The performance of proposed scheme is evaluated using the benchmark databases. A summary of these databases is given in Table-I which is also available in the UCI machine repository [2]. Proposed scheme was simulated on a Pentium-III machine. For computing classification accuracy of PNN, cross validation was used. Each dataset was divided into two folds, one for training and other for testing. We have taken 50% patterns in fold 1 and remaining 50% in fold 2. The number of generations and sizes of populations used in the proposed scheme for different datasets are shown in Table-II. Table-III presents the Times of execution, classification accuracies for different datasets using Proposed Scheme.

TABLE I
DESCRIPTION OF THE DATA SET USED

Data Set	**Total Patterns**	**At-**	**Class-**	**Pattern in Class 1**	**Pattern in Class 2**	**Pattern in Class 3**
Iris	150	4	3	50	50	50
Pima	768	8	2	268	500	-
Bupa	345	6	2	145	200	-

TABLE II
PARAMETERS AND POPULATION SIZE USED IN GA
Probability of Crossover =0.5 and Probability of Mutation =

Dataset	**Population**	**Generation**
Iris	20	15
Pima	60	16
Bupa	25	25

0.3 for all data sets

TABLE III
THE TIMES OF EXECUTION AND RESPECTIVE CLASSIFICATION ACCURACIES OBTAINED THROUGH PROPOSED SCHEME.

Data Set	**Feature**	**Time (In Seconds)**	**Accuracy**
Iris	2	0.193	99.11
Pima	4	0.973	75.64
Bupa	4	0.560	77.34

6. RESULTS AND DISCUSSIONS

Table-III presents the performances (The Times of execution and respective Classification Accuracies) evaluated using proposed scheme. The classification accuracy and the execution cost have been taken as the performance indexes. By observing Table-III, it is noted that in proposed scheme, a very less number of features are capable to produce a higher classification accuracy e.g. in iris data set only 2 features are required to produce 99.11 % accuracy. Similar results are observed for Pima and Bupa dataset. In addition to classification accuracy; execution time is another parameter which is noticeable. In proposed scheme, time of execution for iris dataset is 0.193 sec.

7. CONCLUSIONS

In this paper, we have proposed a novel scheme of feature selection using PNN. A number of recent publications have used PNN for different diversified applications including classification. There have been research publications to reduce the size of a conventional PNN. It is known that PNN grows from the first layer on the basis of number of features (inputs) and partial derivatives produced due to these inputs in the subsequent layers. If number of features is reduced, then the growth of partial derivatives will also stop at immediate next layer and there will be no more partial derivatives onward due to this dead node. To select subsets of features in order to reduce irrelevant / derogatory features, we applied GA. The fitness of GA is measured by computing the classification accuracy obtained by PNN for a selected subset of features. The scheme is tested for three datasets and in each case, proposed scheme outperforms in terms of time of execution as well as classification accuracy. It is observed that the proposed scheme is taking much less time with high classification accuracy. It justifies our investigation. A further extension of proposed scheme is to test it on very large datasets like Microarrays, spatial datasets which will be scope for future research in this direction and can be compared with schemes proposed in the recent literatures. We are working to extend the work and compare with exiting scheme in the literature.

ACKNOWLEDGEMENT

This work is supported by Government of India under UGC granted project “Investigating a Novel Approach for Dimensionality Reduction Using Pattern Recognition Techniques”.

REFERENCES

[1] Amit, Saxena, Nikhil, R.,Pal, Megha, Vora, 2010. Evolutionary methods for unsupervised feature selection using Sammon’s stress function, Springer Journal on Fuzzy Information and Engineering. 2(3). 229-247.

[2] Blake L. and Merz C.J., 2001. "UCI Repository of machine learning databases, "http://www.ics.uci.edu/~mlearn/MLRepository.html.junio".

[3] Debrup, Chakraborty and Nikhil, R.,Pal,2008. Selecting Useful Groups of Features in a Connectionist Framework. IEEE Transactions On Neural Networks. 19(3)

[4] Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

[5] Ladislav Zjavka, 2010. Generalization Of Patterns By Identification With Polynomial Neural Network, Journal of ELECTRICAL ENGINEERING, VOL. 61, NO. 2, 2010, 120–124.

[6] Milos Hauskrecht, Richard Pelikan, Michal Valko, and James Lyons-Weiler, 2007. Feature Selection and Dimensionality Reduction in Genomics and Proteomics, Springer Verlag, pages 149–172.

[7] Misra, B.B., Dehuri, S., Dash, P.K., Panda G., 2008. A reduced and comprehensible polynomial neural network for classification, Pattern Recognition Letters of Elsevier Journal, 29 (2008) 1705–1712.

[8] Misra, B.B., Satapathy, S.C., Biswal, B.N., Dash, P.K., and Panda, G., 2006. Pattern classification using polynomial neural networks, IEEE Int. Conf. on Cybernetics & Intelligent Systems (CIS), 2006.

[9] Mitra P., Murthy, C.A. and Pal, S.K., 2002. Unsupervised feature selection using feature similarity. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(3): 301-312.

[10] Nikhil, R., Pal, 2002. Fuzzy logic approaches to structure preserving dimensionality reduction. IEEE Transactions on Fuzzy Systems. 10(3). 277-286.

[11] Oh, S.K., Pedrycz, W. and Park, B.J., 2003, "Polynomial neural networks architecture: analysis and design," Computers and Electrical Engineering. 29. 703-725.

[12] Pao, Y.H., 1989, "Adaptive pattern recognition neural networks", Addison Wesley, MA.

[13] Quinlan, J.R., 1987. Generating production rules from decision trees. In: Proc. Internat. Joint Conf. on Artificial Intelligence, San Francisco, CA: MorganKaufmann, pp. 304–307.

[14] Raudys, S.J., Jain A.K., 1991. Small sample size effects in statistical pattern recognition: Recommendations for Practioners. IEEE Transactions on Pattern Analysis and Machine Intelligence. 13(2).252-264.

[15] Trunk, G.V., 1979. A problem of dimensionality: A simple example. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1(3).306-307.

[16] A.G.Ivakhnenko, 1971. Polynomial theory of complex systems, IEEE Trans. Syst., Man Cybern-I. 364–378.

Image Segmentation with Statistical Feature and Fuzzy Inference System

Mangesh D. Ramteke[1], Prof. Mohd. Atique[2]
Electronics and Telecommunication Engineering Department,
Sant Gadge Baba University, Amravati(India)
mangeshcool111@gmail.com[1], atique_shaikh@rediffmail.com[2]

Abstract

In this paper, we present a new algorithm that can segment a fuzzy data. This method is based on fuzzy logic and clustering technique. It is proposed that images are described by fuzzy IF ... THEN rules instead of pixel values. This new approach may benefit from recognized fuzzy systems superior incorporation of measurement uncertainties, greater resources for managing complexity and better ability to deal with natural language. The concept of relevance has been proposed as a measure of the relative importance of sets of rules [3]. Based on this concept and on this methodology, a new Fuzzy Clustering of Fuzzy Rules Algorithm is proposed and applied to organize the fuzzy IF . . . THEN rules.

***Index Terms* - Fuzzy clustering, fuzzy techniques, and Image segmentation.**

1. INTRODUCTION

Image segmentation aims to partition an image into regions having certain properties, and it is one of the fundamental image processing tasks. Image segmentation has been applied to many applications, including medical imaging, remote sensing, image compression, and image retrieval. Fuzzy logic based techniques have recently emerged with demonstrated strong potential for segmenting complex images [4,5], According to [6], fuzzy approaches for image segmentation can be categorized into four classes: segmentation via thresholding, segmentation via clustering, supervised segmentation, and rule based Segmentation. Among these categories, rule based approaches are able to take advantage of application dependent heuristic knowledge, and model them in the form of fuzzy rules. In [7], a set of fuzzy rules are established based on fuzzy variables, which are associated with the membership values of pixels obtained by the fuzzy c-mean clustering approach (FCM)

Section 2 provides a brief overview of the technique used to define the fuzzy rules. The processing steps of the proposed methods are presented in sections 3 with conclusions provided in section 4.

2. FUZZY RULES

Most fuzzy logic inference is based on Zadeh's composition rule. This generalizes traditional modus ponens which states that from the proposition

P1: If X is A Then Y is B and
P2: X is A,

we can deduce Y is B. If proposition P2 did not exactly match the antecedent of PI, for example, X is A', then the modus ponens rule would not apply. However, in [8], Zadeh extended this rule if A, B. and A' are modeled by fuzzy sets. In this case, X and Y are fuzzy variables [8] defined over universes of discourse U and V respectively. The proposition P1 concerns the joint fuzzy variable (X,Y) and is characterized by a fuzzy set over the cross product space UxV. This relation is "composed" with the input relation X is A', the result being projected onto the V universe of discourse, providing the meaning for the output Y is B'. The flexibility of this form of reference lies in the methods of translating proposition Pi and in the particular form of composition of fuzzy relations chosen. The fuzzy sets and membership functions are de?ned in the following manner: if X is a collection of objects, then a fuzzy set A in X is de?ned as a set of ordered pairs:

A = { (x, μ_x(x) | x ª X }

In the above equation, A is a fuzzy set and $\mu_x(x)$ is a membership function (MF in short) of x in A . The MF maps each element of X to a continuous membership value between 0 and 1. There are many types of membership functions (trapezoid, Gaussian, generalized bell, sigmoidal, etc.). For our application sigmoidal and Gaussian MF are chosen.

A fuzzy rule base contains fuzzy rules R_i:
R_i: IF (x_1 is A_{i1}) ^ (x_2 is A_{i2}) ^… ^ (x_n is A_{in})
THEN (y is B),

where A_{ij} and B_i are fuzzy sets, x_i and y are fuzzy inputs and output correspondingly. The structure of a rule is the following:

IF Premise THEN Conclusion;

where the premise consists of antecedents linked by fuzzy operator ^ (AND). There are many alternative possibilities to de?ne the fuzzy operator. The most frequently used one is the minimum operation. In the MAX-MIN inference method, the activation degree of the premise is:

w_i=min{ $w_{i1,}w_{i2,\ldots,}w_{in}$ },

where w_{ij} is the intersection between input x_j and fuzzy set Xij in the i^th^ rule. In a fuzzy inference system the output is calculated in the following way: the activation degree w_i is computed for each rule in the rule base. Then the i^{th} output fuzzy set is cut-off by w_i in each rule and the union of these cut-off fuzzy sets is composed, where the union means generally the maximum operation. Then a crisp value from the resultant output fuzzy set is calculated. This process is called defuzzi?cation. There are many ways of defuzzi?cation. The Center of Gravity (COG) method is generally used because it is general and easy to compute. This method calculates the crisp output by the sums of the center of gravities of the conclusions. Thus, a fuzzy inference system can compute output y of an input vector x.

In a fuzzy rules system representing an image, smooth areas of the image may be described by rules with a large radius, while regions with more detail may be represented by a higher number of rules with a smaller area of influence.

The membership function to measure the closeness of a pixel to a region represents the similarity between the pixel to be classified, called the candidate pixel, and the centre of a region based on the gray level intensity. The membership function reflects the axiom that the closer lo a region, the larger the membership value of the candidate pixel. The membership functions of all linguistic variables can be plotted as shown in Fig. 2.1. Whereas Figure 2.2 [illegible] t the inference mechanism.

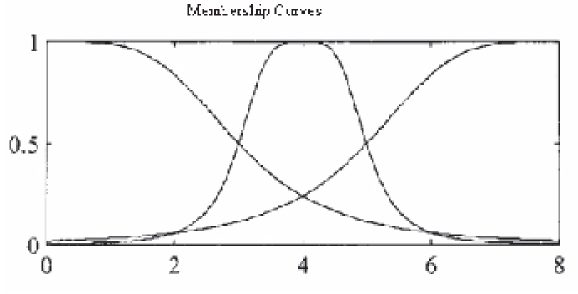


Figure 2.1: Membership functions of the linguistic variables used in the rule based system

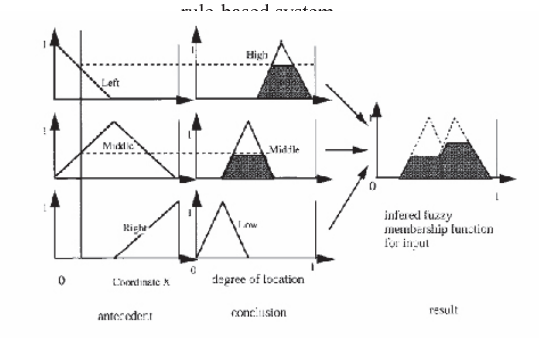


Figure 2.2: The Inference Mechanism

3. RECALL OF FCM CLUSTERING ALGORITHM

Fuzzy c-means called FCM is an unsupervised clustering algorithm, has been applied successfully to a number of problems involving feature analysis, clustering and classifier design, in fields such as agricultural engineering, astronomy, chemistry, geology, image analysis, medical diagnosis, shape analysis, target recognition and image segmentation[9]. The fuzzy extension allows $u_{i(}x)$ to membership function in fuzzy sets u_i and x interval [0 1] such that on

$$\sum_{i=1}^{c} \mu_x(i) = 1$$

For all x in X. In this case, $\{u_1, \ldots\ldots\ldots, u_c\}$ is called a fuzzy c-partition of X.

The fuzzy c-means objective function becomes,

$$J = \sum_{i=1}^{c} \sum_{j=1}^{N} \mu_i^m \parallel x_j - c_i \parallel^2$$

is the center of i^{th} cluster, Where $\{u_1, \ldots\ldots, u_c\}$ is a fuzzy c-partition, (m>1) is a degree of fuzziness and $X=\{x_1, \ldots\ldots\ldots, x_n\}$ represent the set of N data, in segmentation case X represent the pixels gray scales.

3.1 Fuzzy C-Means Algorithm

Step 1- Initialization:

For the number of cluster points *c with the constraint* $2 \leq c \leq N$ fix the fuzzifier. Set l=1. with fixed thresholding parameter $\varepsilon > 0$ and an initial fuzzy c-partition U° ($u^{\circ}{}_1$,............... $u^{\circ}{}_c$)

Step 2: Compute the vector of cluster centers c^l with c^{l-1}

$$c_i^l = \frac{\sum_{k=1}^{N} (\mu_{ik})^m}{\sum_{k=1}^{N} ((\mu_{ik})^m} * X_k$$

Step 3: Update partition matrix U^l with

$$\mu_{ik}^l = \frac{1}{\sum_{j=1}^{c} (\frac{d_{ik}}{d_{jk}})^{2/m-1}}$$

Where d_{ik} represent the Euclidean. d_{ik} = □X_k-C_i□

Step4: Compare U^l to U^{l-1}:

If □U^l - U^{l-1}□≤ ε then stop

Otherwise l=l+1 and go to step 2.

4. THE PROPOSED METHOD

Image can also be represented by means statistical features like Average grayscale value (Ag), Standard deviation (Sd), Variance(Va), Entropy(E), Skewness (Sk), Kurtosis(Ku). Therefore image is having certain modifications as follows. The dimension of the feature space depends on the representation of the image information. Therefore, a proposed segmentation approach combining pixel characterization by a set of statistical features and fuzzy clustering approach is discussed.

As we've mentioned each pixel becomes characterized by a set of statistical features, the proposed approach can be divided into two principal steps. The first consists to characterize each image pixel by a feature vector. Features can be extracted from regions masked by (*w*w*) window. Second step is a clustering procedure of the feature vector, initially extracted, using FCM clustering algorithm. By applying FCM, a partition of the feature vectors into new regions can be found.

This section describes the proposed image segmentation. As depicted in figure (4.2) the system scans the image using a sliding window and extracts a feature vector for each (*w*w*) block. The c-means algorithm is used to cluster the feature vectors into several classes with every class corresponding to one region in the segmented image [BEZ 81]. An alternative to the block-wise segmentation is a pixelwise segmentation by forming a window centered around every pixel. A feature vector for a pixel is then extracted from the windowed block. The spatial scanning order of an image (*M*N*) is performed, from left to right and top to bottom, pixel by pixel. Therefore, the standard fuzzy clustering algorithm will undergo the following modifications especially in two given steps.

Step 2: Compute cluster centers c^l with U^{l-1}

$$c_i^l = \frac{\sum_{k=1}^{N} (\mu_{ik})^m}{\sum_{k=1}^{N} ((\mu_{ik})^m} * X_k$$

Becomes

$$c_i^l = \frac{\sum_{k=1}^{N} (\mu_{ik})^m}{\sum_{k=1}^{N} ((\mu_{ik})^m} * S_k$$

Where S_k represents statistical features vector for k^{th} pixel.

Step 3: Update partition matrix U^l with

$$\mu_{ik}^l = \frac{1}{\sum_{j=1}^{c} (\frac{d_{ik}}{d_{jk}})^{2/m-1}}$$

Becomes

$$\mu_{ik}^l = \frac{1}{\sum_{j=1}^{c} (\frac{D_{ik}}{D_{jk}})^{2/m-1}}$$

With $D_{ik} = \Box S_k - C_i \Box$.

Figure 4.1: Overall Procedure of our Method

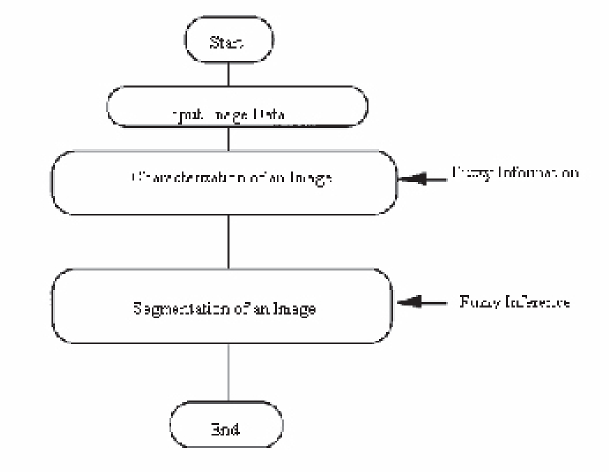
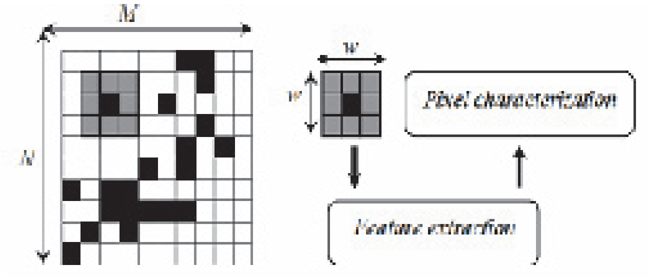


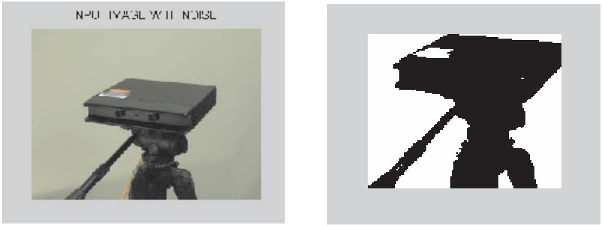
Figure 4.2: Pixel characterization by a feature vector.



5. EXPERIMENTAL RESULTS

In this section, an image is used to test the validity of the proposed method. When working with this algorithm, one has to specify the number of clusters. This number was chosen according to the number of textures in the input image. The segmentation results obtained with this method are shown in figure (5.1). For segmented images, the pixels that correspond to the same cluster are assigned the same gray level.

Figure 5.1: (a) Original image (b) Segmented image



6. CONCLUSION

Image segmentation is a difficult task in image processing. A unique segmentation approach will certainly never be established to be applied to all kinds of images. In this paper we have proposed an unsupervised fuzzy segmentation approach to be applied in image segmentation, aiming to increase the performance of the standard Fuzzy c-mean segmentation technique. Starting from a well known algorithm, fuzzy c-means, we modified its standard use by including the pixel characterization using a set of statistical features containing the most commonly used. Furthermore, the search of other optimal features to characterize texture and the use of sliding window with a variable size are an important perspective of our present work.

7. REFERENCES

[1] Klir, Y., "Fuzzy Sets and Fuzzy Logic", *Prentice Hall PTR, N. J.,* 1995.

[2] Ferdinando Di Martino, Salvatore Sessa, "Implementation of the Extended Fuzzy C-Means Algorithm in Geographic Information Systems" *Journal of Uncertain Systems Vol.3, No.4, pp.298-306*, 2009.

[3] Y. Hata, S. Kobashi, S. Hirano, H. Kitagaki, E. Mori, ''Automated segmentation of human brain MH images

aided by fuzzy information granulation and fuzzy inference," *IEEE 7hns. System., Part C: Applications and Reviews, vol. 30, pp. 381-395*, 2000.

[4] L. K. Huang, M. J. 3. wang, "Image thresholding by minimizing the measure of fuzziness," *Pattern Recognition, vol. 28, pp. 41-51,*1999.

[5] J. C. Bezdek, J. Keller, K. Raghu, N. H. Pal, "Fuzzy models and algorithms for pattern recognition and image processing", *Kluwer Academic Publishers, Boston*, 1999.

[6] Kuo-Lung Wu, Miin-Shen Yang, "Alternative c-means clustering algorithms", *Pattern Recognition* vol. 35, pp. 2267 2278, 2002.

[7] Kaymak, U., and M. Setnes, Fuzzy clustering with volume prototype and adaptive cluster merging, *IEEE Transactions on Fuzzy Systems*, vol.10, no.6, pp.705–712, 2002.

[8] L. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning", *Information Sciences, Part 1, Vol. 8, pp. 199-249; Part 2, Vol. 8, pp. 301-357; Part 3, Vol. 9, pp. 43-80*, 1975.

[9] Gour C. Karmakar, Laurence S. Dooley. "A generic fuzzy rule based image segmentation algorithm". *Pattern Recognition Letters 23,* pp. 1215-1227, 2002.

SVM(Support Vector Machine) Learning to Detect Sentence Boundary as Avoiding Ambiguity of Dot(.) Punctuation in English Text

Shweta Dubey
M.E(C.T.A Pursuing 4th Sem)
Shri Shankaracharya College of Engg. & Tech., Bhilai, C.G.
e-mail : dubeyshweta84@gmail.com

Vivek Dubey
ASSOCIATE PROFESSOR
Shri Shankaracharya College of Engg. & Tech., Bhilai. C.G.
e-mail : vivekdubey22@gmail.com

Tarun Dhar Diwan
M.E(C.T.A Pursuing 4th Sem)
Shri Shankaracharya College of Engg. & Tech., Bhilai, C.G.
taruncsit@gmail.com

Abstract

Soft Computing and Information Communication Technology(ICT) both are main part in information technology and computer science because today all documentations are computerized to store necessary information and database into English language, so natural language processing concept is come from A.I branch of computer science and information technology as part of soft computing and ICT(information communication technology). So SVM(support vector machine) can be used to solve the problem of sentence boundary detection which indicates the machine learning method regarding efficient soft computing and ICT. Hence Computer system may be learned to avoid ambiguity of dot(.) periods in detecting sentence boundary.

***Keywords*—-SVM (support vector machine), true positives, false positives, true negatives, false negatives, precision, recall, f feature, sentence boundary detection.**

1. INTRODUCTION

Basically sentence boundary detection is essential feature of recognizing the correct sentence meaning in text area and NLP, because without valid meaning based documentation has no any information extraction features and communication is generated in efficient successful trend. SVM may be used to extract the feature of sentence boundaries where (SVM) is a machine learning model and which solved the problem of sentence boundary detection by calculation of precision, recall, f feature value. Feature extraction f value is evaluated on the basis of precision and recall value calculation.Sentence boundary detection (SBD) means detecting end of sentence in given or used text. Various type of punctuation marks contain multiple feature at one time which gives sentence boundary ambiguity in text , This type of error of ambiguity can be find by calculating precision, recall values and f feature values of support vector machine and in such a way sentence boundary is detected, where ambiguity is come due to multiple behavior of one punctuation mark or symbols in text at one time, then precision and recall values are calculated to provide better information on what kinds of errors were made by ambiguous punctuation mark. Basically SVM (support vector machine) firstly calculate ambiguity of each segment in input English text by calculating precision value, here segment is one part of input text but not lead to exact sentence boundary . Secondly recall value to find exact sentence boundary lastly f is calculated to extract features of each segment of input text.

2. RELATED WORK

Basically here sentence boundary detection is done on the basis of dependency analysis of each segment of text. Segment of text is not necessary to occur as a boundary of segment so there dependency analysis of previous and next segment of text is done in three type of sentence which are given bellow:

1) *Open dependency based sentence.*
2) *Closed dependency based sentence.*
3) *Without dependency based sentence i.e. Independent sentence.*

In above three type of sentence based ambiguous text has a lot of need of detecting sentence boundary detection and after analysis of this three type of sentence in this paper by using suitable example, it is search that more feature of text segment gives more ambiguity with minimum SBD.

3.CHALLENGES/ PROBLEMS OF SENTENCE BOUNDARY DETECTION

In sentence boundary detection following problems are come as a challenges to detect end of sentence in creating recognize correct meaning of any text of English language during soft computing of information communication technology(ICT).

1. *Recognize tokens by removing white space and special characters from English text.*
2. *Resolving ambiguous separators in numbers to text.*
3. *Resolving ambiguous abbreviation in text.*
4. *Then resolving end of sentences(EOS)(?, !,.).*
5. *Morphology analysis on a corpus database in a form of statistical analysis and word/letter analysis which are part of nlp.*
6. *Correct syntactic and semantic meaning of each word by using grammar i.e. morphology and lexical analysis.*
7. *Avoiding ambiguity in punctuation marks.*

4.OBJECTIVE/GOAL OF PAPER

Finding the solution of sentence boundary detection challenge as avoiding ambiguity of dot(.) punctuation mark. It is done by calculating the error rate where error rate is related to each segment of text which is evaluated to check dependency analysis on the basis of dot(.) punctuation as either it is a abbreviation, termination or name into English language based text.

5. SIGNIFICANT OF PAPER

In this paper ambiguity of dot(.) periods is avoided into each segment of English language based text as to detect sentence boundary where f feature value is detected from SVM as when any word is presence it is counted as 1 otherwise as 0 value and this 0 or 1 value is assign finally assign into vector of SVMs.

6. PROPOSED METHODOLOGY

SVM methodology is support the creation of document vector from the features extraction value of each segment or line of text. Each new word seen by the SVM module is internally assigned to a different coordinate in the vector. The value of each coordinate is zero when the word is absent from a document otherwise one, basically it is a binary arrangement of text segment vector in memory[4][2], So maximum feature f value is created with maximum presence of one word at one time into one segment of text, which represent high maximum ambiguity i.e. maximum precision value, and maximum ambiguity of any word or string into one segment or line of text, decreases the sentence break or boundary detection behavior as a less (minimum) recall value. This is clear by using bellow example of input, output text along precision, recall, f feature extraction value. To apply methodology, calculation is done to calculate true positives, false positives, true negatives, and false negatives values whose description is bellow given. These values are used in recall, precision, f feature extraction value calculation formula.

1. *True Positives: Those who test positive for a condition and are positive (i.e., have the condition).*
2. *False Positives: Those who test positive, but are negative (i.e., do not have the condition).*
3. *True Negatives: Those who test negative and are negative,*
4. *False Negatives: Those who test negative, but are positive.*

Now Precision is the ratio between the number of candidate tokens that have been correctly assigned to a class and the number of all candidates that have been assigned to this class[5][1].
Precision value=true positives / (true positives + false positives)———(1)

Recall is defined as the proportion of all candidates truly belonging to a certain class that have also been assigned to that class by the evaluated system[3].
Recall value=true positives / (true positives + false negatives)————(2)

Finally, the so-called F measure is the harmonic mean of precision and recall (van Rijsbergen 1979).
f measure value =2* precision * recall / (precision + recall)————(3)

7. ANALYSIS

Here if any sentence is closed with proper syntactic and semantic analysis where start and end both are related to only one sentence then this type of sentence is come into without dependency or independent type of sentence, but if end is correct as syntactic and semantic way but starting is depend on previous segment of text then this type of sentience is known as closed dependency based sentence and if both starting and ending of any segment of text are incomplete the this type of segment is known as open dependency based sentence. Therefore sentence boundary is introduced as three type on the basis of above description of analysis i.e.

1. *Strong boundary (independent sentence i.e. without dependency based sentence)*
2. *Weak boundary (open dependency based sentence)*
3. *Absolute boundary (closed dependency based sentence)*

A) *Finding Dependency Structure of Each Sentence to Bellow Used Input Text*

I am doing Ph.D.
Ph.D. is difficult.
Mr. and Mrs.
Mishra are well.
They are Mr. and Mrs.
Tarun is playing a game.
Our college name is S.S.C.E.T.
Mr. M. Mishra is a best guide.

B) *Expected Output t*o Input English Text *After Applying Used Method in This Paper*

I am doing Ph.D.
Ph.D. is difficult .
Mr. and Mrs. Mishra are well.
They are Mr. and Mrs.
Tarun is playing a game.
Our college name is S. S.C.E.T.
Mr. M. Mishra is a best guide.

C) *Finding Four Type of Sentence Dependency Structure*

1) I am doing Ph.D. Ph.D. is difficult. Mr. and Mrs.
(Open dependency based sentence)

2) Mishra are well. They are Mr. and Mrs. Tarun is
(Open dependency based sentence)

3) playing a game. Our college name is S.S.C.E.T.
(closed dependency based sentence)

4) Mr. M. Mishra is a best guide.
(Without dependency based sentence i.e. independent sentence)

D) *Here Precision to each statement*

10/10+7=1.4=10/17=0.588
10/10+3=3.3=10/13=0.769
8/8+6=1.3=8/14=4/7=0.571
7/7+3=2.3=7/10=0.7

E) *Here Recall to each statement*

10/10+2=5=10/12=5/6=0.8333
10/10+2=5=10/12=5/6=0.8333
8/8+1=8=8/9=0.888
7/7+1=7/8=0.875

F) *f measure value*

(2*.588*.833)/(.588+.833)=.979/1.421=0.688
(2*.769*.833)/(.769+.833)=1.602
(2*(.571*.888)/(.571+.888)= 1.014/1.459=0.694
(2*(.875*0.7)/(.875+0.7)=1.225/1.575=0.777

Here input English text has a evaluation of four type of sentences i.e. first two type are open dependency based sentence, third closed dependency based sentence and finally last fourth type of sentence is without dependency based sentence and above calculation proof that max precision gives max ambiguity regarding to abbreviation , punctuation, separator so sentence boundary has a need of evaluating ambiguity and then eliminating it to gain exact sentence boundary to any input text.

Sentence boundary maximally detected to open dependency structure based text because features are maximum with various individual sentence boundaries.

8. RESULT

Table1
Sentence Boundary Detection Result Table
Obtained by Using SVMs

	Recall value	Precision value	F Measure Value
With dependency information (open)	10/10+2 =10/12 =5/6 =0.833 ≈0.83%	10/10+7 =10/17 =0.588 ≈0.6%	2*.588*.833/(.588+.833) =.979/1.421 =0.688 ≈0.7%
With dependency information (open)	10/10+2 =10/12 =5/6 =0.833 ≈0.83%	10/10+3 =10/13 =0.769 ≈0.8%	2*.769*.833/(.769+.833) =1.602 ≈1.6%
With dependency Information (closed)	8/8+1 =8/9 =0.888 ≈0.89%	8/8+6 =8/14 =4/7 =0.571 ≈0.6%	2*(.571*.888)/(.571+.888) =1.014/1.459 =0.694 ≈0.7%
Without dependency Information	7/7+1 =7/8 =0.875 ≈0.88%	7/7+3 =7/10 =0.7%	2*(.875*0.7)/(.875+0.7) =1.225/1.575 =0.777 ≈0.7%

Here f is feature measure i.e. lexical analysis, parsing or morphological analysis. Recall is exact sentence boundary detection value calculation. Precision is ambiguity calculation of each segment of used text.

Hence feature extraction value f is max 1.6% in open dependency based sentence to maximum ambiguity of 0.8% precision value and corresponding exact sentence boundary recall value is minimum i.e. 0.83%.

9.CONCLUSION

Maximum precision value with maximum f feature values generates the minimum recall value where precision is ambiguity during detecting sentence boundary, f feature values is made by ambiguous behavior of punctuation marks. This precision and f feature values are creates a minimum recall value as less sentence boundary of text, i.e. weak boundary extracts various features of text segment with less sentence boundary detection.

10.FUTURE WORK

Machine learning method of SVM must be implemented to compiler design as creating software of sentence boundary detection in all type of ambiguity of punctuation marks and symbols to English , Hindi Text, it will come into automatic sentence boundary detection sentence task and by there meaningful individual sentence are easy to recognize.

REFERENCES

[1] Mehdi M. Kashani, Fred Popowich, & Fatiha Sadat. Automatic transliteration of proper nouns from Arabic to English. The Challenge of Arabic for NLP/MT. *International conference* at the British Computer Society, London, 23October 2006; pp.76-83.

[2] Sarvnaz Karimi, Andrew Turpin, Falk Scholer, Punkt.2006. *English to Persian Transliteration*. SPIRE 2006: 255-266.

[3] Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullender *, Ed. Acm Press Frontier Series. ACM Press*, New York, NY, 19-33.

[4] Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara.1999. Japanese Dependency Structure Analysis Based on Maximum Entropy Models. In *Proceedings of the EACL*, pages 196–203.

An Efficient Algorithm Design for Two Dimensional Pattern Matching

Pawan Patnaik[*], Sanjeev Karmakar[*], Manoj Kumar Kowar[*] Jyoti Singh[#]
pawanpatnaik@yahoo.com, dr.karmakars@gmail.com, mkkowar@gmail.com, jsbhilai@yahoo.com

[*]*Bhilai Institute of Technology,Durg(C.G.)*
[#]*Swami Vivekanand Technical University,Bhiai(C.G.)*

Abstract- A method for designing a two-dimensional array matching machine whose running time is exactly *m+n–1* steps for by text array has been presented in this paper. A deterministic two-dimensional on-line tessellation acceptor is used as a two-dimensional array matching machine. Keyarrays here are not restricted to rectangular ones.

I. INTRODUCTION

During few years, many efficient algorithms to locate all occurrences of any of a finite number of keywords and phrases in an arbitrary text string have been developed [1-4].

Recently, on the other hand, several authors [5-7] have investigated the problem to exact-match subarray identification in more than one dimension.

In the basic two-dimensional array matching problem, two rectangular arrays are given: a 'keyarray' and a 'text array'. The problem is to find all occurrences of the keyarray as embedded subarrays of the text array. Such a problem occurs, for example, in some methods for detecting edges in digital images, where a set of 'edge detector' arrays are matched against the pixel array of the image.

By reducing the array problem to a string matching problem, Baker [5], Bird [6], and Sudo [7] demonstrated that efficient string matching algorithms may be applied to arrays, and described array matching algorithms whose running times are linear in the size of the text array.

Based on these algorithms, this paper presents a method for designing a two-dimensional array matching machine whose running time is exactly $m+n-1$ steps for $m \times n$ text arrays. A deterministic two-dimesional on-line tessellation acceptor [8] is used as a two-dimensional array matching machine. Keyarrays here are not restricted to rectangular ones.

Section II of this paper describes the background of the work presented. Section III deals with the design of the array matching machine for two-dimensional on-line tessellation acceptor (2-dota). In Section IV, a procedure for array matching problem has been discussed. Finally, the conclusions have been drawn in section V.

II. PRELIMINARIES

This section first reviews several terms and notations necessary for string pattern matching machines. The terms used have their usual meaning.

Let Σ be a finite alphabet (i.e., a finite set of symbols) and W ($\subseteq \Sigma^+$)be a finite set of keywords. A string matching problem for W is as follows: Given a string x in Σ^+ called a text string, find all pairs (y, i) in $W \times \{1, 2......, l(x)\}$ such that $x\ (i— l(y) + k) = y\ (k)$ for each $k\ (1\leq k\ \leq l\ (y))$.

A Pattern Matching Machine (PMM) has been developed as a useful device for solving the string matching problem [1]. A PMM for W is a machine which takes as input the text string x in which keywords of W appear as substrings. The PMM consists of a set of states. Each state is represented by a number. The machine processes the text string x by successively reading the symbols in x, making state transitions and occasionally emitting output. The behavior of the PMM is dictated by three functions: a *goto function* g, a *failure function f*, and an *output function output*.

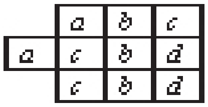
A real-time PMM for W is a deterministic finite automaton whose next move function is outputted when Algorithm 0 described below is applied to the goto and failure functions g, f of the PMM for W^2.

Algorithm 0. Construction of the next move function δ.
Input: Goto function g and failure function f for W.
Output: Next move function δ.
Method:

```
begin
$queue \leftarrow empty;$
for all $a \in \Sigma$ do
    begin
    $\delta(0,a) := g(0,a);$
    If $g(0,a) \neq 0$ then $queue \leftarrow g(0,a)$
    End;
While $queue \neq empty$ do
    begin
    $r \leftarrow queue;$
    for all $a \in \Sigma$ do
        begin
        $s := g(r,a);$
        if s ? $fail$ then
        begin
        $queue \leftarrow s;\ \delta\ (r,a) := s$
        end
    else $\delta\ (r,a) :=\ \ (f(r), a)$
end end end.
```

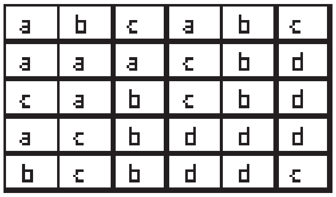
We then give several terms and notations necessary for two-dimensional array matching problems.

Definition 2.1 Let Σ be a finite alphabet. A (two-dimensional) array over Σ is a two-dimension pattern of elements of Σ with the right edge adjusted, as shown in Fig. 2. Let $\Sigma^{(2)}$ denote the set of arrays over Σ, and for each y in $\Sigma^{(2)}$, let r(v) denote the number of rows of y. For each rectangular array x in $\Sigma^{(2)}$, and each i,j ($1 \le I \le r(x), 1 \le j \le c(x)$), where c($x$) denotes the number of columns of x, let $x_{i,j}$ denote the symbol positioned at the ith row and the jth column of x. Furthermore, for each i ($1 \le i \le r(x)$) and each j,j' ($1 \le j \le j' \le c(x)$), let $x\,[i;(j,j')]$ denote the string $x_{i,j}\, x_{i,j-1}\, ...\, x_{ij} \in \Sigma$.



	a	b	c
a	c	b	d
	c	b	d

Fig.1. Keyarray



a	b	c	a	b	c
a	a	a	c	b	d
c	a	b	c	b	d
a	c	b	d	d	d
b	c	b	d	d	c

Fig.2. Text array

III. DESIGN OF 2-DOTA'S AS ARRAY MATCHING MACHINES

This section describes a procedure for designing a 2-dota which solves the array matching problem for the set consisting of only one keyarray.

Let y be a given keyarray over an alphabet Σ. then a 2-dota M_y, each (i, j)-cell of M_y ($1 \le i \le r(x)$, $1 \le j \le c(x)$) such that $x\,(i, j) \sim y$ enters an accepting state has been designed.

3.1. Construction of row next move function

Let W be the set of rows of the given kayarray y. That is, $W = \{y(i)|1 \le i \le r(y)\}$. Let $P(W) = \{p_1, p_2......p|w|\}^2$ and π be a mapping from $\{1, 2,....., r(y)\}$onto $\{1, 2, ..., |W|\}$ such that for each i, $p_{\pi(1)}$is the symbol corresponding to $y(i)$. This subsection gives the algorithm for constructing the row next move function δ_1 (for y) which is exactly the same as the next move function of the real time PMM for W.

3.1.1. Construction of row goto function δ_2

Using Algorithm 1 below, we first constrcut the row goto function δ_1.

Algorithm 1. Construction of the row goto function δ_1.

Input : A keyarray $y \in \Sigma^{(2).}$ (Let $W = (y)\ (i)|1 \le i \le r(y).)$

Output : The row goto function g_1 for y which is the goto function of the PMM for W, the trans-function t^w_c for W, the inverse trans-function h^w for W, the set $P\ (W) = \{p_1, p_2, ..., P|w|\}$, the queue 'QUEUE' whose ith element ($1 \le i \le r\ (y)$) is the symbol $p_{\pi(i)}(eP\in(W)$ corresponding to $y(i)$, and the set $N\,(g_1)$ of node numbers of g_1.

Method : We assume $g_1(s, a) = \text{fail}$ if a is undefined or if $g_1(s, a)$ has not yet been defined, and $t^w_c(s) = e$ when state s is first created. The procedure enter *(u)* inserts into the goto graph a path that spells out it.

```
Begin/*main*/
P(W)←empty; N (g_1)←empty;QUEUE←empty;
s1 :=0;k :=0;
for i := 1 to r(y) do enter(y(i));
for all a∈Σ do if g_1(0,a) = fail then g_1(0,a) :=0
end/*main*/

procedure enter
begin
s :=0;j :=1;
while g_1(s, a) = fail do
begin
s = g_1(s, a) : j: = j+1
end

While j ≤ l do
Begin
s1 :=s1 + 1: N (g_1) := N (g_1) U (s_1)
(g_1) (s, a_l) := s_1; s :=s1 j : = j+1

End;
If t^w_c(s) = e then
Begin
t^w_c(s) =
P (W) = P (W) U {pk}: K : = k+1
end;
QUEUE ←t^w_c(s)
end /*enter*/.
```

3.1.2. Construction of row failure function f_l

Using Algorithm 2 described below, we then construct the row failure function f_l for the given keyarray y, which is the failure function of the PMM for $W = \{y(i)|1 \le i \le r(y)\}$

Algorithm 2. Construction of row failure function f_l.

Input : The row goto function g_1 for y, the trans function t^w_c (Where $W = \{y(i)|1 \le i \le r(y)\}$, and the set $N\,(g_1)$.

Output : The row failure function f_l for y (which is the failure function of the PMM for W) and its inverse function f_1^{-1}, and the trans function t^w.

Method :

```
begin
For all s ∈N (g_1) do
begin
s : = g_1 (0, a);
end;
queue←empty;
```

```
for all
do
begin

if   then
begin
 ;
if   then
begin
 ;
While   do  ;
 ;
 ;
if   then

end end end end .
```

3.1.3. Construction of row next move function

Algorithm 3. Construction of the row next move function .
Input   : The goto function   from Algorithm 1 and the failure function f1 from Algorithm 2.
Output   : The next move function .
Method : Apply Algorithm 0 to   and .

Algorithm 4. Construction of the column goto function   and column failure function .
Input : The queue , the set , and the inverse transfunction from Algorithm 1, and the relational function   from Algorithm 4.
Output : The column goto function   and the column failure function   for the given keyarray y, the set , and the set , where output2 denotes the output function of the PMM obtained from the PMM for   by merging all mergable nodes.

Method : We assume   if a is undefined or if   has not yet been defined.

```
begin
 ;
for all   do
while   do
begin
 ;

push all elements of   to  ;
while   do
begin
 ;
while   do
begin
 ;
while   do  ;
 ;
if   then  ;
end
end;
 ;
while   do
begin

 ;
while   do
begin
 ;

end
end;

end;
 ;
for   to   do
end.
```

Algorithm 5. Construction of the column next move function .
Input : The goto function   and the failure function   from Algorithm 5.
Output : The next move function .
Method : Apply Algorithm 0 to   and   (Note that in this case,   in Algorithm 0 should be replaced by

3.2. Construction of the desired 2-dota

To design a 2-dota   which solves the array matching problem for the set consisting of only one keyarray y over   have been proposed. The 2-dota   obtained by using following Algorithm 7 acts in such a way that, when a text array x in   is presented to   each   - cell (of) satisfying   enters an accepting state.

Algorithm 6. Construction of desired 2-dota .
Input : The trans function   from Algorithm 2, the row and column next move functions   and   for y from Algorithm 3 and Algorithm 6, respectively,   and   (where   denotes the set of node numbers (states) of the directed graph representing), and the set   from Algorithm 5.
Output : The 2-dota .
Method :
1. Let
2. Let   and .
3. For each   and each , let
 ,
 ,
 ,
 ,
where   and   denote the node numbers of the starting nodes of the graphs representing   and , respectively.

Example 3.1. Let y be the keyarray over   (say   as shown in Fig.1. The 2-dota M, obtained from Algorithm 7 is , where
(1) ,
(2) , and

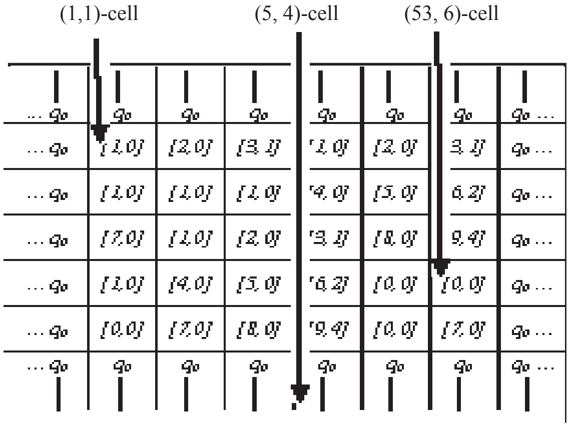
(3) , ,
, ,
, ,
, ,
, ,
And so on.

Let x be the text array as shown in Fig.2. Fig.3 shows the states which the cells of My enter after My has read x. Note that the cells (3,6) and (5,4) have entered an accepting state.

IV. THE ARRAY MATCHING PROBLEM FOR MULTIPLE KEY ARRAYS

In the last section, a procedure for designing a 2-dota which solves the array matching problem for only one keyarray has been described. This section briefly introduces a method for designing a 2-dota which solves the array matching problem for multiple keyarrays.

Fig. 3. State-configuration of 2-dota Mv after Mv has read the x.



... q0	q0	q0	q0	q0	q0	q0	q0 ...
... q0	[1,0]	[2,0]	[3,1]	'1,0]	[2,0]	3,1]	q0 ...
... q0	[1,0]	[1,0]	[1,0]	'4,0]	[5,0]	6,2]	q0 ...
... q0	[7,0]	[1,0]	[2,0]	'3,1]	[8,0]	9,4]	q0 ...
... q0	[1,0]	[4,0]	[5,0]	'6,2]	[0,0]	[0,0]	q0 ...
... q0	[0,0]	[7,0]	[8,0]	'9,4]	[0,0]	[7,0]	q0 ...
... q0	q0	q0	q0	q0	q0	q0	q0 ...

Let , , , , dota (obtained from the procedure described in the last section) which solves the array matching problem for . From My 's, we construct a 2-dota My (which solves the array matching problem for V) as follows, by using a direct product method.

,

where

(i)

,

(ii) ,

(iii) for each and each

,

and

The 2-doata above acts in such a way that, when a text array is presented to , each cell satisfying enter an accepting state with .

V. CONCLUSION

It has been concluded that the array matching problem can be efficiently solved by using a 2-dota. The procedure described here and the behavior of 2-dota's imply that the array matching problem can be solved in exactly steps for m by n text arrays. This can find use in modeling proteins - antibody reactions. This in term may find use in drug designing and DNA matching.

REFERENCES

[1] A.V. Aho and M.J. Corasick, Efficient string matching: An aid to bibliographic search, Comm. ACM 18 (6) (1975) 333-340.

[2] R.S. Boyer and J.S. Moore, A fast string searching algorithm, Comm. ACM 20 (10) (1977) 762-772.

[3] B. Commentz-Weller, A String Matching Algorithm Fast on the Average, Lecture Notes in Computer Science 71 (Springer, Berlin, 1979).

[4] D.E. Knuth, J.H. Morris and V.J. Pratt, Fast pattern matching in strings, SIAM J. Comput. 6(2) (1977) 323-350.

[5] T.P. Baker. A technique for extending rapid exact-match string matching to arrays of more than one dimension. SIAM J. Comput. 7 (4) (1978) 533-541.

[6] R.S. Bird. Two dimensional pattern matching, Information Processing Lett. 6 (5) (1977) 168-170.

[7] M. Sudo, A study of two dimensional pattern matching algorithm, Master's Thesis at Faculty of Engineering. Tohoku University (1977).

[8] K. Inoue and A. Nakamura, Some properties of two-dimensional on-line tessellation acceptors, Information Sci. 13 (1977) 95-121.

[9]. Tomas Polcar and Borivoj Melichar, Two-dimentional pattern matching by two-dimentional online tessellation automata. LNCS 3317, 2005, pp 327-328.

[10] Jan Zdarek and Borivoj Melichar, Two-Dimensional Pattern Matching by Finite Automata. LNCS 3845,2008,PP 329-340

[11] Neetha Sebastian and KAMALA KRITHIVASAN Inference of Regular Expressions from a Set of Srings using Pattern Automation, Journal of Combinatorics, Information and System Sciences, Vol 33, Nos. 3-4, pages 307-322, 2008.

[12] Lian, X., Chen, L. Efficient Pattern Matching over Uncertain Data
Streams In the HKIE Transactions, 16(4), pages 9-18, 2009

[13] Wanli Ouyang, Renqi Zhang and Wai-Kuen Cham, "Fast pattern matching using orthogonal Haar transform ". In IEEE Int. Conf. Computer vision and pattern recognition (CVPR2010), San Francisco, USA, Jun. 2010.

A Critical Review of Offline Handwritten Odia Character Recognition Techniques

Chinmayee Bihari[1], Babita Majhi[2], Ganapati Panda[3]

[1,2]*Dept. of Information Technology, Institute of Technical Education and Research, Siksha 'O'Anusandhan University, Bhubaneswar*

[1]chinmayee_bihari@yahoo.co.in, [2]babita.majhi@gmail.com

[3]*School of Electrical Sciences, Indian Institute of Technology Bhubaneswar*

[3]ganapati.panda@gmail.com

Abstract — **Odia is one of the major languages used in India. An extensive literature reveals that there are many research work has been carried out on English, Arabic, Chinese, Japanese etc, but few work have been reported on Indian Languages. However, on Odia character recognition practically very few work has been conducted. In this paper several processing techniques and methods which have been applied for handwritten character recognition are outlined. The recognition accuracy achieved in each case are also provided. The review work reveals that immediate attention is needed for suggesting suitable methods for recognition of handwritten Odia numerals and characters recognition. It is also observed that suitable feature extraction work has not also been carried. The present review work is modest attempt in this direction.**

Keywords—character recognition, handwritten character recognition and odia character recognition.

I. INTRODUCTION

Handwriting is one of the methods of communication. Hence, it is essential to recognize the handwriting by man or machine. Recognition by man is very easy as compared to machine. Off-line handwritten character recognition is one type of character recognition, where, handwritten characters written in documents are recognized by the system. In last few decades many technical papers on handwritten character recognition on different languages are reported. Also difficulties (on recognitions due to the handwritings of several individuals varies from each others) and several techniques for those character recognitions are presented. Form literature survey a number of research work has been carried out on recognition of Roman, English, Chinese scripts etc, but few papers have dealt on Indian Scripts ([2- 4, 7]). On reviewing work relating to Indian Script recognition ([5, 6]) very little research work has been reported on Odia character recognition [8].

Various sub-processes involved in handwritten Character recognition are: pre-processing, feature extraction, classification and decision making. Pre-processing is the preliminary step of character recognition, in which the handwritten characters are first captured in an image format. Then the normalization is done to make the character image matrix to an N×N matrix for removal of irregularities. Feature Extraction is the process of minimizing the intra-class variability and maximizing the inter-class variability. This is an important step in achieving good performance for a character recognizer. Classification is the process in which the new input data are mapped to the predefined classes to which they belong. Decision making is the process of choosing the final answer to reach a desired level of performance. Handwritten character recognition is one of the most active areas of research with various practical applications in bank cheque reading, postal address reading, automatic data entry, helping blinds to read etc. These systems offer potential advantages by providing an interface that facilitates interaction between man and machine [1].

The basic features of offline character recognition are:

- Flexibility: it should recognize a large number of character patterns.
- Efficiency: it should be efficient.
- Automated Learning: it should have automatic learning capability.
- Online Adaptability: it should have the capability to gather new knowledge of different writer-specific handwritten patterns as it operates.

The review paper is organized into five sections. Section 1 discusses the problem of handwritten character recognition. Section 2 provides an extensive review of character recognition research. In section 3 work carried on Odia character is discussed. Finally Section 4 outlines conclusion of the investigation.

II. REVIEW OF LITERATURE

The first widely used commercialized OCR was the IBM 1418, which was designed to read a special IBM font, 407. In 1900 Russian Scientist Tyuring developed an aid for visually handicapped. The first character recognizers appeared for the development of the digital computers in the middle of the 1940s. During 1950-1990 the character recognition systems available were of low quality. After that the recognition system developed with the combination of image processing, pattern recognition, and artificial intelligence methodologies.

Georgios Vamvakas et al. have proposed an off-line handwritten character recognition system in which he has suggested a new methodology for feature extraction in which the character image is iteratively subdivided so that the resulting sub images have approximate number of foreground pixels [11]. In this case pre-processing has been done using the Niblack's approach [12] in which a character image is taken and considered 1s as foreground 0s for background pixels. At level zero the image is subdivided in four sub-images, taking the divisive point (DP) into consideration. The DP is obtained by the intersection of two lines, such that horizontal line drawn by taking the balance number of foreground pixels on top and bottom part of the line. Also in case of vertical line there is balance number of foreground pixels on left and right part of line. In this way at the first level 16 sub-images and 4 DPs are found. The larger number of DPs shows an improved representation of the character. The classification is performed in two phases: (a) training Phase, (b) recognition phase. In training phase the initial classification is mapped to the highest recognition rate, then the mutually misclassified classes are merged and next the merged classes are distinguished if possible. In the recognition phase, the input data are mapped to the previous classes, if they do not match, then they form a new class. In the decision making process accuracy classification is about 94.73% and 99.03% for CEDAR and MINIST databases respectively.

A hierarchical approach for Bangla words recognition is reported by Basu et al. in 2009[5]. In this case the segmentation and recognition processes are marked as an important aspect for the cursive words, which are the more difficult processes. First the text lines are drawn for the handwritten words from the document images, then segmentation is done over them and finally recognition is carried out. For the feature extraction of the words that belong to the different zones, the character images are enclosed within the minimal bounding boxes. After that different features like longest-run, modified-shadow, octant centriod are applied to different zoned parts. Then the classification is done by a two-pass approach. In the first pass, classes are formed taking the global features into consideration and in second pass, the classes are refined due to the local features. The recognition result in two pass approach is 80.58%.

Hiromichi Fujisawa has proposed a new robust technique for postal address recognition [7]. The printed and handwritten mails are recognized through five principles: hypothesis driven principle, deferred decision/multiple-hypotheses principle, information integration principle, alternative solution principle and perturbation principles. Handwritten character recognition by different methods do not give same accuracy. So it is also an active research area to find better accuracy in the handwriting recognition. In 2008 Tian fu Gao et al. have proposed an method called linear discriminant analysis (LDA)-based compound distances for distinguishing similar characters [13]. The LDA- based compound distance is an extension of compound Mahalanobis function (CMF), where the complementary distance is calculated in one-dimensional space, but in LDA-based method it calculates the distance in high-dimensional space for better accuracy. They also found here the reduced error rates by a factor over 26%. Better normalization gives improved performance for handwritten character recognition, where the character image is brought into a standard shape. Cheng-Lin et al. have emphasized on the normalization method and have proposed a pseudo two-dimensional nonlinear normalization (P2DNLN) for the line density of blurred characters [14]. They have overcome the insufficient shape restoration capability in one-dimensional nonlinear normalization method. Though the P2DNLN adds little computational overhead, it has reduced the error by a factor of 16.4%. For better classification, Luisa Mico has proposed two methods known as LAESA and TLAESA [15], developed four fast nearest neighbour search algorithms to a handwritten character recognition task in order to compare their behavior. These algorithms are very fast and efficient algorithms. Chaudhuri et al. have reported a system on recognition of two popular Indian languages Bangla and Hindi [18]. They have modeled a single system for recognizing these languages due to many common features between them. The different processes like document digitization, skew detection, text line segmentation and zone separation, word and character segmentation, character grouping into basic, modifier and compound character category are implemented by using same algorithms for both languages. But the feature sets and classification tree for error correction are different for both scripts. The word level recognition accuracy for Bangla character is 93.66% and character level is 98.62% where as for Devnagari it is 91.25% in word level and 97.18% *in* character level. Hailong Liu et al. have replaced statistical approach for handwritten character recognition like directional element feature (DEF) by gradient feature extraction which provides higher resolution and modified quadratic discriminant function (MQDF) by including minimum classification error (MCE) [19]. To improve the recognition accuracy they have applied several state-of-the-art techniques. These techniques are used for both handwritten Chinese digit as well as character recognition. The recognition accuracy is 99.54% on MINIST test set and 99.33% on ETL9B test set.

Due to the vast differences of handwritten characters of several individuals the recognition is an NP-hard problem. Therefore Batuwita et al. have applied the fuzzy logic for feature extraction which have the characteristics of flexibility, efficiency and online adaptability property [20]. Mainly they used online adaptable fuzzy method for offline handwritten numeric character recognition.

In 2008 Vamvakas et al. have reported an OCR methodology for historical documents that is printed or handwritten [21]. They have used here three steps for creating a database for training using a set of documents and recognizing the new document images. The different processes used are binarization, top down segmentation to detect text lines, words and characters, clustering to group characters of similar shape. This methodology requires neither any knowledge of the fonts nor the existence of standard database because it can adjust depending on the type of documents that need processing. The recognition rates using Levenshtein distance is about 83.66% and 72.68% for different test sets.

III. RESEARCH WORK ON HANDWRITTEN ODIA CHARACTER RECOGNITION

A method called Hidden Markov Model (HMM) is used by Bhowmik et al. to recognize the Odia numerals [8].One key factors behind using HMM is the states are not known previously, but determined automatically based on a database. The classification accuracy is 95.89% and 90.50% for training and test sets respectively. The number of characters in Odia like other Indian languages is large and there are also many compound characters. There are more than 200 characters in Odia and here the OCR is a complex task. In 2001 Chaudhuri et al. [16] overtook the difficulties of recognizing of Odia characters. They have used several techniques, like skew correction, line segmentation, zone detection, and word and character segmentation. Here the individual characters are recognized by stroke and run-number based features. The boundary-tracing method is used to distinguish among the similar shaped characters. There are two stages of recognizing the characters; in the first part modified shaped characters are recognized and subsequently the rests are recognized. The recognition of compound characters involves also two stages: that is the characters are grouped into small subsets by a feature based tree classifier and the characters in each group are recognized using a sophisticated run-number based matching approach. The individual text lines identifiers provides 97.5% accuracy where as the word and character segmentation provide 97.7% and 97.2% accuracy respectively. In this way their system recognizes the characters with about 96.3% accuracy. Pal et al. have used the curvature feature for the cursive characters in Odia [17]. First the normalization takes place by 49×49 blocks of the input image. There is also a method for curvature features like, bi-quadratic interpolation which has been applied and then the direction of gradient is quantized. Finally the principal component analysis is used to reduce the dimension of the input image. They have used 9556 Odia handwritten character samples and obtained 94.60% accuracy from their proposed system.

IV. CONCLUSION

Handwritten character recognition is an important research area particularly in context of Indian languages. Further these research findings have potential future as they can be applied to many fields. A thorough and detailed study on various reported methods reveal that in case of Odia language the research effort in this direction is minimal. Hence feature extraction and development of improved classifiers for Odia characters are burning issues which need immediate attention.

REFERENCES

[1] A. Amin. “Off-Line Arabic Character Recognition System: State of the Art”, Pattern Recognition, vol. 31, No. 5, pp 517-530, 1998.

[2] T.S. EL-Sheikh, R.M. Guindi, “Computer recognition of Arabic cursive scripts, Pattern Recognition”, vol. 21, pp. 293–302, 1988.

[3] A. Amin, “Recognition of hand-printed characters based on structural description and inductive logic programming”, Pattern Recognition Letters, vol. 24, pp. 3187–3196, 2003.

[4] S. Espana-Boquera, M.J. Castro-Bleda, J. Gorbe-Moya, F. Zamora-Martýnez , “Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models”, IEEE transactions on pattern analysis and machine intelligence, pp. 1-14, 2010.

[5] Subhadip Basu, Nibaran Das, Ram Sarkar, Mahantapas Kundu, Mita Nasipuri and Dipak Kumar Basu, “A hierarchical approach to recognition of handwritten Bangla characters”, Pattern Recognition, Elsevier, vol. 42, pp. 1467-1484, 2009.

[6] U. Pal, T. Wakabayashi, F. Kimura, “Comparative Study of Devnagari Handwritten Character Recognition using Different Feature and Classifiers”, 10th International Conference on Document Analysis and Recognition, pp. 1-5, 2009.

[7] Hiromichi Fujisawa, “Forty years of research in character and document recognition—an industrial perspective”, Pattern Recognition, Elsevier, vol. 41, pp. 2435 - 2446, 2008.

[8] Tapan K Bhowmik, Swapan K Parui, Ujjwal, Bhattacharya, Bikash Shaw, “An HMM Based Recognition Scheme for Handwritten Oriya Numerals**”,** 9th International Conference on Information Technology (ICIT’06), 2006.

[9] B.B. Chaudhuri, U. Pal, M. Mitra, “Automatic Recognition of Printed Oriya Script”, Sadhana, vol. 27, pp. 23–34, 2002.

[10] S. Mohanti, “Pattern recognition in alphabets of Oriyalanguage using Kohonen neural network”, Int. J. Pattern Recogn. Artif. Intell. Vol. 12, pp. 1007–1015, 1998.

[11] Georgios Vamvakas , Basilis Gatos and Stavros J. Perantonis, “Character recognition through two-stage foreground sub-sampling”, Pattern Recognition, Elsevier, vol. 43, pp. 2807-2816, 2010.

[12] W. Niblack, “An Introduction to Digital Image Processing”, Prentice-Hall, Englewood Cliffs, NJ, pp. 115–116, 1986.

[13] Tian-Fu Gao, Cheng-Lin Liu, “High accuracy handwritten Chinese character recognition using LDA-based compound Distances”, Pattern Recognition, Elsevier, vol. 41, pp. 3442 – 3451, 2008.

[14] C.-L. Liu, K. Marukawa, “Pseudo two-dimensional shape

normalization methods for handwritten Chinese character recognition", Pattern Recognition Elsevier, vol. 38, pp. 2242 – 2255, 2005.

[15] L. Mico, J. Oncinar, "Comparison of fast nearest neighbour classifiers for handwritten character recognition", Pattern Recognition Letters, Elsevier, vol. 19, pp. 351–356, 1998.

[16] B.B. Chaudhuri U. Pal M. Mitra, "Automatic Recognition of Printed Oriya Script", IEEE Trans. Computer Vision and Pattern Recognition Unit, pp.795-799, 2001.

[17] U. Pal, T. Wakabayashi and F. Kimura, "A System for Off-line Oriya Handwritten Character Recognition using Curvature Feature", 10th International Conference on Information Technology, IEE Trans, pp. 221-223, 2007.

[18] B. B. Chaudhuri and U. Pal,"An OCR System to Read Two Indian Language Scripts: Bangla and Devnagari (Hindi)", Computer Vision and Pattern Recognition Unit Indian Statistical Institute, IEEE Trans, pp. 1011-1015, 1997.

[19] Hailong Liu and Xiaoqing Ding, "Handwritten Character Recognition Using Gradient Feature and Quadratic Classifier with Multiple Discrimination Schemes", ICDAR'05, IEEE Trans, pp. , 2005.

[20] K.B.M.R. Batuwita, G.E.M.D.C. Bandara, "Fuzzy Recognition of Offline Handwritten Numeric Characters", IEEE Trans, pp., 2006.

[21] G.Vamvakas, B.Gatos, N. Stamatopoulos, and S.J.Perantonis, "A Complete Optical Character Recognition Methodology for Historical Documents", The Eighth IAPR Workshop on Document Analysis Systems, IEEE Trans, pp. 525-532, 2008.

Improving Accuracy of Software Effort Estimation in Software Engineering using Type-2 Fuzzy Logic

H. S. Hota
Dept. of CSIT, Guru Ghasidas Vishwavidyalaya
Bilaspur, India hota_hari@rediffmail.com

Ramesh Pratap Singh
Dept. of Computer Science, D.P. Vipra College, Bilaspur, India
singhramesh30@gmail.com

Abstract

Estimation of effort required for development of software products is inherently associated with uncertainty. Formal effort estimation models like Constructive Cost Model (COCOMO) are limited by their inability to manage uncertainties and imprecision surrounding software projects early in the development life cycle. Type-2 fuzzy logic based cost estimation models are proposed here because they are more appropriate when vague and imprecise information was to be accounted for and was used in this research to improve the effort estimation accuracy. The aim of this research is to study the role of size in precision improvement of effort estimation by characterizing the size of the project using Type-2 fuzzy logic model. From the experimental results, it was concluded that, by fuzzifying the project size using type-2 fuzzy logic model, the accuracy of the effort estimation can be improved and the estimated effort can be very close to the actual effort. At the end a conceptual Multiple Type-2 fuzzy inference model (MT2FIS) has also been proposed which is incorporating other input factors of COCOMO.

Keywords- Constructive Cost Model (COCOMO), type-2 fuzzy logic, Magnitude of Relative Error(MRE), Mean Magnitude of Relative Error(MMRE), Footprint of Uncertainty (FOU), Software effort estimation.

I. INTRODUCTION

The precision and reliability of the effort estimation is very important for software industry because both over estimates and under estimate of the software effort are harmful to software companies. As highlighted by McConnell in [16], several surveys have found that about two-thirds of all projects substantially overrun their estimates. In realty, estimating software development effort remains a complete problem attractive considerable research attention. By adopting a sound estimation process that allows the team and the project manager to reach a consensus on the effort involved in the work, the morale is maintained and the work is much more predictable. It is very important to investigate novel methods for improving the accuracy of such estimates. As a result many models for estimating software development effort have been proposed and are in use.

The best known technique using LOC (Lines of Code) is the COCOMO (Constructive COst MOdel), developed by Boehm[1]. This model, along with other SLOC/SDI based models, uses not only the LOC, but also other factors such as product attributes, hardware limitations, personnel, and development environment. These different factors lead to one or more "adjustment" factors which adjust the direct evaluation of the effort needed. In COCOMO's case, there are seventeen such factors derived by Boehm. This model shows a linear relation between the LOC and the effort. But Constructive Cost Model (COCOMO) is limited by their inability to manage uncertainties and imprecision surrounding software projects early in the development life cycle [17].

Assumptions make estimates more accurate. Fuzzy logic-based cost estimation models are more appropriate when vague, imprecise and uncertain information is to be accounted [2] for suggested the use of fuzzy sets in improving the accuracy of effort estimation through fuzzy sets. But the amount of uncertainty involve in the estimation process, the type-2 fuzzy sets are propose here to use in effort estimation of software development. This study proposed to extend the COCOMO model by incorporating the concept of fuzziness into the measurements of size.

The size of the project in COCOMO [1][3] is represented by fixed numerical values. In fuzzy logic based cost estimates models; this size is represented with fuzzy interval values. In the work in related area, attempts have been made to fuzzify some of the existing project data, [4] compared function point analysis, regression techniques, feed forward neural network and fuzzy logic in software effort estimation. Their results will show that fuzzy logic model will achieve good performance, being performed in term of accuracy only by neural network model. In fuzzy logic model, triangular membership functions will be define for small, medium, large intervals size. [8] First realized the fuzziness of several aspects of COCOMO [9] researched on the application of fuzzy logic to COCOMO and function points' model. Looking into the amount of uncertainty involve in effort estimation process of software, a novel method using type-2 fuzzy set which merely deals with the uncertainty in process is proposed here.

The concept of type-2 fuzzy set was introduced by Zadeh (1975) [15] as an extension of the concept of an ordinary fuzzy set (called type-1 fuzzy set). The basics of fuzzy logic do not change from type-1 to type-2 fuzzy sets, and in general will not change for type-n [5]. A higher type number just indicates a higher degree of fuzziness. Since a higher type changes the nature of the membership functions, the operations that depend on the membership functions change, however, the basic principles of fuzzy logic are independent of the nature of membership functions and hence do not change. Rules of inference, like Generalized Modus Ponens, continue to apply [5].

II. METHODOLOGY

A typical type-2 fuzzy logic system is depicted in fig1.

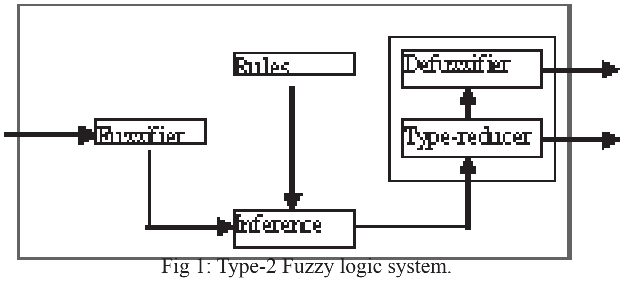


Fig 1: Type-2 Fuzzy logic system.

A type-2 fuzzy set is characterized by a fuzzy membership function i.e. the membership grade for each element of this set is a fuzzy set in (0,1), unlike type-1 set where the membership grade is crisp number in (0,1). Such sets can be used in situations where there is uncertainty about the membership grades themselves i.e. an uncertainty in the shape of the membership function or in some of its parameters.

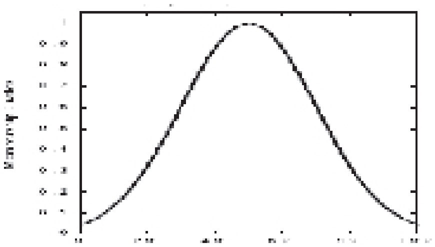


Fig 2: Type-1 membership function.

If for a type-1 membership function, as shown in fig 2, we blur it to the left and to the right, as illustrated in fig 3, then a type-2 membership function is obtained. In this case for a specific value x the membership function (u') takes on different values, which are not all weighted the same, so we can assign an amplitude distribution to all of these points. Doing this for all $x \in X$we create a three dimensional membership function or a type-2 membership function that characterizes a type-2 fuzzy set [14][11]. A type-2 fuzzy set $\tilde{A}$, is characterized by the membership function:

$$\tilde{A} = \{((x,u), \mu_{\tilde{A}}(x,u)) \mid \forall x \in X, \forall u \in J_x \subseteq (0,1)\} \quad \ldots(1)$$

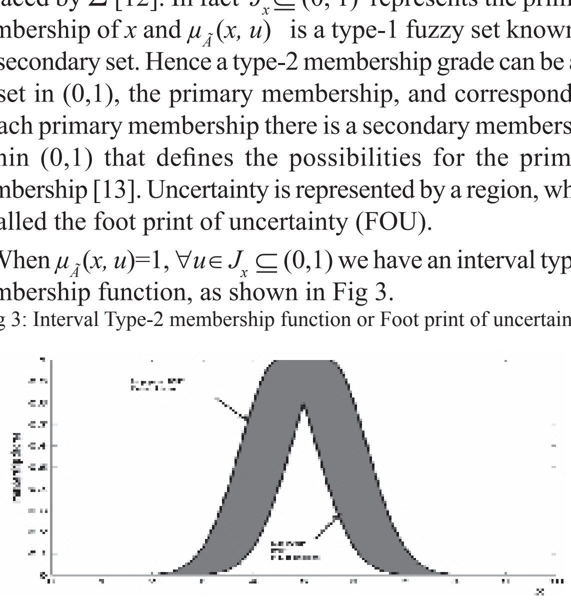
Where $0 \leq \mu_{\tilde{A}}(x,u) \leq 1$ Another expression for $\tilde{A}$ is

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x,u)/(x,u) \quad J_x \subseteq (0,1) \quad (2)$$

Where $\iint$ enotes the union overall admissible input variables x and u. For discrete universe of discourse $\int$ replaced by Σ [12]. In fact $J_x \subseteq (0, 1)$ represents the primary membership of x and $\mu_{\tilde{A}}(x, u)$ is a type-1 fuzzy set known as the secondary set. Hence a type-2 membership grade can be any subset in (0,1), the primary membership, and corresponding to each primary membership there is a secondary membership within (0,1) that defines the possibilities for the primary membership [13]. Uncertainty is represented by a region, which is called the foot print of uncertainty (FOU).

When $\mu_{\tilde{A}}(x, u)=1, \forall u \in J_x \subseteq (0,1)$ we have an interval type-2 membership function, as shown in Fig 3.

Fig 3: Interval Type-2 membership function or Foot print of uncertainty



(FOU).

The uniform shading for the FOU represents the entire interval type-2 fuzzy set and it can be described in terms of an upper membership function $\bar{\mu}_{\tilde{A}}(x)$ and a lower membership function $\underline{\mu}_{\tilde{A}}(x)$.

A common criterion for the evaluation of cost estimation models is the magnitude of relative error (MRE), which is defined in the following equation:

$$MRE_i = \frac{\left[Actual\ \ Effort_i - Predicted\ \ Effort_i \right]}{Actual\ \ Effort} \quad \ldots(3)$$

The MRE value is calculated for each observation i whose effort is predicted. The aggregation of MRE over multiple observations (N) can be achieved through the Mean MRE (MMRE) in as follows:

$$MMRE = \frac{1}{N}\sum_{i}^{n} MRE_i \quad \ldots(4)$$

III. PROBLEM FORMULATION

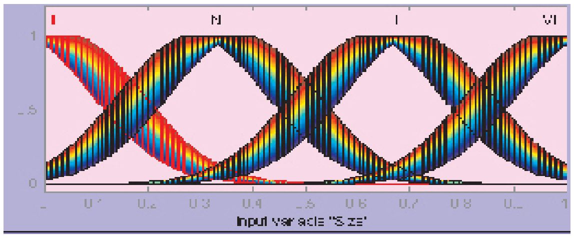
Effort is defined as Person Month (PM) in COCOMO. It determines the effort required for a project based on software project's size in Kilo Source Line of Code (KSLOC) as well as other cost drivers known as scale factors and effort multipliers as given below:

$$PM = K.(Size)^{0.91+0.01 \times \sum_{j=1}^{5} SF_j} \times \prod_{i=1}^{17} EM_i \quad \ldots(5)$$

Where, K is a multiplier constant and the set of Scale Factors (SF) and Effort Multipliers (EM) are defined the model. Here n=15 EM and s=5 SF.

It is important that uncertainty at the input level of the COCOMO model yields uncertainty at the output [7], which leads to gross estimation error in the effort estimation. The problem of software cost estimation relies on a single (numeric) value of size of given software project to predict the effort. However, the size of project is, based on some previously completed projects that resemble the current one. Obviously, correctness and precision of such estimates are limited. It is a principal importance to recognize this situation and come up with technology using which we can evaluate the associated imprecision residing within the final results of cost estimation. The technology endorsed here deals with type-2 fuzzy sets. Among the various input parameters of COCOMO model one important parameter size is consider here. By fuzzifying this using type-2 fuzzy logic we can calculate the effort that will improve the estimation accuracy. Fig 4 shows the fuzzification of size using Gaussian membership function of type-2 fuzzy logic with universe of discourse taken from COCOMO dataset [10]. After fuzzification an equation is formulated as written in equation (7) to calculate fuzzy size.

Fig 4: Representation of input variable size using a Type-2 Gaussian membership function or FOU



Using type-2 fuzzy sets, size of a software project can be specified by distribution of its possible values. Commonly, this form of distribution is represented in the form of a type-2 fuzzy set. This becomes obvious and, more importantly, bears substantial significance in any practical endeavor.

In this research paper, it is projected to characterize the size of the project using type-2 gaussian membership function which gives superior transition from one interval to another. The type-2 fuzzy set is described by equation (1) and (2) such that: The upper membership function is

Upper (FOU(Ã)) = N(m,σ_2 x)

And the lower membership function is

Lower (FOU(Ã)) = N(m, 1; x)σ …(6)

In this research, a new fuzzy effort estimation model is proposed using type-2 gaussian function to deal with the size and to generate type-2 fuzzy membership function and rules. In the next step, we evaluate the COCOMO model using equation (1) and size will be obtain from type-2 fuzzy set, rather than from the classical size. The classical size and membership function defines for the size:

$$\tilde{F}_{size} = f(\overline{\mu}_{\tilde{A}}(x), \underline{\mu}_{\tilde{A}}(x), Size) \quad \ldots (7)$$

For case, f is taken as a linear function, where $\underline{\mu}_{\tilde{A}}(x)$ and are the upper and lower bounds of FOU as shown in fig: 4 the equation (7) can be rewritten as below:

$$\tilde{F}_{size} = [\overline{\mu}_{\tilde{A}}(x) \times Size , \underline{\mu}_{\tilde{A}}(x) \times Size] \quad \ldots (8)$$

IV. EXPERIMENTAL RESULTS

The size of the project is defined and customized to the type-2 Gaussian membership function. In designing the above model, we have used COCOMO dataset [10]. The assignment of linguistic values to the size uses conventional quantification where the values are intervals. In the case of the size attribute, we have defined a type-2 fuzzy set for each linguistic value with a Gaussian shaped membership function as shown in fig 4. The Table 1 shows the sample data of actual effort, COCOMO effort and proposed model effort for different projects. The COCOMO effort was calculated using equation (5) and type-2 fuzzy effort is calculated using equation (7) and (1).

This table clearly shows that effort based on type-2 fuzzy logic model is closed to actual effort. This same is shown in Fig. 5 in from of bar chart. The MRE is calculated using equation (3). For example, the MRE calculated for Project ID 8 for COCOMO and Type-2 fuzzy model is 0.384837 and 0.163423 respectively.

Table 1: Results and comparison of effort estimation in person Months

Then for each model, the Mean Magnitude of Relative Error (MMRE) was calculated. Finally mean of those calculations are used to compare both models. The result of project ID shown in Table 1 shows the MMRE for COCOMO is 0. 0.045256087 and for proposed model the value equals to 0.029655856. It shows the proposed model has MMRE less than COCOMO, so it means the accuracy of proposed model is better than that of COCOMO.

Project ID	Actual Effort	COCOMO	Type-2 Fuzzy logic model
1	2040	1523.55	1834.26
2	1600	1256.12	1389.87
3	243	222.58	241.22
4	156	280.13	201.36
5	61	45.63	54.35
6	599	539.60	615.81
7	8	10.65	8.35
8	1075	661.30	899.32
9	423	348.22	381.69
10	321	284.20	305.10

Table 2: Comparison of COCOMO and Type-2 fuzzy logic model based on MMRE

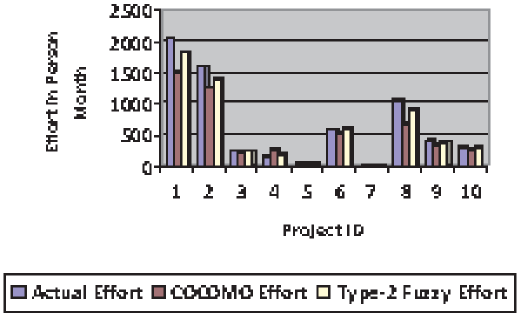
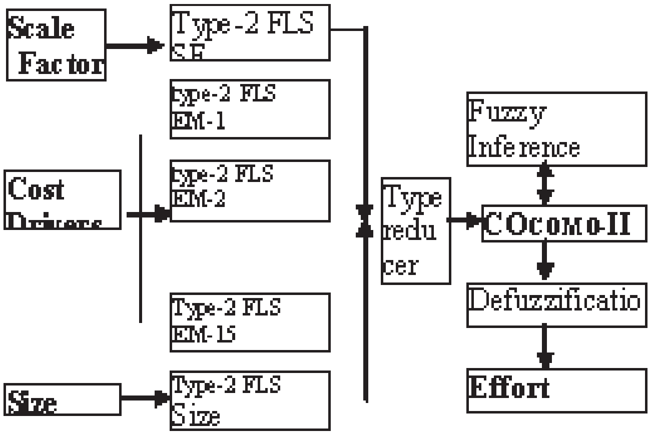


Fig 5: Comparison of effort estimation

Model	Evaluation MMRE
COCOMO	0.045256087
Proposed Model (Type 2 Fuzzy logic model)	0.029655856

V. PROPOSED CONCEPTUAL MODEl

Figure 6: The proposed Multiple Type-2 Fuzzy Logic based Model (MT2FIS) with Inputs: COCOMO II Cost Drivers, Scale Factors and Size. Output: Estimated Effort.



In this paper we have compared proposed model with COCOMO by considering only one input parameter. However we can consider other input parameters in order to achieve more accurate effort. Keeping this concept in mind, a model named with MT2FIS is proposed here. Block diagram of this model is depicted in Fig 6. All these inputs will be fuzzified with type-2 fuzzy membership function and aggregated input will be given to the type reducer in order to convert it into type-1, A rule base will also be developed. Finally after defuzzification the model will produce effort .This effort may be compared with actual and COCOMO effort.

VI. CONCLUSION

In this research work, for the size of the project, its associated linguistic values are represented by type-2 Gaussian membership function. The relative error for type-2 fuzzy logic model is lower than that of the error obtained using COCOMO. From experimental results, it is concluded that, by fuzzifying the size of the project using type-2 fuzzy logic model, it can be proved that the resulting estimate impacts the effort. The effort generated using proposed model gives better result than that of using ordinal COCOMO. This illustrates that by fuzzifying size using type-2 fuzzy logic model, the accuracy of effort estimation can be improved and the estimated effort can be very close to the actual effort. Moreover by capturing the uncertainty of the initial data (estimates), one can monitor the behavior (quality) of the cost estimates over the course of the software project. This facet adds up a new conceptual dimension to the models of software cost estimation by raising awareness of the decision making with regard to the quality of the initial data needed by the models. At last a Multiple type2 fuzzy model(MT2FIS) is proposed with various input parameters of COCOMO to predict more accurate effort for software development .

REFERENCES

[1]. Boehm, B.W.,1981. Software Engineering Economics, Englewood Cliffs, NJ, Prentice-Hall.

[2]. C.S. Reddy and KVSVN Rao “Improving the accuracy of effort estimation through fuzzy set representation of size”, Journal of Computer Science, Vol 5, pp 451-455, 2009.

[3]. Boehm B. W., C. Abts and S. Chulani, 2000. Software Development Cost Estimation Approaches - A Survey, University of Southern California Centre for Software Engineering, Technical Reports, USC-CSE-2000-505.

[4]. MacDonell, S.G. and A.R. Gray, 1997. A Comparison of Modeling Techniques for Software Development Effort Prediction, in Proceedings of the 1997 International Conf. on Neural Information Proceedings and Intelligent Information Systems, Dunedin, New Zealand, Springer-Verlag, pp: 869-872.

[5]. Oscar Castillo and Patricia Melin “Type-2 fuzzy logic: Theory and Application” Springer, 2008.

[6]. Lotfi Zadeh A., 1994. Fuzzy Logic, Nrural Networks and Soft Computing, Communication of ACM, 37(3): 77-84.

[7]. Lotfi Zadeh A., 2001. The Future of Soft Computing in Joint 9th IFSA World Congree and 20th NAFIPS International Conference, Vancouver, Canada.

[8]. Fei Z., and X. Liu, 1997. F-COCOMO: Fuzzy Constructive Cost Model in Software Engineering, in Proceedings of the IEEE International Conference on Fuzzy Systems, IEEE Press, New York, pp: 331-337

[9]. Ryder J., 1998. Fuzzy Modeling of Software Effort Prediction, in Proceedings of IEEE Information

Technology Conference, Syracuse, NY.

[10]. Menzies,T.,2005. Promise software engineering repository. http://promise.site.uottawa.ca/SERepository/

[11]. J.M. Mendel, G.C. Mouzouris, Designing fuzzy logic systems, IEEE Trans. Circuits Systems -II: Analog Digital Signal Process. 44 (November 1997) 885-895.

[12]. J. M. Mendel and R. I. John, “”Type-2 Fuzzy Sets Made Simple,” IEEE Trans. on Fuzzy Systems, vol. 10 (April 2002), pp. 117-127.

[13]. Qilian Liang, Jerry M. Mendel. “Interval type-2 fuzzy logic systems: theory and design,” IEEE Transactions on Fuzzy Systems, vol. 8, no. 5, pp. 535-550, October 2000.

[14]. .M. Mendel, Uncertain Rule-based Fuzzy Logic Systems, Prentice-Hall PTR, Upper Saddle River, NJ, 2001. [18] V. Novák, I. Perfilieva, The Use of Interval Type-2 Fuzzy Logic as a General Method for Fuzzy Systems, IEEE World Congress on Computational. Intelligence., pp, 249 - 253.

[15]. L.A. Zadeh, “The Concept of a linguistic variable and its Applications to Application Reasoning-1,” Information Science, Vol. 8, No. 3, pp. 199-249, 1975.

[16]. Steve McConnell. Rapid development: taming wild software schedules. Microsoft Press, 1996.

[17]. Iman Attarzadeh, Siew Hock Ow. Improving the Accuracy of Software Cost Estimation Model Based on a New Fuzzy Logic Model, World Applied Science, Vol. 8(2), pp. 177-184, 2010.

Fuzzy Logic based Inflow Prediction Model fFor Reservoirs of Mahanadi Basin

Ishtiyaq Ahmad[#1], Ashish Patel[#2], Dr. M.K.Verma[#3], Dr. R.K.Tripathi[#4]

[#1]*Department of Civil Engineering., IT, GGV, Bilaspur (C.G.), India*
ia_friends2000@yahoo.co.in
[#2]*Sub-DIC Bio-Informatics Centre, National Institute of Technology, Raipur (C.G.), India*
ashish111ppp@gmail.com
[#3]*Department of Civil Engineering, National Institute of Technology, Raipur (C.G.), India*
mkseem670@gmail.com
[#4]*Department of Civil Engineering, National Institute of Technology, Raipur (C.G.), India*
rajesh_tripathi64@yahoo.co.in

***Abstract*— Rainfall- inflow relationships are widely used in many hydrological studies, particularly for optimal reservoir operation policy. Due to uncertainty in rainfall; rainfall – inflow relationships should be based on such an approach which best describes the uncertainties. One such emerging technique is fuzzy logic approach which is based on uncertainties. In the present paper fuzzy logic inflow prediction model has been developed for the reservoirs of Mahanadi basin in Chhattisgarh State, India.**

***Keywords*— Fuzzy logic, regression, centroid defuzzification.**

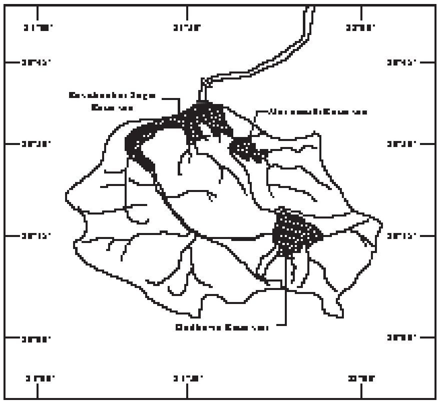
I. INTRODUCTION

Among the various components of a water resources development project, reservoirs are the most important. The success of any reservoir system's planning and operation solely lies with the accuracy of estimation of inflows. The use of regression analysis is the simplest and frequently used approach in the development of rainfall – inflow models [1]. Since the rainfall and inflow measurements show haphazard fluctuations around an average value, the appropriate methodology should be based on uncertainty techniques. One such technique is fuzzy logic approach which is based on uncertainties. The objective of this paper is to propose a fuzzy inflow forecasting model approach as an alternative to regression approach.

II. STUDY AREA

The Mahanadi is one of the important river systems in the country and is one of the twelve major river basins in India. Mahanadi Reservoir Project (MRP) complex is situated in Dhamtari district of Chhattisgarh State and harnesses the water of Mahanadi River, an east flowing river. MRP complex comprises of five reservoirs. MRP consists of two basins, Mahanadi basin and Pairi basin. Ravishankar Sagar, Murumsilli and Dudhawa reservoirs are constructed on Mahanadi River in Mahanadi basin. The catchment areas of Ravishankar Sagar, Murumsilli and Dudhawa reservoirs are 3670, 484 and 621 square kilometer respectively [2,6]. The project serves the purpose of Municipal and Industrial supply and irrigation supply; hence it is multipurpose multi-reservoir system [1]. The index map of Mahanadi Basin is shown in Fig. 1.

Fig. 1: Index map of Mahanadi Basin



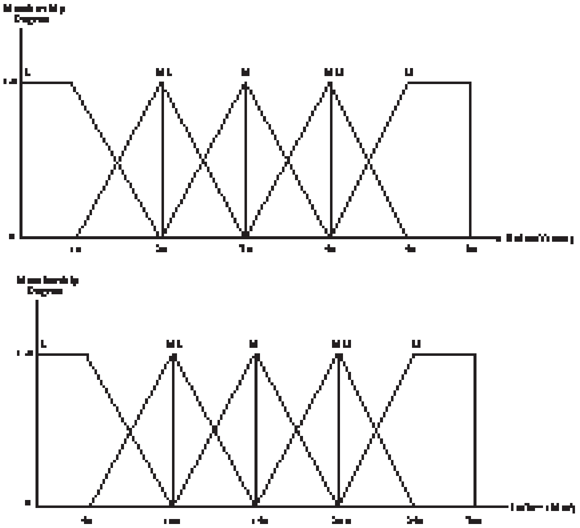
III. FUZZY LOGIC

Fuzzy Logic, a form of logic in which variables can have degrees of truthfulness or falsehood represented by a range of values between 1 (true) and 0 (false). With fuzzy logic, the outcome of an operation can be expressed as a probability rather than as a certainty. For example, in addition to being either true or false, an outcome might have such meanings as probably true, possibly true, possibly false, and probably false. Fuzzy logic is all about the relative importance of precision. It was first introduced in 1965 by Lofti Zadeh at the University of California, Berkeley. Fuzzy logic allows for set membership [3] values to range (inclusively) between 0 and 1, and in its linguistic form, imprecise concepts like "slightly", "quite" and "very". Specifically, it allows partial membership in a set. It is related to fuzzy sets and possibility theory. Fuzzy logic starts with the concept of a fuzzy set [7]. Imprecise, subjective and non-commensurable objectives for reservoir operation can be addressed in an easily interpreted form by Fuzzy logic based optimization approaches.

II. FUZZY MODEL

Many researchers have applied the fuzzy approach to various engineering problems (Mamdani, 1974; Pappis and Mamdani 1977; Sen 1998). The fuzzy logic approach is based on the linguistic uncertain expression rather than numerical uncertainty measures. The basis of fuzzy logic is to consider hydrological variables in a linguistically uncertain manner, in the forms of subgroups, each of which is labeled with successive fuzzy word attachments such as "low", "medium", "high", etc. For instance, in this present paper rainfall and inflow variables are considered as five partial subgroups, namely low (L), medium low (ML), medium (M), medium high (MH) and high (H). A small number of fuzzy subgroups selection leads to unrepresentative predictions whereas a large number imply unnecessary calculations. In the preliminary stage the number of subgroups is selected as five. Five subgroups in each variable imply that there are 5 x 5 = 25 different partial relationship pairs that may be considered between the rainfall and inflow variables. However, many of these relationships are not actually possible. For instance, if the rainfall is "High" it is not possible to state that the inflow is "Low" or even "Medium". Fig. 2 shows the relative positions of the fuzzy words (i.e. "low", "medium low", medium", "medium high", "high") employed to the model. Each one of the middle fuzzy words is shown as a triangle with the maximum membership degree at its apex. The most left and right fuzzy words, namely, "Low" and "High" are represented by trapeziums. It is significant to consider that neighbouring fuzzy subsets interfere with each other providing the fuzziness in the modeling.

Fig 2: Fuzzy subgroups of rainfall and inflow



Since the rainfall-inflow relationship [4], in general, has a direct proportionality feature, it is possible to write the following five rule-bases for the description of fuzzy rainfall-runoff modeling:

Rule 1 → IF Rainfall is L (Low) THEN Inflow is L (Low) or
Rule 2 → IF Rainfall is ML (Medium Low) THEN Inflow is ML (Medium Low) or
Rule 3 → IF Rainfall is M (Medium) THEN Inflow is M (Medium) or
Rule 4 → IF Rainfall is MH (Medium High) THEN Inflow is MH (Medium High) or
Rule 5 → IF Rainfall is H (High) THEN Inflow is H (High)

In order to show two applications of fuzzy inference, in Fig. 3 two rules of the rainfall-runoff relationship are shown with membership degree, μ. For a given rainfall intensity, i=150 mm both L and ML fuzzy subgroups of rainfall variable are triggered. The consequent part of each runoff variable appears as a truncated trapezium for each rule on the right hand side in Fig. 3. The overlapping of these two truncated trapezium indicates the combined inference from these two rules as in the lower part of the same figures, which is represented in Fig. 3 with relevant numbers. In this figure A1and A2 indicate triangular sub areas in the fuzzy inference.

For hydrologic design purposes, it is necessary to deduce from these combined fuzzy subgroups a single value, which is referred to as "defuzzification" [5] in the fuzzy system terminology. The purpose of defuzzification is to convert the final fuzzy set representing the overall conclusion into a real number that, in some sense, best represents this fuzzy set. Although there are various defuzzification methods the most common method is centroid defuzzification [5]. In general, given a fuzzy set with membership degree μ(x) defined on the interval [b, c] of variable x, the centroid defuzzification prediction is defined as:

$$\bar{x} = \frac{\int_b^c x\mu(x)dx}{\int_b^c \mu(x)dx}$$

By applying this formula to the fuzzy inference set in Fig 3, it is possible to obtain a defuzzification value by numerical calculation as,

$$\bar{x} = \frac{\sum_{i=1}^{2} A_i x_i}{\sum_{i=1}^{2} A_i}$$

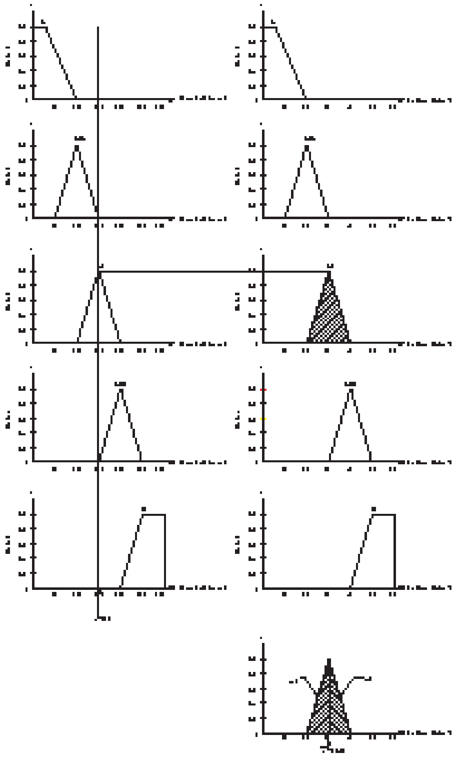
Where x = centroidal x- axis value for triangles and rectangles

A = area of triangle and rectangles

According to above equation the single defuzzified value is calculated as,

$$\bar{x} = \frac{5\times 26.67 + 5\times 33.33}{5+5} = 30$$ which is shown in the Fig. 3.

Fig 3: Rainfall- runoff relationship rules with fuzzy inference set

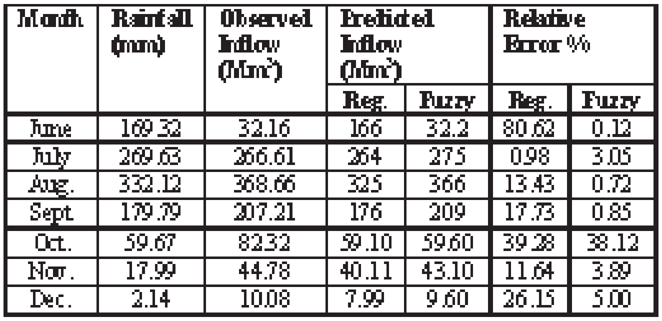


RESULTS AND CONCLUSION

The application of two methodologies (regression and fuzzy) is performed for Ravishankar Sagar, Murumsilli and Dudhawa reservoirs of MRP complex for the average rainfall-inflow data of 17 years for the monsoon months of June-December. The monthly inflow prediction result is shown through Table 1, 2 and 3. The relative error with respect to observed runoff for each runoff prediction through classical regression and fuzzy model is also presented. It is observed that fuzzy logic model prediction yield less relative error as compared to regression model. To conclude it is suggested that in case of rainfall-inflow records existence; it is preferable to apply fuzzy logic approach for inflow estimations from given inflow measurements as uncertainties can be best addressed by fuzzy logic.

TABLE I
Monthly Inflow Prediction Results For Ravishankar Sagar Reservoir

Fig 4: Monthly inflow prediction results for Ravishankar Sagar Reservoir



Month	Rainfall (mm)	Observed Inflow (Mm³)	Predicted Inflow (Mm³)		Relative Error %	
			Reg.	Fuzzy	Reg.	Fuzzy
June	169.32	32.16	166	32.2	80.62	0.12
July	269.63	266.61	264	275	0.98	3.05
Aug.	332.12	368.66	325	366	13.43	0.72
Sept	179.79	207.21	176	209	17.73	0.85
Oct.	59.67	82.32	59.10	59.60	39.28	38.12
Nov.	17.99	44.78	40.11	43.10	11.64	3.89
Dec.	2.14	10.08	7.99	9.60	26.15	5.00

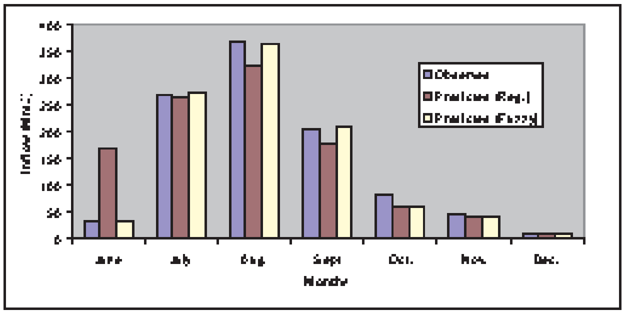
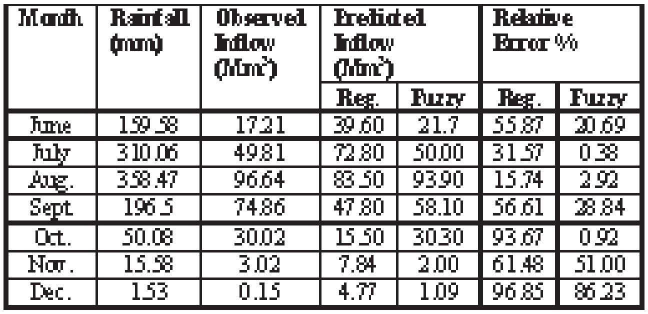


TABLE II
Monthly inflow prediction results for murumsilli reservoir

Fig 5: Monthly inflow prediction results for Murumsilli Reservoir



Month	Rainfall (mm)	Observed Inflow (Mm³)	Predicted Inflow (Mm³)		Relative Error %	
			Reg.	Fuzzy	Reg.	Fuzzy
June	159.58	17.21	39.60	21.7	55.87	20.69
July	310.06	49.81	72.80	50.00	31.57	0.38
Aug.	358.47	96.64	83.50	93.90	15.74	2.92
Sept	196.5	74.86	47.80	58.10	56.61	28.84
Oct.	50.08	30.02	15.50	30.30	93.67	0.92
Nov.	15.58	3.02	7.84	2.00	61.48	51.00
Dec.	1.53	0.15	4.77	1.09	96.85	86.23

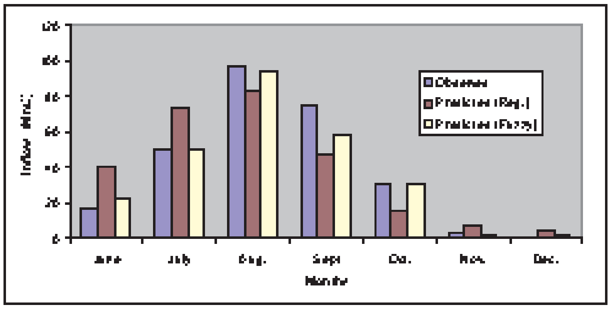
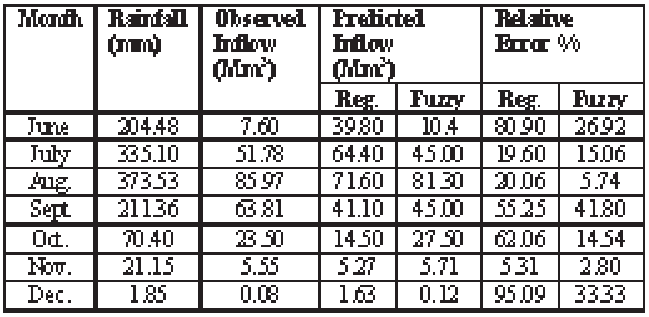


TABLE III
Monthly inflow prediction results for Dudhawa Reservoir
Fig 6: Monthly inflow prediction results for Dudhawa Reservoir



Month	Rainfall (mm)	Observed Inflow (Mm³)	Predicted Inflow (Mm³)		Relative Error %	
			Reg.	Fuzzy	Reg.	Fuzzy
June	204.48	7.60	39.80	10.4	80.90	26.92
July	335.10	51.78	64.40	45.00	19.60	15.06
Aug.	373.53	85.97	71.60	81.30	20.06	5.74
Sept	211.36	63.81	41.10	45.00	55.25	41.80
Oct.	70.40	23.50	14.50	27.50	62.06	14.54
Nov.	21.15	5.55	5.27	5.71	5.31	2.80
Dec.	1.85	0.08	1.63	0.12	95.09	33.33

REFERENCES

[1] Z.Sen. (2003). Fuzzy Awakening in rainfall-runoff modeling. Nordic Hydrology. [Online],Vol. 35(1).pp. 31-34. Available: http://www2.er.dtu.dk/nordichydrology.

[2] Verma, M. K. (2000). Development of Strategies for Optimal Utilization of Water Resources of Mahanadi Reservoir project Complex. Ph. D. Desertation. Devi Ahilyabai University, Indore (M.P.).

[3] G. J. Kilr and T. A. Folger, "Fuzzy Sets, Uncertainty and Information", Prentice hall of India Pvt. Ltd., New Delhi, 2004.

[4] Hoyt W.G. (1986). Studies of relation of rainfall and runoff in United State U.S. Geological Survey Water Supply Paper, 772, 301.

[5] J. T. Ross, "Fuzzy Logic with Engineering Application", McGraw-Hill Inc. New York, 1995.

[6] Central Water and Power Station, Khadakwasla, Pune (India). "Development of Decision Support System for Mahanadi Project", Final Report, Feb., 1994.

[7] L. A. Zadeh. (1965). Fuzzy sets. Information and Control. Vol. 8. pp. 338-353.

[8] Mamdani, E. H., (1974): Application of fuzzy algorithms for simple dynamic plant, Proc. IEE, 121, pp1585-1588.

[9] Pappis, C. P. and Mamdani, E. H., (1977): A fuzzy controller for a trafic junction, IEEE Trans. Syst. Ma Cybern, Vol. 7, No. 10, pp 704-717.

[10] Z. Sen.(1998): Fuzzy algorithm for estimation of solar irradiaiation from sunshine duration, Solar Energy, Vol. 63 , No. 1, pp 39-49.

Classification of Gene Expression Data Using a Single Layer Single Neuron Neural Network

Sashikala Mishra[1], Madhusmita Nayak[2], Kailash Shaw[3], Babita Majhi[4] and Ganapati Panda[5]

[1,2,4]Dept. of Computer Science/Information Technology, Institute of Technical Education and Research, Siksha 'O' Anusandhan University, Bhubaneswar

[1]sasi.iter@gmail.com, [2]madhusmita.nayak0@gmail.com, [4]babita.majhi@gmail.com

[3]Dept. of Computer Science and Engg., Gandhi Engineering College, Bhubaneswar, kailash.shaw@gmail.com

[5]School of Electrical Sciences, Indian Institute of Technology, Bhubaneswar, ganapati.panda@gmail.com

ABSTRACT

The paper proposes a low complexity single layer single neuron neural network (SLSNNN) classifier of microarray data. Two different feature extraction schemes : factor analysis (FA) and principal component analysis (PCA) are proposed to be used in the classifier. The proposed model employs a simple gradient based technique for its training. The performance of model evaluated through simulation study of two real life micro array data exhibits that the SLSNNN-FA provides improved performance compared to the SLSNNN-PCA model.

***Keywords*—** Microarray gene expression data, feature reduction, principal component analysis, factor analysis, classification and functional link artificial neural network.

I. INTRODUCTION

Recent advances in bioinformatics and high-throughput technologies such as microarray analysis have brought a revolution in our understanding of the molecular mechanisms underlying normal and dysfunctional biological processes. Microarray studies and other genomic techniques also stimulate the discovery of new targets for the treatment of disease which is aiding drug development, immunotherapeutics and gene therapy. It also helps the scientists and physicians in understanding of the pathophysiological mechanisms, in diagnoses and prognoses and choosing treatment plans. In a classification study using the training samples representing different classes and the particular choice of the classifier design approach, first the classifier is trained and then the classifier is used to predict the class of new or unseen samples. An important issue in classifier design is selection of proper features.

Recently a two stage classification method was proposed in [1] for microarray data. In the first stage a pre defined number of genes was selected and then passed to the second stage for classification. In [2] a stomach cancer detection system based on artificial neural network (ANN) and discrete cosine transform (DCT) has been proposed. In this investigation a optimal number of DCT coefficients have been computed and an optimal structure of the ANN classifier has been suggested. Identification of significant factors influencing diabetes control, has been made and has been successfully applied to a working patient management system [3]. Correlation based feature selection together with the machine learning algorithm and support vector machine for classification of cancer patient has been studied[4]. A stable classification method known as minimum error distance threshold has been reported for microarray data[5]. Bayesian linear model for microarray data classification based on a prior distribution is proposed in [6]. Combination of genetic algorithm and support vector machines has been employed for multiclass cancer identification [7]. The literature survey reveals that many recent and complex techniques have already been applied for classification of gene expression data. In this paper a simple but efficient single layer and single neuron nonlinear structure is selected as a classifier. Its performance for gene expression data has been evaluated and conclusion has been made.

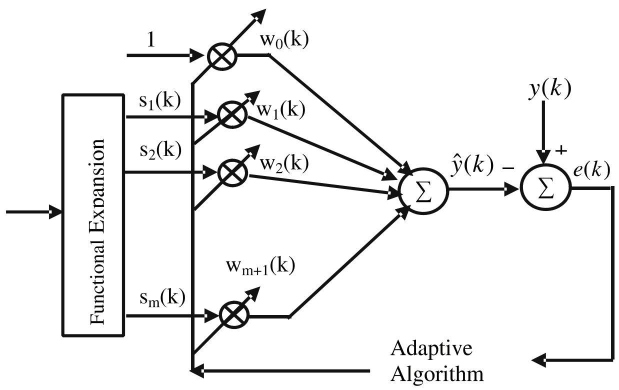
The rest of the paper is organized as follows : Section II deals with the development of single layer single neuron neural network (SLSNNN) based classifier. Relevant data collection and data reduction using principal component analysis and factor analysis are carried out in Section III. The classification task using SLSN model is dealt in Section IV. The results obtained from simulation studies using real life data are presented in Section V. Finally Section VI provides the conclusion of the investigation.

II. SINGLE LAYER SINGLE NEURON NEURAL NETWORK CLASSIFIER

The SLSNNN [8] is a single neuron, single layer architecture capable of imparting nonlinear decision boundaries. In this network the input pattern is mapped to a nonlinear function with more elements which are nonlinearly related to the input elements. The block diagram of a SLSNNN based training system is shown in Fig. 1.

The input signal x(k) is expanded to nonlinear values using trigonometric relations (sine and cosine). These expanded inputs are adaptively weighted by least mean square based learning algorithm. The output obtained is compared with the training signal and the error generated is used to change the weights of the model using gradient based least mean square algorithm [9]. The training is completed after the squared error is reduced to the possible minimum value. At this stage the SLSNNN architecture represents the classification model.

Fig. 1 Basic structure of SLSNNN Classifier



II. FEATURE REDUCTION

A. *Factor analysis(FA)*

Factor Analysis [10] is mainly a data reduction method used to reduce a set of observed variables to a set of unobserved or latent variables. The factor analysis has reduced lung cancer data set from 197 X 581 dimensions to 197 X 197 dimensions and iyer data set from 517 x 11 dimensions to 517 x 4 dimensions. Some samples of the factor score for iyer data set is given in Table 1.

B. *Principal component analysis(PCA)*

Principal component analysis [11] is a technique that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. Using this technique the lung cancer dataset of dimension 197 X 581 is reduced to a dimension of 197 X 81 dimension and iyer data of dimension 517 x 11 reduced to 517 x 3 dimension. Some representative principal components of the data sets are listed in Table 2.

Table 1

Factor score using FA for iyer dataset

-0.06399	-0.26223	0.281123	0.229748
-0.08148	-0.18354	0.20707	0.125544
-0.12239	-0.21501	0.227557	0.161897
-0.25798	-0.32793	0.31913	0.357807
-0.22455	-0.16856	0.314929	0.330515
-0.25153	-0.38564	0.409566	0.406296
-0.28001	-0.19357	0.401695	0.354626
-0.38019	-0.42404	0.500239	0.400831
-0.30062	-0.30521	0.523015	0.454973
-0.22757	-0.14696	0.386858	0.301932
-0.60051	-0.0975	0.494963	0.379598
-0.98964	-0.17252	0.731195	0.447304

Table 2

Principal components using PCA for iyer dataset

-1.76117	-0.5604	0.901152
-2.37866	-1.00503	1.184685
-2.18099	-0.85419	1.090394
-1.0742	0.121962	0.257325
-1.33188	-0.55664	0.608361
-0.73354	0.255268	0.113902

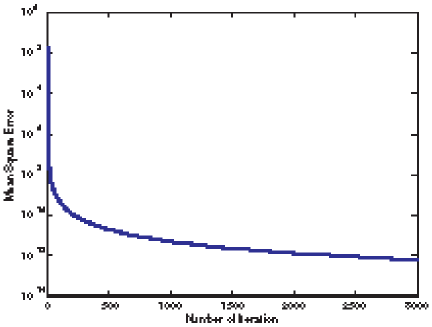
IV. CLASSIFICATION

The reduced features of lung cancer and iyer microdata sets are used as the inputs to the proposed classifier. The lung cancer and iyer data sets contain four and eleven groups/classes of patients. The class numbers are used as the training signal for the classifier. The SLSNNN classifier designed after training is employed to group the patients.

In this model the input vector contains 197 and 4 features with FA and 81 and 3 with PCA for lung cancer and iyer dataset respectively. Each feature is expanded to three trigonometric terms and then fed to the SLSNNN model. The four and eleven outputs obtained from the SLSNNN model in case of lung cancer and iyer dataset are compared with training signal to produce error signal. The mean square error is used as the cost function. The epoch based learning of weights is carried out using the LMS algorithm. The classification results obtained from test operation are shown in Tables 3-6. The convergence characteristics of the classifier during training operation are shown in Figs. 2-5. It is observed that in all cases good convergence performance is obtained. Further it is seen that the PCA-SLSNNN model offers better convergence compared to FA-SLSNNN model.

Fig 2 Convergence characteristics of FLANN model using lung cancer data with FA

Table 3 Classification results obtained from testing using



FLANN and FA for Lung Cancer data

Table 4

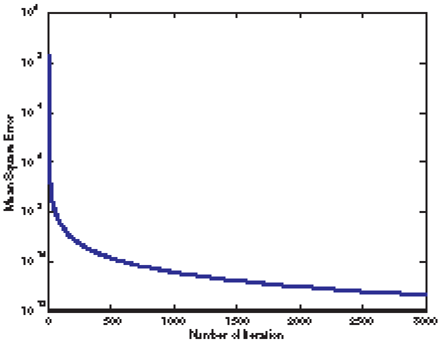


Fig. 3 Convergence characteristics of FLANN model using lung cancer data with PCA

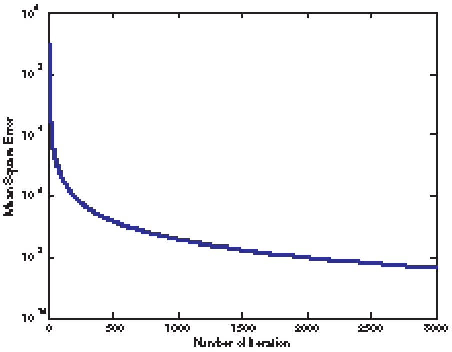


Fig. 4 Convergence characteristics of FLANN model using iyer data with FA

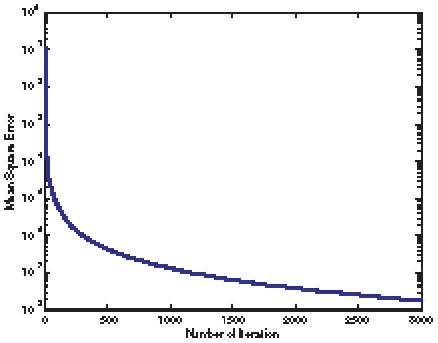


Fig. 5 Convergence characteristics of FLANN model using iyer data with PCA

Classification results obtained from testing using SLSNNN and PCA for Lung Cancer

Classified Cluster4 observations	Cluster1	Cluster2	Cluster3	
Class1	**13**	3	1	2
Class2	1	**7**	2	1
Class3	1	3	**12**	2

Table 5 Classification results obtained from testing using FLANN and FA for Iyer Data

Table 6

Classified Cluster4 observations	Cluster1	Cluster2	Cluster3	
Class1	**14**	2	3	2
Class2	1	**7**	1	1
Class3	0	2	**9**	2

Classification results obtained from testing using FLANNNN and PCA for Iyer Data

Table 7

Classified observa-tions	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
Class1	12	0	1	1	1	0	1	0	0	1	0
Class2	1	16	1	1	1	0	2	0	1	0	0
Class3	1	0	12	1	3	0	1	0	0	1	0
Class 4	2	2	1	15	0	0	0	2	1	0	0
Class 5	1	0	0	0	14	1	0	0	1	1	1
Class 6	0	1	2	1	0	4	2	0	1	2	0
Class 7	1	0	0	0	0	0	10	0	1	2	0
Class 8	0	1	1	0	0	0	2	11	0	0	0
Class 9	0	0	1	0	1	0	1	0	13	0	2
Class 10	1	0	1	0	0	0	0	0	1	12	2
Class 11	1	0	0	1	0	1	1	0	1	1	15
Cumulative	20	20	20	20	20	6	20	13	20	20	20

Comparison of classification results

Classified observa-tions	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
Class1	10	1	0	1	1	0	1	1	0	0	0
Class2	1	13	1	0	1	0	2	1	1	2	0
Class3	2	1	10	2	3	0	1	0	0	2	0
Class 4	2	2	0	12	0	0	0	1	1	0	0
Class 5	1	0	1	1	12	1	2	1	1	1	1
Class 6	0	1	3	0	0	2	1	0	1	0	0
Class 7	1	0	1	0	1	0	8	0	2	2	2
Class 8	1	1	1	1	1	1	2	6	1	3	1
Class 9	0	0	1	1	1	0	2	0	11	0	2
Class 10	1	0	1	1	0	1	0	3	1	10	2
Class 11	1	1	1	1	0	1	1	0	2	0	12
Cumulative	20	20	20	20	20	6	20	13	20	20	20

V. CONCLUSION

The paper proposes an efficient single layer ANN classifier using trigonometric functional scheme of input features. The input features are extracted using FA and PCA techniques. The classifiers are designed using simple LMS based technique. The simulation results exhibit superior performance in case of FA based feature extraction compared to the PCA one. Further it is seen that if the input data is less as in case of iyer data classification performance of both proposed model is poor. Improved classifiers need to be designed in such cases.

Name of data set	Percentage of accuracy	
	FLANN-FA	FLANN-PCA
Lung cancer	70%	67%
Iyer	66%	51.27%

REFERENCES

[1] T.T. Wong and C. HanHsu, “Two stage classification methods for microarray data”, Expert system with applications, vol. 34, pp. 375-383, 2008.

[2] A M. Sarhan, “Cancer classification based on microarray gene expression data using DCT and ANN”, Journal of theoretical and applied information technology, pp. 208-216, 2009.

[3] Y. Huang, P. McCullagh, N. Black and R. Harper, “Feature selection and classification model construction on type 2 diabetic patients' data”, Artificial intelligence in medicine, Elsevier, vol. 41, pp. 251-262, 2007.

[4] Y. Wang,I. V. Tetko, M. A. Hall, E. Frank, A. Facius, K. F. X. Mayer and H. W. Mewes, “Gene selection from microarray data for cancer classification-a machine learning approach”, Computational biology and chemisty, vol. 29, pp. 37-56, 2005.

[5] C. S Li and C Cheng, “Stable classification with applications to microarray data”, Computational statistics and data analysis, Elsevier, vol. 47, pp. 599-609, 2004.

[6] D. Hernandez-Lobato, J. M. Hernandez-Lobato and A. Suarez, “Expectation propagation for microarray data classification”, Pattern recognition letters, Elsevier, vol. 31, pp. 1618-1626, 2010.

[7] S. Peng, Q. Xu, X. B. Ling, X. Peng, W. Du and L. Chen, “Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines”, Federation of European biochemical societies letters, vol. 555, pp. 358-362, 2003.

[8] J. C. Patra, R. N. Pal, B. N. Chatterji, G. Panda, “Identification of nonlinear dynamic systems using Functional Link Artificial Neural Networks”, IEEE Trans. Syst., Man, Cybern. B, vol. 29, issue 2, pp. 254-262, 1999.

[9] B. Widrow, Adaptive signal processing, PHI Publication.

[10] Subhash Sharma, Ajith Kumar, *Cluster analysis and factor analysis,* University of South Carolina, Arizona State University.

[11] S. Lakhina, S. Joseph and B. Verma, “feature reduction using principal component analysis for effective anomaly based intrusion detection on NSL-KDD”, International Journal of Engineering Science and Technology, vol. 2, issue 6, pp. 1790-1799, 2010.

An Enhanced Association Rule Mining Using Apriori

[1]Mohammad Kamran, [2]S. Qamar Abbas, [3]Mohammad Rizwan Baig,
[1] Research Scholar, Integral University, Kursi Road, Lucknow, India
[2] Professor, Ambalika Institute of Management & Technology, Lucknow, India
[3] Professor, Integral University, Lucknow, India
e-mail : mkamran_lko@hotmail.com

Abstract-

The explosive growth of business, scientific and government databases sizes has created a need for new generation tools and techniques for automated and intelligent database analysis. One of the important problems in data mining is discovering association rules from databases of transactions where each transaction consists of a set of items. Apriori algorithm is the best-known association rule algorithm. The most time is consumed by scanning the database repeatedly in apriori algorithm. In this paper, we proposed an algorithm to enhance the efficient of existing algorithm by reducing the times of scanning database. The proposed approach reduces not only the scanning of datasets but also the overall execution time of the algorithm.

Keywords— **Data Mining, Association Rules, Apriori**

I. INTRODUCTION

The rapid improvements in hardware/software devices have enabled markets, business centers, and production units to collect and store relevant data easily and efficiently. Today, all departmental stores, large organizations, manufacturers and even small business companies possess plenty of data reflecting their transactions, operations, or business-relevant activities. Consequently, the complexity and volume of data increase day-to-day. The growth in data complexity requires managers and engineers to be equipped with sophisticated methods to be able to benefit from the valuable knowledge included within the data. One of the most important data mining approaches is the association rule mining which is used to detect hidden affinity patterns in the datasets [1].

The most important step in mining association is generation frequent itemsets. In algorithm apriori the most time is consumed by scanning the database repeatedly. The running time of the algorithm by reducing the times it scans the database far and away is proposed in this paper.

II. ASSOCIATION RULES

Association rule mining is the process of finding patterns, associations and correlations among sets of items in a database. The association rules generated has an antecedent and a consequent. An association rule is a pattern of the form *X & Y ⇒ Z [support, confidence]*, where X, Y, and Z are items in the dataset. The left hand side of the rule X & Y is called the antecedent of the rule and the right hand side Z is called the consequent of the rule. This means that given X and Y there is some association with Z. Within the dataset, confidence and support are two measures to determine the certainty or usefulness for each rule. Support is the probability that a set of items in the dataset contains both the antecedent and consequent of the rule or $P(X \cup Y \cup Z)$. Confidence is the probability that a set of items containing the antecedent also contains the consequent or $P(Z| X \cup Y)$. Typically an association rule is called strong if it satisfies both a minimum support threshold and a minimum confidence threshold that is determined by the user [2]. An important but simple algorithm for finding frequent itemsets is the Apriori Algorithm [2].

III. APRIORI ALGORITHM

The Apriori algorithm is a basic algorithm for finding frequent itemsets from a set of data by using candidate generation. Apriori uses an iterative approach known as a level-wise search because the *k*-itemsets is used to determine the $(k+1)$-itemsets. The search begins for the set of frequent 1-itemsets denoted L_1. L_1 is then used to find the set of frequent 2-iemsets, L_2. L_2 is then used to find L_3 and so on. This continues until no more frequent *k*-itemsets can be found [2].

IV. RELATED WORK

Algorithm APRIORI [3] is one of the oldest and most versatile algorithms of Frequent Pattern Mining (FPM). With sound data structures and careful implementation it has been proven to be a competitive algorithm in the contest of Frequent Itemset Mining Implementations (FIMI) [4]. The performance of Apriori was beaten most of the time by sophisticated DFS algorithms, such as lcm [5], nonordfp [6] and eclat [7].

V. PROPOSED WORK

Based on the frequency of collection of Apriori search algorithm using a layer of the iterative approach is simple and clear, not complex theoretical analysis, but also easy to realize. To generate maximum length for the frequency of collection K, K to the database scans. When the database storing a large number of data services, the limited memory capacity, the system I/O load considerable time scanning the database will be a very long time, so efficiency is very low. To improve the effectiveness, the various studies are mainly towards reducing the amount of computation and by less the number of scanning the database to improve, we introduced a method of improving the algorithm Apriori. The method can be use to find out all the frequent itemsets without scanning DB so many times.

i. Count the occurrence of each item by initially scanning the database record one by one

ii. Suppose I_k and I_m appeared are the items of one record of itemset I=(I_1 ,I_2 ,I_3......I_m) and its probability is $Z_{km.}$
min(Z_k , Z_m)=Z_{km} =Z_k*Z_m ,

iii. Z_{km} is the minimum of the Z_k and Z_m,; if I_k and I_m is total correlation and Z_{km} is Z_k *$Z_{m;}$ if I_k and I_m is total independent
The methodology for probability $Z_{km\ is}$
Z_{km} =(a*min(Z_k, Z_m)+b*Z_k*Z_m)/(a+b); a+b=1
Where "a" is the probability while I_k and I_m are total correlation, and "b" is the probability while I_k and I_m are total independent,

iv. if Z_{km} is more than the threshold value which the user set, then I_k , I_m are the frequent itemsets.

Let Z_1, Z_2…Z_n are the independent probability of every item I_1, I_2…I_n ,the probability for any two item I_k, I_m(Z_k<Z_m) both appeared in one transaction is Z_{km} .

If I_k and I_m are total non-correlation, then as per the definition of association rules it can be concluded that Z_{km} = Z_k*Z_m , if I_k and I_m are total correlation, then Z_{km} is the minimum of the Z_k and Z_m that is Z_k , so , Z_k*Z_m = Z_{km} = Z_k

Let parameter "a" be the probability which I_k and I_m are total correlation, and parameter "b" for total non-correlation. a+b=1, 0 < a, b < 1, then Z_{km} can be defined as formula below:

$$Z_{km} = a* Z_k + b*Z_k * Z_m \tag{1}$$

1. Create a new array ZFI[n], the original value for each element is 0;
Scanning the database,
Calculating the probability of each itemset I_1,I_2, …, I_n respectively and marked by Z_1, Z_2, …, Z_n .
Let each element of the array ZFI[1], ZFI[2], …ZFI[n] be the Z_1, Z_2 ,…, Z_n .which refer to the probability of each itemset I_1, I_2, …, I_n .
The process for calculating the probability of I_i appearing.
(i) Set the pointer at the beginning of the database
(ii) scan each item value from the record
(iii) skip on to next record
(iv) repeat step (ii) and (iii) until end of the data table
(v) set the initial value of the counter so that it can record no. of items present by incrementing the value by 1 i.e. ZFI[i] = ZFI[i]+1
(vi) repeat steps until the end of the datatable so that No. of items/No. of Records in the datatable

To calculate the probability of itemset I_1,I_2, …, I_n appearing we will repeat all the above procedures.

2. Set a minimum value V_1 for the probability of I_i appearing, if the probability of I_i appearing ZFI[i] is larger than V_1 then itemset I_i is a frequent 1-itemset. So, we get some frequent 1-itemset, let "m" be the number, and ZFI[1],ZFI[j],…ZFI[m] be the probability of 1-itemset appearing respectively.
3. Due to the probability of 1-itemset appearing ZFI[1], ZFI[j], …ZFI[m] ,base on the formula (1), then the probability of any two itemset appeared in one datatable can be evaluated. Set a minimum value V_2 for the probability of I_i and I_j appeared synchronously in one dataset, if the probability is larger than V_2 then itemset I_iI_j is a candidate frequent 2-itemset.otherwise set the value of the probability is zero to predigest the later calculation. Let the element of array ZUI_2 [i] record the value of candidate frequent 2-itemsets.
Let V_2 be the minimum probability for candidate frequent 2-itemset, and V_3 for candidate frequent 3-itemset, V_{k-1} for candidate frequent (k-1)-itemset
Set minimum probability V_{k-1} :
V_{k-1}=a*min(ZFI_{k-1}[1],ZFI_{k-1}[2],…ZFI_{k-1}[m])+b*min(ZFI_{k-1}[1],ZFI_{k-1}[2],…ZFI_{k-1}[m])* max(ZFI_{k-1}[1],ZFI_{k-1}[2],…ZFI_{k-1}[m])
4. Recur the above step 1.2.3., from k=2 to n to calculate the probability of k-itemsets I_1,I_2,… ,I_k appearing in one dataset;
5. Scan the database another time to calculate the support of the candidate frequent itemsets which is the result of step 4.
a) Create a new array DMI[m] with each element's original value is zero. (m =number of candidate frequent itemsets);
b) Read and get the data of the database until the end of the database.
c) If there are itemsets I_i,I_j, … ,I_k in any datatable synchronously and I_i≠ 0, I_j ≠ 0…I_k≠0 ; then the support for I_iI_j … I_k DMI[k]= DMI[k]+1.

Recur the above step (b)(c) to calculate the actual support of every candidate frequent itemsets until the end of the database.

6. Find out the frequent itemsets from the candidate frequent itemsets. If DMI[k] is larger than the minimum support which the user set, then output the frequent itemsets.
Step 5., 6. is used to confirm the probability and support of the candidate frequent itemsets which come out by the method of probability evaluation whether satisfy the request of the user.
7. Output the association rule from the result of the step 6.

VI. IMPLEMENTATION AND ANALYSIS

In order to verify the implementation of our methodology and proposed algorithm efficiency we have selected a transaction database from the departmental store and done a comparative analysis with Apriori algorithm. The total database comprises of 9 transactions and 5 itemsets. The itemsets and the respective code are listed below.

TABLE I

Item	Butter	Diaper	Baby Powder	Bread	Umbrella
Code	B	D	P	R	U

As per our proposed algorithm the process of finding frequent itemsets include 3 steps: Firstly it scans the database to come out the probability of frequent 1-itemsets; Then evaluates the probability of candidate frequent 2-itemsets, 3-itemsets …m-itemsets, based on the probability of frequent 1-itemsets; Last it scans the database another time to confirm the frequent itemsets from the candidate frequent itemsets.

The parameter "a", "b" is 5/9 and 4/9 respectively in the simulation of the algorithm.

TID	Itemset
T100	B,D,U
T200	D,R
T300	D,P
T400	B,D,R
T500	B,P
T600	D,U
T700	B,P
T800	B,D,P,U
T900	B,D,P

Fig. 1 A

Calculated the probability of each itemset to find out frequent 1-itemset at the support was 2/9

Itemsets
B
D
P
R
U

Fig. 1 B

Threshold V2 = (5*(2/9)+4*(2/9)*(7/9))/9
= 146/729

From frequent C1 (fig 1B) to evaluate the probability of candidate frequent 2-itemstes (fig 1C)

Itemsets
B
D
P
R
U

Fig. 1 C

2-Itemsets	Probability
{B,D}	438/729
{B,P}	345/729
{B,U}	207/729
{D,P}	345/729
{D,R}	144/729
{D,U}	219/729
{P,U}	195/729

Get the candidate frequent 2-itemsets below (1D):

Fig. 1 D

Set a=5/9,b=4/9;
Threshold V3 = (5*(146/729)+4*(146/729)*(7/9))/9
= 73*146/59049

From frequent C2 to evaluate the probability of candidate frequent 3-itemstes (fig 1E)

Itemsets	Probability
{B,D,P}	69*345/59049
{B,D,R}	73*138/59049
{B,D,U}	73*207/59049
{B,P,R}	69*130/59049
{B,P,U}	69*195/59049
{B,R,U}	69*114/59049
{R,D,P}	73*130/59049
{P,R,U}	65*114/59049
{D,R,U}	73*114/59049
{D,P,U}	73*195/59049

Fig. 1 E

Get the candidate frequent 3-itemsets below (1F):

3-Itemsets	Probability
{B,D,P}	69*345/59049
{B,D,U}	73*207/59049
{D,P,U}	73*195/59049

Fig. 1 F

Fig. 1 Candidate itemset

VII. COMPARISON

The proposed algorithm in this paper and apriori was compared. The algorithm proposed in this paper ahead apriori in reducing the times of scanning the database. It would scan the database k times to find out frequent k-itemsets in apriori while only 2 times in our algorithm by putting forward a concept of candidate frequent itemset. In the table 2 below, it listed the frequent 1-itemsets, 2-itemsets, 3-itemsets for both of the two algorithm respectively. The proposed algorithm in comparison to Apriori would also not miss any frequent itemsets if the probability is likely to occur more in comparison to Apriori which is also evident from the table below.

TABLE II
THE COMPARISON OF THE FREQUENT ITEMSETS

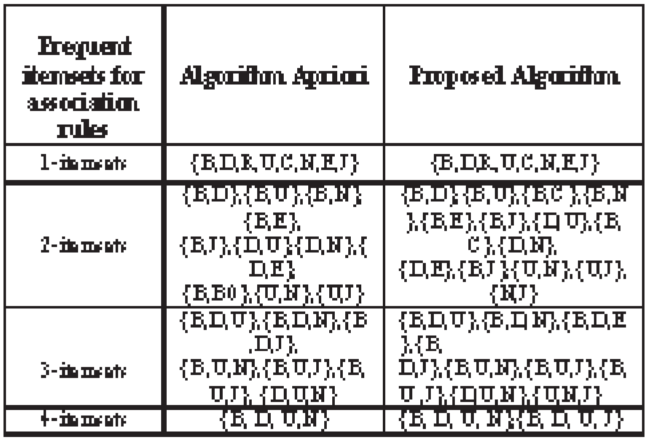
(candidate) frequent itemsets	Apriori Algorithm	Proposed Algorithm
1-itemset	{B,D,P,R,U}	{B,D,P,R,U}
2-itemset	{B,D},{B,P},{B,U},{D,P},{D,R}, {D,U}	{B,D},{B,P},{B,U},{D,P},{D,R}, {D,U},{D,U},{P,U}
3-itemset	{B,D,P},{B,D,U}	{B,D,P},{B,D,U},{D,P,U}

VIII. EXPERIMENTAL AND ANALYSIS

The experiments were conducted on real data sets by the algorithm described in this paper and apriori on Microsoft Server 2003. The datasets was obtained from the UCI repository of machine learning databases. Table 3 listed the candidate frequent 1-itemsets,2-itemsets,3-itemsets of our algorithm and frequent 1-itemsets , 2-itemsets, 3-itemsets of algorithm apriori. We report here only results on some typical data sets, with 10 itesmsets and between 5000 tuples.

The first data set we use has 1000 tuples, then increase 1000 tuples every time. We test the execution time of the algorithms with respect to number of tuples and itemsets. Figure 2 shows that: With tuples from 0 to 5000, when the number of tuples is small, both algorithms have similar performance. However, as the number of tuples grows, the algorithm in this paper takes effect. It keeps the runtime low. In contrast, the algorithm apriori does not scale well under large number of tuples. Figure 3 shows the execution time of both algorithms with respect to the itemsets increased. As the number of itemsets goes up, the runtime of both algorithms has increases and the algorithm in this paper grows slower than algorithm apriori. Algorithm in this paper presents a smoother increasing of runtime than that of algorithm apriori.

TABLE III
THE COMPARISON OF THE FREQUENT ITEMSET IN THE EXPERIMENTS



Frequent itemsets for association rules	Algorithm Apriori	Proposed Algorithm
1-itemsets	{B,D,R,U,C,N,E,J}	{B,D,R,U,C,N,E,J}

2-itemsets	{B,D},{B,U},{B,N}
	{B,E},
	{B,J},{D,U}{D,N},{
	D,E}
	{B,B0},{U,N},{U,J}
		{B,D}{B,U},{B,C },{B,N
		},{B,E},{B,J},{D U},{B,
		C },{D,N},
		{D,E},{B,J }{U,N},{U,J},
		{N,J}

3-itemsets	{B,D,U},{B,D,N},{B ,D,J}, {B,U,N},{B,U,J},{B, U,J}, {D,U,N}	{B,D,U},{B,D N},{B,D,E },{B, D,J},{B,U,N},{B,U,J},{B, U ,J},{D,U,N},{U,N,J}
4-itemsets	{B, D, U,N}	{B, D, U, N}{B, D, U, J}

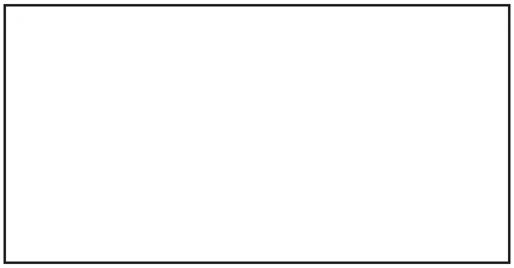


Fig. 2 Performance comparisons with the tuples

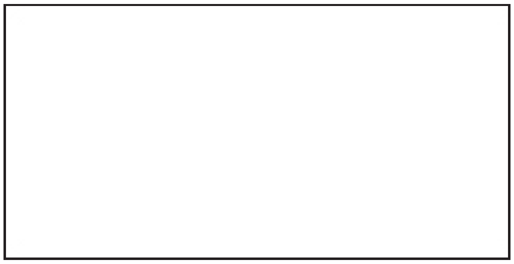


Fig. 3 Performance comparisons with the itemset

IX. CONCLUSION

We have studied the algorithmic aspects of apriori association rule mining. The paper analyzed the inadequacies of the Apriori algorithm, on the basis suggesting for improvement algorithm. This algorithm bases on the structure of apriori algorithm. The main contribution of this work is the performance increase in the process of mining association rules. The method developed in this paper explores efficient mining of association rules by reducing the scanning of database of a dataset. First it scans the database to filter frequent 1-itemsets and get their occurrences respectively. Then it gets the candidate frequent 2-itemset,3-itemsets up to n-itemsets by evaluating their probability on formula (1) in the paper and the result of the first step. Last it scans the database for another time to refine the candidate frequent itemsets to the frequent itemsets. After simulation analysis, we found that proposed algorithm is more excellent than the traditional method Apriori algorithm in the efficiency of performance.

X. FUTURE WORK

A challenge for data mining in general is how to develop a data mining model to overcome loss of information, discover too many obvious patterns and discover knowledge in a specific domain and visualize the rules. Visual data mining is a novel approach to deal with the growing flood of information. By combining traditional data mining algorithms with Visualization Techniques to utilize the advantages of both approaches is also a future topic of interest.

REFERENCES

[1] Agrawal, R., Imielinski, T., and Swami, A., 1993. Mining association rules between sets of items in large relational databases. *Proceedings of ACM SIGMOD international conference on management of data* 1993, 207-216.

[2] Han, Jiawei and Micheline Kamber. 2001. Data Mining: Concepts and Techniques. Morgan Kaufman Publishers.

[3] R. Agrawal, R.Srikant, "Fast Algorithm for Mining Association Rules", proceedings of 20th VLDB conference PP 478-499 September 1994.

[4] B. Goethals and M. J. Zaki. Advances in frequent itemset mining implementations: Introduction to fimi03. In B. Goethals and M. J. Zaki, editors, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'03), volume 90 of CEUR Workshop Proceedings, Melbourne, Florida, USA, 2003

[5] T. Uno, M. Kiyomi, and H. Arimura. Lcm ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets. In B. Goethals, M. J. Zaki, and R. Bayardo, editors, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'04), volume 126 of CEUR Workshop Proceedings, Brighton, UK, 2004.

[6] B. R´acz. nonordfp: An FP-growth variation without rebuilding the FP-tree. In B. Goethals, M. J. Zaki, and R. Bayardo, editors, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'04), volume 126 of CEUR Workshop Proceedings, Brighton, UK, 2004.

[7] L. Schmidt-Thieme. Algorithmic features of eclat. In B. Goethals, M. J. Zaki, and R. Bayardo, editors, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'04), volume 126 of CEUR Workshop Proceedings, Brighton, UK, 2004.

[8] R. Agrawal, J.C. Shafer "Parallel Mining of Association Rules" IEEE Transactions on Knowledge and Data Engineering, Volumes 8, Number 6, PP 962-969, December 1996.

[9] LI Yan-hong. A user-guided association rules mining method and its application. 2004: 320-321.

[10] Y. Aumann and Y. Lindell. A statistical theory for quantitative association rules. KDD'99

[11] R Srikant,Q Vu,R Agrawal. Mining association rules with item constraints. In:Proc of the 1997Third Int'l Conf on Knowledge Discovery in Databases and Data Mining. New port Beach, California: AAAI Press,1997. 67- 73

Text To Speech Synthesize: Emotion in Speech

K.J. Satao [#1], Suresh Kumar Thakur[*2],

*[#1] Information Technology - [*2]Computer Science Engineering, Rungta College of Engineering & Technology, Bhilai, India- Engineering-Rungta College of Engineering & Technology, Bhilai, India*

e-mail : [1]kjsatao@rediffmail.com [2]Sur25011986@gmail.com

Abstract

This paper concerns the process of building an emotional speech synthesizer using the different parameter with speech synthesis system developed. The goal is to build a unit selection synthesis system that can portray emotions with different levels of intensity. To achieve this, the system was based on theoretic frameworks developed by Psychologists to describe emotions

Keywords— **Emotional , synthesizer, intensity, Psychologists.**

I. INTRODUCTION

The overall goal of the speech synthesis research community is to create natural sounding synthetic speech. To increase naturalness, researchers have been interested in synthesising emotional speech for a long time. One way synthesised speech benefits from emotions is by delivering certain content in the right emotion (e.g. good news are delivered in a happy voice), therefore making the speech and the content more believable. Emotions can make the interaction with the computer more natural because the system reacts in ways that the user expects. Emotional speech synthesis is a step in this direction

The implementation of emotions seems straightforward at first but a closer look reveals many difficulties in studying and implementing emotions. The difficulties start with the definition of emotions. Researchers agree that emotions are not as often thought of, just a subjective experience or feeling. An emotion seems to be made up of several components.

- evaluation or appraisal of antecedent event, the meaning of the stimulus for the individual [11]
- physiological change, e.g. pupil dilation or blushing
- action tendencies, flight or fight patterns [9]
- subjective feeling [12]
- expressive behaviour such as non-verbal expressions including facial expression[8] and vocal expression [2]

A. Emotion categories

A common way of describing emotions is by assigning labels to them like emotion denoting words. There are a number of researchers that have compiled lists of emotional words [15]. Of course some of these terms are more central to a certain emotion than others. Also different emotion theories have different methods for selecting such basic emotion words.

In a Darwinian sense the basic emotions have been evolutionarily shaped and therefore can be universally found in all humans. There is a Jamesian extension that expects to find specific patterns for the basic emotions in the central nervous system. The number of basic emotions is usually small. Ekman identified just six basic emotions by looking at the universality of facial expressions. To describe different forms of basic emotions like hot and cold anger, finer grained emotion categories are used which the basic emotions are inclusive of. Scherer [13] suggests that an emotion is more general than another if its appraisal components form a subset of the other emotion. So just ”anger” can be subdivided into ”hot anger” and ”cold anger” depending on the outcomes of particular SEC’s not specified for ”just anger”.

The process of Text-to-Speech is rarely straightforward. Texts are full of heteronyms, numbers, and abbreviations that all require expansion into a phonetic representation. There are many spellings in English which are pronounced differently based on context. Most text-to-speech (TTS) systems do not generate semantic representations of their input texts, as processes for doing so are not reliable, well understood, or computationally effective. As a result, various heuristic techniques are used to guess the proper way to disambiguate homographs, like examining neighbouring words and using statistics about frequency of occurrence

Similarly, abbreviations can be ambiguous. For example, the abbreviation “in” for “inches” must be differentiated from the word “in”, and the address “12 St John St.” uses the same abbreviation for both “Saint” and “Street”. Text to speech systems with intelligent front ends can make educated guesses about ambiguous abbreviations, while others provide the same result in all cases, resulting in nonsensical (and sometimes comical) outputs.

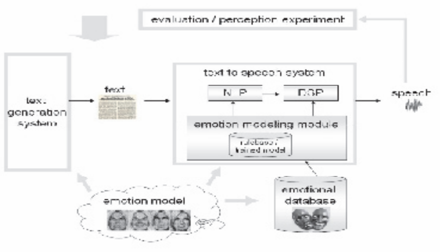


Fig. 1 Aspects of emotional TTS system

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer, and can be implemented in software or hardware. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech. Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diaphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely “synthetic” voice output. The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written works on a home computer. Many computer operating systems have included speech synthesizers since the early 1980s.

II. TEXT TO SPEECH SYNTHESIZER TECHNOLOGIES

The most important qualities of a speech synthesis system are naturalness and intelligibility. Naturalness describes how closely the output sounds like human speech, while intelligibility is the ease with which the output is understood. The ideal speech synthesizer is both natural and intelligible. Speech synthesis systems usually try to maximize both characteristics.

A. *Concatenative synthesis*

Concatenative synthesis is based on the contaminative (or stringing together) of segments of recorded speech. Generally, concatenative synthesis produces the most natural-sounding synthesized speech. However, differences between natural variations in speech and the nature of the automated techniques for segmenting the waveforms sometimes result in audible glitches in the output. There are three main sub-types of concatenative synthesis

B. *Unit selection synthesis*

Unit selection synthesis uses large databases of recorded speech. During database creation, each recorded utterance is segmented into some or all of the following: individual phones, diaphones, half-phones, syllables, morphemes, words, phrases, and sentences. Typically, the division into segments is done using a specially modified speech recognizer set to a “forced alignment” mode with some manual correction afterward, using visual representations such as the waveform and spectrogram.[1] An index of the units in the speech database is then created based on the segmentation and acoustic parameters like the fundamental frequency (pitch), duration, position in the syllable, and neighbouring phones. At runtime, the desired target utterance is created by determining the best chain of candidate units from the database (unit selection). This process is typically achieved using a specially weighted decision tree. Unit selection provides the greatest naturalness, because it applies only a small amount of digital signal processing (DSP) to the recorded speech. DSP often makes recorded speech sound less natural, although some systems use a small amount of signal processing at the point of concatenation to smooth the waveform.

The output from the best unit-selection systems is often indistinguishable from real human voices, especially in contexts for which the TTS system has been tuned. However, maximum naturalness typically require unit-selection speech databases to be very large, in some systems ranging into the gigabytes of recorded data, representing dozens of hours of speech.[2] Also, unit selection algorithms have been known to select segments from a place that results in less than ideal synthesis (e.g. minor words become unclear) even when a better choice exists in the database .[3].

C. *Domain-specific synthesis*

Domain-specific synthesis concatenates pre-recorded words and phrases to create complete utterances. It is used in applications where the variety of texts the system will output is limited to a particular domain, like transit schedule announcements or weather reports.[6] The technology is very simple to implement, and has been in commercial use for a long time, in devices like talking clocks and calculators. The level of naturalness of these systems can be very high because the variety of sentence types is limited, and they closely match the prosody and intonation of the original recordings. Because these systems are limited by the words and phrases in their databases, they are not general-purpose and can only synthesize the combinations of words and phrases with which they have been preprogrammed. The blending of words within naturally spoken language however can still cause problems unless the many variations are taken into account. For example, in non-rhotic dialects of English the “r” in words like “clear” /’kli??/ is usually only pronounced when the following word has a vowel as its first letter (e.g. “clear out” is realized as /?kli???’¨»?t/). Likewise in French, many final consonants become no longer silent if followed by a word that begins with a vowel, an effect called liaison. This alternation cannot be reproduced by a simple word-concatenation system, which would require additional complexity to be context-sensitive.

D. *Formant synthesis*

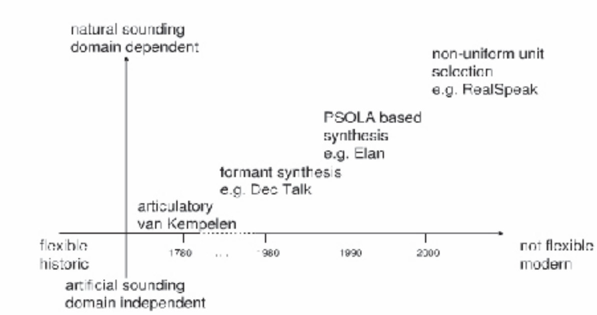
Synthesis does not use human speech samples at runtime. Instead, the synthesized speech output is created using additive synthesis and an acoustic model (physical modelling synthesis) [7]. Parameters such as fundamental frequency, voicing, and noise levels are varied over time to create a waveform of artificial speech. This method is sometimes called rules-based synthesis; however, many concatenative systems also have rules-based components. Many systems based on formant synthesis technology generate artificial, robotic-sounding speech that would never be mistaken for human speech. However, maximum naturalness is not always the goal of a speech synthesis system, and formant synthesis systems have advantages over concatenative systems. Formant-synthesized speech can be reliably intelligible, even at very high speeds, avoiding the acoustic glitches that commonly plague concatenative systems. High-speed synthesized speech is used by the visually impaired to quickly navigate computers using a screen reader. Formant synthesizers are usually smaller programs than concatenative systems because they do not have a database of speech samples. They can therefore be used in embedded systems, where memory and microprocessor power are especially limited. Because formant-based systems have complete control of all aspects of the output speech, a wide variety of prosodies and intonations can be output, conveying not just questions and statements, but a variety of emotions and tones of voice.

E. *Formant synthesis*

Articulatory synthesis refers to computational techniques for synthesizing speech based on models of the human vocal tract and the articulation processes occurring there. The first articulatory synthesizer regularly used for laboratory experiments was developed at Haskins Laboratories in the mid-1970s by Philip Rubin, Tom Baer, and Paul Mermelstein. This synthesizer, known as ASY, was based on vocal tract models developed at Bell Laboratories in the 1960s and 1970s by Paul Mermelstein, Cecil Coker, and colleagues.

Until recently, articulatory synthesis models have not been incorporated into commercial speech synthesis systems. A notable exception is the NeXT-based system originally developed and marketed by Trillium Sound Research, a spin-off company of the University of Calgary, where much of the original research was conducted. Following the demise of the various incarnations of NeXT (started by Steve Jobs in the late 1980s and merged with Apple Computer in 1997), the Trillium software was published under the GNU General Public License, with work continuing as gnu speech. The system, first marketed in 1994, provides full articulatory-based text-to-speech conversion using a waveguide or transmission-line analogy of the human oral and nasal tracts controlled by Care's "distinctive region model".

Fig. 2 Historic Development



III. EMOTION IN SPEECH

Vocal expression has been recognized as one of the primary carriers of affective signals for centuries. Darwin [12] in his pioneering monograph on the expression of emotions in man and animals underlined also this importance of the voice as an affective channel. In recent times studies have been undertaken to find the specific vocal patterns for certain emotions and further, how accurately listeners can infer emotions from the voice. Notably, Scherer [2] has done important work in his studies on acoustic profiles. But many other studies have been undertaken that examine the relationship of vocal expression and emotion. To examine the vocal correlates of emotions, one has to analyse a speech database. The source of the content of such a database has been widely debated [13].

A. *Sources of Emotional Speech*

To obtain authentic emotional speech data is one the biggest challenges in speech and emotion research. The goal is to have a closely controlled corpus with spontaneous speech. Because one cannot have spontaneous speech that is closely controlled, researchers have devised a number of strategies to obtain somewhat natural and spontaneous emotional speech data. The most frequently used and oldest method is to have an actor portray certain emotions .This method has the advantage of control over verbal, phonetic, and prosodic speech content. Because only the emotion is varied in the actors' portrayal, direct comparisons of voice quality and prosody between different affective states are feasible .Another advantage is that it is easier to obtain expressions of full blown and extreme emotions. The problem with an actor-based approach is the ecological validity of the obtained data. Actors might portray only stereotypical expressions of emotions and their portrayal may differ from spontaneous expression in natural circumstances. Banse and Scherer challenge these criticisms on two grounds. Actors actually feel the emotion they are portraying and that natural emotion expression is also "staged" because of the control of oneself required in different social contexts[2]. Nick Campbell [11] proposed a method similar to the actor approach. Instead of having the speaker read the same sentence in different emotions he had emotionally rich carrier sentences read which seemed to evoke genuine emotions. This idea was was further developed in the creation of the Belfast Structured Emotion Database [13] where different carrier paragraphs were written for the different emotions.

The elicitation of authentic emotions in participants in a laboratory has been tried by number of researchers [10]. There are a few techniques that are termed mood induction procedures which can be used in different settings. Few studies in the field of speech and emotion have used these kinds of methods. Johnstone & Scherer [11] have used a computer game in a speaker-centred study. In the case of a computer game the subjects were asked to give a verbal protocol of their current emotional state while playing. This allowed the experimenter to vary variables through the computer game. Recorded speech from spontaneous human interaction is the most natural but also uncontrolled. There are a few studies that are concerned with this kind of data. The Belfast Naturalistic Emotion Database is a collection of recordings from interviews and TV programs. Marc Schrder [14] analysed this corpus in his PhD thesis to find acoustic correlates for emotion dimensions, valence and arousal. As described there are a large variety of methods for obtaining emotionally colored data. Different techniques are suitable for certain investigations.

B. Acoustic Correlates of Emotions

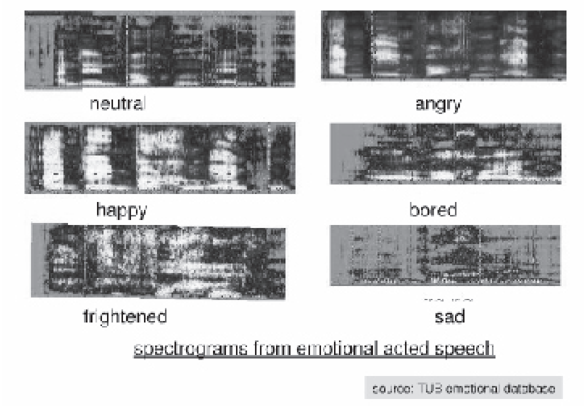
During evolution speech was added to a”primitive analog vocal signalling system”[14]. This means that the study of speech parameters expressing emotions is very complex. Acoustic parameters vary in their function of linguistic information carriers and non-verbal information carriers. Therefore it is not clear which parameters should be measured. Parameters like voice quality are important carriers of emotion in speech but are very difficult to measure. Therefore many studies have focused on measuring different aspects of F0 and intensity [2].Scherer conducted an extensive experiment where actors portrayed 14 different emotions, varying in intensity [2].

It was found that vocal parameters indicated the degree of arousal but also quality aspects or valence. Other studies also found a strong correlation between vocal parameters and arousal, but there is very little evidence for correlation between valence and parameters.

Marc Schrder conducted an extensive analysis of the Belfast Naturalistic Emotion Database and relating the results to three emotion dimensions activation (arousal), evaluation (valence) and power [14].

Fig. 3 Spectrograms from emotional acted speech

IV. CONCLUSIONS



The actual experiment took place on a computer where subjects had to rate the synthesized sentences. The rating was done using a continuos scale represented by a slider bar that was labelled angry on the left side, happy on the right side and neutralin the centre. The program recorded the input on a scale from 0 to 100 where 50 meant neutral, 0 meant angry, and 100 meant happy. It is not correct to say that happy and angry are opposite of each other but for the purpose of this experiment it should not had any effect on the outcome. This type of rating scale had the advantage that it is forced choice with a continuos response. A continuos response was needed because the strength of a given emotion was part of the assessment.

ACKNOWLEDGMENT

Underlying Productions needs to concede T. Dutoit and other giver for initial and continue the IEEE LaTeX style files which have been used in the preparation of this paper Speech and Emotion Research. An overview of Research Frameworks and a Dimensional Approach to Emotional Speech Synthesis.

REFERENCES

[1] Alan W. Black, Perfect synthesis for all of the people all of the time. IEEE TTS Workshop 2002

[2] John Kominek and Alan W. Black. (2003). CMU ARCTIC databases for speech synthesis. CMU-LTI-03-177

[3] Julia Zhang. Language Generation and Speech Synthesis in Dialogues for Language Learning, masters thesis, Section 5.6 on page 54

[4] PSOLA Synthesis.

[5] T. Dutoit, V. Pagel, N. Pierret, F. Bataille, O. van der Vrecken. The MBROLA Project: Towards a set of high quality speech synthesizers of use for non commercial purposes. ICSLP Proceedings, 1996.

[6] L.F. Lamel, J.L. Gauvain, B. Prouts, C. Bouhier, R. Boesch. Generation and Synthesis of Broadcast Messages, Proceedings ESCA-NATO Workshop and Applications of Speech Technology, September 1993.

[7] Examples include Astro Blaster, Space Fury, and Star Trek: Strategic Operations Simulator

[8] Ekman, P. (1993). Facial Expression and Emotion. American Psychologist,48(4):384-392.

[9] Frijda, N. H. (1986). The Emotions. Cambridge University Press, Cambridge,UK.

[10] Gerrards-Hesse, A., Spies, K., & Hesse, F. W. (1994). Experimental Induction of emotional states and their effectiveness: A review. British Journal of Psychology,85:55-78.

[11] Johnstone, T., Banse, R., & Scherer, K. R. (1995). Acoustic Profiles from Prototypical Vocal Expressions of Emotion. Proceedings of the XIIIth International Congress of Phonetic Sciences, 4, 2-5.

[12] Russell, J. A. (1980). A circumplex model of affect. Journal of Personality and Social Psychology. 39:1161-1178.

[13] Schere, K. R (1986) Vocal affect expression: A review and a model for future research. Psychological Bulletin. 99:143-165.

[14] Schrder, M. (2003). Speech and Emotion Research. An overview of Research Frameworks and a Dimensional Approach to Emotional Speech Synthesis. PhD thesis. UniversitŁt des Saarlandes. Saarbrcken.

[15] Whissell, C. M. (1989). The dictionary of affect and language. In Plutchik, R.and Kellerman, H., editors, Emotion: Theory, Research, and Experience. Volume 4: The measurement of emotions, pages 113-131. Academic Press, New-York.

Efficient Density based Clustering Technique for Categorical Datasets

Anil K. Tiwari[#1], Lokesh Kumar Sharma[*2], and G. Ramakrishna[†3]

[#]*Disha College of Information Technology, Raipur, Chhattisgarh, India*
[1]*anil1969_rpr@yahoo.com*

[*]*Rungta College of Engineering and Technology, Bhilai, Chhattisgarh, India*
[2]*lksharmain@gmail.com*

[†]*Department of Computer Science and Engineering, K L University, Vijayawada-India*
[3]*ramakrishna_10@yahoo.com*

Abstract

Density based clustering can identify arbitrary shape clusters and noises. Achieving good clustering performance requires regulating the appropriate parameters in the density based clustering. Basic idea is taken from Optics algorithm and is extended for categorical datasets. Algorithm effectively produces arbitrary shape clusters and identifies noise among the datasets.

***Keywords*— Clustering algorithms, Categorical datasets, Hamming distance.**

I. INTRODUCTION

Clustering has modern application in various domains, such as biomedical data, software engineering economics and others dataset emerging in such domains are often too large and too complex for human analysis these dataset motivates to design algorithm for categorical dataset.

Density based clustering approach uses a local density criterion. Clusters are subspaces in which the objects are dense and separated by subspaces of low density. Advantages of these include time efficiency and ability to find clusters of arbitrary shape. These algorithms leave the users with the responsibility of selecting parameter values for □ (Radius) and MinPts (minimum number of objects) that will lead to high quality cluster.

In this work the key idea of density based clustering is that each cluster contains minimum number of objects (MinPts) with respect to given radius (□). A modified density based clustering algorithm named EDCT is proposed which can handle categorical data. The Algorithm is tested and found to perform well. Rest of the paper is organized as follows. Section II reports related work on clustering. Basics of clustering algorithms for categorical data are discussed in section III. The experiment and analysis are given in section IV. Finally conclusions are given in section V

II. RELATED WORK

Existing clustering algorithms can be categorized on the basis of their procedures such as hierarchical, partitioning density based clustering etc ([1] [2] [3]). Hierarchical algorithms decompose a database D of n objects into several levels of nested partitioning, i.e. a tree that iteratively splits D into smaller subsets until each subset consists of only one object. In such a hierarchy, each node of the tree represents a cluster of D. Partitioning algorithms construct a flat (single level) partition of a database D of n objects into a set of k clusters such that the objects in a cluster are more similar to each other than to objects in different clusters. The Single-Link method is a commonly used hierarchical clustering method explained [1]. Other algorithms which in principle produce the same hierarchical structure have also been suggested [1].

Optimization based partitioning algorithms typically represent clusters by a prototype. Objects are assigned to the cluster represented by the most similar prototype. An iterative control strategy is used to optimize the whole clustering such that, e.g., the average or squared distances of objects to its prototypes are minimized. Consequently, these clustering algorithms are effective in determining a good clustering if the clusters are of convex shape, similar size and density, and if their number k can be reasonably estimated.

Depending on the kind of prototypes, one can distinguish k-means, k-modes and k-medoid algorithms. For k-means algorithms ([1]), the prototype is the mean value of all objects belonging to a cluster. The k-modes [1] algorithm extends the k-means paradigm to categorical domains. k-modes algorithm follows the cluster of spherical shape which is not possible always in real world also number of cluster to be generated are predefined which limits the efficiency of this algorithms

Density-based approaches apply a local cluster criterion and are very popular for the purpose of database mining. Clusters are regarded as regions in the data space in which the objects are dense, and which are separated by regions of low object density (noise). These regions may have an arbitrary shape and the points inside a region may be arbitrarily distributed. A common way to find regions of high-density in the data space is based on grid cell densities [2][4]. A histogram is constructed by partitioning the data space into a number of non-overlapping regions or cells. Cells containing a relatively large number of objects are potential cluster centers and the boundaries between clusters fall in the “valleys” of the histogram. The success of this method depends on the size of the cells which must be specified by the user. Cells of small volume will give a very "noisy" estimate of the density, whereas large cells tend to overly smooth the density estimate. In [4], a density-based clustering method is presented which is not grid-based. The basic idea for the algorithm DBSCAN is that for each point of a cluster the neighborhood of a given radius (ε) has to contain at least a minimum number of points (MinPts) where □ and MinPts are input parameters.

Another density-based approach is WaveCluster [1], which applies wavelet transform to the feature space. It can detect arbitrary shape clusters at different scales and has a time complexity of O(n). The algorithm is grid-based and only applicable to low-dimensional data. Input parameters include the number of grid cells for each dimension, the wavelet to use and the number of applications of the wavelet transform. Hinneburg et al [12] proposed a density based algorithm DenClue for multimedia database. This algorithm uses a grid but is very efficient because it only keeps information about grid cells that do actually contain data points and manages these cells in a tree-based access structure.

Agrawal et al [3] proposed CLIQUE clustering technique. CLIQUE combines the density and grid-based clustering technique for mining in high-dimensional data spaces. Input parameters are the size of the grid and a global density threshold for clusters. The major difference to all other clustering approaches is that this method also detects subspaces of the highest dimensionality such that high-density clusters exist in those subspaces.

Another approach to clustering is the BIRCH method [1] which cannot entirely be classified as a hierarchical or partitioning method. BIRCH constructs a CF-tree which is a hierarchical data structure designed for a multiphase clustering method. First, the database is scanned to build an initial in memory CF-tree which can be seen as a multi-level compression of the data which tries to preserve the inherent clustering structure of data. Second, an arbitrary clustering algorithm can be used to cluster the leaf nodes of the CF-tree. Because BIRCH is reasonably fast, it can be used as a more intelligent alternative to data sampling in order to improve the scalability of clustering algorithms. Zaki et al [15] proposed CLICK clustering algorithm which returns too many clusters or too many outliers. ROCK algorithm introduced by Guha et al [16], which does not require the user to specify the number of clusters. This algorithm exhibits cubic complexity in the number of objects, which makes it unsuitable for large datasets.

III. Ordering Point Structure For Categorical DATASET

The Basic concept of density-based clustering is that for each object of a cluster the neighborhood of a given radius (ε) has to contain at least a minimum number of objects (MinPts), i.e. the cardinality of the neighborhood has to exceed a threshold. Here we use this idea for developing the clustering technique for categorical datasets. In this work, Hamming distance for calculation of distance between categorical objects is used. Some formal definitions for this notion of a clustering are as follows.

Hamming Distance: It can be defined as follows:

$$HD(x,y) = \sum_{j=1}^{m} \delta(x_j, y_j) \quad \text{where}$$

$$\delta(x_j, y_j) = \begin{cases} 1, & \text{if } x_j \neq y_j \\ 0, & \text{if } x_j = y_j \end{cases}$$

Directly density-reachable: Object p is directly density-reachable from object q wrt ε and MinPts in a set of objects D if 1) p ∈ Nε (q) (Nε (q) is the subset of D contained in the ε-neighborhood of q.

2) *Card* (*Nε(q))* ≥ *MinPts* (*Card*(*N*) denotes the cardinality of the set *N*) The condition *Card*(*N*$_ε$*(q))* ≥ *MinPts* is called the "core object condition". If this condition holds for an object *p,* then we call *p* a "core object". Only from core objects, other objects can be directly density-reachable.

Density-reachable: An object *p* is *density-reachable* from an object *q* wrt. ε and *MinPts* in the set of objects *D* if there is a chain of objects $p_1 ... p_n$, $p_1 = q$, $p_n = p$ such that $p_i \in D$ and p_{i+1} is directly density-reachable from p_i wrt ε and *MinPts*. Density-reachability is the transitive hull of direct density reachability. This relation is not symmetric in general. Only core objects can be mutually density-reachable.

Density-connected: Object *p* is *density-connected* to object *q* wrt ε and *MinPts* in the set of objects *D* if there is an object *o* ∈*D* such that both *p* and *q* are density-reachable from *o* wrt ε and *MinPts* in *D*. Density-connectivity is a symmetric relation.

A density-based *cluster* is now defined as a set of density-connected objects which is maximal wrt Density-reachability and the *noise* is the set of objects not contained in any cluster.

Let *D* be a set of objects. A *cluster C* wrt ε and *MinPts* in *D* is a non-empty subset of *D* satisfying the following conditions:

1) Maximality: *p, q* ∈*D*: if *p* ∈*C* and *q* is density-reachable from *p* wrt ε and *MinPts*, then also *q* ∈*C*.

2) Connectivity: *p, q* ∈ *C*: *p* is density-connected to *q* wrt. ε and *MinPts in D*. The algorithm DBSCAN [4], which discovers the clusters and the noise in a database according to the above definitions, is based on the fact that a cluster is equivalent to the set of all objects in *D* which are density-reachable from an arbitrary core object in the cluster (c.f. lemma 1 and 2 in [2][4][11]).The retrieval of density-reachable objects is performed by iteratively collecting *directly* density-reachable objects. DBSCAN checks the ε-neighborhood of each point in the database. If the ε-neighborhood $N_ε(p)$ of a point *p* has more than *MinPts* points, a new cluster *C* containing the objects in $N_ε(p)$ is created. Then, the ε-neighborhood of all points *q* in C which have not yet been processed is checked. If $N_ε$ (*q*) contains more than *MinPts* points, the neighbors of *q* which are not already contained in *C*are added to the cluster and their ε-neighborhood is checked in the next step. This procedure is repeated until no new point can be added to the current cluster *C*.

Density-Based Cluster-Ordering: To introduce the notion of a density-based cluster-ordering, we first make the following observation: for a constant *MinPts*-value, density-based clusters with respect to a higher density (i.e. a lower value for ε) are completely contained in density-connected sets with respect to a lower density (i.e. a higher value for ε). Consequently, the DBSCAN algorithm is extended such that several distance parameters are processed at the same time, i.e. the density-based clusters with respect to different densities are constructed simultaneously.

To produce a consistent result, however, we would have to obey a specific *order* in which objects are processed when expanding a cluster. We have to select an object which is density-reachable with respect to the lowest ε value to guarantee that clusters with respect to higher density (i.e. smaller ε values) are finished first. This algorithm works in principle like such an extended DBSCAN algorithm for an infinite number of distance parameters ε_i which are smaller than a "generating distance" ε ($0 \leq \varepsilon_i \leq \varepsilon$). The only difference is that we do not assign cluster memberships and it uses Hamming distance for calculating the distance between objects. Instead, we store the *order* in which the objects are processed and the information which *would* be used byan extended DBSCAN algorithm to assign cluster memberships (if this were at all possible for an infinite number of parameters). This information consists of only two values for each object: the *Hamming core distance* and a *Hamming reachability distance*, introduced in the following definitions.

Core distance of an object p: Let *p* be an object from a database *D*, let ε be a distance value, let $N_\varepsilon(p)$ be the ε-neighborhood of *p*, let *MinPts* be a natural number and let *MinPts_distance(p)* be the distance from *p* to its *MinPts*' neighbor. Then, the *core-distance* of *p* is defined as

$$= \begin{cases} Undefined \quad if \quad Card(N_\varepsilon(p)) < MinPts \\ MinPts_{distance(p)},\ otherwise \end{cases}$$

The core-distance of an object *p* is simply the smallest distance ε' between *p* and an object in its ε-neighborhood such that *p* would be a core object with respect to ε' if this neighbor is contained in $N_\varepsilon(p)$. Otherwise, the core-distance is UNDEFINED.

Hamming Reachability Distance:

Let *p* and *o* be objects from a database *D*, let $N_\varepsilon(o)$ be the ε-neighborhood of *o*, and let *MinPts* be a natural number. Then, the *reachability distance* of *p* with respect to *o* is defined as

$$Hamming\ reachability\_distance_{\varepsilon\ MinPts}(p,o)$$

$$= \begin{cases} Undefined \quad if \quad |(N_\varepsilon(p))| < MinPts \\ Max(core\_distance(o), distance(o,p)),\ otherwise \end{cases}$$

Intuitively, the Hamming reachability distance of an object *p* with respect to another object *o* is the smallest distance such that *p* is directly density-reachable from *o* if *o* is a core object. In this case, the reachability distance cannot be smaller than the core distance of *o* because for smaller distances no object is directly density-reachable from *o*. Otherwise, if *o* is not a core object, even at the generating distance ε, the Hamming reachability distance of *p* with respect to *o* is UNDEFINED. The Hamming reachability distance of an object *p* depends on the core object with respect to which it is calculated. This algorithm creates an ordering of a database, additionally storing the Hamming core-distance and a suitable Hamming reachability distance for each object. We will see that this information is sufficient to extract all density-based clustering with respect to any distance ε'which is smaller than the generating distance ε from this order.

At the beginning, we open a file for writing and close this file after ending the loop.

Each object from a database is simply handed over to a procedure Expansion of order cluster if the object is not yet processed. The pseudo-code for the procedure is given. The procedure Expansion of cluster order first retrieves the ε-neighborhood of the object passed from the main loop, sets its Hamming reachability distance to UNDEFINED.

Proposed cluster algorithm Efficient Density based Clustering Technique for Categorical Dataset (EDCT) can be summarized in pseudo code as given below.

```
EDCT (Soo, ε, MinPts, Of)
   Of.open();
For( i = 1 ; Soo.size; i++)  {
   Object := Soo.get(i);
IF( NOT Object.Processed )THEN
  Eco(Soo, Object, ε, MinPts, Of)
  Of.close(); }
ExtractDBSCAN-Clustering (Coo, ε', MinPts)
// ε ' must be less than  ε for Coo
Clid := NOISE;
For(i=1; i = Coo.size;i++ ){
Object := Coo.get(i);
IF Object.hamming reachability_distance > ε' THEN
IF Object.core_distance =  ε' THEN
   Clid := nextId(Clid);
   Object.Clid := Clid;
ELSE
   Object.Clid := NOISE;
ELSE
   Object.Clid := Clid; }
Eco(Soo, Object, ε, MinPts,Of);
   neighbors := Soo.neighbors(Object, ε);
   Object.Processed := TRUE;
Object.hamming _reachability_distance := UNDEFINED;
Object.setHamming_core_distance (neighbors, ε, MinPts);
Of.write(Object);
IF Object.Hamming_core_distance<> UNDEFINED THEN
OrderSeeds.update(neighbors, Object);
WHILE NOT OrderSeeds.empty() DO
   currentObject := OrderSeeds.next();
   neighbors:=Soo.neighbors(currentObject, ε);
   currentObject.Processed := TRUE;
currentObject.setHamming_core_distance(neighbors, ε,
MinPts);
Of.write(currentObject);
IF currentObject.Hamming_core_distance<>UNDEFINED THEN
OrderSeeds.update(neighbors, currentObject);
END;
OrderSeeds::update(neighbors, CenterObject);
c_dist := CenterObject.Hamming_core_distance;
FOR ALL Object FROM neighbors DO
IF NOT Object.Processed THEN
new_r_dist:=max(c_dist,CenterObject.dist(Object));
IF Object.hamming_reachability_distance=UNDEFINED THEN
Object.hamming _reachability_distance := new_r_dist;
insert(Object, new_r_dist);
ELSE
IF new_r_dist<Object.hamming _reachability_distance THEN
Object.hamming _reachability_distance := new_r_dist;
decrease(Object, new_r_dist);
END;
```

In above code set of objects is represented by Soo, Organized sorted file by Of, Expansion of cluster order by Eco and through Hamming distance Ordering of objects of clusters is given by Coo and Clusters identification is given by Clid:

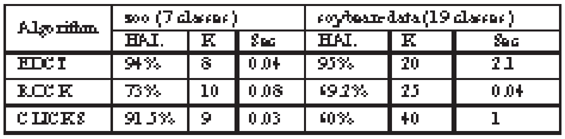
IV. EXPERIMENT AND ANALYSIS

We performed our experiment on UCI dataset Zoo and Soybean [13]. For analysis Hurbert and Arabie Index (HA) [14] is calculated for these datasets. Suppose that U is the solution known or believed to be present in S and V is the clustering result by EDCT, ROCK and CLICKS algorithm. Then a, b, c, d are the numbers of pairs of objects placed in:

a: the same class in U and the same cluster in V.
b: the same class in U but, not the same cluster in V.
c: not the same class in U but the same cluster in V.
d: different classes and clusters in both U and V.

THEN, HA INDEX = $-\frac{a+d}{a+b+c+d}$ [14]

TABLE I
HA Indexes (higher is better), on zoo and soybean data.



Algorithm	zoo (7 classes)			soybean data (19 classes)		
	HAI.	K	Sec	HAI.	K	Sec
EDCT	94%	8	0.04	95%	20	2.1
ROCK	73%	10	0.08	69.2%	25	0.04
CLICKS	91.5%	9	0.03	60%	40	1

In above table HA Index [14] of EDCT is higher than Rest of the two algorithms which delivers more efficient clustering. Number of clusters generated by EDCT is lesser than rest of the algorithms. Running time of EDCT is higher than ROCK and CLICKS for Soybean data where numbers of classes are larger than Zoo data. In light of these results it is evident that EDCT algorithm out performs the other two algorithms.

V. CONCLUSION

This investigation is carried out with a modified density based clustering algorithm named EDCT, which can handle categorical datasets. It works efficiently and deals with arbitrary shape clusters and detects noises. It is tested with bench mark datasets. The results show reasonably good performance. In future studies, an investigation will be carried out on how to apply the algorithm to other applications such as training the codebook or getting the shape of images. We also intend to modify the algorithm to achieve trade-off between limited amounts of accuracy for large gain in efficiency.

REFERENCES

[1] Rui Xu; Wunsch, D," Survey of clustering algorithms", IEEE Tran. On Neural Network, Vol. 16(3), pp. 645 – 678, 2005.

[2] Mihael Ankerst , Markus M. Breunig , Hans-Peter Kriegel , Jörg Sander, OPTICS: ordering points to identify the clustering structure, Proceedings of the 1999 ACM SIGMOD international conference on Management of data, p.49-60, May 31-June 03, 1999, Philadelphia, Pennsylvania, United States

[3] Agrawal R., Gehrke J., Gunopulos D., Raghavan P.: "Automatic Subspace Clustering for High Dimensional Data for Data Mining Applications", Proc. ACM SIGMOD'98 Int.Conf. on Management of Data, Seattle, WA, 1998, pp. 94-105.

[4] Ester M., Kriegel H.-P., Sander J., Xu X.: "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, 1996, pp. 226-231.

[5] Huang Z.: "A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining", Proc. SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Tech. Report 97-07, UBC, Dept. of CS, 1997.

[6] Brecheisen, S., Kriegel, H. P. and Pfeifle, M. (2006) "Multi-Step Density-Based Clustering". Knowledge and Information System 9(3) 284-308.

[7] Brecheisen, S., Kriegel, H. P. and Pfeifle, M. (2006) "Parallel Density-Based Clustering of Complex Objects". LNAI 3918 Springer Verlag 179-188.

[8] Gorawski, M. and Malczok, R. (2006) "AEC Algorithm: A Heuristic Approach to Calculation Density-Based Clustering Eps Parameter" LNCS 4243 Springer –Verlag 90-99.

[9] Gao, S. and Xia, Y. (2006) "GDCIC: A Grid-based Density-Confidence-Interval Clustering Algorithm for Multi-density Dataset in Large Spatial Database" In: Proc. 6th IEEE Int. Conf. on Intelligent Systems Design

Applications) 713-717.

[10] Ma, D. and Zhang A. (2004) “An Adaptive Density-Based Clustering Algorithm for Spatial Database with Noise”. Proc. 4th IEEE Int. Conf. on Data Mining (ICDM 04) 467-470.

[11] Sander, J., Ester, M., Kriegel, H. P. and Xu, X. (1998) Density- Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. Journal of Data Mining and Knowledge Discovery, Kluwer Academic Publishers vol. 2, 169-194.

[12] Hinneburg A., Keim D. (1998) “An Efficient Approach to Clustering in Multi Media database with Noise, Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining, New York City.

[13] Mertz C.J. and Merphy P.(1996) “UCI Repository of achine Learning databases”,http://www.ics.uci.edu/`mlearn/MLRepository.html.

[14] Hubert L. and Arabie P. (1985) “Comparing partitions”. Journal of Classification.193-218.

[15] Zaki M.J. and Peters M. (2005) “CLICK:Clustering Categorical Data using K-partite Maximal Cliques”. TR04-11.CS Dept. RPI.

[16] Guha S., Rastogi R. and Shim K. (2000) “ROCK: A Robust Clustering algorithm for Categorical Attributes”. Information Systems 25(5).

Deterministic Feed Forward Back-Propagation Artificial Neural Network Design to Recognize Internal Dynamics of Chaotic Motion

S Karmakar
Department of Computer Applications, Bhilai Institute of Technology
Bhilai House-491001, Durg (C.G.), INDIA e-mail : karmakar_sanjeev@rediffmail.com

B Varghese
Department of Computer Applications, Bhilai Institute of Technology
Bhilai House-491001, Durg (C.G.), INDIA, e-mail : brvarghese@rediffmail.com

M K Kowar
Department of Electronics & Telecommunication, Bhilai Institute of Technology
Bhilai House-491001, Durg (C.G.), INDIA e-mail : kowar_bit@rediffmail.com

P Guhathakurta
India Meteorological Department, Sivaji Nagar, Pune-411005, INDIA.
e-mail: pguhathakurta@rediffmail.com

ABSTRACT

TMR data time series behaves as chaotic series, represents no periodic behavior and sensitivity to initial conditions, internal dynamics is difficult or impossible to identify, The motion 'looks' random, highly non-linear. To identification of internal dynamics of TMR over very high-resolution geographical region (district), three-layer perceptron feed-forward back propagation neural network model has been developed. This network has input layer (*at the bottom*), one hidden layer (*at the middle*) and output layer (*on the top*). The model has *n* input vectors at the input layer ($x_1 \ldots x_i \ldots x_n$). *p* neurons in hidden layer ($z_1 \ldots z_j \ldots z_p$) and one neuron ($y_k$) in output unit to observe n+1th year TMR data as output target variable $n \times p + p$ trainable weights are used in the network. The neurons output can be obtained as $f(x_j)$ Where *f* is a transfer function (*axon*), typically the sigmoid (*logistic or tangent hyperbolic*) function.

Sigmoid function $f(x)=\dfrac{1}{1+e^{-\delta x+\eta}}$ where δ

determines the slope and η is the threshold. In the proposed model δ =1, η =0 is considered such that $\forall \pm n \in I^{+}$, the output of the neuron will be in close interval [0, 1]. The network has been tested with different values of *p* and *n*. It has been found that $MAD \; \alpha \; P$ and $MAD \propto \dfrac{1}{n}$ And also it has been observed that, three neurons ($p = 3$) and eleven input vector ($n = 11$) offered optimum result in terms of MSE. The network has been trained by using proposed algorithm up to the MSE level 2.161592488977503E-04. It required 75 lakhs epochs (*i.e. three days of parallel processing time through P5 Due processor with 1GB memory*) for training. More than that provided over-trained network. The performance of the model has been identified by comparison of SD of actual data and MAD between actual and predicted data. It has been found that, MAD is one third of SD and CC between actual and predicted values is 0.92 and 0.81 for training (1951-1991) and independent (1992-2004) period respectively. This facts are clearly indicates the skill of developed model and successfully identification of internal dynamics of TMR data time series, is presented through this paper. Required TMR Data have been collected from IMD Pune, India.

Keywords- Neural, deviation, dynamic system, chaotic, stochastic.

Abbreviation- *MAD*: Mean Absolute Deviation (% of Mean), *ANN*: Artificial Neural Network, *TMR*: Total Monsoon Rainfall, *IMD*: India Meteorological Department, CC: Coefficient of Correlation.

INTRODUCTION

Rainfall data time series behaves as chaotic series [1]. In short, data time series (chaos) represented the important principles, no periodic behavior, sensitivity to initial conditions, chaotic motion is difficult or impossible to forecast, the motion 'looks' random and non-linear. A non-linear systems, the change in a variable at an initial time can lead to a change in the same or a different variable at a later time, that is not proportional to the change at the initial time. Neural network technique is being applied by many scientists in the non-linear prediction models. Neural network technique has a strong potential for pattern recognition and signal processing problems and it has the ability to predict for the future value of the time series. This technique has successfully been applied to a variety of problems. It has been shown by Elsner and Tsonis (1992) that neural network can be used successfully to predict a chaotic time series [2]. Application of neural network for short-term prediction of air pollutants (Bonzar *et al*., 1993; Guhathakurta, 1998) has shown interesting results [3]. The neural network technique is also able to learn the dynamics within the time series data (Elsner and Tsonis, 1992) and long range prediction models of monsoon rainfall of India using only time series data of monsoon rainfall have shown some encouraging results (Goswami and Srividya, 1994; Guhathakurta and Thapliyal, 1997, Guhathakurta, 2006)[3]. Karmakar *et al.,* 2008 found that neural network technique is useful both for stochastic and deterministic forecast processes [4]. In deterministic forecast process, rainfall time series is treated as deterministic and even chaotic. It uses rainfall data of the past years to forecast future rainfall. Attempts to predict all- India southwest monsoon rainfall using deterministic forecast were already made by many scientists. Karmakar *et al.,* 2008 have found neural can be successfully applied for districts as well as subdivision level [5]. However, in bad feeling of obtaining some accuracy in prediction during the test period, the success was not so appreciable in the real operational forecasting. In this study various three layers perceptron back- propagation neural networks based on different size of input vector (*n*) and neurons in hidden layer (*p*) to identify internal dynamics of chaotic motion has been observed. The neural network design and its performances is presented in this paper.

ARCHITECTURE OF BACK-PROPOGATION NEURAL NETWORK

Three layers perceptron feed forward back-propagation deterministic artificial neural network models have been developed. In Proposed model, where *n* input vectors ($x_1 \ldots x_i \ldots x_n$) in input layer used to input *n* years data time series. Number of neurons *p* in hidden layer ($z_1 \ldots z_j \ldots z_p$).

One neuron *(y_k)* in output unit used to observe *(n+1)* year *target* data. The neurons output is obtained as *f (x)*. Where *f* is a transfer function, typically the sigmoid (logistic or tangent hyperbolic)

function. Sigmoid function $f(x) = \dfrac{1}{1+e^{-\delta x+\eta}}$ where δ

determines the slope and η is the threshold. We have been considered δ =1, η =0 therefore , $\exists \pm n \in I^{+}$ the output of the neuron will be in close interval $\lfloor 0, 1 \rfloor$.

TRAINING

The process of learning the training set of patterns means the determination of the optimum weights, which minimize the mean square error between the output of the network and the desired value. Most commonly used 'Back-Propagation learning algorithm' (Rumelhart *et al* 1986) is used for the training [6]. We have used three layers in network with 36, trainable weights. Initial value of weights is assigned by random values between ±0.5. The optimum weights are learned through back propagation algorithm's iterative process (epochs). Training of the networks is continued till the mean square error (MSE) becomes less than a pre-assigned value ranging from 0.00005 to 0.00001. The used algorithm is given here with the various phases, where the various parameters used in the training algorithm are as follows-

x: *($x_1 \ldots x_i \ldots x_n$)* Input training rainfall time series t: Output target vector *($t_1 \ldots t_k \ldots t_m$);* δ_k = Error at output unit y_k; δ_j = Error at hidden unit Z_j; α = Learning rate.; V_{oj} = bias on hidden unit j; Z_j= Hidden unit j; w_{ok} = bias on output unit k; y_k= output unit k.

Initialization of weights.

1. Initialization weights to small random values.
2. While stopping conditions is false, do Step 3-10.
3. For each training pair do Steps 4-9.

Feed forward

4. Each input neuron receives the input signal x_i and transmits this signal to all neurons in the layer above i.e. hidden neurons.
5. Each hidden unit *(z_j, j = 1....p)* sums its weighted input signals and sends this signal to all units in the layer above i.e. output units

$$z_{inj} = v_{oj} + \sum_{i=1}^{n} x_i v_{ij}$$

Applying activation function

$$Z_j = f(z_{inj})$$

6. Each output unit *(y_k, k = 1...m)* sum its weighted input signals.

$$y_{ink} = w_{ok} + \sum_{j=1}^{p} z_j w_{jk}$$

Applying activation function

$$y_k = f(y_{ink})$$

Back propagation errors

7. Each output unit *(y_k, k = 1...m)* receives a target pattern corresponding to an input pattern error information term is calculated as

$$\delta_k = (t_k - y_k) f'(y_{ink})$$

8. Each hidden unit *(z_j, j = 1...n)* sums its delta inputs from units in the layer above

$$\delta_{inj} = \sum_{k=1}^{m} \delta_j w_{jk}$$

The error information term is calculated as

$$\delta_j = \delta_{inj} f'(z_{inj})$$

Updation of weights and biases

9. Each output unit *(y_k, k = 1...m)* updates its bias and weights (*j* = 0,..*p*)The weights correction term is given by $\Delta W_{jk} = \alpha \delta_k z_j$

$$\Delta W_{ok} = \alpha \delta_k$$

And the bias correction term is given by

$$W_{j1}(new) = W_{j1}(old) + \Delta W_{j1},$$

$$W_{o1}(new) = W_{o1}(old) + \Delta W_{o1}$$

$$\Delta V_{ij} = \alpha \delta_j x_i$$

Each hidden unit $(z_j, j = 1....p)$ updates its bias and weights ($i = 0,..n$) the weights correction term

$$\Delta V_{oj} = \alpha \delta_j$$

The bias correction term

Therefore

$$V_{ij} = V_{ij}(old) + \Delta V_{ij},$$

$$V_{oj}(new) = V_{oj}(old) + \Delta V_{oj}$$

10. Test the stopping condition, i.e., MSE up to the level of $2x10^{-4}$.

RESULTS AND DISCUSSIONS

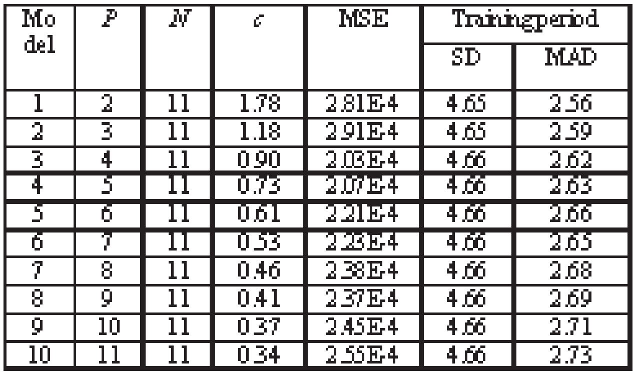
Neural Network with eleven years *TMR* of Ambikapur district as input to observed twelfth year *TMR* as a target variable with unusual number of neurons in hidden layer that is $p = 2$ to 11 have been observed. It has been found that, number of neurons is directly proportional to MAD (% of mean) between actual and predicted values. The relation between neurons (p) and MAD (%of mean) as shown as bellow Equ^n (1) & (2).

$$MAD \propto p \ldots (1)$$

$$MAD = cp \ldots (2)$$

Where c is error constant. The relation within number of neurons

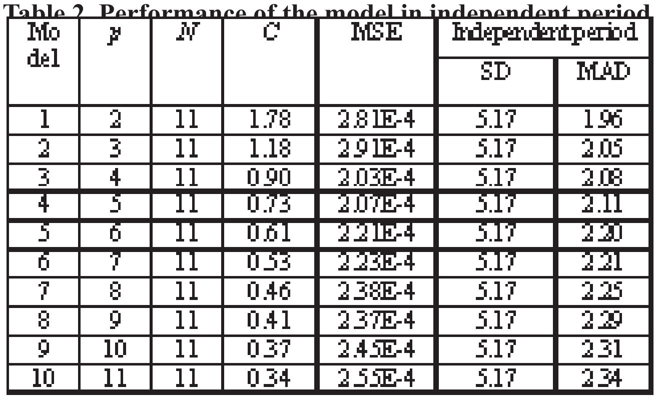
in hidden layer MAD (% of mean) and error constant shown in Table 1&2 .



Model	P	N	c	MSE	Training period	
					SD	MAD
1	2	11	1.78	2.81E-4	4.65	2.56
2	3	11	1.18	2.91E-4	4.65	2.59
3	4	11	0.90	2.03E-4	4.66	2.62
4	5	11	0.73	2.07E-4	4.66	2.63
5	6	11	0.61	2.21E-4	4.66	2.66
6	7	11	0.53	2.23E-4	4.66	2.65
7	8	11	0.46	2.38E-4	4.66	2.68
8	9	11	0.41	2.37E-4	4.66	2.69
9	10	11	0.37	2.45E-4	4.66	2.71
10	11	11	0.34	2.55E-4	4.66	2.73

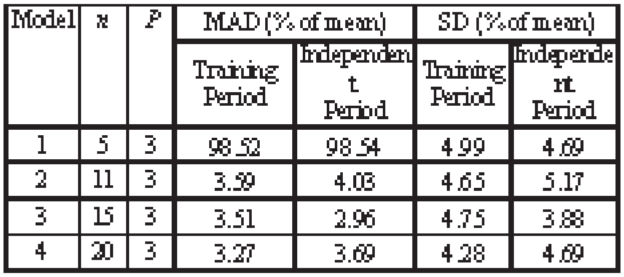
Table 1. Performance of the model in training period

Table 2. Performance of the model in independent period



Model	p	N	C	MSE	Independent period	
					SD	MAD
1	2	11	1.78	2.81E-4	5.17	1.96
2	3	11	1.18	2.91E-4	5.17	2.05
3	4	11	0.90	2.03E-4	5.17	2.08
4	5	11	0.73	2.07E-4	5.17	2.11
5	6	11	0.61	2.21E-4	5.17	2.20
6	7	11	0.53	2.23E-4	5.17	2.21
7	8	11	0.46	2.38E-4	5.17	2.25
8	9	11	0.41	2.37E-4	5.17	2.29
9	10	11	0.37	2.45E-4	5.17	2.31
10	11	11	0.34	2.55E-4	5.17	2.34

Table 3. Performance of various network with different *n* and *p=3*.

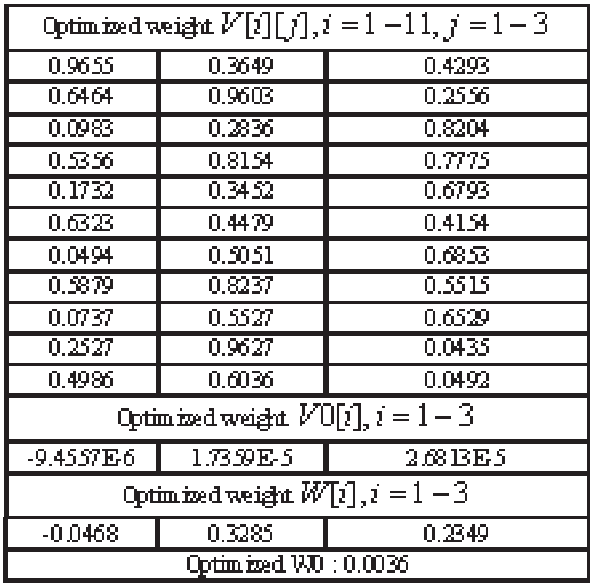


Model	n	P	MAD (% of mean)		SD (% of mean)	
			Training Period	Independent Period	Training Period	Independent Period
1	5	3	98.52	98.54	4.99	4.69
2	11	3	3.59	4.03	4.65	5.17
3	15	3	3.51	2.96	4.75	3.88
4	20	3	3.27	3.69	4.28	4.69

In other hand the relation between number of input and MAD (% of mean) shown in Table 3 . It is indicated that, MAD is inversely proportional to , i.e., shown in Equ^n (3).

It is observed that 11 years *TMR* time series as input, 3 neurons in hidden layer will give optimum result (Table 3). In deterministic forecast, past 11 years *TMR* time series as input to obtain 12th year *TMR*. The random weights of the network between 0 and 1 have been initialized. Optimized weights shown in Table 4 through 75 lakhs epochs with MSE 2.1615924E-04, have been found.

Table 4. Optimized weights after 75 lakhs epochs with MSE 2.1615924E-04.



Optimized weight $V[i][j], i = 1-11, j = 1-3$		
0.9655	0.3649	0.4293
0.6464	0.9603	0.2556
0.0983	0.2836	0.8204
0.5356	0.8154	0.7775
0.1732	0.3452	0.6793
0.6323	0.4479	0.4154
0.0494	0.5051	0.6853
0.5879	0.8237	0.5515
0.0737	0.5527	0.6529
0.2527	0.9627	0.0435
0.4986	0.6036	0.0492
Optimized weight $V0[i], i = 1-3$		
-9.4557E-6	1.7359E-5	2.6813E-5
Optimized weight $W[i], i = 1-3$		
-0.0468	0.3285	0.2349
Optimized W0 : 0.0036		

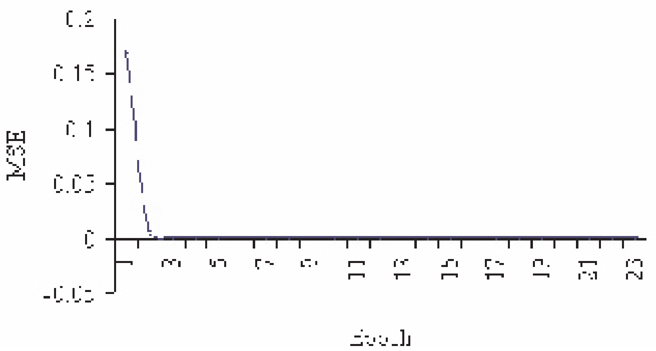
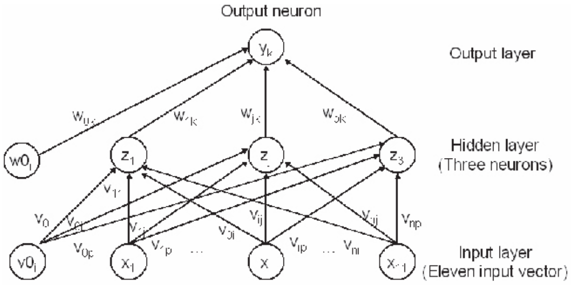


Fig. 1. NN learning curve.

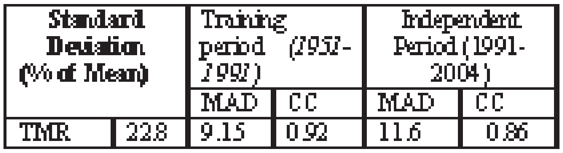
As Fig. 1, the Network presents a good learning curve with a final mean square error (MSE) very close to the optimum value of 0 inside 23 epochs only. Developed ANN model in deterministic forecast is shown as Fig. 2, where eleven input vectors $(x_1 ... x_j ... x_{11})$ in input layer used to input eleven years TMR data time series. Three neurons in hidden layer (z_1, z_j, z_3). One neuron (y_k) in output unit used to observe twelve-year TMR (t) data. 36 trainable have been used in the network. The neurons output is obtained as sigmoid f (xj). MAD (% of mean) between predicted and actual rainfall values during the training (1951-1991) as well as independent period (1992-2004) to observe performance of the models have been calculated. The MAD (% of mean) for the model during the training period and the independent period are given in the Table 6.

Fig. 2. Proposed ANN Model



It has been found that it is one third of the standard deviation (% of mean) in training period and it is just half of the standard deviation (% of mean) in the independent period for the districts *TMR* time series. Correlation between actual and model predicted *TMR* values in training and independent period is 0.86 and 0.83 respectively. This clearly indicates the skill of the models. Pattern of the *TMR* is successfully identified in training and independent period.

Table 6. Performance of the model in training & independent period



Standard Deviation (% of Mean)		Training period (1951-1991)		Independent Period (1991-2004)	
		MAD	CC	MAD	CC
TMR	22.8	9.15	0.92	11.6	0.86

CONCLUSION

Several issues have been found in design an ANN for identification of internal dynamics of chaotic motion i.e., selection of input vectors, selection of number of hidden layer, number of neurons in hidden layer, training of ANN etc. It is concluded that, only one hidden layer is sufficient. Back-propagation has the most important characteristics of a multiplayer network is the number of neurons in the hidden layer(s). If appropriate number of neurons are not used, then, the network will be unable to model complex data, which will cause poor fit result. If too many neurons are used, the training time may become too long, and, result in the network over fit the data which is worse. When over fitting occurs, the network will begin to model random noise in the data. It is concluded that 11 input vector with 3 neurons in hidden layer provided optimum result. Therefore, past 11 years total monsoon rainfall data as input, 3 neurons in hidden layer, one neuron in output layer, trainable weights, and sigmoid function as neuron output for the neural network development. It has been observed that after three days of processing (i.e., 75 lakh epochs) is required to train the network and found satisfactory results. Finally it is concluded that, artificial neural network is one of efficient method in identification of internal dynamics of chaotic motion

ACKNOWLEDGEMENT

We are grateful to Dr. H. P. Das Dy. Director General, India Meteorological Department, Pune, India for encouragement and help. We also thank to reviewer for comments and helpful suggestions.

REFERENCES

[1]. Sujit Basu, "The chaotic time series of Indian rainfall and its prediction.", Proc. Ind. Acad. Sci., 101 27-34 (1991).

[2]. Elsner J. B., Tsonis A. A., "Nonlinear prediction chaos and noise",Bull. Amer. Meteor. Soc. 73 49-60 (1992).

[3]. P Guhathakurta, "Long-range monsoon rainfall prediction of 2005 for the districts and sub-division Kerala with artificial neural network", Current Science, 90 773-779, (2006).

[4]. S karmakar, M K Kowar, P Guhathakurta, "Development of an 8-Parameter Probabilistic Artificial Neural Network Model for Long-Range Monsoon Rainfall Pattern Recognition over the Smaller Scale Geographical Region -District", IEEE Xplore, (2008).

[5]. S Karmakar, M K Kowar, P Guhathakurta, "Development of Artificial Neural Network Models for Temperature Parameters Pattern Recognition over the Smaller Scale Geographical Region-District", International. J. Engg., Res. & Ind. Application, Vol. I, No. 5, (2008), pp 111-121.

[6]. Rumbelhart D., Hinton G. E., and Williams R. J., "Learning internal representation by error propagation, R J Parallel Distributed Processing: Exploration in the Microstructure of Cognition", MIT Press Cambridge 1

Speech Analysis of Chhattisgarhi (dialect) Speech signal of different regions of Chhattisgarh.

Madhuri Gupta & Akhilesh Tiwari
Department of Computer Science & Engg
Shri Shankaracharya College of Engineering & Technology, Bhilai
Email: madhuri_gupta28@yahoo.com, akhilesh_tiwari@rediffmail.com

ABSTRACT:

Speech recognition (also known as **automatic speech recognition** or **computer speech recognition**) converts spoken words to text. The term “voice recognition” is sometimes used to refer to recognition systems that must be trained to a particular speaker—as is the case for most desktop recognition software. Wavelets have scale aspects and time aspects; consequently every application has scale and time aspects. To clarify them we try to untangle the aspects somewhat arbitrarily. This paper presents a brief description about Chhattisgarhi language and dialects commonly spoken at different regions of Chhattisgarh. According to the Indian Government, Chhattisgarhi is an eastern dialect of Hindi. Chhattisgarhi has several dialects of its own, in addition of Chhattisgarhi proper. Overall Chhattisgarhi can be divided into 26 different types of dialects.

From the diffent regional dialects we are trying to analyze the speech signals of speakers with its native place in Chhattisgarh State. A database speech signal is created from different regions of Chhattisgarh State. All speakers are in the age group of 8-12 years from the parent location.

Keywords: Chhattisgarhi language, Speech Recognition, Wavelet Analysis.

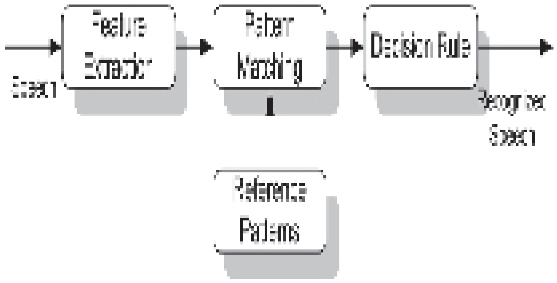
INTRODUCTION

The Chhattisgarhi language, a dialect of eastern Hindi, is a Pre-dominant language in the state, recognized along with Hindi as the official language of the state.

Many tribal and some Dravidian influenced dialects or languages are spoken in various parts of Chhattisgarh.A total of 93 dialects are spoken in the state which together represent all three of India’s major language families except Tibeto-Burman: Munda (Austro-Asiatic language), Dravidian and Indo-European.The speakers are concentrated in the Indian state of Chhattisgarh and in adjacent areas of Madhya Pradesh, Orissa, and Jharkhand. Chhattisgarhi cultural and political movements, with origins going back to the 1920s, affirmed Chhattisgarhi cultural and linguistic identity and sought greater autonomy within India. This came about in 2000 when 16 districts of the state of Madhya Pradesh became the new state of Chhattisgarh.

Speech recognition or more commonly known as automatic speech recognition (ASR) was defined as the process of interpreting human speech in a computer. However, ASR was defined more technically as the building of system for mapping acoustic signals to a string of words. In general, all ASR systems aim to automatically extract the string of spoken Words from input speech signals as illustrated in Figure

Figure 1: Speech Recognition System Concepts



Speech recognition is the process by which a computer (or other type of machine) identifies spoken words.

Basically, it means talking to your computer, AND having it correctly recognizes what you are saying. The following definitions are the basics needed for understanding speech recognition technology.

Utterance

An utterance is the vocalization (speaking) of a word or words that represent a single meaning to the computer. Utterances can be a single word, a few words, a sentence, or even multiple sentences.

Speaker Dependance

Speaker dependent systems are designed around a specific speaker. They generally are more accurate for the correct speaker, but much less accurate for other speakers. They assume the speaker will speak in a consistent voice and tempo.

Vocabularies

Vocabularies (or dictionaries) are lists of words or utterances that can be recognized by the SR system. Generally, smaller vocabularies are easier for a computer to recognize, while larger vocabularies are more difficult. Unlike normal dictionaries, each entry doesn’t have to be a singleword. They can be as long as a sentence or two. Smaller vocabularies can have as few as 1 or 2 recognized utterances (e.g.”Wake Up”), while very large vocabularies can have a hundred thousand or more!

Accuracy

The ability of a recognizer can be examined by measuring its accuracy - or how well it recognizes utterances. This includes not only correctly identifying an utterance but also identifying if the spoken utterance is not in its vocabulary. Good ASR systems have an accuracy of 98% or more! The acceptable accuracy of a system really depends on the application.

Training

Some speech recognizers have the ability to adapt to a speaker. When the system has this ability, it may allow training to take place. An ASR system is trained by having the speaker repeat standard or common phrases and adjusting its comparison algorithms to match that particular speaker. Training recognizer usually improves its accuracy.

Training can also be used by speakers that have difficulty speaking, or pronouncing certain words. As long as the speaker can consistently repeat an utterance, ASR systems with training should be able to adapt.

Problem Definition: Wavelet Analysis: Wavelet analysis is capable of revealing aspects of data that other signal analysis techniques miss aspects like trends, breakdown points, discontinuities in higher derivatives, and self-similarity. Furthermore, because it affords a different view of data than those presented by traditional techniques, wavelet analysis can often compress or de-noise a signal without appreciable degradation.

Indeed, in their brief history within the signal processing field, wavelets have already proven themselves to be an indispensable addition to the analyst's collection of tools and continue to enjoy a burgeoning popularity today.

Problem Definition and Analysis

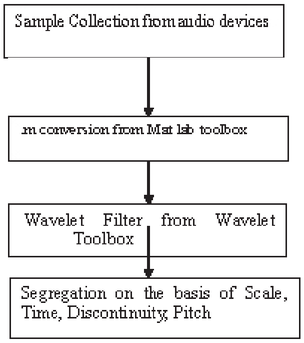
From the regional dialects we are trying to create a model for detecting a locality of speakers in Chhattisgarh State. A database speech signal is created from different regions of Chhattisgarh State. All speakers are from native location of the state in the age group of 8-12 years from the parent location.

The Flow Chart: for creation of this model we need to convert the sample native speaker's signal into .m file from Mat lab Toolbox .After receiving the .m signal of speech the wave let toolbox provide the filtration for that signal. Segregation of signal on the basis of time, scale, discontinuity and pitch. The flow chart shown below

Figure: Working model

Working Model:

The working model defines the creation of Model from the input signal and receiving Result with some accuracy level. The model is shown in below figure



Sample Collection from audio devices
.m conversion from Mat lab toolbox
Wavelet Filter from Wavelet Toolbox
Segregation on the basis of Scale, Time, Discontinuity, Pitch

Figure: Working model

Wavelet Analysis of the Bhujhwari dialect

1. Load the signal: The original signal wave form shown in figure below

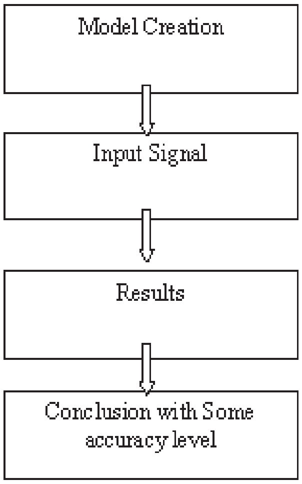
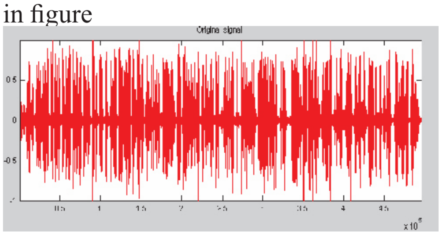


Figure: Original Wave signal of bhujhwari dialect

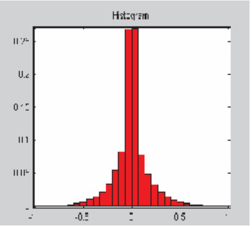
2. Synthesize the signal: The histogram wave signal shown below in figure

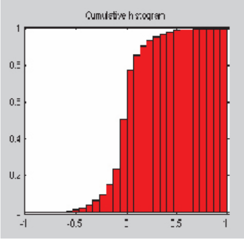


3. Cumulative histogram

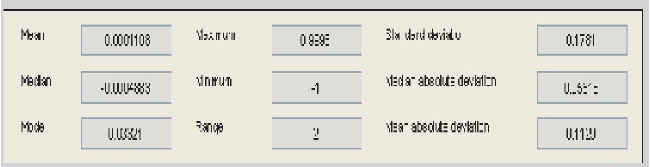
Figure: Cumulative histogram

RESULT





After analyzing the different speech signals of different dia



Mean	0.0001108	Maximum	0.9395	Standard deviation	0.1781
Median	-0.0004883	Minimum	-1	Median absolute deviation	[illegible]
Mode	0.03821	Range	2	Mean absolute deviation	0.1120

For the creation of database on these dialects, I have collected the voice samples of the people from different region of the state.

In this survey a standard set of 15-20 statements have been used which are used in day to day life by people.

These statements have specific words through which we can differentiate the dialects.

People were asked to speak the statements in their own native tone.

By collecting the samples from various region of the state, a database will be formed which will be used for any further implementation of an application in Chhattisgarhi language.

Information included about the person whose voice sample is being recorded:

- Name of the speaker
- Sex
- Age
- Caste
- Educational status
- Residing place

Using this database we will be able to differentiate a voice on the basis of:

- Native Language
- Geographic Region

CONCLUSION

- Speech recognition is the process by which a computer (or other type of machine) identifies spoken words.
- Wavelet analysis is capable of revealing aspects of data that other signal analysis techniques miss aspects like trends, breakdown points, discontinuities in higher derivatives, and self-similarity.
- This model will helpful to find out patterns and parameters for speeches and come out with general model to answer the questions like Locality.
- 22 set of sentences have been used whose conversion in different dialects are recorded for the creation of the database.
- Basically three types of dialects among 26 dialects are used in this model. These dialects are: Baigani, Baheliya,and Bhujwari.
- These standard set of sentences contains almost all major differences in the dialects through which analysis and recognition of these dialects is easily possible.

Application Area:

- The work is useful for identifying age, sex etc.
- Through regional analysis common health profile can be maintained.
- For surveillance of dacoits regional language database is helpful.
- Voice dialing.
- Banking by telephone.
- Telephone shopping.
- Database access services.
- Information services.
- Voice mail.
- Security control for confidential information areas,
- Remote access to computers.

REFERENCES

[1] M., Forsberg, “Why is Speech Recognition Difficult?”. Department of ComputingScience, Chalmers University of Technology, Gothenburg, Sweden, 2003.

[2] Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong, 30-31 Aug. 2008

INDUSTRIAL APPLICATION OF WAVELET ANALYSIS,SEIICHI SHIN,Department of System Engineering, University of Electro-Communications, Chofu, Tokyo, 182-8585, Japan

[2] D., Jurafsky and J.H., Martin, “Speech and Language Processing anIntroduction to Natural Language Processing, Computational Linguistics, and Speech Recognition”. Prentice Hall, Upper Saddle River, NJ, USA, 2000.

[3] M. A. M. Abu Shariah, R. N. Ainon, R. Zainuddin, and O. O. Khalifa,“Human Computer Interaction Using Isolated-Words SpeechRecognition Technology,” *IEEE Proceedings of The InternationalConference on Intelligent and Advanced Systems (ICIAS’07),* Kuala

[4] Lumpur, Malaysia, pp. 1173 – 1178, 2007.

D. S., Jurafsky, and J. H., Martin. *Speech and Language Processing: AnIntroduction to Natural Language Processing, ComputationalLinguistics, and Speech Recognition.*

Prentice Hall, Upper Saddle River, NJ, 2nd edition, 2008.

[5] B., Milner, “A Comparison of Front-End Configurations for Robust Speech Recognition”. *ICASSP’02*, pp. 797–800, 2002.

[6] S.K., Podder, “Segment-based Stochastic Modelings for SpeechRecognition”.

Data Mining Methods for Categorization of student for Learning in an Intelligent Tutoring System

Kiran Mishra[1], R.B. Mishra[2]
Department of Computer Engineering, I.T., B.H.U., Varanasi-221005
e-mail : [1]Kiran.mishra.bhu@gmail.com
[2]ravibm@bhu.ac.in

ABSTRACT

Data mining methods (DM) have been widely used in the classification and categorization problems. In intelligent tutoring system it requires the categorizations of students on the basis of their performances. In this work an application of data mining techniques such as: decision tree (DT) (C5.0 algorithm), ANN model, sensitivity analysis (SA) has been deployed in the data set for the categorizing the student as very good, good, average, below average and poor. Using ANN method relative importance of inputs responsible for category of student has been decided and based on this relative importance an enhanced learning strategy has been developed.

Keywords- Data mining, ANN, Categorization, Intelligent tutoring system, learning.

I. INTRODUCTION

Data Mining is the process of analysing data from different perspectives and summarizing the results as useful information. It has been defined as “the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data” ([1], [2]). The capability to predict/classify a student’s performance is very essential in intelligent tutoring system. A very promising field to attain this objective is the use of Data Mining [3]. In fact, one of the most useful Data Mining tasks in e-learning is Classification/categorization. There are different educational objectives for using classification, such as: to detect students’ misuse or game-playing [4], to discover potential student groups with similar characteristics and reactions to a particular pedagogical strategy [5], to identify learners with low motivation and find remedial actions to lower drop-out rates [6], to predict/classify students when using intelligent tutoring systems [7] etc.

In this paper we are going

- To apply data mining techniques (C5.0) for the categorization of students in terms of Very Good, Good, and Average, Below Average and Poor.
- To determine the common parameters in all the three methods of ANN and also to find out the relative importance of questionnaire types in the categorizations of students.
- Propose a method for enhanced learning and based on most important and least important question types prepare a questionnaire in which ratio of questions depends on relative importance of input types.

Apart from introduction in section 1 rest of the sections have been organized as follows: section 2 presents problem description, section 3 presents data mining methods, section 4 presents developed model for categorization of student, deciding relative importance of input responsible for category of student and an enhanced learning method for improvement of category of student. This section also represents working of the system. Section 5 presents experimentation (Section 5.1 presents experimentation with ANN Model, Section 5.2 presents experimentation with decision tree). Section 6 presents result and section 7 presents conclusion.

II. PROBLEM DESCRIPTION

The problem is firstly described by collecting information of 7 different questions types and performance of students in those question types in seven course wares (CW1- Classes and objects, CW2- Constructors, CW3- Operator overloading, CW4- Inheritance, CW5- Virtual functions, CW6- Managing console I/O operations, CW7- Working with files) focusing on the category of student. Data mining (DM) helps to extract and analyse the meaningful relationship between various question types and it also provides the relative importance of various question types based on 300 records. The types of questions are of A (Analytical), R (Reasoning), D (Descriptive), AR (Analytical-Reasoning), AD (Analytical-Descriptive), RD (Reasoning-Descriptive) and ARD (Analytical-Reasoning-Descriptive). Student’s performance can be VL (very low), L (low), M (medium), H (high) and VH (very high). Category of student take five categorical values i.e. Very Good, Good, Average, Below Average and Poor.

III. DATA MINING METHODS

This section describes the application of DM techniques such as: decision tree (DT) (C5.0 algorithm), ANN model, sensitivity analysis (SA) .The data set has 7 parameters and 200 records but only discrete number of data set is shown in the table 1.

Firstly, the information (input parameters) of different categories of a student from previous work was collected and assigned them levels as VL, L, M, H and VH. It is clear that all the 7 parameters are coded in the symbol Very high (VH), High (H), Medium (M), low (L), and very low (VL).

TABLE 1

A	R	D	AR	AD	RD	ARD	Result
L	M	L	M	M	L	M	Average
M	M	H	M	H	H	M	Good
I	M	H	M	M	H	M	Good
H	M	M	M	H	L	M	Average Below
I	VL	I	I	I	I	I	Average
I	M	M	M	I	M	I	Average
M	H	H	H	L	M	M	Average Below
I	I	I	I	VL	I	VL	Average
I	M	M	I	M	I	M	Average
H	H	M	H	L	M	H	Good
M	M	H	M	L	M	L	Average Below
I	VL	I	I	VL	I	I	Average
M	M	I	M	M	I	I	Average
H	M	M	H	M	H	M	Good
I	M	I	M	I	M	M	Average Below
I	I	I	I	VL	I	VL	Average
L	M	M	L	M	L	M	Average
H	H	M	H	I	M	H	Good
M	M	H	M	I	M	I	Average
L	VL	L	M	L	L	M	Average
H	M	M	L	M	M	M	Average
M	H	M	M	I	M	M	Average
I	M	H	M	M	I	M	Average
M	M	H	M	L	M	L	Average Below
I	VL	I	I	VL	I	I	Average
M	M	I	M	M	I	I	Average
H	M	M	H	M	H	M	Good
L	M	L	M	L	M	M	Average Below
I	I	I	I	VL	I	VL	Average

IV. DEVELOPED MODEL

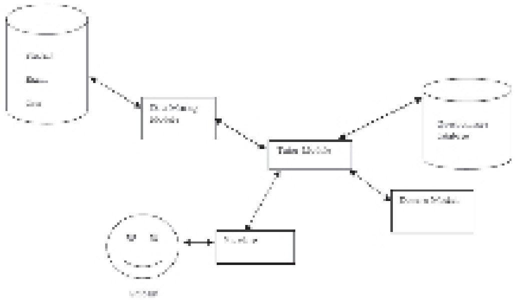
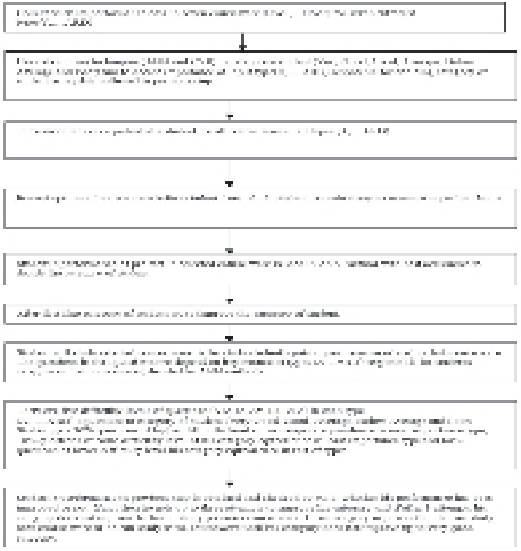


Fig 1 Developed Model

A. Implementation steps for the System

- Data mining module takes students performance data in seven courseware CW1,…,CW7 (CW1- Classes and objects, CW2- Constructors, CW3- Operator overloading, CW4- Inheritance, CW5- Virtual functions, CW6- Managing console I/O operations, CW7- Working with files from C++ curriculum) for seven different types (A,…,ARD) of questionnaire stored in database.
- Data mining module uses data mining techniques (ANN and C5.0) to categorize student (Very Good, Good, Average, Below Average and Poor) and to decide importance of input type (A,…,ARD) responsible for deciding category of student using data taken from previous step.
- Tutor module takes a pretest of a student for all course ware in all types (A,…,ARD) of questionnaires.
- Tutor module presents options of course ware before student from which student can select any courseware as per his choice.
- Tutor module uses student's performance of pre-test in selected course ware through data mining module (C5.0 method) to decide the category of student
- After deciding category of student tutor module tries to improve the category of student.
- Student will study selected course ware. After study student's gets a questionnaire related to that course ware. The types and ratio of questions in that questionnaire is decided by tutor module based on importance of questionnaire types (A,…,ARD) responsible for student's category in that courseware (decided by ANN method).
- There are five difficulty levels of questions (VD, D, Av, Es, VEs) in each type (A,..,ARD) equivalent to category of student (Very Good, Good, Average, Below Average and Poor). Tutor module decides that student will get 30% questions of higher difficulty level (There are five difficulty levels of questions i.e. VD (Very Difficult, Df (Difficult), Av (Average), Es (Easy) and VEs (Very easy) of his category equivalence in most important type of input, 10% questions of same difficulty level of his category equivalence in least importance type and 60% questions of lower difficulty level of his category equivalence in rest of types.
- Student's performance in previous step is obtained and checked by data mining method (ANN method) whether his performance has been improved or not. If not then he gets up to three attempts to improve his category and if after 3 attempts his category does not improve he has to study previous course ware. If his category improves then he can study next course ware or he can study same courseware until his category does not improve up to very good category.

Fig 2 Steps for implementation of the system



V. EXPERIMENTATION

The experimentations were performed with the data mining methods: decision tree, ANN and sensitivity analysis on the dataset of 300 records. The various experimentations with their corresponding results are given below. In all the models given below 300 records have been used and among which 200 records are used for training and 100 records are used for testing the accuracy of models.

A. ANN Model

Three ANN methods Quick, Dynamic based on error back propagation algorithm and radial basis function network (RBFN) are deployed. In all the three methods 7 variables are used as input, which have categorical values. Five categorical variables have five stages while 2 categorical variables have 4 stages. Therefore total number of neuron in input layer of ANN is (5*5+2*4=33) neurons. In this model 200 datasets for training and 100 datasets for testing have used in the artificial neural network model to verify accuracy of the model constructed.

Observation for CW1

Quick Method

In this model among 100 test cases 100 cases are correctly classified where as 0 case is classified incorrectly. Therefore the final accuracy is {((100-0)/100)*100)=100%}100%, and error rate, 0% for the mining results using the artificial neural network model. Parameters used in the neural networks of this study have been shown in Table 2.

Dynamic Method

In this model among 100 test cases 98 cases are correctly classified where as 2 cases are classified incorrectly. Therefore the final accuracy is {((100-2)/100)*100)=98%}98%, and error rate, 2% for the mining results using the artificial neural network model. Parameters used in the neural networks of this study have shown in Table 2.

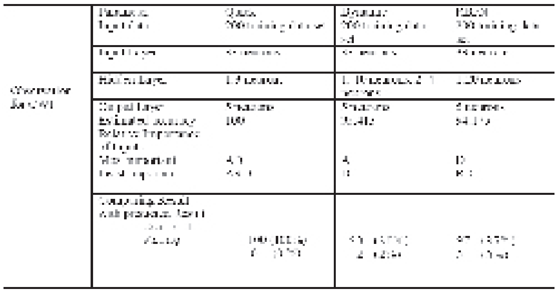
Radial Basis Function Network (RBFN)

In this model among 100 cases 97 cases are correctly classified where as 3 cases are classified incorrectly. Therefore the final accuracy is {((100-3)/100)*100)=97%}97%, and error rate, 3% for the mining results using the artificial neural network model. Observations and parameters used in the neural networks of this study for all seven course wares have been shown in Table 2.

TABLE 2
PARAMETERS AND RESULTS OF ANN

In the same way we have been got parameters of ANN for other courseware. Due to lack of space it has not been provided here.

Result from Observation of table 2



Observation for CW1	Parameters Input data	Quick 200 training data set	Dynamic 200 training data set	RBFN 200 training data set
	Input Layer	33 neurons	33 neurons	33 neurons
	Hidden Layer	[illegible] neurons	1. [illegible] neurons 2. [illegible] neurons	[illegible] neurons
	Output layer Estimated accuracy Relative Importance of Input Most important Least important	5 neurons 100 [illegible] [illegible]	5 neurons [illegible] A B	5 neurons [illegible] D [illegible]
	Comparing Result with predicted field Correct Wrong	100 (100%) 0 (0%)	98 (98%) 2 (2%)	97 (97%) 3 (3%)

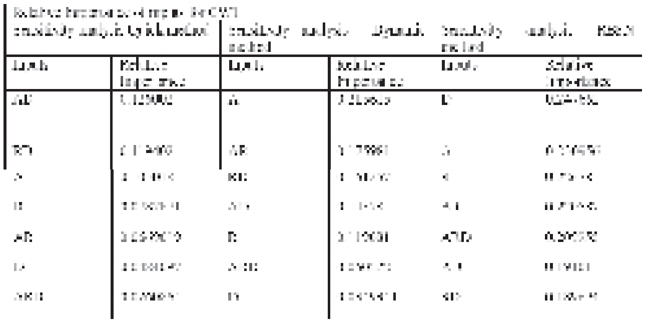
1. For CW1 the best accurate ANN method is Quick method (with accuracy of 100%).
2. For CW2, we can choose any method of ANN as accuracies of all the methods are same (99%).
3. For CW3 we can choose Quick or Dynamic methods as both methods of ANN have equal (100%) accuracy and higher accuracy than RBFN method.
4. For CW4 the best accurate method is Dynamic method (with highest accuracy 99%).
5. For CW5 the best accurate method is Dynamic method (with highest accuracy 100%).
6. For CW6, we can choose any method of ANN as accuracies of all the methods are same (99%).
7. For CW7 the best accurate ANN method is RBFN method (with accuracy of 99%).

TABLE 3
RELATIVE IMPORTANCE OF INPUTS FOR CW1

In the same way we obtain relative importance of inputs for other courseware.

B. Decision tree (C5.0 algorithm)

In this model 200 dataset for training and 100 dataset for testing



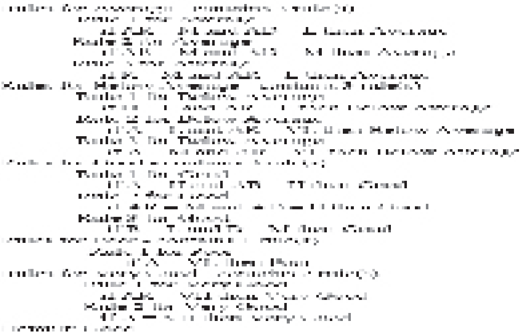
Relative Importance of Inputs for CW1

Sensitivity analysis by Quick method		Sensitivity analysis Dynamic method		Sensitivity analysis RBFN method	
Inputs	Relative Importance	Inputs	Relative Importance	Inputs	Relative Importance
AE	[illegible]	A	[illegible]	D	[illegible]
RD	[illegible]	AE	[illegible]	A	[illegible]
A	[illegible]	RD	[illegible]	[illegible]	[illegible]
B	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
AD	[illegible]	B	[illegible]	ATD	[illegible]
D	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	D	[illegible]	[illegible]	[illegible]

are deployed in decision tree algorithm to calculate the error rate report and obtain its tree-structure, as shown in fig. 3. The rules in the fig. 3 have been generated by the program based upon the tree structure as shown in the Table 4. There are 12 sets of rules in fig 3.

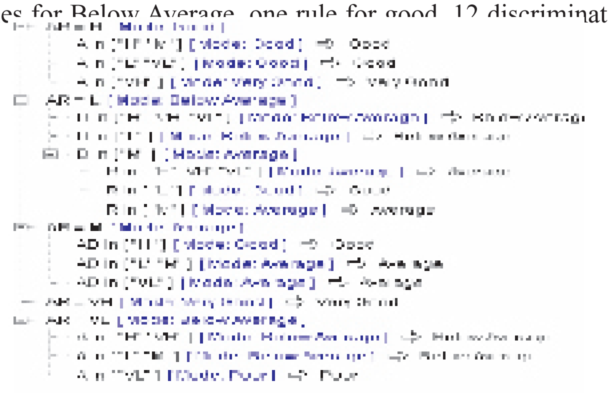
Fig 3 Rules for categorization of students for cw1by decision tree (C5.0 algorithm)

TABLE 4
THE TREE GENERATED BY DECISION TREE (C5.0 ALGORITHM) FOR CW1)



```
Rules for Average - contains 3 rule(s)
          Rule 1 for Average
             [illegible]
          Rule 2 for Average
             [illegible]
          Rule 3 for Average
             [illegible]
Rules for Below Average - contains 3 rule(s)
          Rule 1 for Below Average
             [illegible]
          Rule 2 for Below Average
             [illegible]
          Rule 3 for Below Average
             [illegible]
Rules for Good - contains 3 rule(s)
          Rule 1 for Good
             [illegible]
          Rule 2 for Good
             [illegible]
          Rule 3 for Good
             [illegible]
Rules for Poor - contains 1 rule(s)
           Rule 1 for Poor
              [illegible]
Rules for Very Good - contains 2 rule(s)
           Rule 1 for Very Good
             [illegible]
           Rule 2 for Very Good
             [illegible]
Default : [illegible]
```

In the same way rule set and decision tree for all courseware are generated. After obtaining the rules for category; the testing datasets is used to assess the accuracy of these rules. From the above table it is found that there are two rules for very good, three rules for Good, three rules for average, three rules for Below Average, one rule for good, 12 discriminate rules for five different types of category such as Very Good, Good, Average, Below Average and Poor are shown in Table 4. According to the above-mentioned tables, the overall accuracy of the C5.0 algorithm is 100%.. By default the model predict any student as Average. By default the model predict any student as Good for CW1.



```
AR = H [Mode: Good]
    A in ["H" "M"] [Mode: Good] ⇒ Good
    A in ["L" "VL"] [Mode: Good] ⇒ Good
    A in ["VH"] [Mode: Very Good] ⇒ Very Good
AR = L [Mode: Below Average]
    D in ["H" "VH" "VL"] [Mode: Below Average] ⇒ Below Average
    D in ["L"] [Mode: Below Average] ⇒ Below Average
    D in ["M"] [Mode: Average]
        R in ["H" "VH" "VL"] [Mode: Average] ⇒ Average
        R in ["L"] [Mode: Good] ⇒ Good
        R in ["M"] [Mode: Average] ⇒ Average
AR = M [Mode: Average]
    AD in ["H"] [Mode: Good] ⇒ Good
    AD in ["L" "M"] [Mode: Average] ⇒ Average
    AD in ["VL"] [Mode: Average] ⇒ Average
AR = VH [Mode: Very Good] ⇒ Very Good
AR = VL [Mode: Below Average]
    A in ["H" "VH"] [Mode: Below Average] ⇒ Below Average
    A in ["L" "M"] [Mode: Below Average] ⇒ Below Average
    A in ["VL"] [Mode: Poor] ⇒ Poor
```

VI. RESULT

Fig. 4 to fig. 8 shows the output of the system. Fig. 4 shows questionnaire for student for pre-test provided by questionnaire agent. Fig. 5 shows courseware wise performance of student in pretest. Fig. 6 shows options for study to the student provided by tutor agent. Fig 7 shows category of student decided by data mining agent based on his performance in pre-test questionnaire. Fig. 8 shows improved category of student by using proposed enhanced learning method.

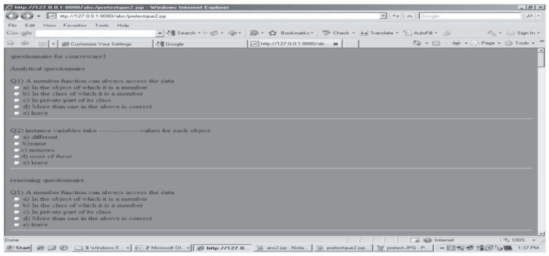
Fig 4 Questionnaire for pre-test

Fig 5 Courseware wise performance of student in pre-test

Fig 6 Options for study after pretest of a student

Fig 7 Category decided by C5.0 from pretest data of student in a selected courseware

Fig. 9. Improved category of student by using system



http://127.0.0.1:8080/abc/pretestque2.jsp - Windows Internet Explorer

questionnaire for courseware1

Analytical questionnaire

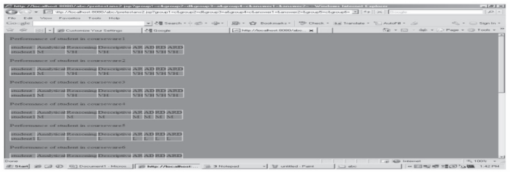
Q1) A member function can always access the data
- ○ a) In the object of which it is a member
- ○ b) In the class of which it is a member
- ○ c) In private part of its class
- ○ d) More than one in the above is correct
- ○ e) leave

Q2) instance variables take -------------values for each object
- ○ a) different
- ○ b)same
- ○ c) nonzero
- ○ d) none of these
- ○ e) leave

reasoning questionnaire

Q1) A member function can always access the data
- ○ a) In the object of which it is a member
- ○ b) In the class of which it is a member
- ○ c) In private part of its class
- ○ d) More than one in the above is correct
- ○ e) leave

In this work we used data mining approach to determine the rules for categorization of student, importance of the inputs and the accuracy of classification and categorization of student.



Performance of student in courseware1

student	Analytical	Reasoning	Descriptive	AR	AD	RD	ARD
student1	M	VH	VH	VH	VH	VH	VH

Performance of student in courseware2

student	Analytical	Reasoning	Descriptive	AR	AD	RD	ARD
student1	M	VH	VH	VH	VH	VH	VH

Performance of student in courseware3

student	Analytical	Reasoning	Descriptive	AR	AD	RD	ARD
student1	M	VH	VH	VH	VH	VH	VH

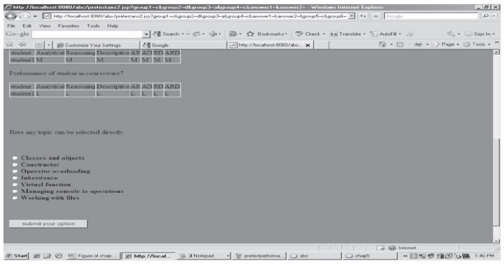
Performance of student in courseware4

student	Analytical	Reasoning	Descriptive	AR	AD	RD	ARD
student1	M	M	M	M	M	M	M

Performance of student in courseware5

student	Analytical	Reasoning	Descriptive	AR	AD	RD	ARD
student1	L	L	L	L	L	L	L

Performance of student in courseware6



http://localhost:8080/abc/pretestans2.jsp?group1=c&group2=d&group3=a&group4=c&answer1=&answer2= - Windows Internet Explorer

student	Analytical	Reasoning	Descriptive	AR	AD	RD	ARD
student1	M	M	M	M	M	M	M

Performance of student in courseware7

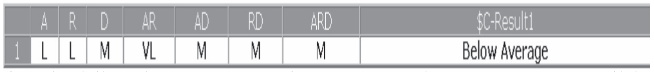
student	Analytical	Reasoning	Descriptive	AR	AD	RD	ARD
student1	L	L	L	L	L	L	L

Here any topic can be selected directly.

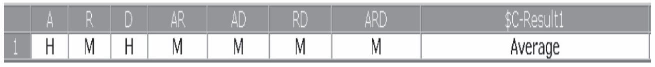
- ○ Classes and objects
- ○ Constructor
- ○ Operator overloading
- ○ Inheritance
- ○ Virtual function
- ○ Managing console io operations
- ○ Working with files

submit your option

We have used ANN models (Quick, Dynamic and RBFN) for deciding importance of Input question types responsible for category of student. A comparative view of all the ANN methods has been represented. A method for improving category of student in intelligent tutoring system by enhanced learning has been developed.



	A	R	D	AR	AD	RD	ARD	$C-Result1
1	L	L	M	VL	M	M	M	Below Average



	A	R	D	AR	AD	RD	ARD	$C-Result1
1	H	M	H	M	M	M	M	Average

REFERENCES

[1] W. Frawley, G. Piatetsky-Shapiro, and C. Matheus, C.. *Knowledge Discovery in Databases: An Overview*. AI Magazine, p. 213-228. 1992.

[2] U.M. Fayyad, G. Pitatesky-Shapiro, P. Smyth, R. Uthurasamy, *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press. ,1996.

[3] C. Romero, S. Ventura, *Educational Data Mining: a Survey from 1995 to 2005*. Expert Systems with Applications, , 33(1), p .135-146, 2007.

[4] R. Baker, , , A. Corbett, K. Koedinger, *Detecting Student Misuse of Intelligent Tutoring Systems. Intelligent Tutoring Systems*. Alagoas,. p. .531–540. 2004.

[5] G. Chen, C.. Liu, K.. Ou, B. Liu,. *Discovering Decision Knowledge from Web Log Portfolio for Managing Classroom Processes by Applying Decision Tree and Data Cube Technology*. Journal of educational Computing Research, 23(3), p. .305–332. 2000.

[6] M. Cocea, S. Weibelzahl, *Can Log Files Analysis Estimate Learners' Level of Motivation?* Workshop on Adaptivity and User Modeling in Interactive Systems, Hildesheim,. p. .32-35, 2006.

[7] W. Hämäläinen, M. Vinni, *Comparison of machine learning methods for intelligent tutoring systems.* Conference Intelligent Tutoring Systems, Taiwan, p. 525–534., 2006.

Segmentation and Characterization of Brain MR Image Regions Using Som and Neuro Fuzzy Techniques

Anamika Ahirwar [1], R.S. Jadon [2]

[1] *Department of Computer Application, Madhav Institute of Technology & Science, Gwalior*
E-mail: aanamika77@gmail.com

[2] *Department of Computer Application, Madhav Institute of Technology & Science, Gwalior*
E-mail: rsj_mits@yahoo.com

ABSTRACT

Medical Image analysis is an active research area today. Due to enhancements in imaging device technologies, now it is possible to capture very high resolution images which are suitable for automated analysis. Segmentation of anatomical regions of the brain is a critical problem. In this paper, a method for segmenting MR images based on SOM neural network is proposed. Firstly, the pixels are clustered, based on their grayscale and spatial features, where the clustering process is accomplished with a SOM network. Clustering separates different regions. These regions could be regarded as segmentation results reserving some semantic meaning. This approach thus provides a feasible new solution for image segmentation like Gray matter, White matter, CSF and tumor. We further characterize the tumor region by extraction of the features like area, entropy, mean and standard deviation.

Keywords—Magnetic resonance Imaging (MRI), Self Organizing Map (SOM), best matching unit (BMU), gray level co-occurrence matrices (GLCM), inverse difference mean (IDM).

1. INTRODUCTION

Magnetic resonance images offer anatomical information for medical examination more accurately than other medical images such as X-ray, ultrasonic and CT images. The segmentation and recognition algorithm of MR images are becoming important for good clinical information. The segmentation technique is extensively used by the radiologists to segment the input medical image into meaningful regions. The specific application of this technique is to detect the tumor region by segmenting the abnormal MR input image. Clustering is one of the widely used image segmentation techniques which classify patterns in such a way that samples of same group are more similar to one another than samples belonging to different groups [1]. Fuzzy C-means algorithm is widely performed because it allows pixels to belong multiple classes with varying degrees of membership. But it is a time consuming process [2]. Image segmentation techniques can be classified as classifier such as self organizing map (SOM) and feature vector clustering or vector quantization. Vector quantization is a very effective model for image segmentation process [7]. Vector quantization portioning an n-dimensional vector space into M regions when all the points in each region are approximated by the representation vector X_i associated with that region. Vector quantization processes in two steps, one is the training process which determines the set of codebook vector according to the probability of the input data and the other is the encoding process which assigns input vectors to the code book vectors. Vector quantization process has been implemented in terms of the competitive learning neural network (CLNN) [5]. Self Organizing Map (SOM) is a member of the CLNNs so this can be the best choice when implementing vector quantization using neural network [3,4]. The importance of SOM for vector quantization is mainly due to the similarity between the competitive learning process used in the SOM and the vector quantization procedure. It is not possible to determine the correct number of regions in the segmented image. FCM algorithm is a popular fuzzy clustering algorithm which classified as constrained soft clustering algorithm. A soft clustering algorithm finds a soft partition of a given data set by which an element in the data set may partially belong to multiple clusters.

2. IMPLEMENTATION OF SOM AND FUZZY C-MEANS

2.1. SOM algorithm

1. Randomize the map's nodes' weight vectors.
2. Grab an input vector.
3. Traverse each node in the map
   1. Use Euclidean distance formula to find similarity between the input vector and the map's node's weight vector
   2. Track the node that produces the smallest distance (this node is the best matching unit, BMU)
4. Update the nodes in the neighbourhood of BMU by pulling them closer to the input vector.

$$W_i(t+1) = W_i(t) + h_{ci}(t) \times \{x(t) - W(t)\}$$

5. Increment t and repeat from step 2 while t < λ, where t is current iteration and λ is limit on time iteration .

2.2. Fuzzy c-means algorithm (FCM)

Step1:

Initialize the membership matrix, U=[u_{ij}].

Step 2:

At k_{th} number of iteration:

Calculate the center vectors c_i with u_{ij}

Step3: $c_i = \sum_{j=1}^{n} u_{ij}^{m} \times x_j \Big/ \sum_{j=1}^{n} u_{ij}^{m}$ U p matrix U for the k_{th} steps and $(k+1)_{th}$ step.

where d_{ij} = $u_{ij} = 1 \Big/ \sum_{k=1}^{c} \left(d_{ij}/d_{kj}\right)^{2/(m-1)}$

Steps4:

If ||U(k+1)-U(k)|| < ε then STOP; otherwise return to step2

3. Feature extraction

The purpose of feature extraction is to reduce the original data set by measuring certain properties or features that distinguish one input pattern from another. Three textural features namely contrast energy and entropy based on the gray level co-occurrence matrices (GLCM) have been used in this work. GLCM $\{P_{(d,\theta)}(i,j)\}$ represents the probability of occurrence of the pair levels(i,j) separated by a given distance *d* at angle θ. In this paper we are considering value of θ=0º and distance (d)=1. The commonly used unit pixel distances and the angles are 0º, 45º, 90º and 135º. The features are calculated using the formulae given below.

Contrast:

Entropy:

$$S_c = \sum_{i=0}^{K}\sum_{j=0}^{M}(i-j)p(i,j)$$

Energy:

$$S_e = -\sum_{i=0}^{K} p(i,j)\log\{p(i,j)\}$$

Mean:

$$S_{sk} = \sum_{i=0}^{K}\sum_{j=0}^{M} p^{2}[i][j]$$

Inverse Difference Moment:

$$S_m = \frac{1}{m \times n}\left\{\sum_{i=0}^{K}\sum_{j=0}^{M}\left(ip[i,j] + jp[i,j]\right)\right\} \qquad i \neq j$$

Standard Deviation:

$$s_d = \sqrt{\left(\left(\frac{1}{(m-1)(n-1)}\right)\sum_{i=1}^{m}\sum_{j=1}^{n}\left(p[i][j]-s_m\right)^{2}\right)}$$

Each set of features are individually normalized to the range of 0 to 255. These features work well especially for MR brain tumor images.

4. RESULTS

The result of the implementation of the neuro fuzzy region classification process is discussed in this section. Any computer aided analysis the execution time is one of the important parameters for analyzing medical images. We have calculated the number of pixels affected by the tumor cells. We worked on six MR brain normal and abnormal images in this paper. The images used are 256x256 gray level images with intensity value ranges from (0 to 255). We have calculated the tumor properties like location, energy, entropy, IDM, contrast, mean, standard deviation and image properties like name of the region, type of the region, average gray value of region, area of that region, centroid of the region. The execution time of this technique found 45 to 65 seconds and detected tumor pixels are 588 (approx). Table-1 and Table-2 shows the input abnormal and normal MR brain image and their image and tumor properties respectively.

5. CONCLUSION

We have implemented a neuro fuzzy based segmentation process to detect brain tumor. We studied the performance of the MRI image in terms of execution time and tumor pixels detected. Fuzzy clustering technique was implemented to detect tumor of brain MRI. We have achieved a higher value of detected tumor pixels. In this paper we have input fixed size and noiseless MR image (size=256x256) only. We can find whether tumor is present or not in the input MR image. We have determined a limit for detected abnormal pixels, if number of abnormal pixels are less than this range than no tumor, else tumor detected.

6. REFERENCES

[1] Wen, P, Zheng, L and Zhou, J, "Spatial credibilistic clustering algorithm in noise image segmentation", IEEE International Conference on Industrial Engineering and Engineering Management, pp: 543 -547, 2007.

[2] Jie Yu, Peihuang Guo, Pinxiang Chen, Zhongshan Zhang and Wenbin Ruan, "Remote sensing image classification based on improved fuzzy c -means", Ceo-Spatial Information Science, vol.11, no.2, pp:90-94, 2008.

[3] Scherf, A. and G. Roberts, 1990. Segmentation using neural networks for automatic thresholding, in: S. Rogers (ed.), Proc. SPlE Conference on Applications of Artificial Neural Networks (Orlando, FL, 1294), pp: 118-124.

[4] Lin, W., E. Tsao and C. Chen, 1991. Constraint satisfaction neural networks for image segmentation, In: T.Kohonen, K. Mkisara, 0. Simula and J. Kangas (eds.), Artificial Neural Networks (Elsevier Science Publishers), pp: 1087-1090.

[5] Martinelli, G., L.P. Licotti and S. Ragazzini, 1990. Nonstationary lattice quantization by a selforganizing Neural network, Neural Networks 3 (4): 385-393.

[6] Hung, M.C. and D.L. yang, 2001. An Efficient Fuzzy C means Clustering Algorithm, Data mining, ICDM 2001, Proceedings IEEE International conference on pp: 225-232.

[7] Ahalt, S.C., A.K. Krishnamurthy, P. Chen and D.E. Melton, 1990. Competitive learning algorithms for Vector quantization, Neural Networks 3 (3): 277-290.

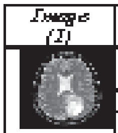
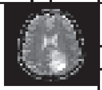
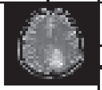
Image (1)	*Image Properties*	WM	GM	CSF	Tumor	*Tumor Properties (Tumor Detection)*	
	Name of region	White matter	Gray matter	C SF	Tumor	Location	Lower Right
	Type of region	Medium	Medium	Medium	Small	Energy	7.6293945312500005E-12
	Average gray scale value of region	34.0	29.0	34.0	52.0	Entropy	-2.2818448713671303E-4
	Area of region (pixels)	17385	10960	17382	588	IDM	7.8125E-7
	Centroid of region	-	-	-	200,152	Contrast	1.5624999999999999E-5
						Mean	1.5624999999999999E-5
						Standard deviation	6.6392111675652282E-7
Image (2)	*Image Properties*	WM	GM	CSF	Tumor	*Tumor Properties (Tumor Detection)*	
	Name of region	White matter	Gray matter	C SF	Tumor	Location	Lower Right
	Type of region	Medium	Medium	Medium	Small	Energy	1.7014579925843832E-11
	Average gray scale value of region	32.0	29.0	32.0	52.0	Entropy	-3.022768067051995E-4
	Area of region (pixels)	16447	11754	16446	444	IDM	5.358367626886146E-6
	Centroid of region	-	-	-	200,152	Contrast	2.1433470507544594E-5
						Mean	2.1433470507544594E-5
						Standard deviation	1.0747314090217674E-6
Image (3)	*Image Properties*	WM	GM	CSF	Tumor	*Tumor Properties (Tumor Detection)*	
	Name of region	White matter	Gray matter	C SF	Tumor	Location	Lower Right
	Type of region	Medium	Medium	Medium	Small	Energy	9.528800842388143E-11
	Average gray scale value of region	31.0	30.0	31.0	59.0	Entropy	-5.981585272008135E-4
	Area of region (pixels)	16265	12030	16270	164	IDM	8.134631403581841E-6
	Centroid of region	-	-	-	184,152	Contrast	4.5553935584005832E-5
						Mean	4.5553935584005832E-5
						Standard deviation	3.02034258405921E-6

Table-2: Shows the input abnormal MR brain images and their properties.

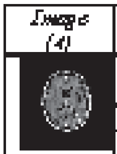
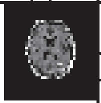
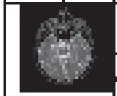
Image (4)	Image Properties	WM	GM	CSF	Tumor	Tumor Properties (Abnormal/Normal)	
	Name of region	White matter	Gray matter	C SF	Tumor	Location	UpperLeft
	Type of region	Medium	Small	Medium	Small	Energy	5.122274160385132E-9
	Average gray scale value of region	45.0	48.0	45.0	-256.0	Entropy	-0.0026573156945395152
	Area of region (pixels)	9048	3952	9056	-8	IDM	4.57763671875E-5
	Centroid of region	-	-	-	184,120	Contrast	2.44140625E-4
						Mean	2.44140625E-4
						Standard deviation	6.579029902225935E-5
Image (5)	Image Properties	WM	GM	CSF	Tumor	Tumor Properties (Abnormal/Normal)	
	Name of region	White matter	Gray matter	C SF	Tumor	Location	UpperRight
	Type of region	Medium	Small	Medium	Small	Energy	4.190951585769653E-9
	Average gray scale value of region	45.0	47.0	45.0	-54.0	Entropy	-0.0026864453000827425
	Area of region (pixels)	9082	4048	9096	-20	IDM	3.0517578125E-5
	Centroid of region	-	-	-	120,168	Contrast	2.44140625E-4
						Mean	2.4414062499999997E-4
						Standard deviation	4.671125922515334E-5
Image (6)	Image Properties	WM	GM	CSF	Tumor	Tumor Properties (Abnormal/Normal)	
	Name of region	White matter	Gray matter	C SF	Tumor	Location	UpperRight
	Type of region	Medium	Small	Medium	Small	Energy	3.725290298461914E-9
	Average gray scale value of region	48.0	38.0	48.0	-99.0	Entropy	-0.0027076061740622863
	Area of region (pixels)	13955	7203	13952	-16.0	IDM	1.52587890625E-5
	Centroid of region	-	-	-	104,72	Contrast	2.44140625E-4
						Mean	2.44140625E-4
						Standard deviation	3.621530344171402E-5

Table-3: Shows the input normal MR brain images and their properties

Static Hand Gesture Recognition System for Sign Languages:Using Error Back propagation neural network & Skin Color Modeling

* Dilip Singh Sisodia
Assistant Professor
Deptt. Of Computer Sc. & Engg.
email : sisodia_dilip@rediffmail.com

Dr. Shrish Verma
Associate Professor & Head
Deptt. Of Information Technology & EC
National Institute of Technology Raipur

ABSTRACT:

In this paper, we propose a system that automatically recognizes and classifies an image of a hand signing a number. A new method for Static hand gesture recognition is proposed which is based on Multilayer Error Back propagation network & pixel intensity based Skin Color detection method. To recognize static hand gestures, the system trained with image data sets of various hand Gestures this image data sets will contain number of images of same hand gestures. After training the network for all the static hand gestures, it should be able to recognize the input Hand gestures taken through already stored media file of gestures or directly from video capturing device like web camera.

INTRODUCTION

Sign language is one form of communication for the hearing and speech impaired. Similar to spoken language, there is no universal sign language. Sign language is itself a separate language with its own grammar and rules. Some signs are expressed as static gestures while others incorporate some dynamic hand movements. For static gestures, the prominent sign is captured within a specific time frame. For dynamic gestures, a sequence of finger and hand positions Needs to be identified and analyzed in order to be recognized. The focus of this paper is on static gestures with a single hand.

In this paper, we have explored a specific area in which artificial neural network & pixel intensity based skin color detection methods are used for recognition of the static hand gestures. I would like to present what I have learned and accomplished for this interesting and exciting topic. I present a simplest approach to find a solution to hand gesture recognition. It will recognize static hand gestures. Previous systems have used data gloves or markers for input in the system.

RELATED WORK

Automatic sign recognition has been investigated since around 1995[19]. Researchers tried a variety of techniques, such as fuzzy logic, neural networks, and Hidden Markov Models (HMMs), mostly on small vocabularies. Vogler and Metaxas achieved a 96% recognition rate on a 22-word vocabulary using parallel HMMs. Chen reported a 92% success rate with whole-word HMMs for a 5113-word vocabulary. Both used instrumented gloves and magnetic trackers to capture their data. Zieren and Kraiss and Bowden use cameras to record signs. They achieve 98% recognition rates for a 152- and a 43-word vocabulary respectively. Zieren and Kraiss use whole-word HMMs, whereas Bowden use Markov chain models to recognize signs [20].

Research on hand gestures can be broadly classified into two categories. The first category, glove-based analysis, employs sensors (mechanical or optical) attached to a glove that transduces finger flexions into electrical signals for determining the hand posture.

The second category, vision based analysis, is based on the way human beings perceive information about their surroundings, yet it is probably the most difficult to implement in a satisfactory way.

After comparing above techniques on various parameters like cost, user comfort, computing power, portability, and accuracy etc. Our preference went out to a vision-based method.

First, with vision-based methods signers do not have to wear hardware (sensors and trackers) on their bodies, so they are less encumbered.

Secondly, vision-based methods have access to the face, as opposed to sensor-based methods. Facial information is important in sign languages; therefore, vision-based methods have more future perspectives [17].

PROPOSED METHOD

The principal assumption of the proposed gesture recognition method is that the images include exactly one hand. Furthermore, the arm is roughly vertical, the palm is facing the camera and the fingers are either raised or not. Finally, the image background is plain and uniform.

The proposed method consists of five main stages:

(a) The first thing for the program to do is to read the image database. A *for* loop is used to read an entire folder of images.

(b) To convert these images in to equivalent Binary images one by one

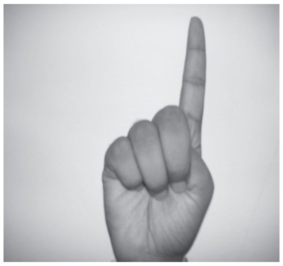
(c) Crop these digitized Binary images & again redraw for same positioning & Orientation

(d) Pass these digitized Binary Images to back propagation Neural Network for Training of the network.

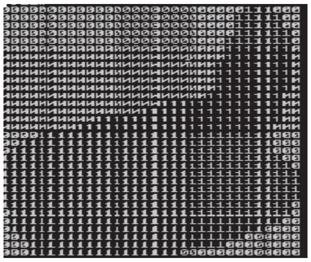
(e) After Successful completion of Training we can Test the network

In the proposed method the Image is converted in to Binary Image with the help of skin color detection algorithm, since color is a robust feature that performs highly even if the conditions of the environment are not exactly the requisite. First of all, color is invariant to rotation and scaling as well as to geometric variations of the hand. Secondly and importantly it allows a simple and fast processing of the input image. On the other hand, skin color varies quite dramatically. It is vulnerable to changing lighting conditions and it differs among people. The perceived variance, however, is really a variance in luminance due to the fairness or the darkness of the skin.

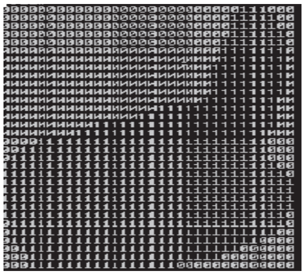
For the common orientation and same positioning of all the Binary images, we will crop the images with the help of image filter, which extract a given rectangular region of an existing Image and provide a source for a new image containing just the extracted region. This cropped Binary image will be passed to Input layer of Back propagation network for Further Training and testing of neural network



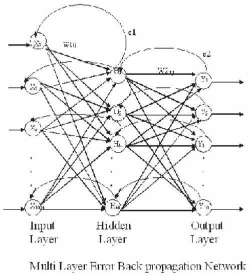
Numeric'1' in



Digitized array



Array of Pixels



Multi layer N/W

For experimental purpose of our project, we have selected and trained a multilayer Back propagation Network to recognize static hand Gestures. The training data set (images) we have used is obtained from a capturing device such as web camera. For example system should learn a static pose of numeric digits '1' (see figure for Sign Language gesture for '1'). We are using a 32x32 resolution for an image.

CONCLUSION & FUTURE WORK

The Artificial Neural Network design in this paper has the ability to recognize hand gestures without affecting either by shift in position or by a small distortion in the shape of input gesture. It also has a function of organization, which processes by means of "Learning with a teacher" (Supervised learning). If sets of input gestures are repeatedly presented to it, it gradually acquires the ability to recognize these gestures. It is not necessary to give any instructions about the categories to which the stimulus gestures should belong. The performance of the network has been demonstrated by simulating on a computer. We do not advocate that the network is a complete model for the mechanism of Hand gesture recognition in the brain, but I propose it as a working design for some neural mechanisms of static gesture recognition.

This work can be extended to recognize static hand gestures faster and more accurately using some robust skin color detection algorithms and performance optimization techniques of neural networks. We can also recognize dynamic hand gestures by using time varying simulation model like Hidden Makov Model (HMM).

One can also track an object like cursor on the screen by positioning the tip of the finger to the cursor location. As one moves the finger in front of the camera, the cursor will move accordingly.

REFERENCES

[1]. Kapuscinski, T.; Wysocki, M.; "Hand gesture recognition for man-machineinteraction" Robot Motion and Control, 2001 Proceedings of the Second International Workshop on 18-20 Oct. 2001 Page(s):91 – 96

[2]. Incertis, I.G.; Garcia-Bermejo, J.G.; Casanova, E.Z.; "Hand Gesture Recognition for Deaf People Interfacing" Pattern Recognition, ICPR 2006. International Conference Volume 2, 2006 Page(s):100 – 103

[3]. [4].Dionisio, C.R.P.; Cesar, R.M., Jr.; "A project for hand gesture recognition" Computer Graphics and Image Processing, 2000. Proceedings XIII Brazilian Symposium on1 7-20 Oct. 2000 Page(s):345

[4]. Licsar, A.; Sziranyi, T.;" Dynamic training of hand gesture recognition system" Pattern Recognition, 2004. ICPR2004. Proceedings of the 17th International Conference on Volume 4, 23-26 Aug. 2004 Page(s):971- 974 Vol.4

[5]. Vladimir I. Pavlovic, Rajeev Sharma, and Thomas S. Huang "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review" IEEE Transactions on Pattern Analysis & Machine Intelligence, Vol. 19, NO. 7, July 1997

[6]. Murphy, O.J., "Nearest neighbor pattern classification perceptrons," Proceedings of the IEEE, vol. 78, no. 10, pp. 1595-1598, October 1990

[7]. Dan W. Patterson."Artificial Neural networks Theory and Applications". Prentice Hall.

[8]. Simon Haykin, "Neural Networks A ComprehensiveFo undation"Pearson Education Asia.

[9]. "A Survey on Pixel-Based Skin Color DetectionTechn iques",Vladimi Vezhnevets, Vassili Sazonov and Alla Andreeva Moscow State University, Moscow, Russia.

[10]. http://java.sun.com/products/archive/j2se/1.0/

[11]. "Java Media Framework", http://java.sun.om/products/javamedia/jmf/index.jsp

[12]. "Processing Image Pixels using Java Getting Started" By Richar G.Baldwin http://www.developer.com/java/other/Article.hp/3403921

[13] Kenny Teng, Jeremy Ng, Shirlene Lim "Computer Vision Based Sign Language Recognition for Numbers"

[14]. "The Gesture Recognition Homepage"

http://www.cybernet.com/~ccohen/

[15]. James MacLean, Rainer Herpers, Caroline Pantofaru,Laura Wood, Konstantinos Derpanis, Doug Topalovic,John Tsotsos, "Fast Hand Gesture Recognition for Rea-Tim-teleconferencing Applications",University of Toronto, University of Applied Science.

[16]. Phung, S.L.; Bouzerdoum, A.; Chai, D. " Skin segmentation using color and edge information" Signa Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on Volume 1, Issue , 1-4 July 2003 Page(s): 525 - 528 vol.1

[17]. Ueda,E.Matsumoto,Y.Imai,M., Ogasawara, T."A hand-pose estimation for vision- based human interfaces"Industrial Electronics, IEEE Transactions on Volume 50, Issue 4, Aug. 2003 Page(s): 676 - 684

[18]. Vamplew, P. and A. Adams, Recognition of Sign Language Gestures using neural Networks. Australian Journal of Intelligent Information Processing Systems, 1998. **5**(2): p. 94-102.

[19]. Waldron, M.B. and S. Kim, Isolated ASL sign recognition system for deaf persons. IEEETransactionsonRehabilitation Engineering, 1995. **3**(3): p. 261-271.

[20]. Ten Holt, g; P Hendriks; TC Andringa; EA Hendriks; MJT Reinders. Automatic recognition of Dutch sign language. Twelvth annual conference of the Advanced School for Computing and Imaging 2006, 157-164.

Hybrid Mechanism for Mutual Authentication

Sridhar S [1], Vimala Devi.K [2]
[1] PG Scholar Dept of Cse (PG), [2] Dept of MCA (PG),
Sri Ramakrishna Engineering College,
Coimbatore, –641022, Tamilnadu, India.
[1] Sri_tag@yahoo.co.in
[2] k.vimaladevi@gmail.com

ABSTRACT

Due to the fast progress of communication technologies, many popular services have been developed to take advantage of the advanced technologies. One of these popular services is wireless communication. Authentication of mobile subscribers is a challenging issue due to increasing security threats as it acts as the first defence against attackers. thus we propose an mutual authentication technique to be used in wireless network We come up with a novel authentication mechanism, called Group Registration, which reduces the huge bandwidth consumption between Visitor Location Register (VLR) and HLR; and overloaded Home Location Register (HLR) makes it suitable for reducing the computation and communication cost of the mobile users as compared to the existing authentication schemes. An efficient and secured channel is developed in the cryptosystems implementing the encryption function and keyed one-way function by cryptographic library JCE (Java Cryptography Extension) API in Java. Encryption function is implemented by symmetry-based AES encryption algorithm. The Keyed one-way function which follows HMAC standard (RFC 2104) is implemented by secure hash function SHA-2.

Keywords— Authentication, global mobility network, Information security, Mutual Authentication, roaming, symmetric cryptosystem, wireless communication, Wireless network security.

I. INTRODUCTION

One of the main challenges of the heterogeneous mobile environments is to allow users to access their services any time, any place, and anywhere, independent of the networks and devices being used .An important requirement for roaming is to make it a seamless experience for end-users. The prerequisite of seamless roaming is transparent end-user authentication and security across different access network technologies.

The proposed authentication technique consists of the following two phases:

•*Nested one-time secret mechanism-setup phase* [1], which is designed for mutual authentication between a mobile user and the system (a VLR and HLR) was carried out as first phase, the second part is tailored for mutual authentication between a mobile user and a VLR when the user does not leave the service area of the VLR.

• *Group Registration -setup phase*, in which the location registration cost, is efficiently reduced by reporting the location changes to the HLR for multiple mobile terminals (MTs) in a single location update request message

A. *Review of current authentication protocol for GSM*

There are three drawbacks in the authentication protocol for GSM as follows [8], [10], [12]:

•*Lack of Mutual Authentication:* There is no mutual authentication mechanism between mobile stations and base stations (VLR). GSM only provides unilateral authentication for the mobile stations. However, the identity of VLR cannot be authenticated. It is therefore possible for an intruder to pretend to be a legal network entity and thus to get the mobile stations' credentials.

•*Storage overhead:* Every time The VLR must turn back to the HLR to make a request for another set of authentication parameters when the MS stays in the VLR for a long time and exhausts its set of authentication parameters for authentication. There is bandwidth consumption between the VLR and HLR.

•*Bandwidth consumption:* Every mobile station in the particular VLR has n copies of the authentication parameters. The parameters are stored in the particular VLR database, and then space overhead occurs.

B. Attacks and assumptions.

The messages transmitted in wireless communication networks are exposed in the air, so malicious parties in wireless environments have more opportunities to eavesdrop or intercept these transmitted messages [1], [3]

The various attacks are

•*Masquerade attacks*: Intruder acting as the base station [3]

•*Replaying Attacks*: An intruder plans to impersonate a legal User and to obtain the authentication key by replaying the user's transmitting contents in the roaming environment [7].

•*Impersonating Attacks*: An intruder can impersonate the visited network V to the roaming user U, which results in U being cheated [7].

The various assumptions are

- It is assumed that there is a secure channel between the VLR and HLR over a fixed network which is already set up.
- It is assumed that the clocks of each mobile user and the system are synchronized and the transmission time between them is stable [2], [12], [13].
- The protocols are all based on asymmetric cryptosystems, which are less efficient than symmetric key cryptosystem

[2], [7].

From the analysis it is found that, the user U has to shift to a new VLR the old long vectors are discarded, which will waste the resources and increase the computation and communication cost for the system. Implementing the Group Registration phase will decrease the cost when compared to the conventional strategy.

This paper proposes a hybrid mechanism for the mobile network that provides mutual authentication and group registration, which aims in the bandwidth reduction .The symmetric encryption and SHA-2, provides a secure channel for communication.

This paper is organized as follows. Previous authentication schemes in Section II and Our basic idea constituting the review of the Hybrid Mechanism for mutual authentication [1]. Group Registration scheme for mobile communications is illustrated in Section III. In Section IV, we propose the future work. A concluding remark is given in Section V.

II. REVIEW OF PREVIOUS WORK

In this section, we will review the various techniques related to authentication, attacks and cryptosystems in the mobile network

In 1995, Dan Brown [2] proposed the privacy and authentication technique (P&A) using the simple challenge /response mechanisms. VLR performs the autonomous authentication by the user's triplets. The weakness is band limited channels of users broadcast signals and public key cryptosystem.

In 1995, Nigel Jefferies [3] proposed the need for mutual authentication between the user and the Network operator in complete security architecture for third-generation systems.

In 1993, M. Rahnema [4] proposed the need for frequency usage in cellular communication. The main drawback is the multipath fading effect due to the usage of Time stamp for entire communication.

In 1997, Suzuki and Nakada [5] used an interactive authentication mechanism with a symmetric cryptosystem .The assumption is that, once the roaming service is setup, the authenticated data travels only between the VLR and user mobile .This weakens the scheme as it increases the number of network signals

In 2000, Buttyan et al. [6] pointed out the masquerade and impersonate attack in [5].in which two Rounds of transmission is needed between U and V as well as between the V and H by assuming that home network is trusted. The drawback is that it is vulnerable to Replaying attack.

In 2007, M. Al-Fayoumi et al. [10] proposed a mutual authentication scheme that uses the public key cryptosystem at the initial and symmetric cryptosystem for future communication .The performance is increased by reducing the communication times but not the computation cost

In 2003, Hwang and Chang [7] proposed a mutual authentication technique that reduces the number of transmissions in authentication phase and reduces the mobile complexity by introducing a mechanism called "Self Encryption". It uses the symmetric cryptosystems .The main weakness is that the intruder intercepts the information about participants' locations.

In 2003 C.C. Lee, M. S. Hwang, and W. P. Yang [8] proposed the need for mutual authentication and consumption of bandwidth in wireless network, it suffers from the Data integrity, Non-repudiation, End-to-end confidentiality, Traffic confidentiality problems. It uses Timestamp for entire communication.

V. Kalaichelvi and R. M. Chandrasekaran [11] proposed an authentication technique that is concerned with device authentication rather than user authentication, the embedded key of sim is replaced by password thus it over comes the dependency on sim card. This scheme uses the nonce based approach so no clock synchronization is needed. The weakness is the use of RSA algorithm, which is highly deterministic.

In 2010, K.Phani Kumar et al. [13] proposed a scheme, in which Mutual authentication in between MS and VLR is provided for every communication, drawback is the use of Timestamp, in which the clock are to be synchronised

From the detailed study, in section III we came to know that need of mutual authentication [2]-[4], [7], [8], [10], [11], [13] and bandwidth consumption [4], [8] in the wireless communication. We propose a hybrid approach and Group Registration scheme that provide mutual authentication using symmetric key cryptosystem [5], [7] and make use of time stamp [9], [12], [13], one-way key function, and nonce [11] approach in the mobile authentication scheme. This proposed technique reduces the bandwidth consumption. The proposed protocol which makes use of Group registration prevents common attacks such as Masquerade attacks, Replaying Attacks and Impersonating Attacks [5]-[7].

III. OUR IDEA

In this section we will review the Hybrid Mechanism for mutual authentication [1], and will introduce our group Registration authentication scheme for the mobile environment

C. *Review of nested one-time secret mechanism.*

There are three basic approaches to achieve mutual authentication between two entities, say Alice and Bob. They are timestamp-based approach, nonce-based approach and the One-time secrets-based approach.

In this section we will review the Hybrid Mechanism for mutual authentication [1], and will introduce our group Registration authentication scheme for the mobile environment

1) Timestamp-based approach

[1], [9], [12], [13] Timestamp-based approach, Alice prepares a request message Request with the current time s tamp T_A and sends the symmetric encrypted E_K (Request, T_A)with key K to Bob. Bob decrypts and obtains (Request, T_A). Difference of Bobs current time and TA should not be greater than the maximal transmission time from Alice to Bob i.e.,

T_A is still fresh, and then Bob believes that E_K(Request, T_A) is produced by Alice and authenticates him. Similarly Bob is authenticated by T_B (shown in fig 1). This is by the assumption that the clocks of Alice and Bob are synchronised and has a stable transmission time

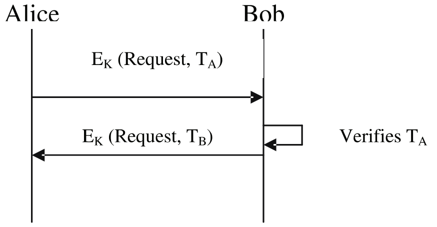


Fig 1: Timestamp-based Authentication approach

1) Nonce-based approach

In a nonce-based mutual authentication scheme[1],[5],[11], Alice prepares a request message Request and a randomly chosen string N_A, and then sends E_K (Request,N_A) to Bob, where N_A is a nonce produced by Alice. After receiving E_K (Request, N_A), Bob decrypts it to obtain (Request, N_A) .Bob prepares a response message Response B and a randomly chosen string N_B, and then sends E_K (Response B, N_A, N_B) to Alice, where N_B is said to be a nonce produced by Bob. Alice decrypts E_K (Response B, N_A, N_B) to obtain and checks if is identical to the one she chosen before. If true, Alice believes that E_K (Response B, N_A, N_B) is produced by Bob in this session, and thus Alice authenticates Bob. Alice is authenticated if N_B is equal to the one chosen by Bob (shown in fig 2).

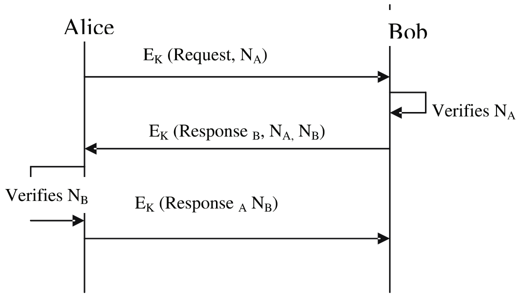

Alice
Bob
E_K (Request, N_A)
Verifies N_A
E_K (Response $_B$, N_A, N_B)
Verifies N_B
E_K (Response $_A$ N_B)

Fig 2: Nonce –based Authentication approach

1) One time secrets-based approach [1], [5].

Let K be the secret key shared between Alice and Bob in advance, We assume that Alice and Bob have successfully finished (j-1) times of mutual authentication and they have negotiated a common secret R_{j-1} in the (j-1) th authentication where j>=2.Now, they are about to perform the jth authentication. Alice computes R_j =F (K, R_{j-1}) and sends E_K (Requests, R_j) to Bob, where F is a public one-way hash function and Request is a request message. The string R_j is called the one-time secret of the jth authentication. Bob decrypts E_K (Request, R_j) to obtain Request and R_j. Bob checks if Request is correct and Rj is equivalent to the hashed value of his stored (K, R_{j-1}). If true, Bob authenticates Alice. Bob prepares a response message Response and sends E_K (Response, R_j) to Alice. Alice performs the decryption operation on E_K (Response, R_j) to acquire Response and R_j. Alice checks if Response is correct and is equal to the one she computed before. If true, Alice authenticates Bob. Finally, they replace R_{j-1} with R_j in their computers or devices

The comparisons of the three authentication mechanisms (ie., timestamps, one-time secrets, and nonces) are summarized in TABLE II.

In the GSM system, two authentications have to be performed i.e., mutual authentication between VLR and HLR and mutual authentication between system and each user. Each VLR and HLR are connected in a wired network can authenticate using timestamp-based approach without suffering from the clock synchronisation problem. In the wireless environment it is not possible to have a clock synchronisation so nonce-based approach is used to authenticate the system to each mobile user.

The nested one time secret mechanism the nonce based approach is used to negotiate the initial value of one time secret, this reduces the computation cost if the user remains in the same VLR .The combined authentication approach is shown in fig 3.

B. Group Registration authentication scheme

In mobile networks, the location of a mobile user needs to be traced for successful and efficient call delivery. In existing cellular networks, as a mobile user changes his/her location area, a location registration request is sent to the home location register (HLR) to update the user profile to point to the new location area. With a large number of mobile subscribers, this conventional registration strategy will incur a high volume of signalling traffic. We propose a new location registration strategy, called Group Registration, which efficiently reduces the location registration cost by reporting

TABLE II
COMPARISONS OF THE THREE AUTHENTICATION MECHANISMS

	Timestamps	One-Time Secrets	Nonces
Assumptions:	1.clock synchronization 2.stable transmission time	The previous Authentication must be successfully finished	None
Suitable for:	The authentication between VLR and HLR	The authentication between a user and the system for the authentication process after the initial one	The initial authentication between a user and the system

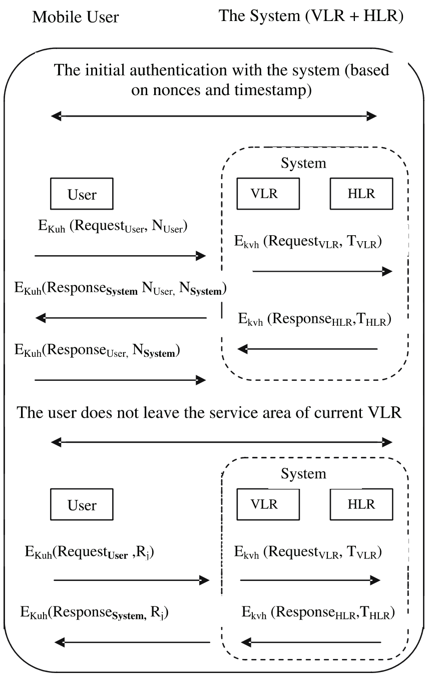

Mobile User
The System (VLR + HLR)
The initial authentication with the system (based on nonces and timestamp)
System
User
VLR
HLR
E_{Kuh} (Request$_{User}$, N$_{User}$)
E_{kvh} (Request$_{VLR}$, T$_{VLR}$)
E_{Kuh}(Response$_{\mathbf{System}}$ N$_{User,}$ N$_{\mathbf{System}}$)
E_{kvh} (Response$_{HLR}$,T$_{HLR}$)
E_{Kuh}(Response$_{User,}$ N$_{\mathbf{System}}$)
The user does not leave the service area of current VLR
System
User
VLR
HLR
E_{Kuh}(Request$_{\mathbf{User}}$,R$_j$)
E_{kvh} (Request$_{VLR}$, T$_{VLR}$)
E_{Kuh}(Response$_{\mathbf{System,}}$ R$_j$)
E_{kvh} (Response$_{HLR}$,T$_{HLR}$)

Fig 3: The Hybrid Mechanism for mutual authentication

Location changes to the HLR for multiple mobile terminals in a single location can update request message.

Specifically, the IDs of the Mobile terminals newly moving into a VLR are buffered and sent to the HLR for location update in the route response message of the next incoming call to any mobile terminals in the Location area. An analytic model is developed and numerical results are presented. It is shown that the proposed Group Registration strategy can achieve significant cost reductions when compared to the conventional strategy and the local anchor strategy over a wide range of system parameters.

Moreover, the Group Registration strategy implementing the frequency reuses results in much smaller call delivery latency than the local anchor strategy.

We adopt SHA-256, which has a 256-bit output, to implement the one-way hash function, and we also implement the random-number generator by SHA-256 in the protocols, where performing one SHA-256 operation takes about the same time as performing one symmetric encryption or decryption operation.

In general, the length of the identity of every mobile user is usually less than 128 bits. Thus, we let the length of the user's identity be 128 bits. Encryption function can be implemented by symmetry-based encryption algorithm AES, Encryption with AES is based on a secret key with 256 bits.

IV.CONCLUSION

In this paper, we have shown out the drawbacks in the existing authentication protocol. This paper proposes to improve the performance Group Registration technique based on hybrid mechanisms. It results in lesser bandwidth consumption and reduces the computation and communication cost. The proposed scheme can withstand the replay attack and the impersonating attack on mobile communications .From analysis it is proved that the proposed method is not only secure against various known attacks, but also more efficient than previously proposed schemes. In the simulation environment, the most likely way is to build an IMS environment is to test the implementation. It can be done by building an EAP authentication environment and implementing the protocol on OpenIMS, Xsupplicant and Free Radius.

REFERENCES

[1] Chun-I Fan, Pei-Hsiu Ho, and Ruei-Hau Hsu "Provably Secure Nested One-Time Secret Mechanisms for Fast Mutual Authentication and Key Exchange in Mobile Communications" I5/ACM Transactions on Networking, VOL. 18, NO. 3, JUNE 2010.

[2] D. Brown, "Techniques for privacy and authentication in personal communication systems," I5 Personal Commun., vol. 2, no. 4, pp. 6–10, Aug. 1995.

[3] N. Jefferies, "Security in third-generation mobile systems," IEE Coll.Security Netw., pp. 8/1–8/5, 1995.

[4] M. Rahnema, "Overview of the GSM system and protocol architecture," I5 Commun. Mag., vol. 31, no. 4, pp. 92–100, Apr. 1993.

[5] S. Suzuki and K. Nakada, "An authentication technique based on distributed security management for the global mobility network," I5 J. Sel. Areas Commun., vol. 15, no. 8, pp. 1608–1617, Oct. 1997.

[6] L. Buttyan, C. Gbaguidi, S. Staamann, and U.Wilhelm, "Extensions to an authentication technique proposed for the global mobility network," I5 Trans. Commun., vol. 48, no. 3, pp. 373–376, Mar., 2000.

[7] K. F. Hwang and C. C. Chang, "A self-encryption mechanism for authentication of roaming and teleconference services," I5 Trans.Wireless Commun., vol. 2, no. 2, pp. 400–407, Mar. 2003.

[8] C. C. Lee, M. S. Hwang, and W. P. Yang, "Extension of authentication protocol for GSM," IEE Proc., Commun. vol. 150, no. 2, pp. 91–95, 2003.

[9] C. C. Chang, J. S. Lee, and Y. F. Chang, "Efficient authentication protocol of GSM," Comput. Commun., vol. 28, no. 8, pp. 921–928, 2005.

[10] M. Al-Fayoumi, S. Nashwan, S. Yousef, and A. R. Alzoubaidi, "A new hybrid approach of symmetric/asymmetric authentication protocol for future mobile networks," in Proc. Wireless Mobile Comput., Netw. Commun., 2007, pp. 29–29.

[11] V. Kalaichelvi and R.M. Chandrasekaran, "Secure authentication protocol for mobile," Proc. Comput., Commun. Netw., pp. 1–4, 2008.

[12] K. P. Kumar, G. Shailaja, A. Kavitha, and A. Saxena, "Mutual authentication and key agreement for GSM," in Proc. ICMB, 2006, p. 25.

[13] K. Ammayappan, A. Saxena, and A. Negi, "Mutual authentication and key agreement based on elliptic curve cryptography for GSM," in Proc.ADCOM, 2006, pp. 183–186.