A Genetic Algorithm for Feature Selection in a Neuro-Fuzzy OCR System

Shamik Sural, Data-Core Systems, 3700 Science Center, Philadelphia, PA 19104, USA.
P.K.Das, Dept. of Comp. Science & Engineering, Jadavpur University, Calcutta 700032, India.

Abstract

We have worked on the development of a character recognition system in the soft computing paradigm. In this paper we present a genetic algorithm used for feature selection with a Feature Quality Index (FQI) metric. We generate feature vectors by defining fuzzy sets on Hough transform of character pattern pixels. Each feature element is multiplied by a mask vector bit before reaching the input of a multilayer perceptron (MLP). The genetic algorithm operates on the bit string represented by the mask vector to select the best set of features. The method has been tested with three benchmark data sets and the results show a fast convergence of the genetic algorithm.

Keywords : Feature Selection, Feature Quality Index, Genetic Algorithm, Multilayer Perceptron, Optical Character Recognition

1. Introduction

Any pattern recognition system typically consists of selection and extraction of useful features from a pattern and use of a classifier to distinguish it from a set of similar looking patterns. A pattern can have a large number of measurable attributes, all of which may not be necessary for uniquely identifying it from other patterns in a particular domain of classification problem using a chosen classifier. Good features enhance within-class pattern similarity and between-class pattern dissimilarity. Thus, the selection of measurable attributes is a crucial step in pattern recognition system design. Devijver and Kittler [4] have stated that the reason for feature selection is "to curtail the effect of the 'curse of dimensionality' phenomenon on the complexity of the classifier". In an attempt to develop a neuro-fuzzy classifier for character recognition [17] using multilayer perceptrons [10], we felt that the need for feature selection is more prominent in the connectionist framework as the classification power of a neural network is implicitly stored in a set of inter-connection weights.

Ruck et al [15] have developed an algorithm for feature ranking using multilayer perceptrons. The sensitivity of a neural network output to its inputs is used by them to rank the features. Priddy et al [14] have presented a probability of error based method for determining the saliency of input features and hidden nodes. They have shown that the partial derivative of the output nodes with respect to a given input feature yields a sensitivity measure for the probability of error. This partial derivative provides a saliency metric for determining the sensitivity of the feedforward network trained with a mean squared error learning procedure to a given input feature. Belue and Bauer [1] have developed a method that takes into consideration the saliency of a feature relative to the saliency of a known irrelevant feature. To establish a working procedure for determining which features are significant, a noise variable is included as a feature input along with the original inputs to represent an absolutely insignificant piece of information. The neural network is trained with the original features as well as the added feature. The saliency of all the features is then computed. The training and computation of saliency is updated a number of times. A confidence interval constructed around the average saliency of the injected noise is used to identify features that contribute better to classification. Pal and Chintalapudi [12] have proposed a connectionist model for selection of a subset of good features for pattern recognition problems. Each input node of an MLP has an associated multiplier, which allows or restricts the passing of the corresponding feature into the higher layers of the net. A high value of the attenuation factor indicates that the associated feature is either redundant or harmful. The network learns both the connection weights and the attenuation factors. At the end of learning, features with high value of the attenuation factors are eliminated.

De *et al* [3] have suggested an MLP-based feature selection technique using a Feature Quality Index. The FQI based feature ranking process uses the concept that the influence of a feature on an MLP output is related to the importance of the feature in discriminating among classes. The impact of the q^{th} feature on the MLP output out of a total of 'p' features is measured by setting this feature value to zero for each input pattern $\mathbf{x_i}$, i = 1,2,...,n. FQI is defined as the deviation of the MLP output with q^{th} feature value set to zero from the output with all the features present. We have used a genetic algorithm (GA) to aid the feature selection process using FQI in an optical character recognition system. In section 2 of this paper we

Sixth International Conference on Document Analysis and Recognition (ICDAR2001), Seattle, 2001, pp. 987-991.

give a brief overview of the neuro-fuzzy classifier designed by us. The genetic algorithm for feature selection is presented in section 3. Finally we present the results and draw conclusions in section 4 of the paper.

2. Overview of the neuro-fuzzy classifier for character recognition

We use Hough transform [6] to extract features from each character pattern both during training of an MLP and recognition. An important observation on Hough transform is that it provides three characteristics of a straight line in an image. These are the values of ρ, θ and count of a (ρ-θ) accumulator cell used for Hough Transform implementation. If an input character pattern is corrupted by noise, some of the features may be missed out due to the thresholding done on the accumulator cell counts. To overcome this problem, instead of thresholding, we define a number of fuzzy sets to extract information from the Hough transform accumulators. These fuzzy set membership functions are listed in table I for θ values in the first quadrant. Similar fuzzy sets are defined on the (a,b,c) accumulator cells for circle extraction where (a,b) denotes the centre of a circle and c, its radius. These set definitions are shown in table II. Based on the basic fuzzy sets defined on Hough transform accumulator, a number of new fuzzy sets are next synthesized using t-norms defined in table III. We use the standard intersection: i(p,q) = min(p,q) as the t-norm [9]. The height of each synthesized fuzzy set is used to define a feature element and the set of 'n' such feature elements constitute an n-dimensional feature vector for a character. Initially we have n=20 features as shown in table III. The feature vectors extracted from all the characters form the input of the MLP.

We define the MLP outputs to represent fuzzy pattern classes and the MLP learns the degree by which an input feature vector belongs to each of these classes. When the MLP is trained with the sample character patterns, the expected outputs corresponding to each input pattern is computed based on a distance measure between the input feature vector and the feature vector of the character represented by the particular output unit. Consider a P-class problem domain with P nodes in the output layer of the MLP where each character pattern is represented by an n-dimensional feature vector $\overline{F}_i$. The Euclidean distance between $\overline{F}_i$ and other feature vectors is calculated as follows.

$$d_{ik} = [\sum_j (F_{ij}-F_{kj})^2]^{1/2} \quad k = 1,2,..., P \quad j=1,2,..,n \qquad (1)$$

The k^{th} expected output of the MLP for the input vector $\overline{F}_i$ is defined as:

$$O^i_{k(exp)} = \mu_k(\overline{F}_i) = 1/[1+(d_{ik}/f_{den})^{fpow}] \qquad (2)$$

It is seen that, $\mu_k(\overline{F}_i) \in [0,1]$, $\mu_k(\overline{F}_i) = \mu_i(\overline{F}_k)$, $\mu_k(\overline{F}_k) = 1$, and $d_{ik} \geq d_{il} \Rightarrow \mu_k(\overline{F}_i) \leq \mu_l(\overline{F}_i)$. Further, for $f_{den} \rightarrow 0$ and $f_{pow} \rightarrow \infty$, the fuzzy MLP output reduces to a conventional MLP output with $O^i_{k(exp)} = 1$ for i = k, and 0 otherwise. The MLP is trained with the input fuzzy feature vectors and fuzzy expected outputs by the back propagation algorithm [16]. As is evident, there is little initial information explicitly available on the usefulness of the features defined by us. We use a genetic algorithm to select the best set of features from the 20 features defined initially for the multilayer perceptron.

3. Genetic algorithm for feature selection using FQI

The Feature Quality Index defined in the introductory section can be written as:

$$FQI_q = \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{O_i} - \mathbf{O_i}^{(q)}\right\|^2. \qquad (3)$$

Here $\mathbf{O_i}$ and $\mathbf{O_i}^{(q)}$ are the output vectors with all the p features present and with the q^{th} feature set to zero, respectively. The features are ranked according to their importance as $q_1, q_2, ..., q_p$ if $FQI_{q1} > FQI_{q2} > ... > FQI_{qp}$. In order to select the best p' features from the set of p features, ${}^pC_{p'}$ possible subsets are tested, one at a time. The quality index $FQI_k^{(p')}$ of the k^{th} subset S_k is measured as

$$FQI_k^{(p')} = \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{O_i} - \mathbf{O_i}^{k}\right\|^2. \qquad (4)$$

Here $\mathbf{O_i^k}$ is the MLP output vector with $\mathbf{x_i^k}$ as the input. $\mathbf{x_i^k}$ is derived from $\mathbf{x_i}$ as follows.

$$x_{ij}^k = \begin{cases} 0 & \text{if } j \in S_k \\ x_{ij} & \text{otherwise} \end{cases} \qquad (5)$$

A subset S_j is selected as the optimal set of features if $FQI_j^{(p')} \geq FQI_k^{(p')} \ \forall k; k \neq j$. It is observed that the value of p' should be pre-determined and that ${}^pC_{p'}$ number of possible choices are to be verified to arrive at the best feature set. It is also evident that no *a priori* knowledge is usually available to select the value of p' and an exhaustive search is to be made for all values of p'; $p' = 1,2,..., p$. The number of possible trials then becomes $(2^p - 1)$ which is prohibitively large for high values of p. We

use genetic algorithm [5] for fast selection of the best feature set based on Feature Quality Index. To do this, we define a mask vector **M** where $M_i \in \{0,1\}$; i=1,2,...,p and each feature element q_i, i = 1,2,...,p is multiplied by the corresponding mask vector element before reaching the MLP input as shown in Fig. 1. The MLP inputs may then be written as follows.

$$I_i = q_i M_i ; \; i = 1,2,...,p.$$

$$= \begin{cases} 0 \text{ if } M_i = 0 \\ q_i \text{ otherwise} \end{cases} \quad (6)$$

Thus, a particular feature q_i reaches the MLP if the corresponding mask element is set to one. To find the sensitivity of a particular feature q_j, we have to set the mask bit M_j to zero. When we select the k^{th} subset S_k of the feature set $\{q_1, q_2, ...,q_p\}$, all the corresponding mask bits are set to zero and the rest are set to one. When the feature set multiplied by these mask bits reaches the MLP, we get the effect of setting the features of the subset S_k to zero and calculate the value of FQI_k. It should be kept in mind that the k^{th} subset thus chosen may contain any number of feature elements and not a pre-specified p' number of elements. Starting with an initial population of strings representing the mask vectors, we use a genetic algorithm with *reproduction*, *crossover* and *mutation* operators to determine the best value of the objective function. The objective function is the FQI value of the feature set S_k selected with the mask bits set to zero for the selected features and is given by

$$FQI_k = \frac{1}{n} \sum_{i=1}^{n} \left\| \mathbf{O_i} - \mathbf{O_i}^k \right\|^2 .$$

In this process, we solve both the problems of pre-determining the value of p' and searching through the ${}^{p}C_{p'}$ possible combinations for each value of p'. After running the genetic algorithm for a sufficiently large number of generations, the mask string with the best objective function value is determined. The feature elements corresponding to the mask bits zero are chosen as the selected set of features.

4. Results and conclusions

We have first tested the GA based feature selection method on two benchmark data sets, namely, Iris data and Crude Oil data [7]. We have checked both the intra-group ranks as well as overall score for each set of features where the intra-group rank is the ranking obtained for a fixed number of features. The overall rank is used for the process of feature selection. It has been found that for the Iris data set, the best two features are 2 and 3 and the least important feature is the feature 1. For Crude-oil data set, 3 and 4 are the best features and 2 is the least important feature. It must be mentioned that the importance of a feature depends on the particular classifier chosen for the classification of patterns. The results match with those obtained in most of the experiments done by other researchers [8,11].

The number of features in the Iris data set is 4 while that in the Crude-oil data set is 5. The effectiveness of the genetic algorithm in selecting the best set of features cannot be properly judged from such a small number of features. We have, therefore, used the genetic algorithm to select the best set of features from Mango-leaf data [2,13], which contains 18 features. We have first performed an exhaustive search through all the 2^{18} possible combination of features to find the best feature set for a particular MLP configuration. The best combination was found to be $(q_1q_2q_3q_4q_5q_8q_9q_{10}q_{11}q_{12}q_{13}q_{14}q_{16}q_{17}q_{18})$. We have then used the genetic algorithm to select the best set of features for the same MLP configuration. It has been found that the best set of features (which is indeed the best set as known from the exhaustive search) is selected within 2000 generations (the best value is first obtained within 800 generations) by the genetic algorithm where each generation has a population of 18 chromosomes. The fitness function is evaluated 36,000 number of times during this process. The time taken is much less compared to that required for the exhaustive search.

Once the effectiveness of the GA based feature selection method is established, we use it for feature selection in the OCR system developed by us. The GA parameters are: Chromosome Length = 20, Population Size = 20; Crossover Probability = 0.67 and Mutation Probability = 0.02. The results of the GA over different generations are shown in table IV. It is seen that the feature selection process has converged well within 4000 generations (The best value is first obtained within 2500 generations) during which the FQI has been evaluated 80,000 number of times. This is much less compared to 2^{20} number of times required for an exhaustive search. The selected set of features is (1,3,5,6,7,9,10,11,12,13,15,16,17,19,20). The number of features is reduced from 20 to 15 which is a reduction of 25%. The MLP is next trained with only this set of features for classification. The time and resource requirements during classification are, thereby, greatly reduced, without affecting the recognition accuracy.

It is concluded that the genetic algorithm makes the feature selection process highly efficient in the optical character recognition system. Since there is no gradient information present in the feature values, it is otherwise difficult to formulate a directed search for feature selection. The genetic algorithm as proposed here is a generic method and hence, can be used for feature selection in other pattern recognition problems also.

Table I Fuzzy set membership functions for line extraction from a character pattern of height X and width Y.

Fuzzy Set	Membership Function	Notation
Long line	count/$[(X^2+Y^2)^{1/2}]$	LL
Short line	2LL if count $\leq [(X^2+Y^2)^{1/2}]/2$ 2(1-LL) if count $> [(X^2+Y^2)^{1/2}]/2$	SL
Nearly horizontal line	$\theta/90.0$	HL
Nearly vertical line	1-HL	VL
Slant line	2HL if $\theta \leq 45.0$ 2(1-HL) if $\theta > 45.0$	TL
Line near top border	ρ/X if HL > VL 0 otherwise	NT
Line near bottom border	1-NT if HL > VL 0 otherwise	NB
Line near vertical centre	2NT if (HL > VL and $\rho \leq X/2$) 2(1-NT) if (HL > VL and $\rho > X/2$) 0 otherwise	NVC
Line near right border	ρ/Y if VL > HL 0 otherwise	NR
Line near left border	1-NR if VL > HL 0 otherwise	NL
Line near horizontal center	2NR if (VL > HL and $\rho \leq Y/2$) 2(1-NR) if (VL > HL and $\rho > Y/2$) 0 otherwise	NHC

Table II Fuzzy set membership functions for circle extraction from a character pattern of height X and width Y.

Fuzzy Set	Membership Function	Notation
Large circle	c/(X/2)	LC
Small circle	2LC if $c \leq (X/4)$ 2(1-LC) if $c > (X/4)$	SC
Centre near right border	a/Y	CRB
Centre near left border	1-CRB	CLB
Centre near horizontal mid-point	2CRB if $a < (Y/2)$ 2(1-CRB) otherwise	CHM
Centre near top border	b/X	CTB
Centre near bottom border	1-CTB	CBB
Centre near vertical mid-point	2CTB if $b < (X/2)$ 2(1-CTB) otherwise	CVM
Centre near mid-point	(2CHM)CVM	CMP
Dense circle	count/$2\pi c$	DC
Sparse circle	2DC if count $\leq \pi c$ 2(1-DC) Otherwise	PC

Table III Synthesized fuzzy set definitions

Srl. No.	Synthesized Fuzzy Set	Definition (i ≡ t-norm)
1.	Long slant line	i(TL,LL)
2.	Short slant line	i(TL,SL)
3.	Nearly horizontal short line near vertical center	i(HL,i(SL,NVC))
4.	Nearly horizontal short line near top border	i(HL,i(SL,NT))
5.	Nearly vertical long line near left border	i(VL,i(LL,NL))
6.	Nearly vertical long line near right border	i(VL,i(LL,NR))
7.	Nearly horizontal long line near top border	i(HL,i(LL,NT))
8.	Nearly horizontal long line near bottom border	i(HL,i(LL,NB))
9.	Nearly vertical long line near horizontal centre	i(VL,i(LL,NHC))
10.	Nearly vertical short line near horizontal centre	i(VL,i(SL,NHC))
11.	Large dense circle with centre near mid-point	i(LC,i(DC,CMP))
12.	Large sparse circle with centre near mid-point	i(LC,i(PC,CMP))
13.	Large sparse circle with centre near bottom border on horizontal mid-point	i(LC,i(PC,i(CB B,CHM)))
14.	Small sparse circle with centre near left border on vertical mid-point	i(SC,i(PC,i(CL B,CVM)))
15.	Small dense circle with centre near top border on horizontal mid-point	i(SC,i(DC,i(CT B,CHM)))

Srl. No.	Synthesized Fuzzy Set	Definition (i ≡ t-norm)
16.	Small sparse circle with centre near top left border	i(SC,i(PC,i(CTB,CLB)))
17.	Small sparse circle with centre near top right border	i(SC,i(PC,i(CTB,CRB)))
18.	Small sparse circle with centre near bottom border on horizontal mid-point	i(SC,i(PC,i(CBB,CHM)))
19.	Small sparse circle with centre near mid-point	i(SC,i(PC,CMP))
20.	Small dense circle with centre near mid-point	i(SC,i(DC,CMP))

Table IV Performance of the genetic algorithm for feature selection in the OCR system.

No. of Generations	Best Objective Function Value	Best Fit String
10	1.690104	01010100001000100000
20	1.759118	01100101001000100000
50	1.759118	01100101001000100000
100	1.790014	01000101001000000000
500	1.820325	01010101000101000100
1000	1.899543	01000001000101000100
2000	1.903486	01010001000010000100
2500	1.917685	01010001000001000100
3000	1.917685	01010001000001000100
4000	1.917685	01010001000001000100

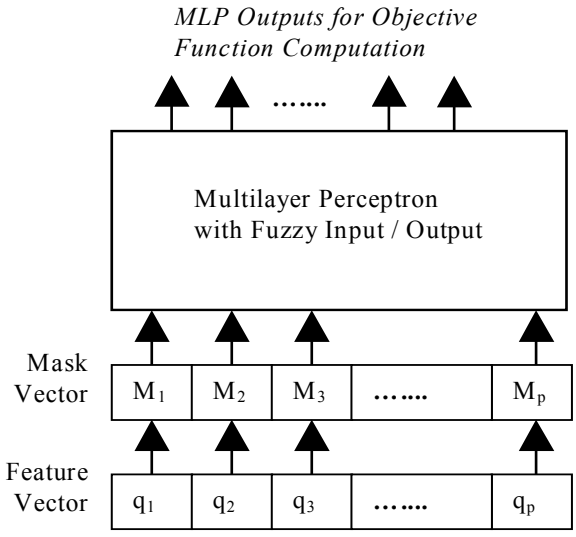

Fig. 1. Schematic diagram of the GA-based feature selection method.

References

1. Belue, L.M. and K.W.Bauer Jr. (1995). "Determining input features for multilayer perceptrons", *Neurocomputing* Vol. 7, 111-121.
2. Bhattacharjee, A. (1984). *Some aspects of mango (Mangifora Indica L) leaf growth features in varietal recognition*. Master's Thesis, Calcutta University, Calcutta, India.
3. De, R., N.R.Pal and S.K.Pal (1997). "Feature analysis: Neural network and fuzzy set theoretic approaches", *Pattern Recognition* Vol. 30, 1579-1590.
4. Devijver, P.A. and J.Kittler (1982). *Pattern recognition – A statistical approach*, Prentice Hall, London.
5. Goldberg, D.E. (1989). *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley. Reading, Mass.
6. Illingworth, J. and J. Kittler (1988). "A Survey of the Hough Transform", *Computer Vision, Graphics and Image Processing* Vol. 44, 87-116.
7. Johnson, R.A. and D.W.Wichern (1992). *Applied multivariate statistical analysis*, 3rd edition, Prentice Hall, Englewood Cliffs. NJ, USA.
8. Keller, J.M. and D.J.Hunt (1985). "Incorporating fuzzy membership functions into the perceptron algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 7, 693-699.
9. Klir, G.J. and B. Yuan (1995). *Fuzzy sets and fuzzy logic - Theory and applications*, Prentice Hall Inc., Englewood Cliffs, NJ, USA.
10. Lippmann, R.P. (1987). "An introduction to computing with neural nets", IEEE ASSP magazine, 4-22.
11. Pal, N.R. (1999). Soft computing for feature analysis, *Fuzzy Sets and Systems* Vol. 103, 201-221.
12. Pal, N.R. and K.K.Chintalapudi (1997). "A connectionist system for feature selection", *Neural, Parallel and Scientific Computation* Vol. 5, 359-382.
13. Pal, S.K. and B.Chakraborty (1986). Fuzzy set theoretic measures for automatic feature evaluation, *IEEE Transactions on Systems, Man and Cybernatics* Vol. 16, 754-760.
14. Priddy, K.L., S.K.Rogers, D.W.Ruck, G.L.Tarr and M.Kabrisky (1993). "Bayesian selection of important features for feedforward neural networks", *Neurocomputing* Vol. 5, 91-103
15. Ruck, D.W., S.K.Rogers and M.Kabrisky (1990). "Feature selection using a multilayer perceptron", *Journal of Neural Network Computing*, 40-48.
16. Rumelhart, D.E., G. E. Hinton and R. J. Williams(1986). "Learning internal representation by error propagation," in: D. E. Rumelhart and J. L. McClelland, Eds., *Parallel Distributed Processing : Explorations in the microstructure of cognition*, Vol. 1 : Foundations, Chapter 8, MIT Press.
17. Sural, S. and P. K. Das (1999). "Fuzzy Hough transform and an MLP with fuzzy input/output for character recognition," *Fuzzy Sets and Systems* Vol. 105, 489-497.