Improving Performance of Nearest Neighborhood Classifier Using Genetic Programming

Abdul Majid, Asifullah Khan and Anwar M. Mirza
Faculty of Computer Science & Engineering, GIK Institute of Engineering Science & Technology, Topi-23460, Swabi, PAKISTAN
{ majid, akhan & mirza }@giki.edu.pk

Abstract

Nearest neighborhood classifier (kNN) is most widely used in pattern recognition applications. Depending on the selection of voting methodology, the problem of outliers has been encountered in this classifier. Therefore, selection and optimization of the voting methodology is very important. In this work, we have used Genetic Programming (GP) to improve the performance of nearest neighbor classifier. Instead of using predefined k nearest neighbors, the number of men and women in the first two quartiles in Euclidean space are used for voting. GP is, then, used to evolve an optimal class mapping function that effectively reduces the outliers. The performance of modified nearest neighborhood (ModNN) classifier is then compared with the conventional kNN for gender classification problem. Receiver Operating Characteristics curve and its Area Under the Convex Hull (AUCH) are used as the performance measures. Considering the first three and first five eigen features respectively, ModNN achieves AUCH equal to 0.985 and 0.992 as compared to 0.9693 and 0.9795 of conventional kNN respectively.

1. Introduction

Classification is an important part of machine learning that has attracted much of the research endeavors. Various classification approaches, such as, k-means, neural networks, decision trees, and nearest neighborhood have been developed and applied in many areas. K nearest neighbor classifier (kNN) is simple and attractive in many classification problems [14].

We have chosen gender classification problem based on frontal facial images. This data is very complex and highly nonlinear. Moghaddam et al. [25] have chosen support vectors based classifiers to enhance the classification accuracy rate. They have investigated that binary classification of this data with high accuracy is very difficult. Pre-determination of k nearest neighbors for the kNN classifier is not easy, for such a non-linear data. Consequently, the accuracy of kNN classifier greatly suffers due to inappropriate choice of k. Usually iterative methods are used to select a suitable k for a particular feature subset [1]. Still the effects of outliers are not reduced considerably. One way to reduce the outliers in kNN are to modify the basic assumption of locally constant conditional probabilities. This assumption can introduce bias in high dimensions due to the curse of dimensionality. Domeniconi, et al. [2] have proposed an adaptive nearest neighbor (NN) classification method to reduce the error introduced due to this bias. Trevor et al. [3] have proposed a modified NN classifier by shrinking neighborhoods in directions orthogonal to the local decision boundaries and stretching them parallel to the boundaries in order to enhance linearity in data. Scott et al. [4] have attached weights to the instances, in order to further modify the structure of feature space. Various other attempts have also been made to use distance measures other than the Euclidean distance for NN classifier [5].

In order to improve the performance of different conventional classifiers through Genetic Programming, we have divided our work in two stages. In the first stage, we have improved the performance of the individual classifiers, such as, Linear Discriminant Analysis and Mahalanobis Distance Based classifiers [20]. While, in the second stage, we have evolved an optimal homogenous and heterogeneous combination of NN classifiers having better performance than the individual ones [24]. Our current work is related to the first stage, in which improvement in kNN classifier is made through GP. It is shown that while keeping the basic assumption of locally constant conditional probabilities, outliers can be reduced by genetically evolved optimal mapping function. This function performs mapping from feature space to the class space based on the NN distribution of training samples in Euclidean space. This improvement in kNN classifier is made by the reduction of outliers through an optimization technique. Receiver Operating Characteristics (ROC) Curves and Area Under the Convex Hull (AUCH) are important tools to analyze the performance of classifiers at different operating conditions. Consequently, AUCH is used to score and rank different classifiers in all generations of GP simulation and achieves an optimal/near-optimal classifier.

In section 2, we have briefly discussed the working of classification system while section 3 describes GP. Our methodology of improving the NN classifier is discussed in Section 4. Section 5 elaborates the implementation details, while results and conclusion are discussed in section 6 and 7 respectively.

2. Classification System

Classification systems usually consist of three main stages: preprocessing, feature extraction and classification stage [14]. Preprocessing is the stage of simplifying subsequent operations without losing relevant information. The purpose of feature extractor is, to reduce the data by measuring certain properties that are useful for classification. Classifiers use these features to assign an object to a category. Usually, feature extraction is carried out using statistical methods e.g. Principal Component Analysis (PCA), Independent Component Analysis (ICA) and transformation techniques. Previously, we have investigated Discrete Cosine Transformation (DCT) for feature extraction [21]. Now, we are using PCA for feature extraction of facial images. Classifiers are employed to classify the frontal view images only. Here, we are not using any preprocessing stage, the raw pixel intensity of frontal view images are given directly to feature extraction stage, in order to compare classifiers performance in an adverse environment. Generally, preprocessing stage (isolating face from background) makes the classification system computationally expensive.

2.1 Feature Extraction

Stanford University medical students database [19] is used as input to evaluate the performance of classification algorithms. These images comprise of the frontal-view of students. This database consists of 200 male and 200 female, gray scale images of size 128x128 pixels.

PCA technique is widely used in statistics for dimension reduction, data compression, and multivariate data projection. It projects a high dimensional data to a lower dimensional subspace by finding the directions where the variance is maximal [14]. We have used Sirovich and Kirby method to compute the principal components for all training and test images [6],[7],[8]. In this method, data set is projected on N eigenbasis (eigenfaces) to obtain first N principal components. PCA has the important property of highest energy concentration than any other projection. It has maximum power of expression of data by capturing the highest amount of variation within a data [14].

Scaled down jackknife [8] scheme is employed to utilize the image database and check the performance of the classifiers. Training to test ratio of 1:3 is selected. First randomly choosing 50 males and 50 females to use as our training set, rest of images are left for testing. This process is repeated several times in order to increase their statistical significance.

2.2 kNN Classifier

Let $\mathbf{x} = \left(x_1, \ldots, x_n\right)^t \in \Re^n$ be a feature vector, represented by n dimensional test sample. In order to build classifier-mapping functions that assign $\mathbf{x}$ to the correct class from a given set of training samples. Nearest neighbor classification is a simple and appealing approach for this binary classification problem. One has to find a set of k nearest neighbors in the training set for a test sample $\mathbf{x_0}$ and then classify $\mathbf{x_0}$ according to the most common class among the k neighbors. kNN classification method uses different neighborhoods for each individual query $\mathbf{x_0}$ so that all points in the neighborhood are close to the query [13]. A crucial issue in NN classifier is the choice of neighbor size k. Large choice of k may improve classification accuracy, but at the same time can increase bias in the approximation of conditional probability function [16]. For comparison purpose, we have used several choices of k from 5-11.

2.3 Performance Evaluation Criteria

Performance of a particular classifier is assessed in terms of true positive rate (TPR) and false positive rates (FPR). TPR is obtained by dividing the number of correct positive cases by the total number of positive cases. Similarly FPR is, the number of incorrect positive cases divided by the total number of negative cases. Although, a single figure of merits is useful when comparing a classifier under different conditions but one of the greatest assets of testing is lost because they don't characterize the classifier over its entire operating range [9]. The selection of operating threshold is application-specific, depending on the maximum acceptance of false and true positives. When a graph is plotted between TPR and FPR for different threshold values, the resulting curve is called ROC curve. ROC curve summarizes how well a classifier has performed under different operating conditions for a particular problem.

For a classifier, one needs TPR to be as close to one as possible, while FPR to be the lowest. Since both axes are normalized to [0,1] range, therefore, area under the ideal ROC curve will be one. A classifier is an optimal one if area under its ROC curve is near to one. Scott et al. [10] have shown that Maximum Realizable ROC (MRROC) is the convex hull of the classifier's ROC. Consequently, AUCH of ROC curve is taken as a measure of the performance of a classifier [11], [22].

In order to calculate the output of conventional kNN classifier against different thresholds and to measure the confidence in decision, the procedure adopted is, the discrete output of classifier is scaled to [0,1] range. The number of examples of men in the nearest neighbor is subtracted from that of women. The difference is normalized and is considered as the output of kNN. The increasing threshold T [0,1] is applied to this output to obtain ROC curve. Therefore, If the output is greater than the threshold, the test sample is considered to be a woman otherwise it is considered a man.

3. Genetic Programming

Genetic programming is a type of evolutionary algorithms that are based on the mechanism of natural selection and natural genetics. These evolutionary algorithms try to mimic Darwin's principals of natural selection to evolve an optimal solution out of solution space. In context of classification, they come under the category of stochastic methods, where randomness plays a crucial role in search and learning [14].

First, we have to represent a candidate using a data structure in the form of a GP tree (figure 4). We create a random population of classifiers to represent a solution space. Next, score each classifier on a classification task, such as measuring accuracy on a set of labeled examples. This process is described as finding the fitness of each classifier using the fitness function. Next the survival of fittest is implemented by retaining the best classifier. The rest are deleted and replaced by the offspring's of these best classifiers. Complete detail of GP simulation block diagram is shown in figure 1. The best retained classifiers and the offspring make a new generation. Some offspring may have high score than their parents. The whole process is repeated for the subsequent generations. Each new generation has a slightly higher score than the previous one. The process is stopped when a single classifier in a generation gets a score that exceeds a desired value. In this way, the solution space is refined and converged to the optimal/near optimal solution [17], [18].

Operators Used to Create Offspring: To produce a new generation, mainly three operators: replication, mutation and crossover are used in genetic programming. Replication is the copying of an individual into the next generation. In mutation, a small part of an individual's genome is changed which brings diversity in the solution space and helps to avoid trapping in local minima/maxima. On the other hand, crossover creates an offspring by exchanging genetic material, usually between two individuals. In fact, crossover tries to mimic recombination and sexual reproduction. Crossover mainly helps converging to an optimal solution.

Population Fitness Evaluation: GP evaluate the population of individual candidates using the fitness function as scoring criteria. The user, according to the application, defines this function. The better an individual is performing, the better its survival is. So higher are the chances of producing children for the next generation.

Candidate Solution Representation: In GP simulation, an individual candidate solution is represented through tree structure. The terminals in the trees act like inputs to a program or like independent variables in a function. Terminals may be constants or variables. Non-terminals nodes are functions that process given values. Functions are usually composed of statements, operators. These terminals and functions make GP tree. The initial population of a GP simulation is formed by randomly generating trees. Functions and terminals in GP simulation should be powerful enough to represent an individual solution to the problem. Most trivial functions being used are *PLUS, MINUS, DIVISION, AND, TIMES* and *EXP* etc.

4. Methodology Adopted

In order to improve the performance of NN classifier, we consider the reduction of outliers as an

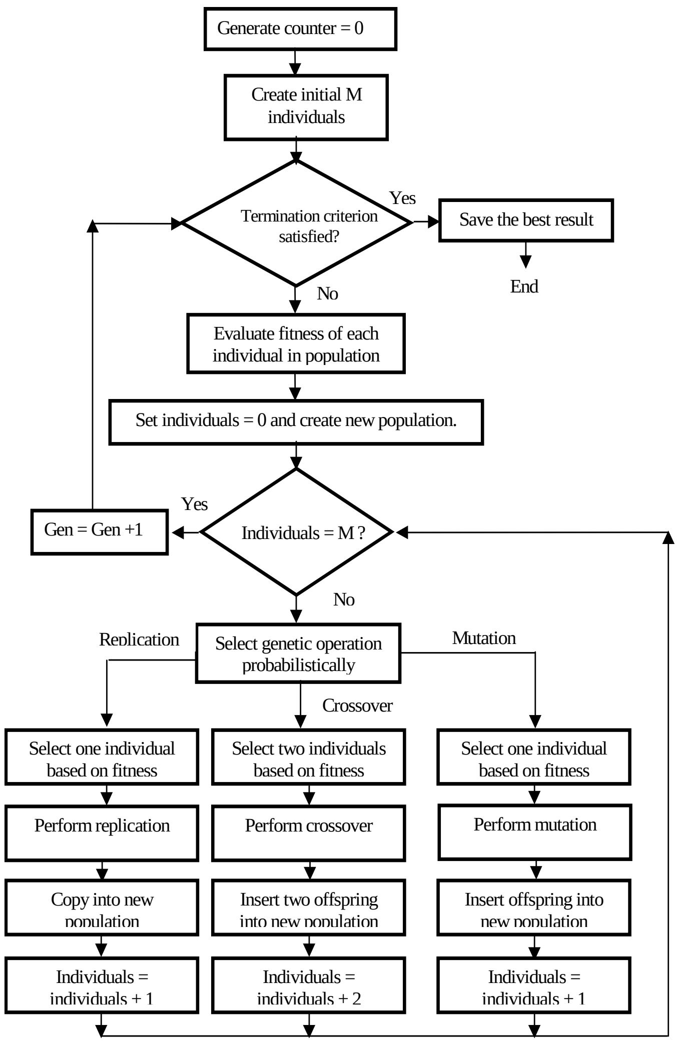

Figure 1 GP simulation block diagram

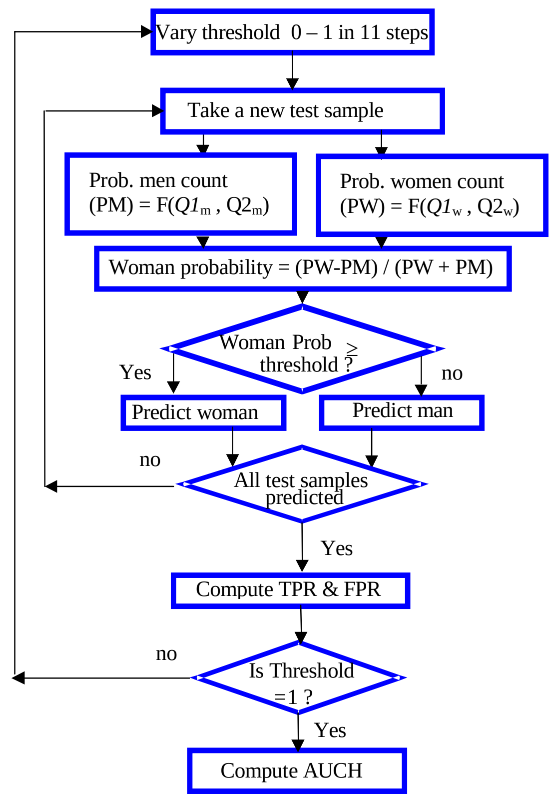

Figure 3 Fitness evaluation module

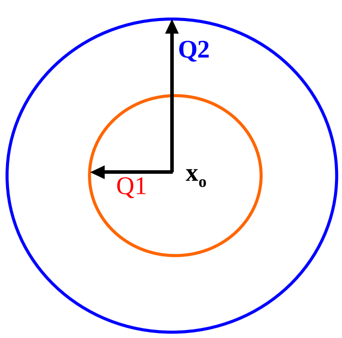

Figure 2 NN classifier diagram

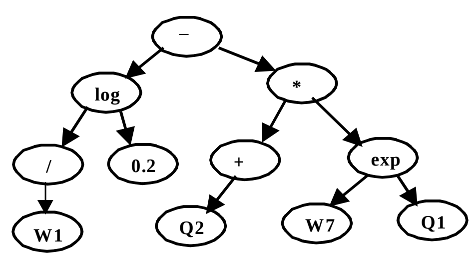

Figure 4 A GP tree representation

optimization problem. For this purpose, voting methodology is modified and then GP is used to evolve such mapping functions that effectively reduce the outliers by learning the distribution of training samples.

4.1 Voting Methodology

In this experiment, neighbors in the first two quartiles are taken for the selection of neighbors in Euclidean space. Let *Q1* and *Q2* be the first two quartile distances of two annular strips as shown in figure 2. Classifier decision is based on the maximum voting which in tern depends on the counts of each class in these strips. The counts of man in quartile *Q1* is denoted by $Q1_m$ and that in *Q2* by $Q2_m$. Similarly the woman counts in the two quartiles are $Q1_w$ and $Q2_w$. The counts of man and woman in each strip may have different weightage. More weightage is automatically given to the smaller strip, depicting the higher effect of nearby samples.

4.2 Optimal Class Mapping Function

Providing the man/woman counts ($Q1_m$, $Q1_m$, $Q1_w$,$Q2_w$) in each strip and weights *W0 - W10* between [0,1], GP evolves such combination of counts and weights that reduces the effect of outliers. The fitness evaluation procedure of evolved NN classifier during GP simulation is shown in figure 3. One candidate is picked from GP population and the performance of each individual mapping function is tested on the whole threshold range of [0,1]. The individual has to predict for all the test samples in order to calculate TPR and FPR corresponding to different thresholds to plot ROC curve. AUCH for that individual is found out. A mapping function's prediction about a test sample is carried out by providing the values of $Q1_m$ and $Q2_m$ as inputs to mapping function in order to compute the probable man counts (PM). Similarly, probable woman counts (PW) are computed for the same mapping function. Woman probability is computed by dividing the difference of PW and PM with its sum. The woman probability, then, is compared with the threshold and if it is greater than or equal to the threshold, the test sample is predicted to be a woman otherwise a man.

5. Implementation Details

We have used GPLAB software [15] for simulation studies. All the necessary parameter settings to tune GP are shown in table 1. This table shows the GP language adjusted for a specific problem of gender classification, while GPLAB automatically use other parameter values. In order to adjust the GP simulation through GP parameters, one needs to define suitable functions, terminals and fitness criteria.

GP Configuration: Four binary floating arithmetic operators (+, -, *, and protected division), *LT, GT, LOG, EXP* and *ABS* are used as functions. The counts of each class in these quartile neighborhoods are provided as variable terminals. While 11 constant weights *W0 - W10* are used as constant terminals.

GP Fitness Criteria: A unique combination of class counts *Q1, Q2* in the two strips and weights *W0-W10* represent a possible mapping function in a GP population. Its fitness is obtained through AUCH of ROC curve. The greater the AUCH is, the better the individual's performance is.

Table 1 GP Parameters Selection

Objective:	To evolve an NN classifier with maximum AUCH
Function Set:	+, -, *, protected division, *GT, LT, EXP* and *LOG*
Terminal Set:	Constants terminals: *W0 - W10* Variable terminals: $Q1_m$,$Q2_m$, $Q1_w$, $Q2_w$
Fitness :	AUCH of 11 ROC points.
Selection:	Generational
Wrapper:	Positive if >= 0, else Negative.
Population Size:	300
Initial Tree Depth	6
Initial population:	Ramped half and half
Operator prob. type	Variable
Sampling	Tournament
Expected no. of offspring	rank85
Survival mechanism	Keep best
Real max level	14
Termination:	Generation 55

6. Results and Discussion

Figure 5 shows the accuracy versus complexity of the best-evolved mapping function up to 55 generations. It is observed that as generations pass by, there is improvement in fitness of the best individual. This improvement is achieved at the cost of complexity in a GP tree. That is, with the increase in fitness of the best individual, its genome's total number of nodes and average tree depth also increases.

Figure 6 and 7 shows the Maximum Realizable ROC curves of the ModNN and kNN classifiers for 1-3 and 1-5 feature subset respectively. It is observed that the ModNN performs better than the conventional kNN in terms of AUCH under all conditions. In

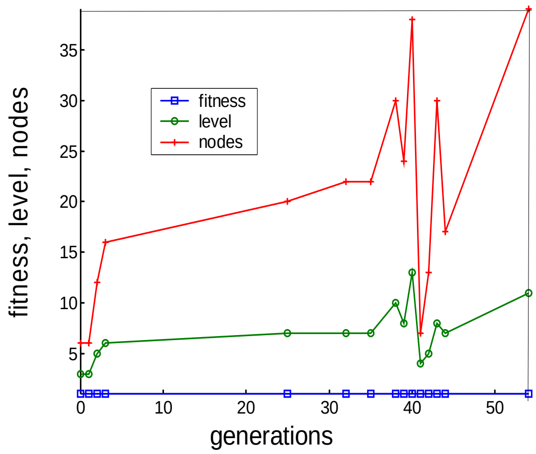

Figure 5 Accuracy vs. complexity graph

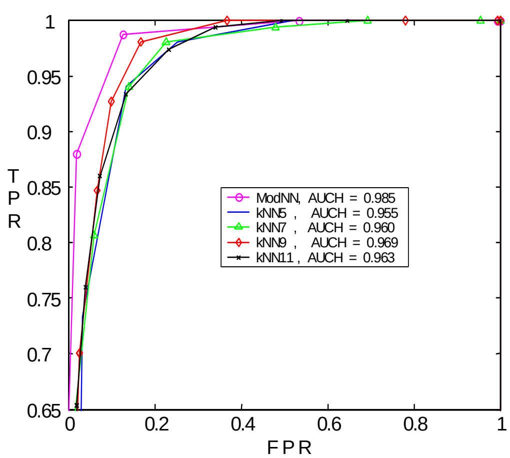

Figure 6 MRROC curves for 1-3 features

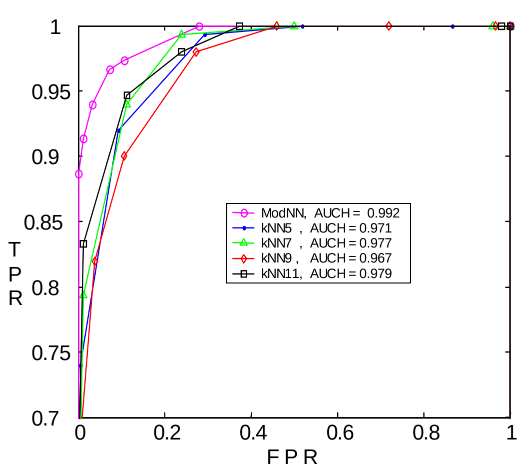

Figure 7 MRROC curves for 1-5 features

MRROC curves, the points having low FPR and high TPR (i.e. points in upper left corner) are the most desirable points [12]. In figure 7, it can be observed that ModNN achieves a high TPR ≈ 0.9 even at FPR ≈ 0. Therefore, ModNN has another advantage over kNN classifier. This type of situation is highly desirable in those applications, where the cost of FPR is highly important. Such as, in the classification of malignant and benign tissues of a very weak patient before cancer therapy, we cannot afford high FPR for a weak patient [23].

Figure 8 and figure 9 show the performance of NN classifiers in term of bar charts for 1-3 and 1-5 feature subset respectively. It is observed that in both feature subsets ModNN performs better than conventional kNN classifiers with different choice of k (k = 5, 7, 9, and 11). From figure 8 and 9, it is clear that there is no consistency in the performance of conventional NN classifiers under different feature subsets. For 1-3 feature subset, k = 9 is an appropriate choice. However, there is degradation in the performance of kNN9 for 1-5 feature subset. Now kNN11 is performing better. That is how, a suitable k is chosen heuristically. On the other hand, there is consistency in

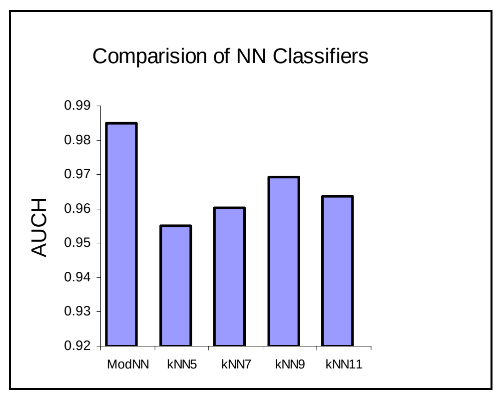

Figure 8 AUCH for 1-3 features

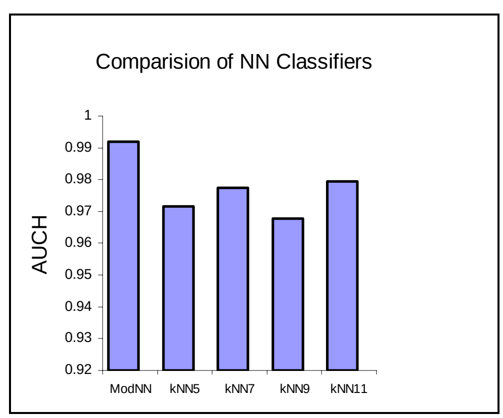

Figure 9 AUCH for 1-5 features

the performance of ModNN classifiers under different feature subsets. AUCH performance of ModNN classifier increases, with the increase of feature subset. That is, with the availability of more information, we are more certain in classification decision.

Thus, ModNN classifier performs better than conventional kNN classifiers and it is independent of the appropriate choice of k. The problem of appropriate choice of k is solved by changing the voting methodology and optimization through GP.

A typical expression, in prefix form, of the best mapping function achieved is:

=/((*(+(*(Q2,W4),+(/(+(W7,*(/(+(W7,W4),+(Q1,W4)),*(*(Q2,W4),W7))),*(Q1,W4)),W7)),W10),W7),/(+(W4,*(Q2,W4)),*(Q1,W2))).*

This function depends on counts *Q1*, *Q2* in each quartile and different weight constants *W1-W7*.

7. Conclusions

The performance of NN classifier is enhanced by changing the voting methodology and evolving an optimized classifier through genetic programming for gender classification problem. This method of improving the performance of nearest neighborhood classifier is not specific to this application data and NN classifier [20]. Our method in general can be applied to any classification data and a classifier. GP can be used to tune to that data by evolving best optimal class mapping function in order to improve classification accuracy. This method is computationally expensive in the evolution stage. However once an optimal mapping function is evolved, then, its use is not expensive in the test stage.

Sponsorship:

This work is sponsored by the Ministry of Science and Technology and Higher Education Commission, Government of Pakistan under the scholarship grants No. 17-6 (0246) /Sch / 2001 and No. 17-6 (176) /Sch / 2001.

References

[1] D. Wettschereck and T. Diettericch, "Locally adaptive nearest neighbor algorithms", *Advances in Neural Information Processing System, Morgan Kaufmann* Publishers, San Mateo, CA, 1994.

[2] C. Domeniconi, J. Peng, and D. Gunopulos, "Locally adaptive metric nearest neighbor classification", IEEE transaction on *Pattern Analysis and Machine Intelligence*, 2002.

[3] T. Hastie, R. Tibshirani, "Discriment adaptive nearest neighbor classification", *Advance in Neural Information Processing Systems*, 1994.

[4] S. Cost and S. Salzberg, "A weighted nearest neighbor algorithm for learning with symbolic features", *Machine Learning*, 1993.

[5] R. Short and K. Fukunanga, "Optimal distance measure for nearest neighbor classification", IEEE transaction on *Information Theory*, 27:622-627, 1981.

[6] M. Burton, V. Bruce and N. Dench, "What is difference between men and women? Evidence from facial Measurement," *Perception*, 22,153-176, 1993.

[7] H. Abdi, D. Valentin, B. Eldmen and J. A O'Toole (1995), "More about difference between men and women: evidence from linear neural net work and principal component approach," *Perception*, 24. 539-562, 1995.

[8] M. Kirby and L. Sirvich, "Application of Karhunen Louve procedure for the application of human" IEEE, 12:103-108, 1990.

[9] C. Lincoln, "Pose independent face recognition", PhD thesis, Department of Electronic System Engineering, University of Essex. Dec. 2002.

[10] M. J. J. Scott, M. Niranjan and R. W. Prager, "Realizable classifiers: Improving operating performance on variable cost problems," *Ninth British Machine Vision* Conference, Volume 1, 304-315, UK, September 1998.

[11] W. B. Langdon and B. F Buxton, "Genetic programming for combining classifiers," In GECCO, 2001, *Morgan Kaufmann.*

[12] Z. Sun, X. Yuan, G. Bebis, and S. Louis, "Neural-network-based gender classification using genetic search for eigen-feature selection," *IEEE International Joint Conference on Neural Networks*, May, 2002.

[13] Y. Wu, K. Ianakiev, and V. Govindaraju, "Improvements in k-nearest neighbor classifications". In W. Kropatsch and N. Murshed, editors, International Conference on *Advance in pattern recognition*, volume 2, pages 222–229. Springer Verlag, 2001.

[14] R. O. Duda, P. E. Hart, and D. G. Stork. "*Pattern Classificati*on", John Wiley & Son s, Inc., New York, second edition, 2001.

[15] http://gplab.sourceforge.net

[16] L. Devroye, L. Gyorfi and G. Lugosi, "*A problistic theory of pattern recognition*", Springer Verlog Inc., New York, 1996.

[17] D. Goldberg, "*Genetic algorithm in search, optimization, and machine learning*," in Reading. Reading, MA: Addison-Wesley, 1989.

[18] W. Banzhaf, P. Nordin, R. E. Keller and F. D. Francone, "*Genetic Programming An Introduction: On the Automatic Evolution of Computer Programs and Its Applications*", Morgan Kaufmann Publishers, Inc. San Francisco, California, 1998.

[19] http://white.stanford.edu/~diclaro/ee368a/code/male .zip

[20] A. Khan, A. Majid and A. M. Mirza "Combination and Optimization of Classifiers in Gender Classification Using Genetic Programming", *International Journal of Knowledge-Based Intelligent Engineering Systems*, vol. 8, 2004, (to appear).

[21] A. Majid, A. Khan, Anwar M. Mirza, "Gender Classification using Cosine Discrete Transformation: A Comparison of Different Classifiers", *Proceedings of the International Multi-topic (INMIC 2003), IEEE Conference*, Dec. 2003.

[22] A. Khan, S. A. Husain, Anwar M. Mirza and Asmatullah," Optimizing ROC Curves Using Genetic Programming", *Proceedings of the TechCom conference*, Karachi, 2003.

[23] A. Khan, "Optimizing ROC Curves Using Genetic Programming", Ms. thesis, *GIK institute of Engineering Science and Technology*, 2003.

[24] A. Majid, A. Khan and Anwar M. Mirza "Optimization and Combination of Nearest Neighborhood Classifiers Using Genetic Programming", submitted in (INMIC 2004), IEEE Conference, Dec. 2004.

[25] B. Moghaddam and M. H. Yang, "Learning Gender with support faces", IEEE Transaction on *Pattern Analysis and Machine Learning*, vol. 24, No. 5, May 2002.