ED 417 708 IR 018 775

TITLE Learning Machine, Vietnamese Based Human-Computer Interface.
INSTITUTION Northwest Regional Educational Lab., Portland, OR.
PUB DATE 1998-00-00
NOTE 62p.; Session 6 of "Information Technology in Education and Training (IT@EDU98). Proceedings"; see IR 018 769. For sessions 1-6, see IR 018 770-775.
PUB TYPE Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE MF01/PC03 Plus Postage.
DESCRIPTORS *Algorithms; Character Recognition; Color; Educational Technology; Foreign Countries; Human Factors Engineering; *Information Technology; Knowledge Base for Teaching; *Machine Translation; Mathematics; Optical Scanners; Programming; *Vietnamese; Word Recognition
IDENTIFIERS Neural Networks

ABSTRACT

The sixth session of IT@EDU98 consisted of seven papers on the topic of the learning machine--Vietnamese based human-computer interface, and was chaired by Phan Viet Hoang (Informatics College, Singapore). "Knowledge Based Approach for English Vietnamese Machine Translation" (Hoang Kiem, Dinh Dien) presents the knowledge base approach, which consists of concepts such as things, actions, relations, and attributes, organized on the structure of inheritance hierarchy. "A Learning Algorithm for Feature Selection Based on Genetic Approach" (Nguyen Dinh Thuc, Le Hoai Bac) presents a genetic algorithm that chooses relevant features from a set of given features, for feature selection based on the correlation among the features and between every feature and given target curve. "Artificial Neural Network for Color Classification" (Tran Cong Toai) examines several neural network models, their learning schemes, and their effectiveness in color classification. "Synthesizing and Recognizing Vietnamese Speech" (Hoang Kiem, Nguyen Minh Triet, Vo Tuan Kiet, Thai Hung Van, Luu Duc Hien, Bui Tien Len) presents algorithms applied successfully in Vietnamese isolated word recognition and Vietnamese synthesis. "On-Line Character Recognition" (Nguyen Thanh Phuong) presents a real-time handwriting character recognition system based on a structural approach. "Data Mining and Knowledge Acquisition from a Database" (Hoang Kiem, Do Phuc) considers how to use multi-dimensional data model (MDDM) for mining rules in a large database. "Genetic Algorithm for Initiative of Neural Networks" (Nguyen Dinh Thuc, Tan Quang Sang, Le Ha Thanh, Tran Thai Son) describes a procedure for initiative of neural networks based on genetic algorithms, based on the correlation between every weight and error function. (SWC)

SESSION 6

Friday, 16 January 1998

Session 6: Learning machine, Vietnamese based human - computer interface

Chair:
Dr. Phan Viet Hoang, Informatics College, Singapore

6-1.**Knowledge based approach for English Vietnamese Machine Translation**
Hoang Kiem, Dinh Dien, University of Natural Sciences,HCMC, Vietnam

6-2.**A learning algorithm based on genetic algorithms approach**
Nguyen Dinh Thuc,Le Hoai Bac, University of Natural Sciences, HCMC, Vietnam

6-3.**Artificial neural network for color classification**
Tran Cong Toai, University of Technology, HCMC, Vietnam

6-4.**Synthesizing and recognizing Vietnamese Speech**
Hoang Kiem, Nguyen Minh Triet, Luu Duc Hien,University of Natural Sciencs , HCMC, Vietnam

6-5.**On- line character recognition**
Nguyen Thanh Phuong, University of Natural Sciences, HCMC, Vietnam

6-6.**Data mining and knowledge acquisition from a database**
Hoang Kiem, Do Phuc
University of Natural Sciences, HCMC, Vietnam

6-7. **Genetic Algorithm for initiative of neural networks**
Nguyen Dinh Thuc, Tan Quang Sang, Le Ha Thanh, Nguyen Thanh Son
University of Natural Sciences, HCMC, Vietnam

KNOWLEDGE - BASED APPROACH FOR ENGLISH - VIETNAMESE MACHINE TRANSLATION

Hoang Kiem, Dinh Dien
University of Natural Sciences,HCMC, Vietnam.

Abstract

The most difficult point in the machine translation is the elimination of the ambiguity of the natural language. This paper will present an knowledge-based approach to solve the above-mentioned ambiguity and the application of the knowledge base in the English-Vietnamese machine translation. This knowledge base consists of such concepts as things, actions, relations, attributes and so on is organised on the structure of inheritance hierarchy. We have experimented this approach in the English-Vietnamese machine-translation system with the effective solutions to the lexical and structural ambiguities.

1. Introduction

In our current age of the information boom, the techno-scientific information mainly written in English has become more and more diversified and the question of how to be able to keep in touch with that source of information is indispensable to all of us. Therefore, the design of an English-Vietnamese automatic translation plays a practically meaningful role in the present background of Vietnam. This paper will present an effective solution which has been applied in the English-Vietnamese automatic translation to the elimination of the ambiguities in the English source texts in order to create the Vietnamese destination documents of desired quality. Due to the time limit , its first stages have been restricted to the techno-scientific documents only which , however , will be expanded into other fields in the near future without any modification to the original approach .

2. Summary of the English-Vietnamese machine translation system

This is a microcomputer-based program which deals with the automatic translation of the natural language from English into Vietnamese. First of all, the program will be input with grammatically correct sentences or paragraphs which then will be automatically translated into the Vietnamese ones accordingly after a careful analysis in terms of vocabulary, syntax, semantics and so on. Thanks to the derivative rules, the built-in dictionary and the shallow semantic analysis based on the knowledge-base, the program will generate the Vietnamese sentences or paragraphs of equivalent meanings.

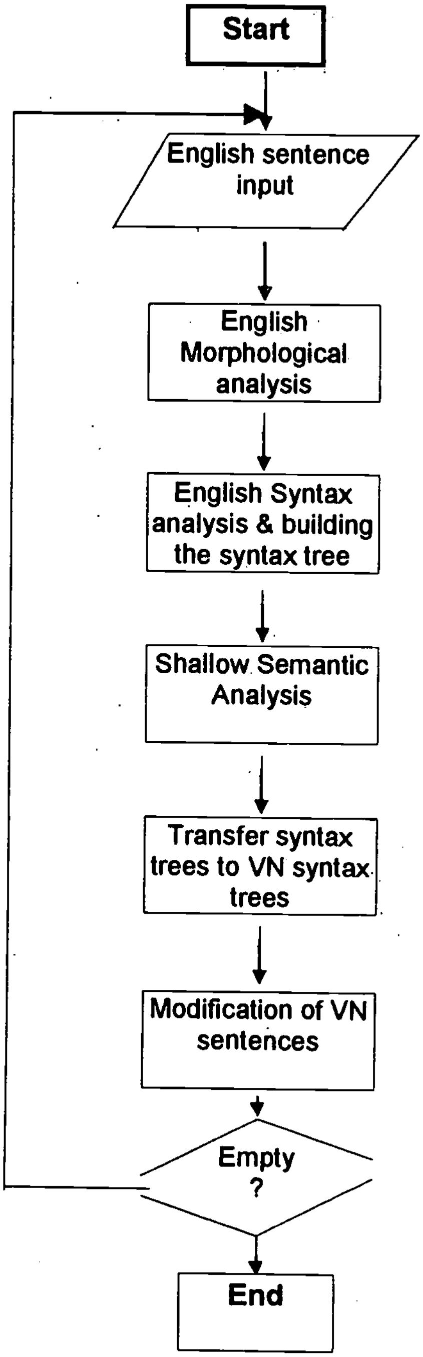

Start
English sentence input
English Morphological analysis
English Syntax analysis & building the syntax tree
Shallow Semantic Analysis
Transfer syntax trees to VN syntax trees
Modification of VN sentences
Empty ?
End

Fig.1 : Flowchart of the program

These English sentences which are referred to as the source ones might be simple or complex ones in all forms, tenses and voices etc. Because a certain word in English as well as Vietnamese may have different parts of speech, meanings depending on its syntactical position in the sentence and the actual context. Therefore, translating the natural language cannot be simply the consultation of dictionaries nor the word by word translation. This notion is more exact with the translation of two languages of such different categories as English and Vietnamese. The main issue is to make the machine understand the source sentence and then, the destination sentence will be created with the desired meaning. Because of this reason, we have to analyse the syntax, semantics of the source sentence and combine the exterior language knowledge input by us. Thereafter, the program will be able to look for the destination sentences which share the most suitable syntax and semantics.

The system has been equiped with the following blocks (Fig.1).

1. Morphological Analysis : to analyse the English morphology of all words of input English sentences.
2. English syntax trees : to analyse the English syntax of an input English sentence for the formation of all possible syntax trees.
3. Shallow Semantic Analysis : to select the optimal syntax tree from several above possible syntax trees based on the context and the meaning of each word.
4. Vietnamese syntax trees : to transfer from English optimal syntax trees to Vietnamese ones with reasonable word orders.
5. Generation and Modification of Vietnamese sentences : to generate and to modify the Vietnamese sentences which arise from the most reasonable syntax tree, mainly based on the contextual properties of the concerned words.

3. The knowledge-based dictionary system

The electronic English-Vietnamese dictionary of this system consists of approximately 10000 most frequently used root words, 2,000 phrases and idioms (based on the frequency dictionary) together with 1,000 terms used in the computer science. All of them are classified in accordance with grammatical characteristics (such as : parts of speech, sentence patterns, their positions and functions in the sentences), their approriate conceptual attributes (such as : *human beings* or *things, space* or *time, quick / slow, good / bad, negative / positive*, and their related fields, etc.).

3.1. The conceptual tree

Concepts are considered as an aggregate of instances which can be divided into various hierachical classes. The instances of each class always share the similar features which are fully inherited from the higher classes .In addition, all the instances of the same class are in logical relation with each other. These relations might be similarity, contrast or interdependence, agent and so forth .The parents classes contain all the common features for the lower ones belonging to them. For example, in the classes of the living things, there are zoology and botany, in the zoology, there exist human beings and animals which will be sub-divided into mammals, reptiles, birds etc.

These conceptual attributes have been well-organised in the structure of inheritance trees, thanks to which most of the ambiguities will be solved in a relatively complete manner. These concepts are structurised in the Entity - Relation model, which is to say: the concept of *zoology* includes such sub - strata as : *human beings* and animals ,the stratum of *human beings* is then divided into *masculine* and *feminin* or into the *time - related* attributes of *the old* and *the young* and so forth ; the next stratum inherits the whole attributes from the parents one. This detailed classification serves to clarify the ambiguities, such is shown in the phrase : *old man and children* which can be analysed to be of these four syntax trees:

a. (Old man) and children

b. Old man and old children

* c. (Old human beings) and children

d. Old human beings and old children

In this case, thanks to the *young* attribute in *children* which is opposite to the *old* attribute in the *old - young* relation which finds itself in the *time* stratum, we omit sentences b and d. Next, thanks to the *masculine* attribute in *man* which isn' t parallel to the *neutral* attribute in *children* in the *gender* relation, we omit sentences a. Whereas, in the sentence *old man and woman* we will choose b due to the balance priority in a phrase with the conjunction *and* :

a. (Old man) and woman

* b. Old man and old woman

c. (Old human beings) and woman

d. Old human beings and old woman

These concepts may be the conditional attribute for a certain meaning or may be the consequentional one for which determines the choice of a certain word. All the poly - semous words are contextually analysed in a systematical manner for their their respective meanings.

* *For example, the word he may be*:

- *that old person* in case it falls on the attribute of *old* and *respect.*
- *that old guy* in case it falls on the attribute of *old* and *non-respect.*
- *that young person* in case it falls on the attribute of *young* and *respect.*
- *that young guy* in case it falls on the attribute of *young* and *non-respect.*

* *For example, the word save may be* :

- *Save* as a verb means *help* in case its object is a noun related to *human beings* (such as : *life, soul,* etc.).
- *Save* as a verb means *economize* in case its object is a noun related to *money* or *time*.
- *Save* as a verb means *cut down* in case its object is a noun related to an abstract concept such as : *difficulties, tiredness*, etc.
- *Save* a a verb means *preserve, store* in case its object is a noun related to the *data* in computers.

3.2. Codification and dictionary consultation

The tree structure is used for the insertion of all the roots, for example : The words : *a, abort, an, and, ant, as, ass* are stored as follows :

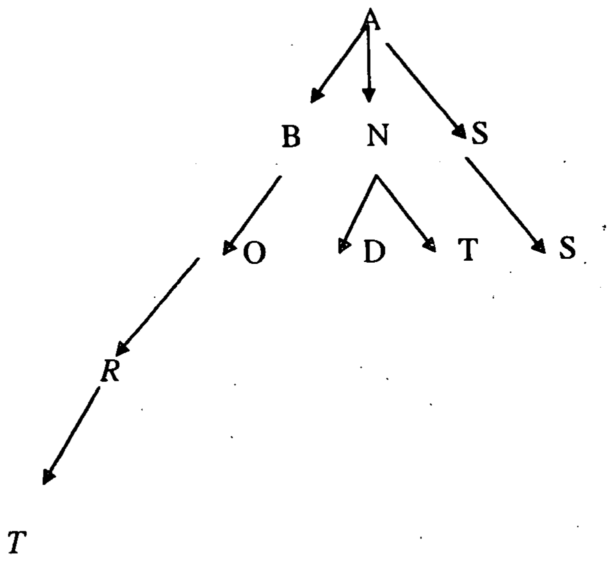


Fig.2 : representation of : *a(bort,n(d,t),ss)*

Each node will be one of following 5 cases

1. The word has not yet been completed and still continued
2. The word has been completed and still continued
3. The word has not yet been completed and branched
4. The word has been completed and branched
5. The branch has been ended

In case the node has a matching end (matching the consulted word) the pointer will be pointed to the equivalent information area of that word in the file containing its full details In case it has been consulted to the node end or no more route to be further consulted the translator considers that word as non-existent.

3.3. Analysis of the morphology

Morphological Analysis : to analyse the English syntax of an input English sentence for the formation of possible syntax trees. In this step, we make use of the anlysis table of Earley. That is, every word / phrase will be at the same time grammatically analysed for all the possible cases which gradually diminish and in the end exist only the acceptable candidates. Thereafter we apply the rules of contextual analysis for the most suitable

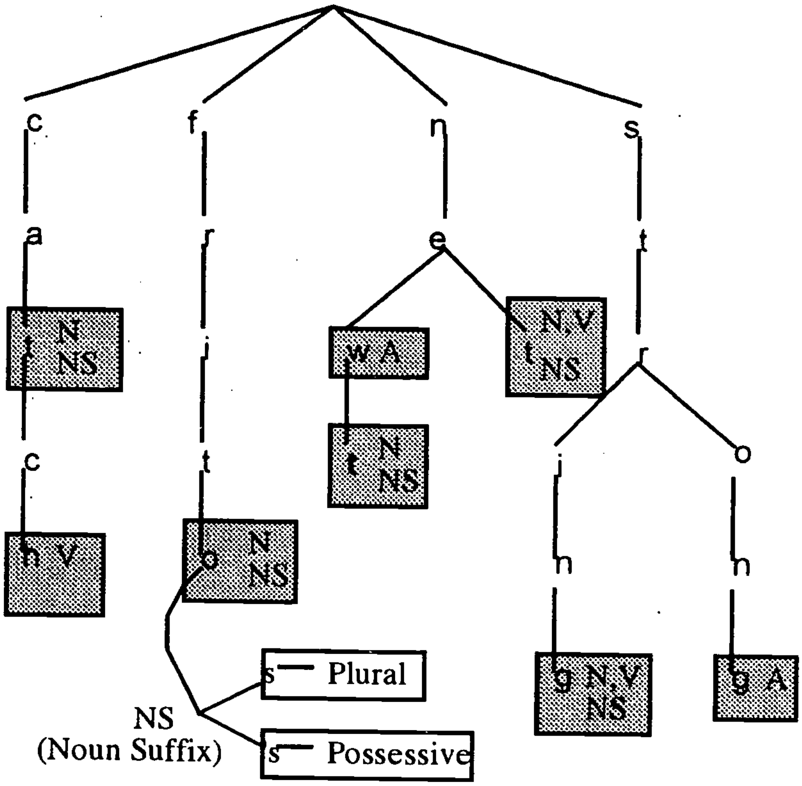


Fig.3. Morphological analysis of *"fritos"*

To minimize the dictionary size, we enter only the roots and their derivatives are treated as deleting such subbordinate ones (affix) as: prefixes, suffixes, terminason conjugations (-ed, -ing, - ly, etc.) until the roots are left (in formity with the transformation rule) to be consulted with, after which the meanings of the roots and the transformed ones will be combined together. Such exceptional cases (as irregular verbs, abnormal noun plurals, etc.) are stored in the dictionary as the roots.

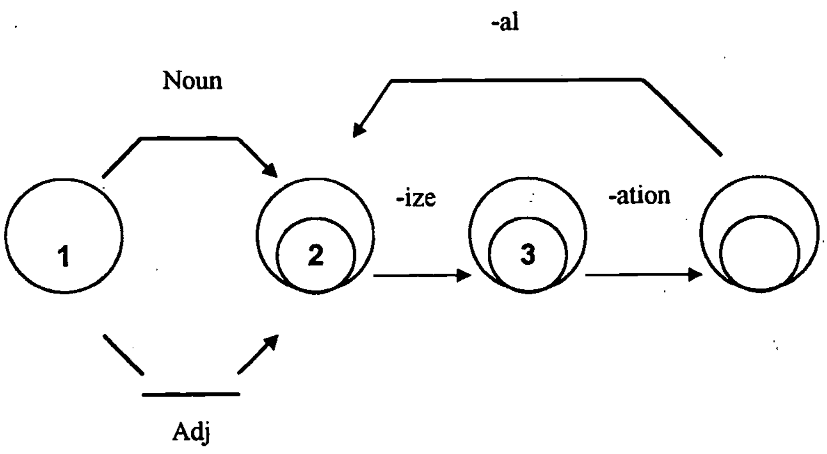


Fig.4 : transition diagram of *computerizational*

3.4. Grammatical Analysis of English sentences

In this step, we make use of the anlysis table of Earley, that is, every word / phrase will be at the same time grammatically analysed for all the possible cases which gradually diminish and in the end exist only the acceptable candidates. Thereafter we apply the rules of contextual analysis for the most suitable. In case more than one syntax tree can be chosen, we have to refer to the complementary clues in the neighboring sentences for the best. In the worst case that no choice is possible, the frequency thumb will be resorted to for the most frequent. In case that no syntax tree is obtained after the grammatical analysis, we have no choice but apply the word-by- word translation of which the result is naturally not so much perfect.

3.5. Generating Vietnamese sentences

The source syntax tree of English then will be translated into the destination one of Vietnamese (in applying the rules of order, grammar, etc.) to be matched with the meaning of each node for a complete Vietnamese sentence. In the course of translating the syntax tree, special words will possess their own special attributes to inform the program to adjust destination syntax trees. For example, the order n the English sentences is usually known as the demonstrative adjectives stand in front of nouns, such as : new book, however, general election or nothing new.

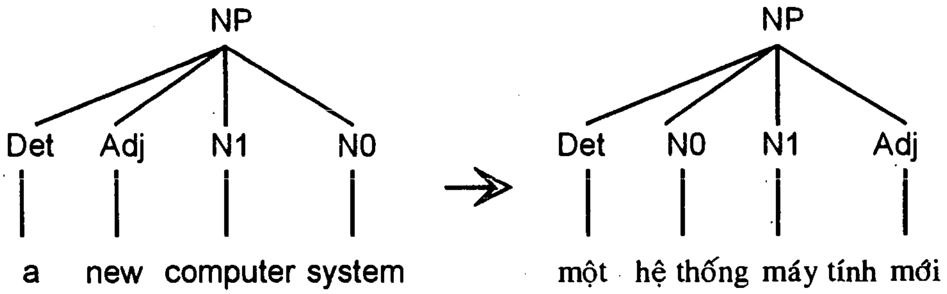


Fig.5. Modification of Vietnamese syntactic tree.

4. Results

So far, this system is fully able to solve most of the ambiguities in the aspects of semantics and lexicology as follows.

1. Parts of speech

There are words such as "love" which might be a verb and a noun will certainly be easily distinguished by our Machine Translation System.

For example : He *loves her with an absolutely faithful love.*

That excellent student can program a lot of very complicated programs.

The contents of this abstract is very abstract .

2. Semantics

There are the words which have the same parts of speech but completely different meanings such as "to be", "old", "ask", "field" and so on.

Ex : *He is scolded by his wife but he is praised by his lover.*

I will lend that old man this old book.

Our company asks that customer to arrive at our office to ask several matters.

I eat rice, plant rice and export rice.

The word order in Vietnamese can be modified to reach the suitability of 90 per cent. Our program is capable of synthesizing the new words based on the meanings of the root morphemes. From *normal*, we can derive new words such as *normalize*, *normally*, *normalization*, *non-normal* etc.

For example : *Normally, I always normalize non-normal problems.*

In addition to the translation of the technical and scientific documents in the fields of Informatics, Electronics, Mathematics, Linguistics, this system is capable of translating other conventional documents. Please refer to the enclosed annexes.

In next stage, this system will connect to the character recognization system in order to translate texts automatically.

At present, this system is able to translate English sentences into Vietnamese ones (in the Computer Science field as well as other conventional sentences) based on the 25 verb patterns of A.S. Hornby at the rate of 0.6 s for every sentence and roughly 15s for every text page with the exactness of 60% - 70%.

- It can handle all types of sentences (e.g. : complex as well as simple ones in any tense, voice, form, etc.) and various kinds of ambiguity (e.g. : the appropriate syntax-based part of speech , the context-based meaning , etc.)

- The system can be used as a spelling checker. Its consultation speed has reached 100 words/ 3 seconds on AT-386DX 40MHz, in the DOS medium.

Acknowledgement

During the course of the realisation of this work , we have received numerous valuable supports and encouragements from various persons and organizations . Especially , We would like to express our heart-felt gratitude as well as our sincere thanks to :

- Mr.Eduart Hovy who has granted us with his useful recommendations and instructive documents.
- and other professors who have been granting us with their disinterested assistances and advice.

Reference

[1] A.S.Hornby : Oxford advanced learner's dictionary. Oxford -1989.

[2] Alfred V.Aho - Jeffrey D. Ullman : The theory of parsing, translation and compiling, Vol 1. Prentice-Hall - 1972

[3] Eduard Hovy : How MT works. Byte, Jan 1993.

[4] Gilbert K. Krulee : Computer Processing of Natural Language. Prentice-Hall - 1991.

[5] J. C. Catford : A linguistic theory of Translation. Oxford - 1967.

[6] Richard Sproat. Morphology and Computation. MIT - 1992.

[7] UNESCO, Proceedings of the Regional Workshop on : Processing of Asian Language, Sep 26-28, 1989.

[8] Victoria Fromkin - Robert Rodman - Peter Collins - David Blair. An introduction to language. HRW - 1990.

A LEARNING ALGORITHM FOR FEATURE SELECTION BASED ON GENETIC APPROACH

Nguyen Đinh Thuc , Le Hoai Bac
University of Natural Sciences, HCMC, Vietnam

Abstract

In this paper, we present a genetic algorithm for feature selection. The algorithm chooses relevant features from a set of given features. The selection based on the correlation among the features and correlation between every feature and given target curve. The proposed algorithm is used for a real application, the classification rate could be increased by about ten percent.

1. Introduction

The choice of features as input variables for data analysis systems is a very important tast and difficult. If meaningful and relevant features are used, the error of the system decreases while the recognition rate and the classification abilities increase. Various methods can give a clue on the relevant features for a data analysis system. They are mostly based on *linear relations* like the correlation, *the principal component analysis, the Fisher disciminant analysis* (see [Pernot and Vallet, 1991] and [Battiti,1994]). Somes other methods based on *neural network* also useful (see [Krisnapuran and Lee, 1992]. The general problem of *"feature selection"* can be stated as follows:

> *"Given an initial set F of N features, find the subset S (F of k < N features that maximizes correlation of each chosen feature with the target curve and minimizes correlation among the chosen features themselves"*

In this paper, we present a procedure based on genetic algorithms to solve this problem. The algorithm chooses relevant features *S* from a set of given features *F*. The basis for these calculations is the correlation among the features and the correlation between every feature and a given target curve.

2. Genetic algorithm and selecting feature

2.1 Genetic algorithm

Genetic algorithms are powerful direct search optimization tools designed in imitation of some principles of genetics and natural evolution, name reproduction, crossover, mutation and selection (see [Goldberg, 1989] and [Holland,1992]). The idea of genetic algorithms is to use a population of possible solutions, which are changed by reproduction and mutation and selected for the next generation according to their fitness. So various paths to the optimum are investigated at the same time and information about these paths can be exchanged. This is called the implicit parallelism of genetic algorithms (see [Goldberg, 1989]).

The four major steps in preparing to use the genetic algorithm on fixed-lenght strings to solve a given problem:

(1) determining the presentation schema,

(2) determining the fitness measure,

(3) determining the parameters and variables for controlling the algorithm. The primary paramaters for controlling the genetic algorithm are the population size and the maximun number of generations to be run,

(4) determining the way of desgnating the result and the criterion for terminating a run.

Once these steps for setting up the genetic algorithm have been completed, the genetic algorithm can be run.

The three steps in executing the genetic algorithm operating can be summarized as follows:

(1) Randomly create an initial population of individual fixed-lenght strings.

(2) Iteratively perform the following substeps on the population of strings until the termination criterion has been satisfied:

(a) Evaluate the fitness of each individual in population.

(b) Create a new population of strings by applying at least the first two of the following three operations. The operations are applied to individual string(s) in the population chosen with a probability based on fitness.

- Copy existing individual strings to the new population.
- Create two new strings by genetically recombining randomly chosen substrings from two existing strings.
- Create a new string from an existing string by randomly mutating at one position in string.

(3) The best individual string that appeared in any generation is designated as the result of the genetic algorithm for the run. This result represent a solution (or approximate solution) to the problem.

2.2 Genetic algorithm for optimal feature selection

To use genetic algorithm for optimal feature selection, the individuals in the algorithm are bitstrings of lengt N, where each bit represents a feature. A value *"zero"* means that the feature is not used and a *"one"* says that the correspongding feature is used. The number k of chosen features may be fixed in the first step, as we did in the contribution. But it may also be variable and hence be optimized by the genetic algorithm in order to find the best number of features.

Various authors have proposed fitness functions (see [Klir and Yuan, 1995] and [Pal and Bezdek, 1994]). In this paper, we consider fitness function studied by Lars A. Ludwid and Adolf Grauel (see [Ludwig and Grauel, 1996]).

Suppose, we have given the feature vector:

$$f = (f_1, f_2, \ldots, f_N)$$

and, a target curve g have also given.

The idea is to maximize the correlation between the chosen features and target curve and at the same time to minimize the correlation among the chosen features. Therefore, on the one hand the correlation among the features f_i and f_j

$$R = (R_{ij})$$

which is a N x N matrix, shall be calculated. On the other hand the correlation between the feature fi and the target curve g

$$r = (r_i)$$

which is a vector with N components, shall be computed.

Now, fitness function is defined as follows:

$$F(X) = (\textstyle\sum_{i=(1,N)} (x_i\, r_i^2\,))\, /\, k - 2((\textstyle\sum_{i=(1,N)} (x_i \sum_{(j<i)} x_j\, R_{ij}^2))\, /\, k(k-1)$$

where $X = (x_1, x_2, \ldots, x_N);\ x_i \in \{0,1\}$ is an invidual of population in genetic algorithm.

3. Application: Analizing medical data

We successfully implement artifical neural network for bone data classification. In this case, a three-layer neural network [Hoang Kiem and Nguyen Dinh Thuc, 1995] is used, which is trained by backpropagation with 30 training patterns. This data set have derived from [Peingold, Nelson and Partiff, 1992]. The data are to be classified in one of 12 classes, so the network always has 12 output neurons. The number of input neurons depends on the type of pre-proccessing.

In our application, we calculated 13 different features and wand to choose the five best features from all. The proposed feature selection algorithm is used. With this optimal features the classification rate of neural network increased from 87.5% to 96%. The results are showed in tables 7.1 and 7.2.

References

[1] Battiti, Using mutual information for selecting features in supervised neural net learning. IEEE Transaction on Neural Networks, 5(4):533-550 -1994

[2] Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading/MA, USA - 1989.

[3] Hoang Kiem and Nguyen Dinh Thuc, Analyzing Data Via Neural Network, Proc. Inter. Comf. on Analysis and Mechanics of Continuous Media, 1995, pages 217-221- 1995.

[4] Holland, Adaption in Natural and Artificial Systems. MIT Press, Cambridge/MA, USA - 1992.

[5] Klir and Yuan,Fuzzy sets and fuzzy lgic. Theory and applications. Prentice Hall PTR - 1995.

[6] Krisnapuran and Lee, Fuzzy-connective-based hierarchical aggregation networks for decision making. Fuzzy Sets Systems, 46:11-17 -1992.

[7] Ludwig and Grauel, Genetic Algorithm for Optimal Feature Selection, Proc. of EUFIT'96, pagers 457-460, 1996, Germany -1996.

[8] Pal and Bezdek, Fuzzy Sets, neural network, and soft computing- 1996, pages 194-212 .

[9] Peingold, Nelson and Partiff, Composite index of Skeletal Mass: Principal Components Analysis of Regional Bone Mineral Densities, Journal of Bone anf Mineral Research, Vol. 7, 1992.

[10] Pernot and Vallet, Determining the relevant parameters for the classification on a multi-layered perception: Application to radar data. In Proc. 1991 Int. Conference Artificial Neural Networks ICANN-91, pages 797-802, Espoo, Finland.

Table 7.1

stt	lsbmd			NeuralNet's output			fmbmd			NeuralNet's output		
1	0	1	0	0.12	0.88	0.1	0	1	0	0.119	0.881	0.1
2	0	1	0	0.089	0.911	0.1	0	1	0	0.12	0.88	0.1
3	0	1	0	0.103	0.896	0.1	0	1	0	0.12	0.881	0.1
4	0	1	0	0.114	0.886	0.1	0	1	0	0.106	0.895	0.1
5	0	1	0	0.1	0.9	0.1	0	1	0	0.129	0.871	0.1
6	1	0	0	0.909	0.091	0.1	1	0	0	0.909	0.091	0.1
7	0	1	0	0.096	0.905	0.1	0	1	0	0.089	0.911	0.1
8	0	1	0	0.13	0.869	0.101	0	1	0	0.132	0.869	0.101
9	0	1	0	0.104	0.895	0.1	0	1	0	0.099	0.902	0.1
10	1	0	0	0.862	0.138	0.1	1	0	0	0.876	0.124	0.1
11	0	1	0	0.093	0.907	0.1	0	1	0	0.124	0.876	0.1
12	0	1	0	0.103	0.897	0.1	0	1	0	0.13	0.87	0.1
13	0	1	0	0.11	0.89	0.1	0	1	0	0.117	0.883	0.1
14	1	0	0	0.913	0.086	0.1	1	0	0	0.895	0.105	0.1
15	0	1	0	0.098	0.902	0.1	0	1	0	0.12	0.881	0.1
16	0	1	0	0.073	0.927	0.1	0	1	0	0.066	0.934	0.1
17	1	0	0	0.944	0.056	0.1	1	0	0	0.944	0.056	0.1
18	0	1	0	0.097	0.902	0.1	0	1	0	0.112	0.889	0.1
19	0	1	0	0.095	0.905	0.1	0	1	0	0.135	0.865	0.1
20	0	1	0	0.097	0.903	0.1	0	1	0	0.112	0.888	0.1
21	0	1	0	0.069	0.931	0.099	0	1	0	0.074	0.925	0.099
22	0	1	0	0.108	0.892	0.1	0	1	0	0.135	0.865	0.1
23	0	1	0	0.097	0.903	0.1	0	1	0	0.12	0.88	0.1
24	0	1	0	0.097	0.904	0.1	0	1	0	0.139	0.861	0.1
25	0	1	0	0.099	0.901	0.1	0	1	0	0.108	0.893	0.1
26	0	1	0	0.079	0.921	0.1	0	1	0	0.073	0.927	0.1
27	1	0	0	0.862	0.138	0.1	1	0	0	0.866	0.134	0.1
28	0	1	0	0.1	0.9	0.1	0	1	0	0.129	0.871	0.1
29	0	1	0	0.103	0.898	0.1	0	1	0	0.109	0.891	0.1
30	0	1	0	0.093	0.906	0.1	0	1	0	0.104	0.896	0.1
31	0	1	0	0.104	0.896	0.1	0	1	0	0.15	0.85	0.1
32	0	1	0	0.096	0.904	0.1	0	1	0	0.099	0.901	0.1
33	0	1	0	0.11	0.89	0.1	0	1	0	0.114	0.886	0.1
34	0	1	0	0.102	0.898	0.1	1	0	0	0.141	0.859	0.1
35	0	1	0	0.099	0.901	0.1	0	1	0	0.127	0.874	0.1
36	0	1	0	0.103	0.897	0.1	0	1	0	0.108	0.892	0.1
37	0	1	0	0.095	0.905	0.1	0	1	0	0.106	0.894	0.1
38	0	1	0	0.1	0.9	0.1	0	1	0	0.113	0.887	0.1
39	0	1	0	0.101	0.9	0.1	0	1	0	0.14	0.86	0.1
40	0	1	0	0.096	0.904	0.1	0	1	0	0.128	0.872	0.1
41	0	1	0	0.083	0.917	0.1	0	1	0	0.077	0.923	0.1
42	0	1	0	0.114	0.885	0.1	0	1	0	0.093	0.908	0.1
43	0	1	0	0.102	0.898	0.1	0	1	0	0.117	0.883	0.1
44	0	1	0	0.1	0.9	0.1	0	1	0	0.129	0.871	0.1
45	0	1	0	0.103	0.897	0.1	0	1	0	0.145	0.855	0.1
46	0	1	0	0.099	0.901	0.1	0	1	0	0.141	0.859	0.1
47	0	1	0	0.123	0.877	0.1	0	1	0	0.175	0.824	0.1
48	0	1	0	0.1	0.9	0.1	0	1	0	0 129	0.871	0.1
49	0	1	0	0.107	0.893	0.1	0	1	0	0.146	0.854	0.1
50	0	1	0	0.108	0.892	0.1	0	1	0	0.135	0.865	0.1

Table 7.2

stt	trbmd			NeuralNet's output			wtbmd			NeuralNet's output		
1	0	1	0	0.302	0.692	0.1	0	1	0	0.1	0.908	0.092
2	0	1	0	0.34	0.66	0.1	0	1	0	0.1	0.886	0.114
3	1	0	0	0.861	0.14	0.1	0	1	0	0.1	0.906	0.094
4	0	1	0	0.014	0.986	0.1	0	0	1	0.1	0.123	0.877
5	1	0	0	0.947	0.053	0.1	0	1	0	0.1	0.904	0.096
6	1	0	0	0.998	0.002	0.1	0	1	0	0.1	0.908	0.092
7	1	0	0	0.898	0.098	0.1	0	0	1	0.1	0.094	0.907
8	1	0	0	0.894	0.109	0.101	0	1	0	0.101	0.898	0.102
9	1	0	0	0.652	0.354	0.1	0	1	0	0.1	0.892	0.108
10	1	0	0	0.966	0.032	0.1	0	1	0	0.1	0.906	0.094
11	1	0	0	0.914	0.087	0.1	0	1	0	0.1	0.898	0.102
12	1	0	0	0.914	0.086	0.1	0	1	0	0.1	0.909	0.092
13	1	0	0	0.958	0.041	0.1	0	0	1	0.1	0.096	0.904
14	1	0	0	0.896	0.11	0.1	0	1	0	0.1	0.892	0.108
15	1	0	0	0.946	0.055	0.1	0	1	0	0.1	0.902	0.098
16	0	1	0	0.024	0.976	0.1	0	1	0	0.1	0.91	0.09
17	1	0	0	0.815	0.18	0.1	0	1	0	0.1	0.895	0.105
18	1	0	0	0.919	0.083	0.1	0	1	0	0.1	0.901	0.099
19	1	0	0	0.749	0.249	0.1	0	1	0	0.1	0.891	0.109
20	1	0	0	0.928	0.074	0.1	0	1	0	0.1	0.899	0.101
21	1	0	0	0.949	0.051	0.099	0	1	0	0.1	0.898	0.103
22	1	0	0	0.863	0.136	0.1	0	1	0	0.1	0.909	0.091
23	1	0	0	0.945	0.056	0.1	0	1	0	0.1	0.903	0.097
24	1	0	0	0.784	0.213	0.1	0	1	0	0.1	0.892	0.109
25	1	0	0	0.884	0.119	0.1	0	1	0	0.1	0.896	0.104
26	0	1	0	0.083	0.919	0.1	0	0	1	0.1	0.102	0.898
27	1	0	0	0.966	0.034	0.1	0	1	0	0.1	0.902	0.098
28	1	0	0	0.922	0.079	0.1	0	1	0	0.1	0.906	0.094
29	1	0	0	0.841	0.156	0.1	0	0	1	0.1	0.106	0.894
30	1	0	0	0.938	0.065	0.1	0	1	0	0.1	0.894	0.106
31	1	0	0	0.922	0.076	0.1	0	1	0	0.1	0.901	0.1
32	1	0	0	0.775	0.225	0.1	0	1	0	0.1	0.896	0.104
33	1	0	0	0.962	0.037	0.1	0	0	1	0.1	0.091	0.909
34	1	0	0	0.928	0.071	0.1	0	1	0	0.1	0.905	0.096
35	1	0	0	0.944	0.056	0.1	0	1	0	0.1	0.904	0.096
36	1	0	0	0.861	0.137	0.1	0	0	1	0.1	0.102	0.898
37	1	0	0	0.928	0.075	0.1	0	1	0	0.1	0.896	0.104
38	1	0	0	0.85	0.15	0.1	0	1	0	0.1	0.9	0.1
39	1	0	0	0.927	0.073	0.1	0	1	0	0.1	0.902	0.098
40	1	0	0	0.878	0.122	0.1	0	1	0	0.1	0.895	0.105
41	0	1	0	0.179	0.824	0.1	0	0	1	0.1	0.098	0.902
42	0	1	0	0.236	0.767	0.1	0	1	0	0.1	0.88	0.12
43	1	0	0	0.893	0.108	0.1	0	1	0	0.1	0.902	0.098
44	1	0	0	0.942	0.058	0.1	0	1	0	0.1	0.907	0.093
45	1	0	0	0.939	0.06	0.1	0	1	0	0.1	0.901	0.099
46	1	0	0	0.875	0.124	0.1	0	1	0	0.1	0.897	0.103
47	1	0	0	0.866	0.131	0.1	0	1	0	0.1	0.892	0.108
48	1	0	0	0.929	0.071	0.1	0	1	0	0.1	0.905	0.095
49	1	0	0	0.927	0.071	0.1	0	1	0	0.1	0.905	0.095
50	1	0	0	0.791	0.205	0.1	0	1	0	0.1	0.91	0.09

ARTIFICIAL NEURAL NETWORK FOR COLOR CLASSIFICATION

Tran Cong Toai
University of Technology, HCMC, Vietnam

1. Introduction

Undoubtedly, the best color classifier is the human being. However applications which require on - line classification of color and selective color adjustments, such as would be the case for color television signals, a substitute to the human classifier would be needed fortunately, a group of classifiers modeled after the biological brain (artificial neural networks) have been developed and studied for years. The hope was to achieve human like performance . We have yet to achieve such a goal. The challenge is to understant how these various neurons interact to achieve the ability of ision, hearing, touch, movement, etc ... One such application, which will be introduced in the forthcoming sections, is color classification. In this option we shall examine several neural network model, their learning schemes, and their effectiveness in color classification.

2. The Perceptron

Figure 1 shows what is believed to be a mathematical model for a single neuron. The node sums N weighted inputs, applies some threshold value θ , and passes the result through a nonlinearity. The output can be either 1, representing the firing of a neuron, or 0 (-1 can also be used). The nonlinearity functions commonly used are the sigmoid function and the limiter (Threshold logic)

The structure in Fig.1 is known as the perception, and is basically a linear classifier that is able of separating two disjoint classes as shown in fig 2. The output of the perceptron can be written as.

$$y = f(\alpha) \quad (1)$$

where

$$\alpha = \sum_{i=0}^{N-1} w_i x_i + \theta \quad (2)$$

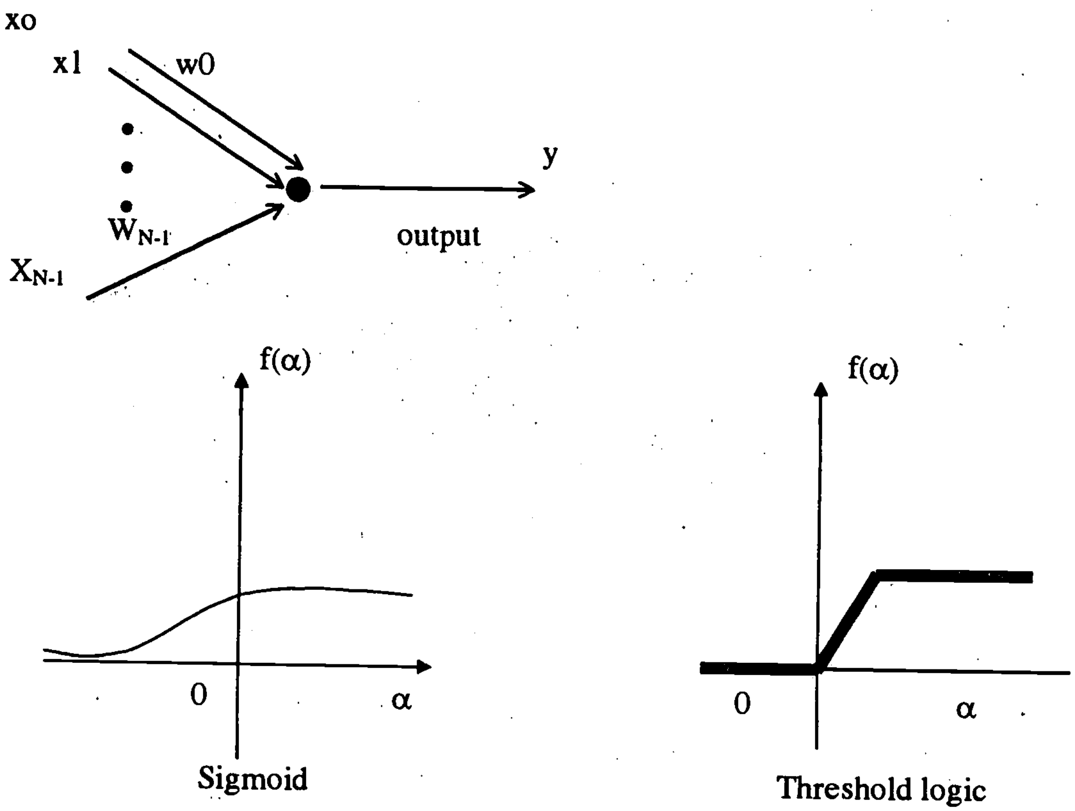


Figure 1: Computational element of a neural structure

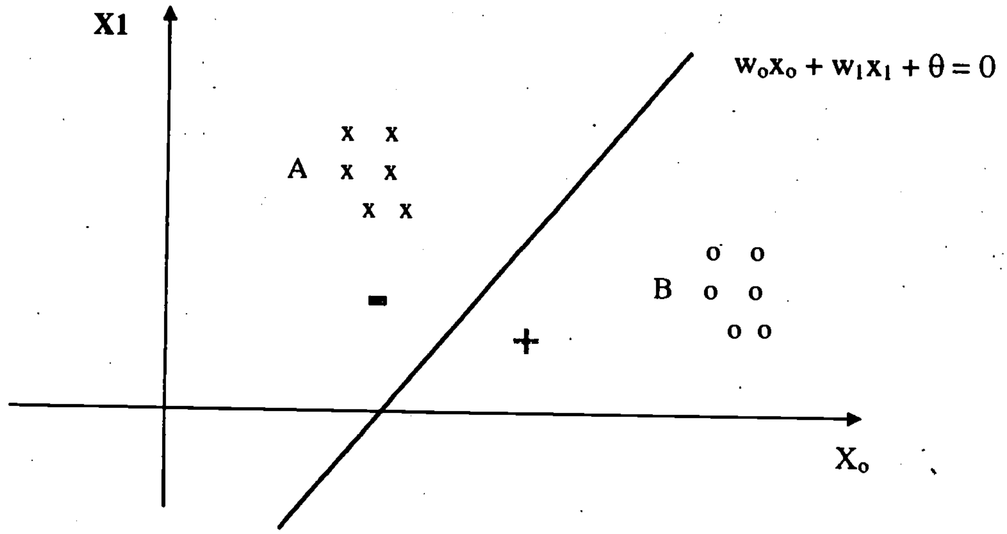


Figure 2: A simple decision Function of two classes

A scheme that determines the weights {wo, w1, ..., w_{N-1}} such that f(α) separates the two classes A and B is, not surprisingly, called a learning scheme. θ is called the threshold, and is usually a fixed number between 0 and 1.

To derive a learning scheme we shall consider for now a perception with just two inputs:

$$\alpha = w_o x_0 + w_1 x_1 + \theta \quad (3)$$

x_o could represent the color feature x of the chromaticity diagram and x1 the color feature y. If we wish the perceptron to separate two color, A and B , then we should expect an output of, say 1 if (xo, x1) belonged to color A and o if the inputs belonged to color B. Alternatively, one can write.

$$\text{if } (x_0,x_1)_p \in A \qquad dp = 1$$

$$\text{if } (x_0,x_1)_p \in B \qquad dp = 0$$

Where the subscript p denotes a pattern reading for (x_0,x_1) and dp denotes the desired output for that pattern. If (w_0,w_1) are known then the actual output y can be calculated from Eq (1). The error for that pattern reading can be given by

$$Ep = 1/2\ (dy - yp)^2 \qquad (4)$$

The problem then becomes the minimization of Ep with respect to wo and w1for all pattern input $(x_0,x_1)p$ such that Eq. Provides the correct separation between the two classes as seen in Fig.2. Ep is a nonlinear function of the variables wo and w1, and hence nonlinear schemes need to be used to minimize it.

If y is given by the sigmoid function

$$y = f(\alpha) = \frac{1}{1 + e^{-\alpha}}$$

Then by differentiating Ep with respect to wo we get

$$\frac{\delta Ep}{\delta Wo} = - (dp - yp)\ yp\ (1 - yp)\ x_{op}$$

and with respect to w_1 we get

$$\frac{\delta Ep}{\delta w_1} = - (dp - yp)\ yp\ (1 - yp)\ x_{1p}$$

The steepest descent algorithm can be used to obtaim the values of the weights as follous:

a. Set the weights (w_0,w_1) and θ to small random values. At iteration K:

b. Present an input value (x_0,x_1) and specify the desired output: 1 if it belongs to one class and 0 if it belongs to the other.

c. Calculate the actual output y

d. Calculate

$$\delta = (d - y)\, y\, (1 - y)$$

e. Calculate the gradients:

$$\nabla E = \left| \begin{matrix} \frac{\delta E}{\delta W_o} & \frac{\delta E}{\delta W_1} \end{matrix} \right| = [\ -\delta x_o - \delta x_1]$$

f. Adjust the weights using the recursive equation

$$W^{(k+1)} = W^{(K)} - \eta \nabla E^{(K)}$$

where $W^{(K)} = [w_0,w_1]^{(K)}$ = weights at iteration K and η is a positive fraction less than 1.

g. Present new value of input or, if data has all been read, recycle same set of data. Go to step 2 and repeat until the weights stabilize, ...

$$\left| wi^{(k+1)} - W^{(k)} \right| \le \varepsilon \qquad i = 0.1$$

Convergence is sometimes faster if a momentum term is added and weights are smoothed by

$$W^{(k+1)} = W^{(K)} - \eta \nabla E^{(K)} + \alpha (\nabla E^{(K)} - \nabla E^{(K-1)})$$

where $0 < \alpha < 1$

The above algorithm is known as the delta rule, and has been used extensively in the literature. Althought the algorithm can produce weights for the classifier, it requires a large number of iterations to converge. The choice of the two parameters α and η seems to be rather arbitrary. To allow you to examine the performance of the delta rule algorithm we present next a C program designed for a perception with two inputs.

- **Program TO.C**

```
# include <stdio.h>

# include <stdlib.h>

# include <math.h>

# include <conio.h>

# include <io.h>
# define eta 0.8
# define alpha 0.2
void main()
    {
    unsigned int d[200];
    unsigned int N, ind, iter, I;
    float W[z], x[2], x1[200], x2[200], net, E;
    float dEp[2], sum, y, theta, dEpold[z], delta ;
    File  * fptr;
    Char filename[14];
        clrscr() ;
        N = 0
        iter = 0;
        gotoxy(1,1);
        printf (" Enter file name containing data") ;
        scanf ("%S", filename);
        fptr = fopen (filename, "r");
        if (fptr == NULL)
            {
             printf (" file%does not exist ", filename);
             exit(1);
            }
        while(fscanf(fptr, " %   d ", &X1[N], &X2[N], &d[N])! = EOF))
            N++ ;
        fclose (fptr);
            srand(1) ;
        W[0] = (float) rand () / 32768.00 ;
            srand(2) ;
        W[1] = (float) rand() /32768.00 ;
        theta = 0.1 ;
        i= 0;
        sum = 0.0 ;
        ind = 1 ;
        gotoxy(1,10);
        printf (" Press esc to exit before convergence") ;
while (ind)
        {
        X[0] = X1[i];
        X[1] = X2[i];
        gotoxy (1,3) ;
        printf (" Iteration # %5d", iter);
        Net = W[0] * X[0] + W [1] * X [1] + theta ;
        if (net >= 20)  E = 0.0  ;
        else E = EXP (- (double)net);
        y = 1.0/ (1.0 + E);
        delta = (d[i] - y ) * y* (1.0 - y) ;
        dEp[0] = X[0] * delta ;
        dEp[1] = X[1] * delta ;
        if (I = = 0)
                {
        w[0]+ = eta * dEp[0];
        w[1]+ = eta * dEp[1] ;
        dEpold[0] =  dEp[0] ;
        dEpold[1] = dEp[1];
        }
        else
                {
                w[0]+= eta * dEp[0] + alpha * (dEp[0] - dEpole[0]) ;
```

```
            w[1]+= eta * dEp[1] + alpha * (dEp[1] - dEpole[1]) ;
            dEpole[0] = dEp[0];
            dEpole[1] = dEp[1];
            }
    Sum+ = fabs (( double) (d[i] - y)) ;
    I ++
    if (I >= N)
    {
    gotoxy (1,6);
    printf (“ Square error = %f”, Sum);
    I = 0; Sum = 0 ;
    iter ++ ;
    }
if (d[i] = = 1)
gotoxy (1,4) ;
else
gotoxy (1,5) ;
printf (“ %d %f”, d[i], y) ;
if (( I = = N) && (Sum < = 1.0 e$^{-1}$)
    {
    gotoxy (1,7) ;
    printf (“ \n W[0] = %f  W[1] = %f”, W[0], W[1]);
    exits(1);
    }
if (Kbhit() ! = 0)
    {
    gotoxy(1,7) ;
    if (getch() = = 27)
    {
            printf ("\n W[0] = %f W[1] = %f" , W[0], W[1]);
            exits (1);
            }
        }
    }
}
```

Available on the accompanying disk is a data file TO. DAT obtained from the chromaticity diagram. Using $\eta = 0.8$ and $\alpha = 0.2$ it took almost 200 iterations for the square error to drop from ~ 28 to ~ 9.55. After 15.000 iterations the error reached a lecel slightly less than one, and wass still decreasing, ... convergence was not reached yet. Changing $\eta = 0.2$ và $\alpha = 0.8$ slowed down the convergence. Before we discuss better techniques for teaching a perceptron we will have to provide you with a tool for collecting data.

References

[1] KS. Knudsen and L. T Bruton " Mixed Multidimensional Filters" PROC - 1990

[2] L.T Bruton and N. R. Bartley " The Design of Hight selective adaptive Three - Dimensional Recursive cone filters " July - 1987

[3] Sid Ahmed, " Image Processing" - International Editions 1995

SYNTHESIZING AND RECOGNIZING VIETNAMESE SPEECH

Hoang Kiem, Nguyen Minh Triet, Vo Tuan Kiet, Thai Hung Van
Luu Đuc Hien, Bui Tien Len
University of Natural Sciences , HCMC, Vietnam

Abstract

Nowadays, the art and science of speech recognition and synthesis have been researched in the information technology world-wide. Each country has its language. It is the purpose of this paper to present some algorithms applied successfully in Vietnamese isolated word recognition and Vietnamese synthesis. In recognition part, we show some fundamental concepts of signal processing used as a pre-processing step. The next step is spectral feature extraction which are Linear Predictive Coding coefficients and Ceptral coefficients. We discuss two methods of recognition: pattern matching and vector quantization. The algorithm composed from time normalization and dynamic time warping theories applied in matching two difference-duration utterance is also reported.

RECOGNITION

Introduction

Sound, one the oldest means of communication in human being, has been the complex automatic processing problem in natural man-machine interface. It has taken a lot of time to input the data and control the system by keyboard and now computer is an indispensable tool in many business.

Recently, existing speech recognition technologies has been proven and shown adequate for simple tasks involving small vocabularies (tens or hundreds of words) and suitable for limited applications (typically, recognition of a set of digits or commands). This paper is intended to be an introduction to the theories and approaches applying a Vietnamese speech recognition system, both practical and experimental. The fundamental structure of the Vietnamese isolated word recognition system is depicted in fig. 1.

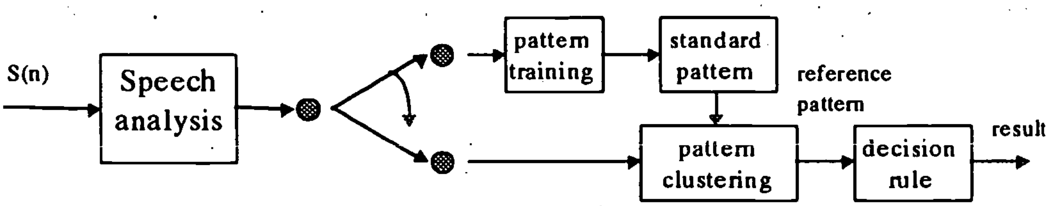


Fig. 1

Pre-processing

There are four step in preprocessing. Initial, these algorithm was tried in the 386 DX PC and 8 bit Sound Blaster Card. The signal was sampled at the rate 8 kHz. First, the signal is cut to reduce the segments at which the value is zero. These segments occur et the beginning and end of the signal. That is the time no sound before and after a word pronounced. Second, we use quantization method to quantize the amplitude of signal by 8 bit to 3 bit. In experiments, it is proved that the sound still keep its quality and saves much computational time. 256 values in squeezed to 8 values by logarit transfer function:

3 bits	0	1	2	3	4	5	6	7
8 bits	0	13	28	44	62	81	103	127

Then the signal is preemphasized in order to flatten the speech signal and reduce computational instabilities:

S'(n) = S(n) - A.S(n-1)

We use A = 0.95.

Finally, as usual, Hamming window is utilized to limit the segment of signal that is concentrated of spectral energy.

S''(n) = S'(n).W(n)

where

$$W(n) = \begin{cases} 0.54 + 0.46\cos(\frac{\Pi n}{N}), |n| \le N \\ 0, else \end{cases}$$

Feature extraction

Linear Predictive coding method is applied in this step. The order of the analysis system in autocorrelation analysis is chosen as

$$R(m) = \sum_{n=0}^{N-1-m} S(n).S(n+m) \quad ; m=0,1,...,p$$

The feature set F = {R(0),...,R(p)} is autocorrelation coefficients.

Then LPC coefficients are converted from autocorrelation coefficients by Durbin's method. The Ceptral coefficients {C_k}, 1≤k≤M, were then computed recursively by using the following relations:

$$c_k = -a_k - \frac{1}{k}\sum_{n=1}^{k-1}(k-n)c_{k-n}a_n$$

Where {a_k} are the LPC coefficients.

In experiments, we use the Euclidean distance, LPC distance proposed by Itakura, Spectral distance, and Mahalanobis distance. In these, the last one, Mahalanobis, gave the best results in time and accuracy

Time alignment

One of the problems of measuring the distance of the two vectors is the different duration. This is similar to putting a point in n-dimension space into m-dimension space (where m≠n). It is solved when the dimension of one vector is contracted or expanded relative to the other. In this domain, Linear time Alignment method and Dynamic time Warping were used. But in experiments it was found that the linear time alignment alone did not give accuracy. As well, the dynamic time warping method consumed much of time because of the recursive algorithms. An algorithm based on the idea of the Dynamic time warping together the Linear time alignment method was developed as follow:

We suppose T and R are the test and reference pattern:

$$T = \{ T(1), T(2), \ldots\ldots\ldots, T(NT) \}$$

$$R = \{ R(1), R(2), \ldots\ldots\ldots, R(NR) \}$$

where NT and NR are the dimensions of vector T and vector R

Linear time alignment is defined as follow:

$$m = w(n) = (n-1)\frac{(NR-1)}{(NT-1)} + 1$$

Applying the function w(n), the element w(n) of the vector R correspondent to the element m linearly is determined, but, in experiment, it was found that this m element did not express the real value respectively. So the neighbour elements will be considered by measuring the distance from them to element m of the vector R. The element which has the minimum distance to m will be selected. Then the element m of the vector R will be matched with the element i of the Vector T, where i satisfy:

$$i = \min(d(m,j)), \qquad j = w(n)-J, \ldots, w(n)+J$$

where J is selected by experiment. It depends on the order of the autocorrelation method. J=10 was used. The greater J is, the greater is the cost of time the algorithm consumes.

Training

In this step, an algorithm that is not training but selecting is built. The pattern is selected from N patterns by the method that we call Dynamic Average. First, the distance for the first pattern to the second one is measured. This measuring is repeated to the rest (N-1) of the patterns. Then the average of these distances is computed. The work will be done again for every pattern in the training set. At the end, every pattern has an average distance of its own. We then obtain the new average value, DAVG, by computing these N average distances. Finally, the pattern whose average distance is nearest to the DAVG will be considered as the standard pattern of the class:

$$DAVG(c) = \frac{\sum_{i=1}^{n} AVR(i)}{n}$$

where

$$AVR(i) = \frac{\sum_{i=1}^{n} D(P(i), P(j))}{n} \quad , i = 1, ..., n$$

where D is the distance measurement between two pattern.

The pattern that is nearest to the DAVG will be chosen as the standard stored in the speech dictionary with maximum of 20 words:

$P(s) = \{ P(i) , i = (1,n) \mid P(i)\text{-}DAVG < P(j) - DAVG \quad \forall j \in (1,n) \}$

Decision rule

We use Bayes rule :

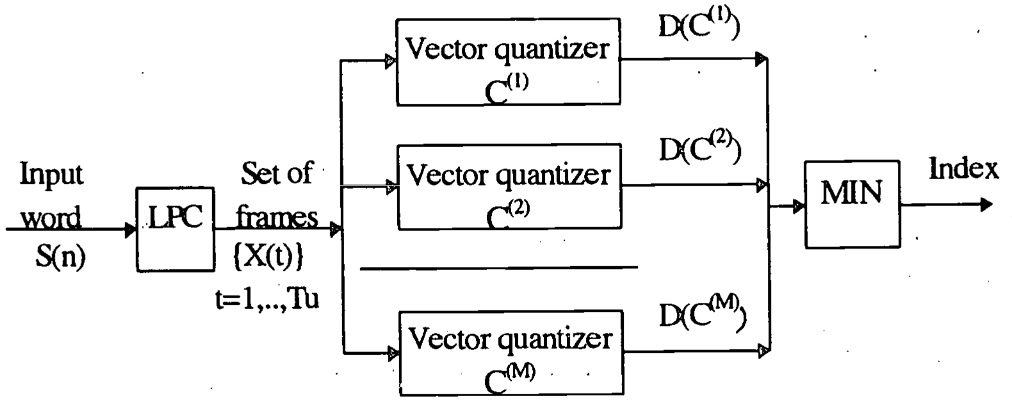
$P(t/s) = \max_t(P(t/s)) = P(t).P(s/t)/P(s)$

whenever there is more than one pattern nearest to the test pattern, the Bayes rule and KNN rule are applied.

Vector quantization

The result of the LPC analysis is a seri of vectors characteristic of the time-varying spectral characteristics of the speech signal. For convenience, we denote the spectral vector as v_l , l=1,2,...,L, where each vector is a p-dimensional vector.

Vietnamese speech recognition system is contructed by the following diagram:



Where

- $C^{(1)}$, $C^{(2)}$,..., $C^{(M)}$ are M codebooks, each codebook responds to a word in the dictionary.

- $D(C^{(i)})$ is a average distortion score which is computed in formula:

$$D(C^{(i)}) = \frac{1}{Tu}\sum_{t=1}^{Tu} d(x_t, \hat{x}_t^{(i)})$$

with

$$\hat{x}_t^{(i)}) = \arg(\min(d(x_t, \hat{y}_j^{(i)}))$$

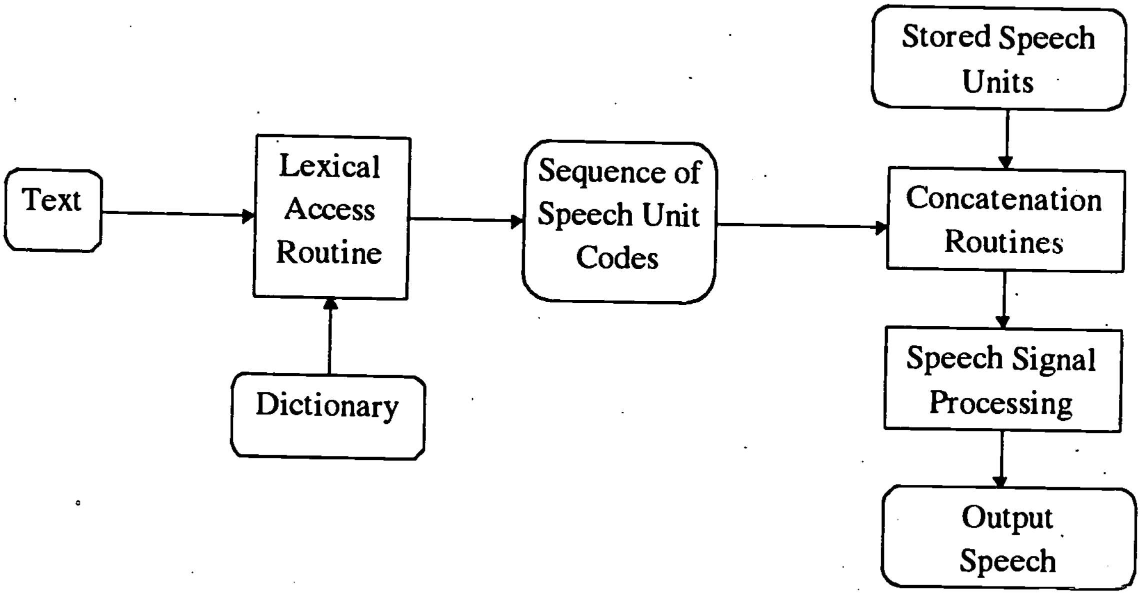
The important problem is how to built a codebook. In practice, we choose 64 entries for a codebook. For each word, we record some samples. After feature analysis by LPC, we obtain a spectral vector including some frames from each sample. Each frame is called a training vector. Then gathering all of frames from the samples of a word, we have a set of training vectors used to build a codebook for that word, We apply the Isodata clustering algorithm to cluster the set of training vectors into a set of 64 entries for the codebook.

SYNTHESIS

Introduction

Speech synthesis involves the conversion of an input text into speech waveforms

Speech synthesis can be characterized by the size of the speech units they



Block diagram of the speech synthesis

concatenate to yield the output speech as well as by the method used to code, store and synthesize the speech. Using large speech, such as phrases and sentences, can give high-quality output speech but requires much memory. Efficient coding methods reduce memory needs but often degrade speech quality. So it is important to choose the speech unit size and method of synthesis.

Vietnamese speech synthesis system based on word concatenation

As discussed above, the main problem in a synthesizer is to choose an appropriate size of the speech unit. For choice, we not two elements:

- The size of the set of units (size of stored dictionary).

- The naturalness of the output speech.

In the Vietnamese synthesizer, we choose words as speech units. It word respond to a waveform recorded from a person. Thus, the Vietnamese synthesizer will need the dictionary including about 6800 entries, approximately 28 MBs. This approach solve the second element: the naturalness of the output speech. However, to yield an output speech smoothly, we need to analyze intonation, rhymth in each sentence of entire text.

RESULTS AND APPLIES

After improving and completing several algorithms, the quality of " Recognizing and systhesizing system" is upgrade obviously. Here by ,there are some applies :

Recognition:

* Recognition Vietnamese speech:

- Previous system: recognize number from 0 to 9 only , manually and consequently.

- Improved system: can recognize from 20 to 50 words, directly and continually. User can speech 4 to 5 words continually and in consequently.

* Morever, Morse Code recognizing is also built:

The system can be recognized directly and continually, or indirectly and inconsequently.

System can also confirm what kind of signal(Signal are typed by hand or machine.)

Synthesis

Building a perfect system which can produce not only northern but southern female voice as well.

Applies which have been put into use:

Our group have applied in many branches.

1. In 116/108 operator: (Automatically answering)

- Problems: the number of operator is very high (appr. 180) -> there are a demand of automatically answering system.
- Solving: build an interactive programme in synthesis and tone-tpuch recognization.
- System will look for database and answer automatically to satisfy user demand.

2. Military applies

- Organition: Researching Institude of Department of Defence and so on.
- Problems: the demand of confirming coordinate of planes in the sky: solders input the coodinate into the computer which control the rockets (a set of coordinate includes 3 parts).
- Solving: establishing an automatically number recognition (from 0 to 9). The accurateness is 99.5%

3. Training and education applies

Setting up a system which trains foreigners in studying Vietnamese.

4. Morse code

Building an automatically Morse Code recognition.

5. In controlling and computing branch

Setting up a system which is controlled by speech, especially in Vietnamese.

Future developing project and short-term project

1. Complete all applies which have been put into use.

2. Widen applies: recognize 20-50 words continualy; synthesis female voice.

3. Studying and improving the ability of recognition about 100 words and synthesis female, adult, child voice and so on.

ON-LINE CHARACTER RECOGNITION

Nguyen Thanh Phuong
University of Natural Sciences, HCMC, Vietnam

Abstract

This abstract presents a real-time handwriting character recognition system based on a structural approach. After the preprocessing operation, a chain code is extracted to represent the character. The most important operation is using a processor for string comparison. The average computation time to recognize a character is about 0.25 to 0.35 seconds. During the learning step, the user define 26 characters and 10 digits to be recognized by the system. The experimental test shows a high accuracy (~99%).

1. Introduction

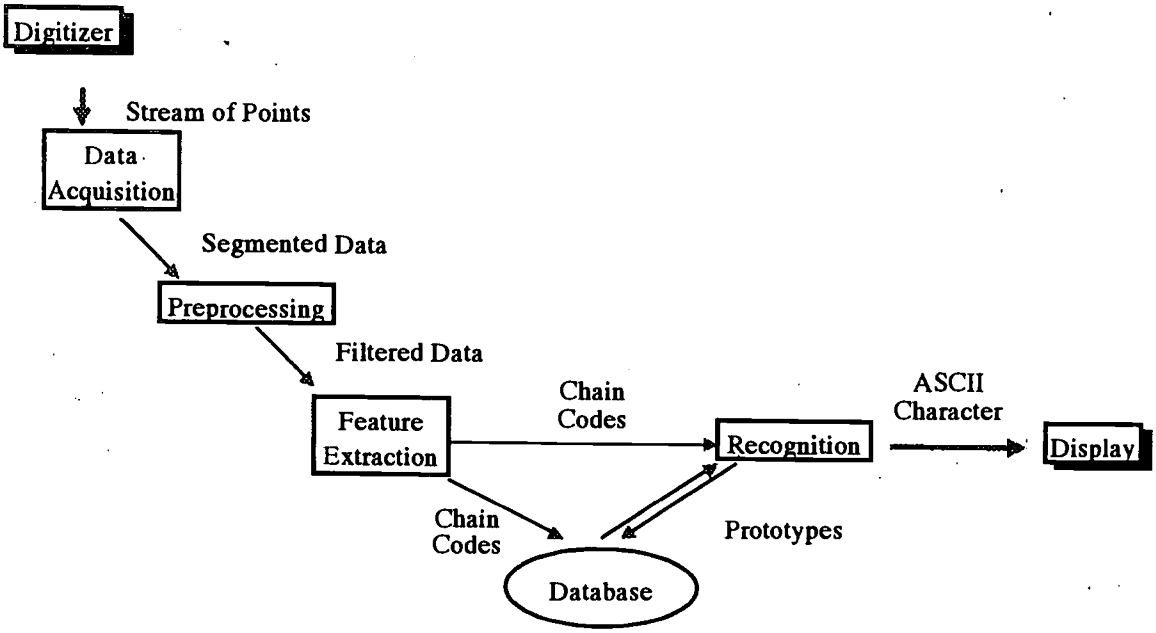
At present, most computer applications solely use the keyboard as a means of entering data or the mouse as a means of pointing to objects. The digitizer, however, enables the user to combine writing, pointing and drawing in a very natural way. The digitizer with a reliable recognition software is the best choice for laptops and palmtops which have energy limit and inadequate space for a keyboard.

For these reasons, our automatic handwriting recognition system has been developed with the following exclusive goal: to isolated character recognition which entails a number of difficulties, the most important of which is variability. The same character can be written in many distinct ways by different writers, or even by the same writer.

A recognizer must be able to recognize isolated characters without the use of boxes. Many applications will not use boxed–input (the use of boxes to separate characters). Box–input makes the users uncomfortable with the process.

Isolated character recognition is performed by a specialized processor that compares a chain code with those in a database and selects the best matches. A personal database is defined for each user. Thus, the user does not have to respect any constraints and can use the system with his own natural handwriting.

A data flow diagram of the system is shown below. Firstly, the data corresponding to the character is digitized and stored. Next, a number of preliminary process are applied to the data in order to prepare it for feature extraction. If the character is processed during the learning operation, its chain code is stored in the database. If the character is recognized, its chain code is compared to those in the database and the system provides the result.



2. Acquisition, segmenting and preprocessing

The digitizing device used in this study is a graphic tablet ("EasyPainter"-Genius Co.) and the resolution is up to 1016 LPI. The pen driver can report the location of the pen at least 100 times per second. This rate ensures that the true path of the pen is reported accurately enough to support the recognizer.

2.1.Segmenting

A segmentation process is applied to isolate the different characters. We use two methods for this process. The first one is based on temporal information. When a given time has elapsed since the last pen contact with the table surface, the acquisition is considered to be completed. It provides a fairly natural process for printing, recognizing and then continuing with more text entry. The second is spatial segmentation. User must write every character disconnected from another character. The strokes used to write each character must be adjoining (< threshold δ) or overlapping strokes.

2.2.Smoothing

The smoothing operation eliminates noise caused by the tablet, trembling while writing, etc. We use two methods for smoothing.

Method 1: this method uses an averaging formula to transform the point coordinates X_i and Y_i to suitable positions, according to the following equation:

$$X_i = (X_{i-3} + 3X_{i-2} + 6X_{i-1} + 7X_i + 6X_{i+1} + 3X_{i+2} + X_{i+3}) / 27$$

$$Y_i = (Y_{i-3} + 3Y_{i-2} + 6Y_{i-1} + 7Y_i + 6Y_{i+1} + 3Y_{i+2} + Y_{i+3}) / 27$$

When the distance and the deviation between points are not considerable, this averaging formula obtains a good result.

Method 2: this method is to reduce high deviation which may influence the feature extraction process.

$$X_{i\_new} = \frac{X_{i\_old} + X_{(i\_old)+1}}{2}$$

$$Y_{i\_new} = \frac{Y_{i\_old} + Y_{(i\_old)+1}}{2}$$

Through this process, with normal rate of handwriting, the deviation is much reduced. For points retained in regions of great curvature, changes are not considerable.

2.3.Spatial Filter

The special filter has 3 functions:

- To eliminate overlapped points.
- To eliminate continuous points with distances (d) less than a fixed threshold (δ).
- To add automatically one or more points if the distance between two original points is more than $n \times \delta$ ($n \geq 1$).

The formula for calculating the new coordinates is :

$$\boxed{\begin{aligned} X &= X_{i-1} + n \times \delta \times \sin\phi_i \\ Y &= Y_{i-1} + n \times \delta \times \cos\phi_i \end{aligned}}$$

where $n = 1 \rightarrow (d / \delta)$ and ϕ_i is calculated by the following formula:

$$\boxed{\phi_i = \tan^{-1} \frac{Y_i - Y_{i-1}}{X_i - X_{i-1}}}$$

However, the consecutive points at cups and corners must be retained at high density. Therefore, we do not use the above formulas when the following condition is true:

$$\min(360 - |\phi_i - \phi_0|, |\phi_i - \phi_0|) \geq \phi$$

where ϕ_0 is the tangent angle for the last point retained

ϕ_i is the tangent angle for the current point.

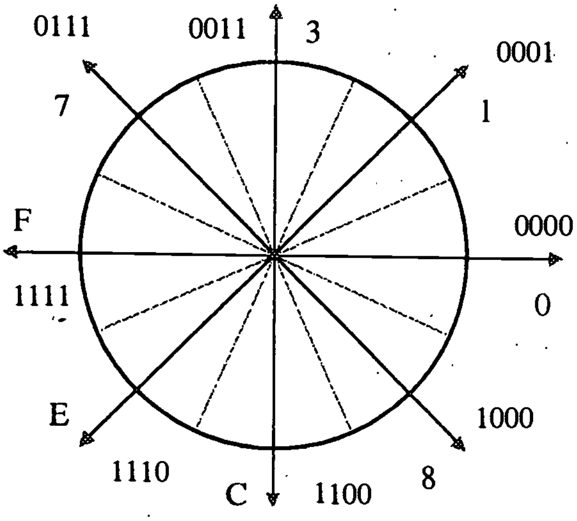
ϕ is a threshold.

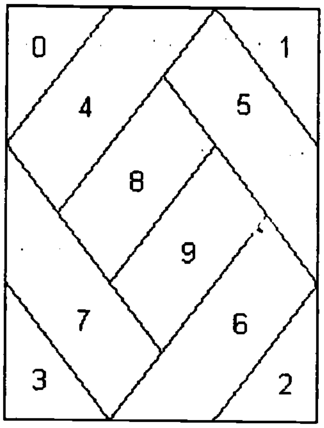
Thus, the spatial filter process eliminates the dynamic information (velocity, acceleration, etc.). In other words, this process renders the system completely independent of the rate of writing.

3. Feature extraction

The most important problem in recognition of handwritten characters is that of extraction of features which are barely affected by handwriting distortion. The following conclusion (derived from paper "Model of Handwriting Process and its Analysis" by S. Kondo and B. Attachoo, Tokai University, Japan) affirms that "the stroke-structure of a character is a quite important factor in the handwriting process". The theory of feature extraction from our study is based on that conclusion.

The structure chosen for feature representation is a string. A character is coded as a string of direction and position symbols, which belong to a basic descriptive alphabet. In fact, if a character has many strokes then each stroke of the character will be described by a component chain code. Thus, the character will be represented by the set of its consecutive component chain codes.





The basic descriptive alphabet (Φ) comprises 8 direction symbols and 10 position symbols.

$$\Phi = \{0_{dir}, 1_{dir}, 3_{dir}, 7_{dir}, 8_{dir}, C_{dir}, E_{dir}, F_{dir} ;\ 0_{pos}, 1_{pos}, 2_{pos}, 3_{pos}, 4_{pos}, 5_{pos}, 6_{pos}, 7_{pos}, 8_{pos}, 9_{pos}\}$$

The direction symbols are extracted as follow. Two points (X_i, Y_i) and (X_{i+1}, Y_{i+1}) define a vector. A vector's slope is given by:

$$\Omega = \tan^{-1} \frac{Y_{i+1} - Y_i}{X_{i+1} - X_i}$$

where $\Omega \in \{0_{dir}, 1_{dir}, 3_{dir}, 7_{dir}, 8_{dir}, C_{dir}, E_{dir}, F_{dir}\}$

The character is surrounded by a rectangle. The starting point of a vector on a fixed cell means that the position symbol is a code from 0_{pos} to 9_{pos}.

A unit of a chain code comprises 2 components: (Position, Direction).

In the coding algorithm, not every primitive object is changed into an equivalent vector. If the latter vector has the same position and direction as the previous one, then only one vector is retained in the chain. This retains the accuracy of the data while overlapping data has been eliminated.

Moreover, with the above algorithm, every character regardless of size, is equally coded. In other words, the recognizing process is completely independent of character size.

4. Learning

Everyone has his own handwriting. The same character can be written in many distinct ways even by the same writer. Each user is required to make his own personal database (Learning phase of the system). Thus, the writer does not have to respect any constraints and can use the system with his own natural handwriting.

The user must write 10 specimens of each character he defines. To enhance the system's operation (rate, storage, accuracy, ...), the user is advised to train the system with his most common way of writing.

5. recognizing

This is the decisive stage. The recognition problem has become the comparison between two strings, which is not simpler but clearer.

The personal database had been loaded into memory before the recognition process. Two characters with the same strokes (one is in memory, another is being recognized) will be compared with each other. This operation reduces the searching field during the comparison phase. Thus, calculation time and recognition errors are reduced.

Matches will be produced one by one after each comparison and the ten highest ones will be retained and updated every time. After the database is inspected, a set of characters correspondent to those matches is produced. The system calculates the sum of the score corresponding to each character and selects the candidate with the highest sum as the classified result.

Elastic matching algorithm:

The conception of string comparison, therefore, is relative (rarely strings have the same length and conventional linear order). In other words, there is no algorithm which is able to produce absolutely exact results in every case, but generally the results are achieved at a relative degree of accuracy. The matter now is to define how high the relative accuracy is. Besides, the algorithm must prove itself flexible and elastic to an object whose variability is instinctive.

Based on the idea:"In the case of the same characters, when a given time has elapsed since the first pen contacts with the tablet surface, the spot of pen on the generated character must approximate the equivalent spot on the standard character", the algorithm is established on the following fundamental:

"During the comparison process, the n^{th} code in the A string will be searched in the B string, from $(n^{th} - \delta)$ to $(n^{th} + \delta)$" where δ is a coefficient of oscillation.

On processing, the primitive chain code is not used directly. It is transformed into a new one by gathering the codes with the same position symbol into a sub chain which starts with that position symbol, followed by direction symbols.

The calculation of matches is made on the following standard:

```
If the same position (symbol):
    Comparing the subchain of direction symbols for resulting the "Res1_dir"
    Res = Res + Res1_dir
Else
    Res = Res - 1
```

"Res1_dir" is calculated as followed:

```
Res1_dir = Res1_dir + Max(-1, Res2_dir)
```

"Res2_dir" is calculated as followed:

Using logical operator "XOR" against a pair of direction symbols
The result = 0 → Res2_dir = 1
=1,2,4,8 → Res2_dir = 0.75
Otherwise → Res2_dir = −1

The final result of recognition is calculated by:

Result = Res / Max(Length of String A, Length of String B)

6. Experimental test

The following figure shows the set of 36 characters used in our experiments:

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9			

The data mentioned below represents the results of 5 participants. For each character the system is trained under 10 specimens and written 100 times in the test. Totally, each participant is required to write (36 × 10 + 36 × 100) = 3960 characters. A test takes about 3 hours per writer.

Writer	Alpha character (%)	Digit character (%)
1	99.31	99.60
2	99.31	99.90
3	99.31	99.50
4	98.92	99.30
5	99.27	(no test)

The next page shows detail recognition results for each character. In each column, we indicate the number of times the character is assigned to each class. Each alpha character was written 500 times and each digit 400 times.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	0	1	2	3	4	5	6	7	8	9
A	500																																			
B		484		11														5																		
C			484									16																								
D		2		498																																
E					500																															
F						500																														
G							499						1																							
H								500																												
I									498											2																
J										496										4																
K											500																									
L												499														1										
M													500																							
N														499						1																
O															480						17	3														
P				6												494																				
Q																	500																			
R											1		2			1		495					1													
S			1																499																	
T										3										497																
U								1													493	6														
V												3										497														
W																					2	6	492													
X																						3		497												
Y																									500											
Z											2															498										
0																											398						1		1	
1																												400								
2																													397					3		
3																													1	397				1		1
4																															399	1				
5																																400				
6																																	400			
7																																		400		
8																											1			1					398	
9																														2		3				395

Comparison of some systems*

Authors	Type of characters	Features	Classification	Number of writers	Number of characters per writer	Number of prototypes	Recognition results
Loy & Landau	Alphanumeric	• Direction • Position	Syntactic - statistic	1	2656	18 per way of writing	98.8
Lu & Brodersen	Alphanumeric and others	• Direction • Velocity	Elastic matching processor	4	500	1	96
Mandler	Alphanumeric	• Direction • Position • Height/width of character	Elastic matching	12	780	3	96.2
Noubound & Plamandon	Alphanumeric and others +, −, (,), ...	• Direction • Position	String comparison processor	15	1770	16	96
Tappert	Alphanumeric and others +, −, (,), ...	• Direction • Position	Elastic matching	9	1441	3	97.3

* Data is derived from the paper "A Structural Approach to On-Line Character Recognition: System Disign and Application" by F.Noubound and R.Plamondon - International Journal of Pattern Recognition and Artificial Intelligence Vol. 5, No. 1 & 2 (1991) 311-335.

Through the test, we have some observations as follows:

The rate of errors on each character (if any) ranges between 1% and 3%; it is up to 7% → 10% for several characters (which came from 2 of the 5 participants).

The errors are generally due to careless writing, divergency from specimens or ambiguity of characters. Those due to the interruption of adjoining (inadvertent occuring during writing process) are not mentioned in the result. Some have an unidentified cause.

The greater the difference between the specimens, the higher the accuracy of the recognition.

7. Conclusion

From the high degree of accuracy (approximately 99%) of the experimental test result, we ascertain that the theory of handwriting recognition in this abstract is convincing. It can be considered as foundational for the next developments of the system.

The use of contextual information, perhaps, will help to solve the problem of the classification of ambiguous characters.

Besides the feature extraction techniques, the elastic matching algorithm must be improved as well as the problem of the inadvertant interruption of adjoining. These are subjects for future study.

Acknowledgements

The author wishes to thank Prof. Dr. Hoang Kiem, Head of Department of Information Technology, The College of Sciences, The Vietnam National University – Ho Chi Minh City for his valuable assistance. – Appreciation goes to PhD. Tran Vinh Phuoc – The College of Technical Sciences and to all those who have kindly supported the author during this study.

Reference

[1] Fathallah Nouboud and Rejean Plamondon, "A structural approach to on-line character recognition: system design and applications", International Journal of Pattern Recognition and Artificial Intelligence Vol. 5, No. 1 & 2 (1991) 311-335.

[2] Shozo Kondo and Boonwat Attachoo, "Model of Handwriting Process and its Analysis" (1986), Department of Communications Engineering, Faculty of Engineering Tokai University, Japan.

[3] "Programmer's Reference", Microsoft Windows for Pen Computing.

[4] "Moderation of the Recognition Process", Microsoft Windows for Pen Computing, Article – Stephen Liffick (Microsoft Corporation, 20 - 3 - 1992).

[5] "Using the Symbol Graph", Microsoft Corporation, Artical – Eric Berman (1992)

DATA MINING AND KNOWLEDGE ACQUISITION FROM A DATABASE

Hoang Kiem, Do Phuc

University of Natural Sciences, HCMC, Vietnam.

Abstract

Data mining is the discovery of interesting relationships and characteristics that may exist implicitly in database. In this paper, we consider how to use multi-dimensional data model (MDDM) for mining rules in a large database. These rules will help us to develop the knowledge for deductive applications [6]. We will create a MDDM and the operations for processing it. After that, we develop two applications for discovering the interesting patterns and the association rules in a database.

1. Introduction

MDDM is a multi-dimensional data model that were used popularly in statistical field, such as the ANOVA method for analyzing the variance among factors. This model is very convenient for mathematical operations [3][5] and in some applications, MDDM can reduce the data redundancy. In temporal databases, MDDM with 3 dimensions are used in modeling the database with the time attribute in the third dimension. Many papers and works related to this fields [1][2][3][5] have proved the efficiency of MDDM in calculation.

In this paper, we will mix the ideas from the above methods .We will define MDDM, database and mathematical operations for MDDM and after that we use MDDM for mining the interesting patterns from data in a decision tree structure and the association rules from a database. Our interest is to discover special relationships and patterns in database, such as :

There are 67% of patients who have concurrently the following characteristics :

"Flu = Yes and Headache=yes and Temperature = very_high", or

There are 59% of customers that buy product X also buy product Y .

There are 67% of customers that buy product X then buy product Y next week.

2. MDDM

In traditional relation database, a relation can be visualized as a two dimensional data structure, one dimension for attributes and another for the tuple of database. This model is not suitable for the statistical and mathematical operations [3][5]. In order to utilize the power of statistical and mathematical methods for discovering interesting patterns and association rules in a database, we define a mixed MDDM as follows:

Let D1,D2,....Dn be the sets of discrete values .

X1,X2,....Xn are the variables ,

D1=Dom(X1) , D2=Dom(X2) ,...., Dn=Dom(Xn).

We consider MDDM as a mapping f(X1,X2,.....,Xn) from B to U :

$$f : B \text{ ----------> } U$$

$$(X1,X2,...Xn) \qquad y = f(X1,X2, \ldots Xn)$$

where B = D1xD2x......Dn is the Cartesian product of D1,D2,.....,Dn

U is a subset of R (real number set).

We call D1xD2x,..Dn is the domain of definition, U is the value set of MDDM f .

We consider (X1,X2,....Xn) as the coordinate and f(X1,X2,.....Xn) as the value of cell. The number of cells in MDDM will be Card (D1)*Card (D2)* ....*Card(Dn) . MDDM is totally defined by giving the domain of definition and the value set of f(X1,X2...,X4).

3. Some database operations in MDDM

Based on the theory of relational database and statistics, we define some database operations, such as projection, selection, filtering, aggregation ....

3.1. Filtering

Filtering FT(f,g,T) with threshold T and MDDM f,g is a database operation for creating another MDDM g(X1,X2,....Xn) from f(X1,X2.....,Xn) as follows :

$$g(X1,X2,....Xn) = \begin{cases} f(X1,X2,..........Xn) & \text{if } f(X1,X2,...........Xn) > T \\ 0 & \text{elsewhere} \end{cases}$$

After filtering FT(f,g,T) , g and f have the same domain of definition .

3.2. Projection

Project PT(f,g,c) where f ,g are MDDM and c is the projection condition.

PT(f,g,c) is defined as follows :

Let f(X1,X2,.......X3) be a MDDM and

D1=Dom(X1) , D2=Dom(X2) = ,... Dn=Dom(Xn).

Let $D1' \subseteq D1$, $D2' \subseteq D2 \ldots Dn' \subseteq Dn$. D1',D2', ...Dn' will contain the values of D1,D2,... Dn that meet the condition listed in parameter c of PT(f,g,c).

We create a MDDM g as follows :

$$g : D1' \times D2' \times \ldots Dn' \text{ -----------> } U'$$

where

$$g(X1,X2,...Xn) = f(X1,X2,....Xn)$$

for each $(X1,X2,...Xn) \in D1' \times D2' \times ....Dn'$

The content of cell (X1,X2,...Xn) will be equal the content of cell (X1,X2,...Xn) for each $(X1,X2,...Xn) \in D1' \times D2' \times ....Dn'$.

The domains of definition of f and g are different, but their dimensions are equal..

4. Some mathematical operations of MDDM

Because the value of f(X1,X2,....Xn) is a real number , we can do mathematical and statistical operations. We define two kinds of operations : global for the whole MDDM and local operation for a part of MDDM.

4.1. Global operation[5][3]

We can do global mathematical operations to the content of all cells of f .For example we can divide the content of all cells of MDDM f by a number r <> 0 and create a new MDDM g that has the same domain of definition with f :

$$g(X1,X2.....,Xn) = f(X1,X2.....,Xn) / r$$

for each $X1 \in D1$; for each $X2 \in D2$....., for each $Xn \in Dn$

Besides of division, we can do multiplication ,addition, subtraction or another statistical operations .

We can create a MDDM h that has the same domain of definition of g(X1,X2.....,Xn) and f(X1,X2.....,Xn) , the value set of h will be defined as follows :

$$h (X1,X2.....,Xn) = f(X1,X2.....,Xn) \oplus g(X1,X2.....,Xn)$$

for each $X1 \in D1$; for each $X2 \in D2$....., for each $Xn \in Dn$

where $\oplus$ is a mathematical operation such as +,-,* , / .

4.2 Mathematical operations by freeing one dimension of MDDM

In this kind of mathematical operations, we free one dimension that means we fix n-1 dimensions of MDDM by assigning n-1 particular values to n-1 variables of these dimensions and extract a one dimensional table. MDDM. If we fix n-1 variables X2 = X21 ; X3 = X31.. Xn=Xn. and extract a MDDM g, as follows :

$$g(X1) = f(X1, X2 = X21 ; X3 = X31.. Xn=Xn1.)$$

where g(X1) is an one dimensional table.

We can calculate the sum ,average, maximum, minimum all the cell contents in this table, the result of this operation will be a real number.

We denote SF for summing operation ,AF for average ... and in this example we have :

$$SF(f, X1=all, X2=X21, \ldots Xn=Xn1) = \Sigma X1$$

$$\text{for each } X1 \in D1$$

where the value "all" of X1 means " for each value of X1 in D1 = Dom(X1) ".

4.3. Local mathematical operations by freeing two dimensions of MDDM

If we free 2 dimensions of MDDM f by fixing the value of n-2 variables of MDDM such as, X3= X31;X4=X41......Xn=Xn1 , we can extract a function g from f defined as follows :

$$g(X1,X2) = f(X1=all,X2=all\ X3=X31, \ldots Xn=Xn1)$$

where g(X1,X2) is a two dimensional table.

Because g(X1,X2) is a two dimensional table, we can do mathematical operations along the dimension X1or X2. For example , we can create an one dimensional table h(X1) from g by summing along the dimension X1 , that means:

$$h(X1=a) = SF(g,X1=a,X2=all) \text{ ; for each } a \in Dom(X1)$$

or we can create MDDM k :

$$k(X2=b) = SF(g,X1=all,X2=b) \text{ ; for each } b \in Dom(X2)$$

We can calculate the sum ,average, maximum, minimum the content of all the cells in this table.

We can free m variables (2 < m < n) and define another MDDM that has m dimensions where n is the dimension of the original MDDM. From the new MDDM, we can define the mathematical operations based on the operations defined in item 4.

5. Using MDDM for mining interesting patterns in a database

5.1. Application to discovering pattern from data in a decision tree structure

In this application, we use MDDM for extracting the interesting patterns from data in a decision tree structure. The induction learning from decision tree is a good model for implementation. MDDM will be used for modeling the decision tree and we used the above mathematical operations for discovering the interesting patterns. Our application depends on the number of instances that has the same attribute values. We hold that the more number of instances of a particular pattern we have ,the more belief we get about this pattern. We use the joint probabilities of the attribute values in a pattern to discover the interesting pattern. We also study the R measure [4] for implementing our methods. The methods listed in [4] base on the numbers of instances that have a specified values .From these numbers , we can infer the probabilities related to the occurrence of instances and classes. In the following sections, we use MDDM for calculating these numbers. Our MDDM also satisfies the measure of Gini that was based on the theory of entropy and the methods that use probabilities and statistics for data classification.

The information table for mining the patterns might be as follows [4]:

Instance	Temperature	Headache	Flu	Number of Instances
e 1	normal	yes	no	n1
e 2	high	yes	yes	n2
e 3	very_high	yes	Yes	n3
e 4	normal	no	No	n4
e 5	high	no	No	n5
e 6	very_high	no	Yes	n6
e 7	high	no	No	n7
e 8	very_high	yes	Yes	n8

where

n1 is the number of instances X(Temperature = normal, Headache = yes ,Flu = no) in DB

......

n8 is the number of instances X(Temperature = very_high, Headache = yes ,Flu = yes)

From this table , we define Dl for Temperature ,D2 for Headache ,D3 for Flu as follows :

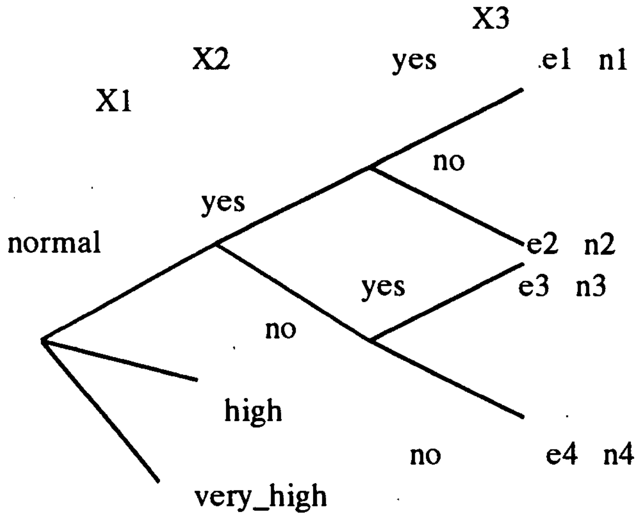
Dl = {normal,high,very_high} D2={yes,no} D3 = { yes , no }

The domain of definition of MDDM f will be DlxD2xD3.

The value set of MDDM will be U={n1,n2,n3,n4,n5,n6,n7,n8} .

Let n be the sum of n1,n2 ..... n8 ; n is also the number of tuples in the data table.

Our MDDM is a good tool for modeling the decision tree structure .From the data table, we can create a tree that a part of this tree is depicted in the following figure:



Based on the Bayesian classification method, the probability of e1 will be calculated as follows:

p(e1) = p(X1=normal)*p(X2=yes/X1=normal)*p(X3=yes/(X1= normal ,X2=yes)

We calculate the following probabilities based on conditional probability and Bayesian classification method:

p(X1=normal)= v1/n where v1 is the number of instances that have X1=normal

p(X2=yes/X1=normal) =v2/v1 where v2 is the number of instances that have X1=normal and X2=yes.

p(X3=yes/(X1= normal ,X2=yes) =n1/v2

That means : e1(X1,X2,X3) will be :

P(e1) = (v1/n)*(v2/v1)*(n1/v2) = n1/n.

We use our global mathematical operations for calculating the value n and divide all cell of f by n to create a new MDDM g. The content of cell in MDDM g will be the occurrence probability of a particular instance. If we want to extract only the interesting patterns that have the probability higher that T, we can use the filtering operation defined in item 3.1. After filtering with the threshold T equal to 50%, we can extract some interesting patterns as follows:

There are 55% of patients who have Flu=yes ,Temperature = High , Headache = Yes

There are 15% of patients who have Flu=no ,Temperature = Normal , Headache = Yes

We can change the threshold T for filtering the necessary concepts.

If we want to calculate the value v1 (the number of instances that have X1=normal), we can use local mathematical method mentioned in 4.2 by freeing two variables X2,X3 or you can free X3 for calculating the value v2 (the number of instances that have X1=normal and X2=yes).

5.2. Application to mining the association rules

In this application, we will use MDDM for mining the association rule. One association rule might be

"There are 80% of customers that buy product X also buy product Y" ,or

"There are 10% of customers that buy product X then buy product Y next week"

.We use MDDM for extracting these patterns. The following information table will contains the TID and Items of customers, as follows :

TID	Item
100	1
100	2
101	3
101	1

TID is transaction Id and Item is the Item number .In TID 100 , there are Items 1,2 ; In TID 101 , there are Items 1,3 .One TID might contain many items.

From this information table, we create a MDDM f that has the domain of definition as follows:

$$D1 = \{100,101\} \ ; D2 = \{ 1,2,3\}$$

In this case, MDDM will be a two dimensional table. If we have the time attribute, MDDM will have three dimensions.

The value set of f will be {0,1} .The content of cell (TID,Item) will be 0 or 1 ; 0 means that TID does not contain this Item ; and 1 if yes .

$$\begin{array}{c|ccc} & & \text{Item} & \\ \text{TID} & 1 & 2 & 3 \\ \hline 100 & 1 & 1 & 0 \\ 101 & 1 & 0 & 1 \end{array}$$

We use template matching method for counting the number of TIDs containing a group of particular Items .We create a MDDM p (Item); p is a pattern that have items {1,3}.The domain of definition of p will be D2 ={1,2,3} .We defined p as follows :

$$p(X) = \begin{cases} 1 & \text{if } X = \text{Item 1 or 3} \\ 0 & \text{elsewhere} \end{cases}$$

That means p(1)= 1 ; p(2)=0 ; p(3) =1.

The template matching is defined as follows :

$$f3(X1) = \begin{cases} 1 & \text{if } \Sigma f(X1,X2)*p(X2) >= n \\ & \qquad \text{for each } X2 \in Dom(X2) \\ 0 & \text{elsewhere} \end{cases}$$

for each $X1 \in Dom(X1)$

where * is a multiplication and n is the number of non zero cells of p .Then we apply the global mathematical operations for summing the content of cell in f3(X) , The result of this summing operation will be the number of TIDs that contain Items 1 ,3 .

6. Conclusion

The MDDM with the facilities of mathematical and database operations has proved the power of organizing and processing data for mining the interesting patterns and rules in a database. This model is very easy for mathematical calculations and implementation.

Besides of these above applications, we already applied this model for mining the sequential pattern in a temporal database. We continue to study the conversion from relational database model to MDDM, the application of MDDM for discovering the fuzzy rules in a fuzzy system, the joining MDDM with the neural network model for classifying or clustering data in data mining field [3].

7. References

[1] Abduillah Uz Tansel , Temporal databases - theory-design-implementation, 1993

[2] Rakesh Agrawal , Fast Algorithms for mining Association rules ,1994

[3] IBM workshop , Data mining ,San Diego, 1996

[4] Ho Tu Bao , The R measure , Hanoi ,IoIT conference ,1996

[5] Jay L. Devore , Probability and Statistics for Engineer and Sciences ,1987

[6] Hoang Kiem,Do Phuc :Embedding an inference engine to a DBMS ,1996 Hanoi,IoIT

[7] Raymond T. Ng Efficient and Effective Clustering Methods for spatial Data Mining,1994

[8] Jiawei Han , Attribute Oriented Induction in Data mining ,1996

GENETIC ALGORITHM FOR INITIATIVE OF NEURAL NETWORKS

Nguyen Dinh Thuc, Tan Quang Sang
Le Ha Thanh, Tran Thai Son
University of Natural Sciences, HCMC, Vietnam

Abstract

In this paper, a procedure for initiative of neural networks based on genetic algorithms is presented. The basis of the calculations is the correlation between every weight and error function. In an examplary real would application for the proposed procedure, the classification rate could be increased by about ten percent.

1. Introduction

Most neural network learning systems use some forms of the back-propagation [1,2]. In practice, however, the back-propagation algorithm encounters tow main difficulties: (1) its rate of convergence is very slow and (2) it does not always guarantee the global minimum to the error function [3,4]. The back-propagation algorithm is based on the steepest descent approach for minimization of the error function. With this approach, the weights are modified in the direction in which the error function decreases most rapidly. This direction is determined by the gradient of the error function at the current point in the weights space. There are two situations in which the steepest descent might result in a slow rate of convergence. The first is when the magnitude of the conponents of the gradient vector very small. The second is when the direction of the gradient vector may not point directly toward the minimum of the error function.

Genetic algorithms are implicit parallism algorithms and powerful direct search optimization tools. To use the genetic algorithm, we must:

(1) Determining the presentation schema;

(2) Determining the fitness measure;

(3) Determining the parameters and variables for controlling the algorithm;

(4) Determining the way of desgnating the result and the criterion for terminating a run.

Once these steps for setting up the genetic algorithm have been completed, the genetic algorithm can be run.

The three steps in executing the genetic algorithm operating can be summarized as follows:

(1) Randomly create an initial population of individual fixed-lenght strings.

(2) Iteratively perform the following substeps on the population of strings until the termination criterion has been satisfied:

(a) Evaluate the fitness of each individual in population.

(b) Create a new population of strings by applying at least the first two of the following three operations. The operations are applied to individual string(s) in the population chosen with a probability based on fitness.

- Copy existing individual strings to the new population.
- Create two new strings by genetically recombining randomly chosen substrings from two existing strings.
- Create a new string from an existing string by randomly mutating at one position in string.

(3) The best individual string that appeared in any generation is designated as the result of the genetic algorithm for the run. This result represent a solution (or approximate solution) to the problem.

2. Genetic algorithm for initiative of neural networks

We tried to build up a learning algorithm for neural network based on genetic algorithm. In this algorithm genetic, each individual represents a weight vector. It is a bit string of length *N*, where each bit represents a weights. The fitness function is the error function of the neural network.

Various authors have proposed the procedures for the main operations of genetic algorithm, In this paper, we consider the procedures studied by De Jong [5]. These procedures are represented as following:

Crossover

```
Input: $L_1,L_2$ - two weight vectors;
Output: L - a new invidual;
(0)    i← 1;
(1)    Randomly_choose($L_1,L_2,W_1,W_2$);
(2)    L[i] ← create($W_1,W_2$);
(3)    i ← i+1;
(4)    If (i<N) Then Goto (1)
       Else Return L;
```

Mutation

```
Input: L - a weight vector;
```

```
Output: L' - a new individual;
For i←1 to N Do
        L'[i] ←L[N-i+1];
```

3. Application: Analyzing chemical data

We succesfully implement artifical neural networks for data classification and apply to analyzing chemical data (Phenol) (see Bang1, Bang2, Bang3). In this case, a three-layer perceptron [2] is used. The network has 26 input neurons, 22 hidden neurons and 1 output neuron. To compare, this network is trained by back-propagation using weight matrix that initated by proposed genetic algorithm and using randomty weight matrix. After the 4000th iteration, the error of the network using proposed algorithm is 0.00017 and the error of the network without using this algoritm is 0.0021.

The network is trained with 73 training patterns. Then, this network is used to classify patterns, The result of the classification is showed in the appendix table.

References

[1] B.Muller, J.Reinhardt - Neural Network - Springer, 1990.

[2] H.Kiem, N.D.Thuc - Analyzing Data Via Neural Network - Proc. Inter. Conf. on
Analysis and Mechanics of Continuous Media, 1995, pp. 217-221.

[3] R.A.Jacob - Increased rates of convergence through learning rate adatption
Neural Networks, Vol.1, pp. 295-307, 1988.

[4] H.A.Malki and A.Moghaddamijoo - Using the K-L Transformation in The Back-
Propagation Training Algorithm - IEEE, p.162-165, 1991.

[5] De Jong, K.A. - An analysis of the behavior of a class of genetic adaptive systems-Dissertation Abstract Internatinal, 1975.

Table 1:

n	logp	sigma	char_1	char_2	char_3	char_4	char_5	char_6	char_7
1	1.48	0.00	-0.253	0.078	-0.213	-0.092	-0.166	-0.097	-0.157
2	2.12	-0.17	-0.254	0.080	-0.152	-0.092	-0.163	-0.100	-0.155
3	2.56	-0.15	-0.256	0.084	-0.152	-0.091	-0.164	-0.099	-0.157
4	2.64	-0.15	-0.255	0.086	-0.153	-0.089	-0.165	-0.098	-0.158
5	3.36	-0.01	-0.256	0.095	-0.044	-0.095	-0.176	-0.088	-0.219
6	1.63	0.06	-0.237	0.046	0.068	-0.135	-0.140	-0.106	-0.191
7	2.20	0.23	-0.245	0.092	-0.150	-0.085	-0.159	-0.094	-0.152
8	2.35	0.23	-0.245	0.112	-0.256	-0.067	-0.165	-0.083	-0.159
9	1.85	0.78	-0.204	0.149	-0.172	-0.034	-0.178	-0.047	-0.232
10	1.60	0.66	-0.241	0.126	-0.104	-0.054	-0.171	-0.072	-0.165
11	2.07	0.42	-0.252	0.122	-0.204	-0.040	-0.180	-0.061	-0.230
12	2.08	0.50	-0.255	0.133	-0.234	-0.047	-0.181	-0.067	-0.178
13	0.65	-0.66	-0.269	-0.010	0.112	-0.201	-0.107	-0.159	-0.160
14	0.81	-0.37	-0.249	0.066	-0.007	-0.169	-0.131	-0.127	-0.133
15	2.19	0.45	-0.215	0.141	0.147	-0.036	-0.186	-0.054	-0.236
16	0.49	0.00	-0.256	0.073	-0.149	-0.076	-0.167	-0.094	-0.161
17	0.44	0.00	-0.253	0.080	-0.163	-0.066	-0.168	-0.095	-0.159
18	2.12	-0.07	-0.253	0.082	-0.217	-0.031	-0.165	-0.094	-0.161
19	2.65	-0.07	-0.253	0.082	-0.220	-0.027	-0.164	-0.095	-0.161
20	3.36	0.06	-0.252	0.078	-0.205	0.013	-0.156	-0.009	-0.153
21	9.53	0.00	-0.253	0.082	-0.221	-0.026	-0.165	-0.095	-0.161
22	1.91	0.34	-0.247	0.105	-0.256	0.129	-0.202	-0.073	-0.172
23	2.48	0.37	-0.248	0.089	-0.212	-0.023	-0.160	-0.089	-0.156
24	2.89	0.35	-0.248	0.076	-0.125	-0.230	-0.134	-0.095	-0.197
25	1.85	0.71	-0.246	0.096	-0.157	0.060	-0.193	-0.073	-0.221

Table 2 :

n	area3	area4	area5	area6	area7	energy	homo	lumo	igc
1	34.72	39.70	39.91	40.04	38.02	-1419.9194	-9.11457	0.39772	-0.43
2	76.43	33.24	39.91	39.91	37.41	-1702.2584	-8.99656	0.37040	-0.27
3	112.37	29.76	40.18	40.14	37.55	-1983.1092	-8.99670	0.38613	0.18
4	135.61	31.28	40.08	39.77	37.58	-2129.3567	-9.02080	0.35021	0.35
5	173.40	23.33	39.10	39.77	33.03	-2622.2650	-8.58464	-0.21799	1.09
6	45.15	36.90	39.91	39.91	43.31	-1429.6447	-9.19489	0.04553	0.28
7	68.20	34.65	39.57	39.91	37.62	-1403.3284	-9.25947	-0.02966	0.28
8	80.59	32.96	39.37	39.91	36.67	-1388.8362	-9.30252	-0.04923	0.50
9	87.80	31.28	39.91	39.91	33.91	-1593.4948	-9.95399	-1.01354	0.67
10	76.54	34.38	40.04	39.91	37.62	-1619.8617	-9.55649	-0.50910	0.03
11	80.96	31.82	40.11	39.70	34.79	-1681.3539	-9.50049	-0.43367	0.48
12	117.00	29.12	39.23	39.91	36.74	-1957.9220	-9.39398	0.45833	0.08
13	67.49	34.28	39.37	39.91	34.11	-1584.8574	-8.01997	0.62018	0.94
14	51.67	34.72	39.37	39.91	37.55	-1523.5630	-8.88516	0.29692	0.75
15	94.87	30.27	40.18	38.83	34.18	-1796.3698	-9.50781	-0.45712	-0.62
16	136.58	21.64	38.89	39.84	37.82	-2131.3311	-9.04435	0.32572	NESd
17	92.41	29.39	39.57	39.91	37.01	-1803.8812	-9.04486	0.34418	-0.95
18	27.30	81.81	32.70	39.70	37.75	-1702.6365	-9.02546	0.38971	0.23
19	26.36	118.73	22.92	39.57	37.89	-1983.0629	-9.01597	0.40417	0.23
20	19.35	176.70	24.00	39.77	37.75	-2625.8289	-8.79831	-0.34179	1.35
21	26.70	509.83	22.32	40.04	37.95	-5648.7091	-9.01942	0.40101	NESd
22	33.64	47.78	38.42	39.91	37.95	-1431.9258	-9.31854	0.04191	0.47
23	29.66	74.35	34.45	39.07	37.95	-1403.7806	-9.30041	0.03908	0.96
24	29.46	98.30	32.09	39.91	34.51	-1376.7021	-9.35918	-0.06962	1.12
25	35.32	91.59	26.03	39.70	34.92	-1617.1112	-9.47541	-0.42754	0.51

Table 3:

n	volu_1	volu_2	volu_3	volu_4	volu_5	volu_6	volu_7	area1	area2
1	71.21	239.74	277.90	204.29	95.82	1087.77	371.33	48.72	57.63
2	56.88	319.43	336.27	382.76	285.81	123.38	1607.93	39.40	47.90
3	125.73	344.52	380.07	432.68	324.86	157.44	1751.66	34.48	41.67
4	78.60	288.80	369.02	325.41	164.79	1760.52	599.73	33.85	40.43
5	211.38	739.48	475.40	455.30	586.77	67.75	24.80	34.32	42.42
6	71.58	863.73	381.76	318.53	275.47	204.29	95.37	48.08	57.19
7	72.72	291.51	479.40	275.01	95.55	201.33	1211.85	42.01	51.12
8	333.59	239.69	648.52	241.83	19.03	180.57	1085.73	33.06	49.23
9	88.47	880.67	436.93	395.88	348.73	258.56	121.06	39.82	48.93
10	47.28	416.72	415.15	295.08	180.60	87.27	293.04	42.10	51.00
11	88.47	880.62	382.98	392.91	349.19	259.61	121.62	41.58	50.48
12	83.99	109.11	1004.44	589.08	559.66	499.41	361.75	30.34	37.83
13	87.01	308.92	1125.21	119.26	257.93	351.44	396.89	43.07	52.18
14	80.34	298.82	313.82	313.34	231.67	107.13	1344.01	45.30	54.61
15	58.06	394.62	359.81	1600.59	286.78	131.24	387.63	39.98	48.48
16	134.50	355.48	447.18	380.73	308.98	160.80	1632.10	38.73	46.22
17	107.82	326.92	370.27	350.73	270.98	132.54	1487.72	39.40	47.90
18	96.40	888.55	364.47	416.60	111.24	220.20	392.15	48.93	58.04
19	114.67	936.82	274.48	336.58	356.20	424.13	446.12	48.66	57.37
20	156.10	374.26	567.83	434.10	350.24	185.94	1902.12	48.77	57.68
21	456.47	1248.61	1187.04	1021.76	398.56	841.04	1295.49	49.71	58.21
22	70.64	289.74	352.54	261.88	96.22	201.33	1211.99	48.88	57.99
23	71.25	863.40	97.38	362.84	307.49	349.50	374.33	49.04	58.15
24	71.45	863.60	368.94	695.31	274.12	204.29	97.59	49.04	58.15
25	88.55	880.70	461.24	335.76	345.10	258.35	121.23	49.71	58.01
26	80.12	872.27	413.46	407.11	315.56	231.67	109.78	49.04	57.95

Bộ dữ liệu Phenol 2.1

	Kết xuất trong		
Stt	mẫu	mạng	
1	0.334211	0.33	ok
2	0.376316	0.433	
3	0.494737	0.459	ok
4	0.539474	0.53	ok
5	0.734211	0.728	ok
6	0.521053	0.521	ok
7	0.521053	0.486	ok
8	0.578947	0.599	ok
9	0.623684	0.635	ok
10	0.455263	0.477	ok
11	0.573684	0.533	ok
12	0.468421	0.464	ok
13	0.694737	0.697	ok
14	0.644737	0.573	
15	0.284211	0.29	ok
16	0.197368	0.232	ok
17	0.507895	0.539	ok
18	0.507895	0.501	ok
19	0.802632	0.804	ok
20	0.571053	0.541	ok
21	0.7	0.675	ok
22	0.742105	0.751	ok
23	0.581579	0.574	ok
24	0.428947	0.437	ok
25	0.468421	0.482	ok
26	0.347368	0.381	ok
27	0.305263	0.317	ok
28	0.276316	0.313	ok
29	0.228947	0.297	
30	0.410526	0.378	ok
31	0.434211	0.48	ok
32	0.597368	0.518	
33	0.502632	0.502	ok
34	0.573684	0.577	ok
35	0.615789	0.614	ok
36	0.615789	0.633	ok
37	0.810526	0.791	ok
38	0.452632	0.452	ok
39	0.592105	0.604	ok
40	0.626316	0.621	ok
41	0.671053	0.689	ok
42	0.955263	0.866	
43	0.618421	0.605	ok
44	0.584211	0.545	ok
45	0.518421	0.487	ok
46	0.423684	0.44	ok

	Giá trị	
Stt	thực	dự đoán
1	0.815789	0.826
2	0.692105	0.454
3	0.542105	0.516
4	0.894737	0.852
5	0.763158	0.713
6	0.915789	0.873
7	1	0.884
8	0.784211	0.813
9	0.986842	0.912
10	0.784211	0.796

These Proceedings are printed with the support of Informatics College, Singapore

U.S. DEPARTMENT OF EDUCATION

OFFICE OF EDUCATIONAL RESEARCH AND IMPROVEMENT (OERI)

EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

REPRODUCTION RELEASE (Specific Document)

I. DOCUMENT IDENTIFICATION

Title: IT@EDU98: INFORMATION TECHNOLOGY IN EDUCATION TRAINING. PROCEEDINGS. JANUARY 15-16, 1998, HO CHI MINH CITY, VIETNAM

Author(s): Dr. Thomas Owens and others

Corporate Source (If appropriate): Northwest Regional Educational Laboratory

Publication Date: 1998

II. REPRODUCTION RELEASE

In order to disseminate as widely as possible timely and significant materials of interest to the educational community, documents announced in the monthly abstract journal of the ERIC system, Resources in Education (RIE), are usually made available to users in microfiche and paper copy (or microfiche only) and sold through the ERIC Document Reproduction Service (EDRS). Credit is given to the source of each document, and, if reproduction release is granted, one of the following notices is affixed to the document.

If permission is granted to reproduce the identified document, please CHECK ONE of the options and sign the release below.

CHECK HERE ➧ XX Microfiche (4" x 6" film) and paper copy (8½" x 11") reproduction

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Northwest Regional
Educational Laboratory
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

OR □ Microfiche (4" x 6" film) reproduction only

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY
____
____
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Documents will be processed as indicated provided reproduction quality permits. If permission to reproduce is granted, but neither box is checked, documents will be processed in both microfiche and paper copy.

"I hereby grant to the Educational Resources Information Center (ERIC) nonexclusive permission to reproduce this document as indicated above. Reproduction from the ERIC microfiche by persons other than ERIC employees and its system contractors requires permission from the copyright holder. Exception is made for non-profit reproduction of microfiche by libraries and other service agencies to satisfy information needs of educators in response to discrete inquiries."

SIGN HERE ➧

Signature: Jerry D. Kirkpatrick Printed Name: Jerry D. Kirkpatrick

Organization: Director, Institutional Development and Communications
Northwest Regional Educational Laboratory ~~Position:~~

Address: 101 S.W. Main St., Suite 500 Tel. No.: (503) 275-9517
Portland, OR Zip Code: 97204 Date: 3/13/98

III. DOCUMENT AVAILABILITY INFORMATION (Non-ERIC Source)

If permission to reproduce is not granted to ERIC, or, if you wish ERIC to cite the availability of the document from another source, please provide the following information regarding the availability of the document. (ERIC will not announce a document unless it is publicly available, and a dependable source can be specified. Contributors should also be aware that ERIC selection criteria are significantly more stringent for documents which cannot be made available through EDRS.)

Publisher/Distributor: __________
Address: __________
Price Per Copy: __________ Quantity Price: __________

IV. REFERRAL TO COPYRIGHT/REPRODUCTION RIGHTS HOLDER

If the right to grant reproduction release is held by someone other than the addressee, please provide the appropriate name and address:

____
____