

Copyright © 2010 Institute of Electrical and electronics Engineers, Inc.

All Rights reserved.

Personal use of this material, including one hard copy reproduction, is permitted.

Permission to reprint, republish and/or distribute this material in whole or in part for any other purposes must be obtained from the IEEE.

For information on obtaining permission, send an e-mail message to stds-igr@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

Individual documents posted on this site may carry slightly different copyright restrictions.

For specific document information, check the copyright notice at the beginning of each document.

A Neural-Evolutionary Approach for Feature and Architecture Selection in Online Handwriting Recognition

Brijesh Verma and Moumita Ghosh

School of Information Technology, Griffith University-Gold Coast Campus, Australia

E-mail: b.verma@griffith.edu.au

Abstract

An automatic recognition of online handwritten text has been an on-going research problem for nearly four decades. It has been gaining more interest due to the increasing popularity of hand-held computers, digital notebooks and advanced cellular phones. However for these input modalities to be economical and user friendly the recognition rate should be very high for real time use. Also, the large number of writing styles and the variability between them makes the handwriting recognition problem a very challenging area for researchers. Many researchers have proposed a number of novel techniques for online handwriting recognition. However, an acceptable classification rate has not been achieved yet and there is a lack of techniques, which can find appropriate features, architecture and network parameters for online handwriting recognition. In this paper we propose a novel neuro-genetic technique to improve classification accuracy through the selection of appropriate features and network parameters for online handwriting recognition. The technique incorporates an evolutionary approach for finding the most significant features, network architecture and its parameters.

1. Introduction

In practical pattern recognition problems, a classification function learned through an inductive learning algorithm assigns a given input pattern to one of the existing classes of the systems. Usually the representation of each input pattern consists of features since they can distinguish one class of patterns from another in a more concise and meaningful way than offered by the raw representation. In many applications, it is not unusual to find problems involving hundreds of features. However, it has been observed that, beyond a certain point, the inclusion of additional features leads to a worse rather than better performance. Moreover, the choice of features to represent the patterns affects several aspects of pattern recognition problem such as accuracy, required learning time and necessary number of samples. Therefore the main goal of feature subset selection is to reduce the number of

features used in the classification while maintaining acceptable classification accuracy. The main aim of this research is to find the combination of most significant features and proper selection of network parameters such as weights, number of hidden units in online handwriting recognition.

Recent comparative studies of feature selection algorithms applied to machine learning include those carried out by Dash and Liu [1], Gordon and desJardins [2], Siedlecki and Sklansky [3], Jain and Zongker [4] and Kohavi and John [5].

In general, feature selection algorithms have two components: an evaluation function that scores candidate feature sets, and a search engine for finding those sets. Given a set of features the selection algorithm will examine a series of sets of features, and choose the one that maximises the evaluation function. When examining the current state of the art, one finds that feature selection algorithms fall broadly into two different frameworks, wrappers and filters, this categorisation being determined by the nature of the evaluation function.

2. Research methodology

The research methodology can broadly be classified into three modules, such as feature extraction, feature subset selection, and neural network based classifier.

2.1. Feature extraction

The module takes normalized sequence of captured coordinates $(x(t), y(t))$ as input and computes a sequence of feature along this trajectory. The features can be broadly divided into three categories, local feature, neighbourhood feature, and global feature. In local feature, the local properties of the points in the trajectory are considered. In neighbourhood feature, six consecutive points are considered to analyse the neighbourhood characteristic of the trajectory. In global features all the points in the trajectory are considered to analyse the overall properties. The following 12 features were investigated in this research.

2.1.1. Writing direction. Writing direction is a local

feature. It is the angle between the line joining two consecutive points with the horizontal axis. The local writing direction at a point $(x(t), y(t))$ is described using cosine & sine [6].

$$\Delta x(t) = x(t+1) - x(t-1) \quad (1)$$

$$\Delta y(t) = y(t+1) - y(t-1) \quad (2)$$

The angle involved in this construction is shown in Figure 1.

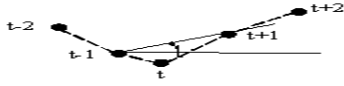


Figure 1 Writing direction

2.1.2. Curvature. Curvature [6] is a local feature. It is the angle between the line joining $(t-2)$ th point and t th point and the line joining t th point and $(t+2)$ th point. It is shown in Figure 2.

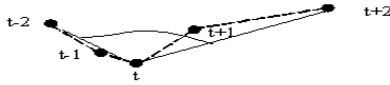


Figure 2 Curvature

2.1.3. Aspect. Aspect is a local feature. It characterizes the ratio of height to width of the boundary box containing the preceding and succeeding points of $(x(t), y(t))$. The aspect of the trajectory in the vicinity of a point $(x(t), y(t))$ is described by the following equation [6]:

$$A(t) = \frac{\Delta y(t) - \Delta x(t)}{\Delta y(t) + \Delta x(t)} \quad (5)$$

2.1.4. Stroke length. Stroke length is a local feature. It represents the length of the line joining two consecutive points. The length is normalized with respect to the total length of the body. The length is calculated using the Euclidian distance between the two points.

2.1.5. Slope. Slope is a local feature. The slope represents the angle between the line joining $(t-1)$ th point and t th point and the horizontal axis. The slope is described in Figure 3.

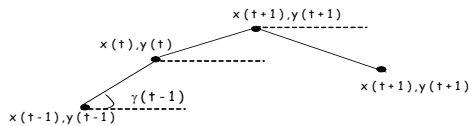


Figure 3 Slope

2.1.6. Pen up/ down. Pen up or pen down represents the feature whether the pen is in contact with the writing pad at t th point.

2.1.7. Curliness. Curliness is a neighborhood feature. Six consecutive neighbors are considered. The

neighbors in either direction are considered, i.e. three in forward direction and three in backward direction. Curliness $C(t)$ is the feature that describes the deviation from the straight line in the vicinity of $(x(t), y(t))$ [6]. It is based on the ratio of the length of the trajectory and the maximum side of the boundary box:

$$C(t) = \frac{\sum_i l_i}{\max(\Delta x, \Delta y)} \quad (7)$$

Where l_i denotes the length the line segment joining two consecutive point.

2.1.8. Linearity. Linearity is a neighborhood feature. The average square distance between every point in the vicinity of $(x(t), y(t))$ and the straight line joining the first and last position in the vicinity is called Linearity [6]. Linearity $LN(t)$ can be defined as follows:

$$LN(t) = \frac{1}{N} \sum_i d_i^2 \quad (8)$$

2.1.9. Angle of vicinity. The slope of the straight line joining the first and the last point in the vicinity of $(x(t), y(t))$ is described by the cosine of its angle.

2.1.10. Number of ascenders. Number of ascenders is a global feature. It denotes the number of point above the base line.

2.1.11. Number of descenders. Number of descenders is a global feature. It denotes the number of point below the base line.

2.1.12. Vertical position of start/end point. Vertical position is a global feature. It represents the vertical position of start and the end point with respect to the base line.

2.2. Feature subset selection

Feature selection algorithm can be classified into two categories based on whether or not the feature selection is performed independently of the learning algorithm used to construct the classifier. If feature selection is done independently of the learning algorithm, the technique is said to follow a filter approach. Otherwise it is said to follow the wrapper approach.

2.2.1. Genetic algorithm in feature selection. Genetic algorithm (GA) is a class of search methods deeply inspired by the natural process of evaluation. In each iteration of the algorithm (generation), a fixed number (population) of possible solutions (chromosomes) is generated by means of applying certain genetic operation in a stochastic process

guided by a fitness measure. The most important and commonly used genetic operators are recombination, crossover and mutation.

2.2.2. Chromosome representation. Canonical genetic representation is chosen for feature selection. In canonical GA, a chromosome is represented through a binary string. If a bit is a 1, means that the corresponding feature is selected. Otherwise the feature is omitted in that particular iteration. Mutation operator operates on a single string and changes a bit randomly. Crossover operates on two parent strings to produce two off-springs.

2.2.3. Selection mechanism. The selection mechanism is responsible for selecting the parent chromosome from the population and forming the mating pool. The selection mechanism follows the survival-of-the fittest mechanism in nature. It is expected that a fitter chromosome receives a higher chance of surviving on the subsequent evolution while the weaker chromosome will eventually die. The fitness values for all the chromosomes are normalized before sending to the selection function. The normalization is done to obtain fitness for all the features within certain range. The roulette wheel selection mechanism is used.

2.2.4. Fitness evaluation. The fitness evaluation determines the confidence level of the optimized solution. The aim of this research consists of optimizing two objectives: minimization of number of features and minimization of error rate of the classifier.

2.2.5. Filter approach. In the filter approach, the training phase and the evaluation phase work separately (Figure 4). The neural network is first trained with the data. Then the trained neural network is used as a classifier to calculate the fitness of the individual population. In the evaluation phase the population is initialised randomly. To calculate the fitness of individual population, the feature vector is multiplied by the individual population. If a particular feature is not selected, that place holds zero value. So the feature is multiplied by zero and neutralising its effect on fitness.

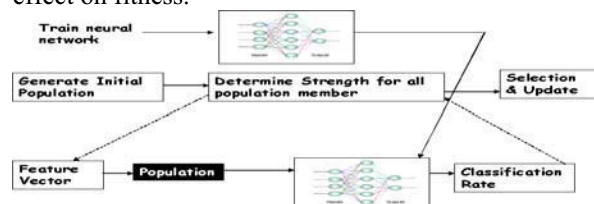


Figure 4 Filter approach architecture

2.2.6. Wrapper approach. In the wrapper approach, the training phase and the evaluation phase work together (Figure 5). In the evaluation phase the population is initialised randomly. For each member in the population, if the bit position holds a zero value the feature is assigned to zero and a new data set is created. With that dataset the neural network is trained. So for individual member in the population, there are individual neural network that has to be trained with the separate dataset. Then that trained neural network is used to calculate the fitness. The fitness is calculated in the same way as in filter approach. The stopping condition for training the neural network is to be equal for all the members in the population and it is taken as the classification error. The stopping criterion of the genetic algorithm is the number of generation. The neural network has to be trained for all the members in the population in each generation. This approach involves the computational overhead of evaluating a candidate feature subset by executing a selected learning algorithm on the dataset using each feature subset under consideration.

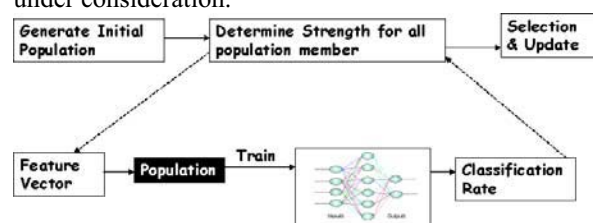


Figure 5 Wrapper approach architecture

2.2.7. Simultaneous search for feature and architecture selection. In this approach, input features for the ANN and its weights are found using a parallel simultaneous approach (Figure 7). The two levels of parallelisms are obtained through two different genetic approaches, one for its feature selection module and the other for finding the weights of the ANN. The first module is based on the canonical GA and the second module is based on the evolutionary algorithm. The two methods are connected based on the fitness for the input model. The connectivity is shown in Figure 6. The fitness for the canonical GA model depends on the fitness for the evolutionary approach, which finds the weight values for the network. The stopping of the evolutionary algorithm depends only on the limited number of generations, so that a limited time complexity can be achieved to find the fitness for the canonical GA. The stopping criterion for the weight selection algorithm is the classification error value and the stopping criterion for the feature selection algorithm is the maximum number of generation.

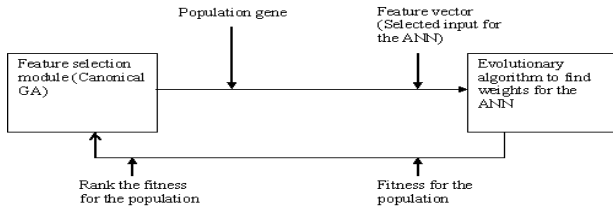


Figure 6 Connection Diagram for parallel-simultaneous approach

Real number encoding scheme is used for the weight selection module. A chromosome is represented through a real number the weights between input and hidden layer, and hidden and output layer. For mutation, a small random value between 0.1 and 0.2 is added to all the weights.

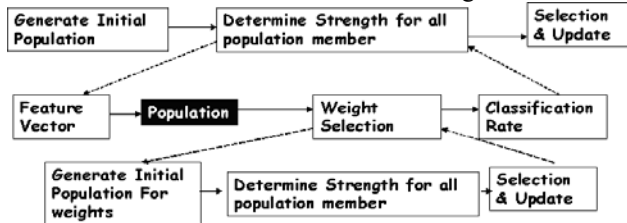


Figure 7 Parallel simultaneous approach architecture

3. Database description

In this paper, a subset of the UNIPEN dataset (lower case, upper case and digits) is used. The length of training dataset was 5000 and testing dataset was 500.

4. Experimental results

The proposed approach has been implemented in C++ and UNIX. A number of experiments were conducted on different datasets, however a detail description is included only for the UNIPEN lower case dataset. The results are listed in the following sections.

4.1. Results for filter approach

To check the time complexity and the classification error, we run the algorithms in two different steps: the number of hidden neurons is fixed and we increase the number of iteration to train the neural network and the number of generation is increased to test the results for the feature selection phase. Table 1 shows the result with trained neural network with five thousand iterations. The classification rate for the neural network was 96.45% on training data set and 86.23% on testing data set. The classification rate fluctuates in the selection phase with the selection of different features.

Table 1 Filter approach results

# population	# generation	Training	Testing
30	40	91.92	82.47
	50	92.94	83.42
50	40	93.57	83.30
	50	93.87	83.81

4.1.1. Performance analysis for filter approach.

The following figure (Figure 8) shows the behavior of feature selection phase. Figure 8 shows that the number of features selected in three cases was almost same. For the UNIPEN lower case dataset, the number of feature selected was 12, for uppercase dataset it was 12, and for the digit dataset it was 11.

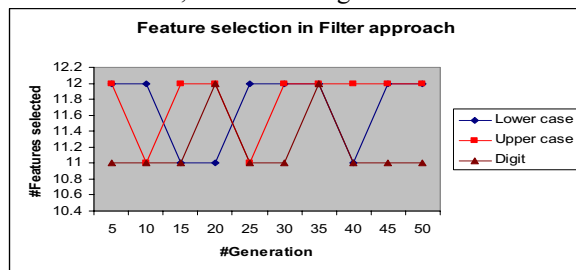


Figure 8 Feature selection in Filter approach

4.2. Results for wrapper approach

The results for wrapper approach are shown in Table 2. The classification rates are reported with varied number of populations and generations. The neural network is trained within the selection phase. BP algorithm was used to train the neural network. Only the best results are shown in the table to show the behaviour of the algorithm in different environment. The RMS error goal and the number of iterations were fixed for all chromosomes to train the network. To check the time complexity and classification error, we run the algorithms in two different steps: (1) the number of hidden neurons is fixed and we increase the number iteration to train the neural network, (2) the number of hidden neuron is increased adaptively in five different steps, and the number of generation is increased to test the results for the feature selection phase.

Table 2 Wrapper approach

# iteration	# hidden neurons	Training	Testing
500	30	89.85	78.35
	40	90.94	80.03
2000	30	96.94	85.60
	40	97.23	90.49

4.2.1. Performance analysis for wrapper approach.

The following figure (Figure 9) shows the behavior

of feature selection phase. Figure 9 shows that the number of feature selected in three cases has decreased almost steadily. For the UNIPEN lower case dataset, the number of features selected decreased from 12 to 9, for uppercase dataset, it was 12 to 9, and for the digit dataset it was 11 to 8.

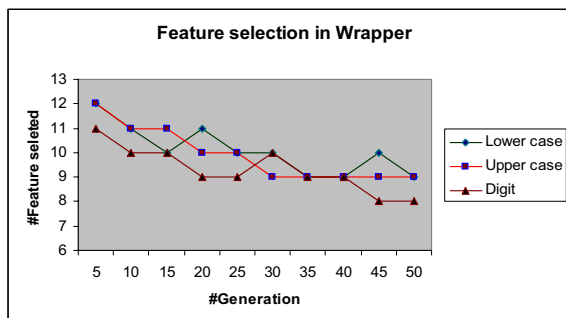


Figure 9 Feature selection in Wrapper approach

4.3. Results for simultaneous search approach

The results for coevolution approach are shown in Table 3. The results show the variation in classification rate on training and testing data set with the variation in different features. The classification rates are reported with a varied number of population and generations. The neural network parameters are chosen arbitrarily in the initial phase. Only the best results are shown in the table to show the behaviour of the algorithm in different environment. The number of generation and the population lengths were fixed for all chromosomes to train the network. To check the time complexity and classification error, we executed the algorithm using steps same as described in section 4.2.

Table 3 Simultaneous search approach

# iteration	# hidden neurons	Training	Testing
500	30	89.85	78.35
	40	90.94	80.03
2000	30	95.94	89.60
	40	95.23	89.49

4.3.1. Performance analysis for co-evolution approach. The following figure (Figure 10) shows the behavior of feature selection phase. As may be seen from Figure 10, the number of feature selected in three cases has decreased almost steadily. For the UNIPEN lower case dataset, the number of feature selected decreased from 12 to 9, for uppercase dataset it was 12 to 9, and for the digit dataset it was 11 to 8.

5. Conclusions

In this paper a novel methodology was proposed that uses combination of GA based automatic feature selection and wrapper based neural network classifier. The proposed method was applied for real-time online handwriting data set, and was compared with other existing methodologies. Three approaches were developed in this research. Among those the wrapper approach with back propagation algorithm has produced the best results (recognition rate - 98.7%) for the digit dataset. For the UNIPEN lowercase and upper case datasets, the highest recognition rates achieved were 91% and 93% respectively.

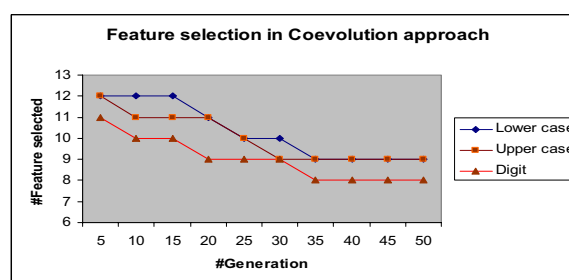


Figure 10 Feature selection in Coevolution approach

6. References

- [1] M. Dash and H. Liu, "Feature selection for classification", *Intelligent Data Analysis*, 1997, vol.1, no. 3.
- [2] D.F. Gordon and M. desJardins, "Evaluation and selection of biases in machine learning", *Machine learning*, 1995, vol. 20, pp. 1-17.
- [3] W. Siedlecki and J. Sklansky, "On automatic feature selection", *International Journal of Pattern Recognition and Artificial Intelligence*, 1998, vol. 2, no. 2, pp. 197 - 220.
- [4] A.K. Jain and D. Zongker, "Feature selection: evaluation, application and small sample performance", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1997, vol. 19, no. 2, pp. 153 - 158.
- [5] R. Kohanvi and G.H. John, "Wrappers for feature subset selection", *Artificial Intelligence*, 1997, vol. 97, pp. 273 - 324.
- [6] S. Jaegar, S. Manke, J. Reichert and A. Waibel, "Online Handwriting Recognition: The NPen++ Recognizer", *International Journal of Analysis and Recognition*, 2001, vol. 3, no. 3, pp. 169 - 180.
- [7] P. Hajela and C.Y. Lin., "Genetic search strategies in multicriterion optimal design", *Structural Optimization*, 1992, vol. 4, pp. 99-107.
- [8] L.S. Oliveira, N. Benahmed, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Feature Subset Selection Using Genetic Algorithms for Handwritten Digit Recognition", *Proceedings of the 14th Brazilian Symposium on Computer Graphics and Image Processing*, IEEE Computer Society, Florianópolis-Brazil, 2001, pp. 362-369.