# COMBINATION AND OPTIMIZATION OF CLASSIFIERS IN GENDER CLASSIFICATION USING GENETIC PROGRAMMING

ASIFULLAH KHAN, ABDUL MAJID and ANWAR M. MIRZA

*Faculty of Computer Science & Engineering, GIK Institute,*

*Ghulam Ishaq Khan (GIK) Institute of Engineering Science & Technology,*

*Topi-23460, Swabi, PAKISTAN*

*{akhan, majid, mirza}@giki.edu.pk*

## Abstract

In this paper, we have investigated the problem of gender classification using frontal facial images. Four different classifiers, namely K-means, k-nearest neighbors, Linear Discriminant Analysis and Mahalanobis Distance Based classifiers are compared. Receiver operating characteristics (ROC) curve along with the area under the convex hull (AUCH) have been utilized as the performance measures of the classifiers at different feature subsets. To measure the overall performance of a classifier with single scalar value, the new scheme of finding the area under the convex hull of AUCH of ROC curves (AUCH of AUCHS) is proposed. It has been observed that, when the number of macro features is increased beyond 5, the AUCH saturates and even decreases for some classifiers, illustrating the curse of dimensionality. We then used genetic programming to combine classifiers and thus evolved an optimum combined classifier (OCC), producing better performance than the individual classifiers. We found that using only two features, the OCC has comparable performance to that of original classifier using 20 macro features. It produces true positive rate values as high as 0.94 corresponding to false positive rate as low as 0.15 for 1: 3 train to testing ratio. We also

observed that heterogeneous combination of classifiers is more promising than the homogenous combination.

***Keywords:*** Gender Classification; Principal Component Analysis; Eigenface; Jackknife Technique; Receiver Operating Characteristics Curve; Area under the Convex Hull; Genetic Programming.

## 1. Introduction

Classification is a mapping function from feature space to class labels. It is an important component of intelligent systems with wide range of applications. Gender classification is a 2-class problem in which one has to predict an image as that of a man or woman. It is an easy task for humans, but a challenging task in computer vision. An improvement in gender classification bolsters the performance of many related downstream applications like face recognition.

Generally two types of gender classification approaches are employed in practice: one in which geometrical features are used is called geometry-based approach and the other which does not use geometrical features, but perform classification using training images is called appearance based approach. We have used the appearance-based approach in conjunction with the Principal Component Analysis (PCA) for feature extraction. We neither isolate faces from the background and nor we use image normalization techniques as used by Moghaddam et., al. [12] in the image preprocessing stage. We simply give the image as it is to the PCA feature extraction stage in order to compare the performance of different classifiers under these adverse conditions.

Reasonable work has been done previously in gender classification [2-4, 8], but search for an improved gender classification system is still going on. We try to address this issue of improving classification accuracy by automatically finding optimal combination of different classifiers using Genetic Programming (GP). Previously Langdon et., al. [14] have used this

idea in data mining . We are using this idea in gender classification and have shown that improved results could be obtained as compared to Langdon et., al.[14], if added constraint is imposed on the fitness criteria of particular combination of classifiers using GP.

The second problem that we address in this paper is that in gender classification studies, classifiers are mostly not compared on the whole threshold range. Abdi et., al. [2-4] and Moghaddam et., al. [12] have compared different classifiers for gender classification, but they have not characterized them on the whole operating range. In this work we not only compare four different classifiers on varying threshold, but also study their behavior at different feature subsets. We also purpose an overall performance measure for a classifier, when both the threshold and feature subset is varied. Two different *train to testing ratios* have been used to reckon the general behavior of the classifiers. We argue that 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.

The paper is organized as follows: In section 2, we briefly describe classification systems followed by a discussion of proposed AUCHS Curve in section 3. Next in section 4 we describe the proposed scheme for combining classifiers using GP. In section 5 implementation details of the proposed schemes is discussed. Results and discussion are presented in section 6. Finally we finish with conclusions in section 7.

**2. Classification Systems**

Classification systems are largely used to obtain useful information from large data. They are an important component of intelligent systems and have wide range of applications. Classification systems usually consist of two main stages: a Preprocessing stage and a Classification stage. Feature extraction is a preprocessing stage that can avoid the curse of dimensionality or improve the generalization ability of classifiers. While in the classification

stage, classification algorithms are applied on the extracted features to perform mapping from feature space to class space.

### 2.1 *Feature extraction*

Faces are represented as features to a classification algorithm. But the major difficulty with representing faces as a set of features is that it assumes some priori knowledge about what are the features and what are the relationships between them that are essential for the task of gender classification. Burton et., al. [8] showed the difficulty in finding a set of the features useful in discriminating accurately between male and female. They showed that no simple set of features could predict the gender of faces. However, Abide et., al. [3] showed that comparable gender categorization performance could be obtained using a posteriori features automatically derived from statistical structure of a set of learned faces. These features are the eigen-vectors or principal component of the pixel cross-product matrix of a set of faces. They can be obtained directly [7,9] or via a linear-autoassociator [4].

Since the seminal work of Abide et., al. [2-4], in case of appearance based approach, mostly PCA and Independent Component Analysis (ICA) are used for feature extraction. PCA is a well-known method for feature extraction, 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. In ICA however, the directions in the input vector space where the signal components are independent random variables or at least as independent as possible are identified. ICA produces basis vectors that are statistically independent (not just linearly decorrelated, as is the case in PCA).

To reduce the dimensionality of the face images, principal components are extracted through Karhunen Loeve Transformation (KLT) [7, 9]. The KLT is used because it has the important property that the projection of the data set on the first $N$ principal components has the highest energy concentration than any other $N$ components projection. Therefore, it captures the

highest amount of variation in a data set, more than any other linear transform for a fixed number of components.

### 2.2 *Classifiers*

The four basic classifiers compared at gender classification problem are: K-means, k-nearest neighbor (kNN), Linear discriminant analysis (LDA) and Mahalanobis distance based (MDB) classifier. Some of the classifiers are modified to output a range of values, rather than just giving 0/1 values (in order to evaluate their confidence in decision). In all cases the output of each classifier is scaled to 0-1 range and the selected threshold T ($0 \leq T \leq 1$) is then applied to this output. The combined classifiers are then obtained by combining the best two classifiers using GP.

K-means algorithm tries to exploit natural separation of the data for classification. It does not use information regarding the gender of each person to perform the classification. First *K* random data points are supposed as center points, where *K* is the number of classes that one would like to find. In our case of gender classification, *K* is equal to two: one class for males and the other for females. The algorithm computes the Euclidean distance from each training point to each center point. The training point is grouped with the center point that is closest to it. Once all the training points are grouped, a new center point is calculated for each group based on the mean value of all the data points in the group. Since the center points shift due to the mean operation, all the training points are regrouped based on the new centers, and the new centers are recalculated. This process is continued until the center points do not move. Grouping data based on the Euclidean distance between a test point and two center points is like dividing the data with a hyper-plane that splits the two center points.

LDA tries to achieve a set of weights that represent a hyper plane, which splits the data into two classes. Finding the optimal value for these weights is straightforward: given a

training set data matrix $\mathbf{M}$, and a vector $\mathbf{g}$ that specifies the gender of each member of the training set (for instance: a 0 for males and a 1 for females), the unknown vector of weights *w* can be found by solving the linear system:

$$\mathbf{M}^{\mathbf{T}}\mathbf{W} = \mathbf{g} \tag{1}$$

Once $\mathbf{W}$ is calculated by multiplying $\mathbf{g}$ by an appropriate pseudo-inverse of $\mathbf{M}$, determining the gender of a test face is accomplished by computing the inner product $\langle \mathbf{x}, \mathbf{W} \rangle$ of $\mathbf{x}$ and $\mathbf{W}$. For example, for threshold=0.5, if $\langle \mathbf{x}, \mathbf{W} \rangle$ <= 0.5, the test face is classified as a female, otherwise male.

In case of MDB, instead of using Euclidean distance to classify data in two classes, we use the Mahalanobis distance. The Mahalanobis metric represents the distance from the mean group value that has a constant covariance; so in two-dimensions, this distance is given by an ellipsoid. As an example, a cut at a certain height through a two-dimensional Gaussian distribution represents a Mahalanobis curve. The equation for this metric is:

$$r^2 = (\mathbf{x} - \mathbf{m}_\mathbf{x})^T \mathbf{C}_\mathbf{x}^{-1} (\mathbf{x} - \mathbf{m}_\mathbf{x}) \tag{2}$$

Where $\mathbf{m}_\mathbf{x}$ and $\mathbf{C}_\mathbf{x}$ represents the mean and covariance matrix respectively. The classification process consists of calculating the Mahalanobis distance of a test point to the mean of the two groups and then deciding which mean is the closest one.

In nearest neighbor approach, one has to measure the distance of the test sample to every training sample, rather than just the distance to the mean training sample like in K-means. The test sample is then assigned to the class, which has the shortest distance. This is the special case of shortest K-nearest neighbor approach where K =1. Where K>1 the method is made sensitive to outliers. Rather than choosing the class with the shortest distance to a test sample, the class having majority of samples among the k-nearest neighbors of the test sample is chosen, i.e. a type of voting is performed. This help smooth out the distribution,

lessening the effect of outliers. Both of these nearest neighbor approaches have the practical disadvantage that all the projected training samples must be stored and searched during the testing phase.

### 2.3 *Performance evaluation of a classifier*

Performance of a particular classifier can be estimated in term of true positive and false positive rates. True positive rate (TPR) represents the number of correct positive cases divided by the total number of positive cases whereas, false positive rate (FPR) is the number of negative cases predicted as positive cases, divided by the total number of negative cases. When a graph is plotted between TPR and FPR for different threshold values, the resulting curve is called Receiver operating characteristics (ROC) curve. ROC curve summarizes how well a classifier has performed under different operating conditions, for a particular problem [6].

Single figures of merits are useful when comparing a classification system under a number of different conditions or settings. However, one of the greatest assets of testing is lost because they don't characterize the system over its entire operating range [1]. Hence ROC curves are plotted by adjusting the classification threshold and computing TPR and FPR at each threshold. The selection of operating threshold is then application-specific, depending on the maximum acceptance of false and true positives. The ROC curve thus provides the operator a degree of freedom to select the operating point which best accomplish his requirements, or simply reject the system outright if it is unable to meet their needs. In the absence of an application-specific operating point, the equal error rate (ERR) can be used to provide a single feature of merit [1]. This is the point on the ROC curve where the likelihood of a false positive and false negative are equal.

To obtain an ROC curve, each classifier is tuned by changing its sensitivity level (threshold). When the sensitivity is at the lowest level, the classifier neither produces false alarms, nor detects positive cases, i.e. the origin of the ROC. When the sensitivity is increased, the classifier detects more positive examples but may also start generating false alarms (false positives). Ultimately the sensitivity may become so high that the classifier always claims each case is positive [14]. This corresponds to the top right hand corner (point (1, 1)) of the ROC. A classifier based on simply making random guesses will have an operating point somewhere on the diagonal line between the origin and top right hand corner (point (1, 1)).

Of course one wants the true positive rate to be as high as 1, and the false positive rate to be as low as 0, i.e. points at the top left corner of the ROC curve. Both the Y-axis and X-axis are normalized (range 0-1), therefore the area under the ideal ROC curve will be 1. Consequently a given classifier is said to be an optimal one for a given problem, if the area under its ROC is near to 1. Scott [11] showed that a "maximum realizable" ROC is the convex hull of the classifier's ROC. As a result AUCH of an ROC curve is taken as a measure of the performance of a classifier.

## 3. An Overall Performance Measure of a Classifier

We know that each classifier has a sensitivity level (threshold). This sensitivity level is varied from 0 to 1 with step 0.1, generating an ROC curve for each feature subset as shown in Fig. 3. Area under Convex Hull (AUCH) is obtained for each of the resulting ROC curve. In other words varying feature subset give us different ROC curves for the same classifier. These different ROC curves obtained for the same classifier have different AUCH. When these AUCH are plotted against number of features, we obtain curves like shown in Fig. 4. We call these curves the areas under the convex hulls (AUCHS) curves.

A question arises here: is there any scalar value that roughly summarizes the performance of a classifier varying both threshold and feature subset (a scalar value summarizing the AUCHS curve)? We propose the idea of finding area under the convex hull of AUCHS curve (AUCH of AUCHS). This is used to summarize the overall performance of a classifier for different macro feature subsets and thresholds. Note that AUCH of AUCHS is obtained by including only point (0, 0) to the AUCHS curve (not point (1, 1)). Taking AUCH of ROC curve, we include both these points, but here in case of AUCHS, since we do not expect point (1,1), therefore it is not considered when we compute the AUCH of AUCHS.

## 4. Proposed Scheme for Combining Classifiers Using GP

Classifiers are usually combined to improve classification performance. But there are no general rules as to how a number of classifiers should be combined in a best possible way. One approach is to generate a large number of classifiers and then to select the best combination [13]. Boosting techniques are also used for combining classifiers [15]. But in Boosting normally a single classifier is improved by iteratively retraining it. Here in this work, optimal combinations of fixed classifiers are automatically generated using GP. They are fixed, because we do not retrain them, as is done in boosting. Also Boosting is in general applied by assuming the classifier is operated at a single threshold, producing a single pair of TPR and FPR on each retraining. Consequently it produces a single point on the ROC rather than a curve required as a metric for judging the performance of a classifier.

Genetic programming (GP) is an optimization technique based on the concepts of Darwinian evolution. A population of individuals, each representing a potential solution to the problem to be optimized, is taken into consideration in order to derive an optimal solution. The solution offered by each individual is assigned a fitness value (a value which shows how well that solution performs). This fitness of an individual is proportional to its probability for

reproducing new individuals. The new individuals replace less fit members of the population, and so the overall population fitness improves with each generation. We have used GP to produce automatically a combined classifier that extracts more useful information than the individual classifiers. This combined and optimized classifier should have better ROC curve and hence large area under the convex hull of its ROC curve, representing higher classification performance.

Combination of classifiers can be carried out in two ways, namely homogenously and heterogeneously. In homogenous combination, two or more classifiers of the same type (trained on the different feature subsets) are combined using GP. While in heterogeneous combination, two or more classifiers of different types (MDB with LDA trained on the same/different feature subsets) are combined using GP. In the former case the composite classifier is like a polynomial, which is a function of only one classifier (see Fig. 2). Whilst in the later case it is a function of two or more classifiers (multiple dependencies).

## 5. Implementation

In order to practically implement and evaluate different classification algorithms, we have used the Stanford University medical student image database [5]. These images comprise only the frontal-view of a person's face. This database consists of 200 male and 200 female grey scale images of size 128x128 pixels. Examples of some frontal face images are shown in Fig. 1.

Scaled down jackknife [2,10] scheme is employed to utilize the image database and check the performance of the different algorithms. Two train to testing ratios (1:3 and 1:9) are selected. This demands first randomly choosing 50 males and 50 females to use as our training set, leaving the rest of the images to test the algorithms. This is done 4 separate times, thus allowing all of the images to be in a training set exactly once. The algorithm

results are averaged across all test sets to increase their statistical significance. For the 2nd train to testing ratio of 1:9, 20 male and 20 female Images are randomly picked for training and the remaining images are left for testing. This process is carried out 10 times to avoid any sort of biasing in picking the faces. We have used Intel Pentium IV machine with a processor speed of 2.0 GHz for our simulations.

Fig. 1. (a) .(here)

Fig. 1. (b) .(here)

### 5.1 *Obtaining ROC and AUCHS curves*

Each classifier is tested on the test set and its prediction for each test image is stored. The TPR and FPR are computed and the threshold is then varied in steps of 0.1. For the whole threshold range of 0 to 1, TPR is plotted against FPR obtaining an ROC curve and it's AUCH. The feature subset is then varied to obtain different ROC curves and consequently different AUCH for the same classifier. AUCH is then plotted against feature subset to obtain AUCHS curve for each classifier.

### 5.1 *Development of OCC*

Four binary floating arithmetic operators (+, -, *, and protected division), if less than (IFLT), if greater than (IFGT), and ABS are used as conventional functions in the GP tree. The classification algorithms to be combined are represented as special unary functions, with their threshold supplied as their single argument (see Fig. 2). It means that these classifiers are acting like the independent variables in a particular polynomial. Note that we have combined only two classifiers: LDA and MDB using GP. First LDA classifiers (evaluated at different

feature subsets) are combined among themselves using GP to evolve an optimized classifier. Then both LDA and MDB classifiers (trained on the same feature subsets) are combined to do the same. The terminal T represents the current value of the threshold that is applied to the classifier evolved by GP. About 200 constant between -1 and +1 are also used as terminal (see table 1).

Fig. 2. (here)

Table 1. (here)

First an initial population of 200 polynomials is generated. Each new individual (polynomial) is tested on each testing example with the threshold parameter (T) taking values from 0 to 1 with step size 0.1. For each threshold value the true positive and false positive rates are calculated. Since a classifier can always achieve a zero success rate and 100% false positive rate, the points (0, 0) and (1, 1) are always included. These plus the eleven true positive and false positive rates are plotted and the AUCH is calculated. This area is representing the fitness of the individual GP program. The larger the AUCH, the better the individual has performed [14]. Polynomials having good AUCH will have more chances of reproduction.

It is observed that using only this AUCH as fitness criteria, usually better ROC curves are not produced. These ROC curves have only one or two points having yet higher TPR values. The rest of the points have quite low TPR values (reclining below the convex hull, see curve OCC3 in Fig. 8). This is due to the fact that, if there is only one point having high TPR and low FPR value (i.e. upper left corner), the convex hull will include this point and the point (1,1) (no matter where the other points lie). This ROC is a maximum realizable one, not the actual maximum ROC curve. What it means is that to practically achieve the realizable points on this MRROC, one has to find again another combination of classifiers according to Scott et., al. [11]. We tried to achieve the maximum ROC curve, rather than the MRROC

curve during evolution by forcing GP to evolve such combinations that have almost all points reclining on the convex hull.

This is achieved by keeping the accumulative sum of the TPR and FPR values at the 11 ROC points for each candidate. The candidate, whose accumulative sum of TPR > 8.5 and that of FPR < 3.8, is then given bonus fitness. This extra constraint not only produced ROC curves with mostly having all points reclining on the convex hull, but also narrowed the search space. This is because, once a candidate that fulfils this criteria is evolved, it is given bonus fitness and thus is retained and reproduced by the inherent mechanism (survival of the fittest) of GP. Hence in coming generations, only its children and neighbors from the whole solution space occupy most of the population. In fact it is a kind of dragging the search space towards the desired solution.

**6. Results and Discussion**

The classifiers are first compared by varying the threshold and obtaining ROC curves (see Fig. 3). Starting from the first feature, the feature subset is increased up to 20 and the corresponding AUCH values for each classifier are obtained (see Fig. 4 and 5). This is repeated by considering 50 macro features for 1:3 train to testing ratio, as shown in Fig. 6. AUCH of AUCHS for the three different cases (1:3 and 1:9 train to testing ratio with 20 macro features, and 1:3 train to testing ratio with 50 macro features) is shown in Fig. 7.

Fig. 3. (here)

In Fig. 4, it can be observed that the AUCH saturates for all of the classifiers except K-means approximately at macro feature subset 5. This is because, using more macro features can yield a better representation of the original image, but it doesn't guarantee the

minimization within class scatter in eigenface space. The AUCH of K-means even decreases with an increase in macro feature subset, illustrating the curse of dimensionality. This may be due to the fact that gender is not the main separation factor in data. Skin color, glasses or no glasses, close picture may have the stronger separation effect than gender and are most likely to be exploited by K-means for separation of data. Increase in feature subset and keeping the training examples fixed further enhance this effect and so results in performance degradation for K-means.

Fig. 4. (here)

The LDA performs better than the rest of classifiers in terms of AUCH. Its performance in terms of AUCH of AUCHS is also better than the rest of classifiers. This is because of the fact that AUCH of AUCHS also depends heavily on high ordinate values corresponding to low abscissa values. In Fig. 5, with train to testing ratio equal to 1:9 i.e. training examples being reduced, it can be observed that before saturation, the performance of MDB improves and becomes comparable to that of LDA. However after the saturation is reached, the performance of MDB degrades as compared to LDA. This is a different behavior compared to previous case of train to testing ratio equal to 1:3. The performance of kNN classifier improves with decreasing training example (see Fig. 5). This may be due to the fact that the effect of outliers in kNN classifier decreases as training examples are decreased. With train to testing ratio equal to 1:3, and feature subset variation increased up to 50 (see Fig. 6), it can be observed that the pattern almost remains the same for all classifiers above feature subset equal to 20.

Fig. 5. (here)

Fig. 6. (here)

In Fig. 7, as discussed earlier, we present the whole summary of the classifiers performance at different threshold and feature subset with a single scalar value of AUCH of AUCHS. It is observed that the overall performance of LDA improves slightly with increasing training samples. While that of kNN and MDB decreases with increasing training samples. When the feature subset variation is increased beyond 20 up to 50, AUCH of AUCHS increases for all of the classifiers. This was expected for all the classifiers except K-means, because K-means performance degrades with increase in feature subset. This unexpected increase in AUCH of AUCHS for K-means happened because the values are greatly affected by the ordinate values corresponding to initial abscissa values. Hence AUCH of AUCHS does not depict the decrease in performance of K-means.

Fig. 7. (here)

To develop an OCC, we first combined only the LDA classifier with itself by using it as a unary function in GP tree (homogeneous combination). It is observed that using only first two features, the combined classifier has better performance than the original LDA classifier (see curve OCC1 and LDA1 in Fig. 8). This is due to the fact that GP is able to evolve a classifier, which has better discrimination power then the original ones. GP is then used to evolve an optimized classifier, that is a function of both MDB and LDA i.e. both MDB and LDA are used as unary functions in GP tree (heterogeneous combination). The optimized classifier obtained (using only first two features) in this case has higher AUCH than both of the previous cases (see curve OCC2 in Fig. 8). Curve OCC2 is obtained by giving bonus fitness to the polynomials fulfilling the added constraint, while curve OCC3 is obtained without using the idea of added fitness. It can be observed from Fig. 8 that curve OCC2 has not only higher AUCH than curve OCC3, but can also be considered a maximum ROC curve, rather than a maximum realizable ROC curve as it has almost all points reclining on the convex hull. While observing Fig. 8, at first glance one can feel that OCC3 is performing

better than OCC2, as it seems to be higher than the rest of the curves. But observing closely, it is found that OCC2 encompasses larger area due to two facts. The first one is that for TPR equal to about 0.7, it has zero FPR, while OCC3 has some non zero FPR. The 2nd fact is that OCC2 quickly reaches to the highest value of TPR =1 at about 0.37 FPR. While OCC3 reaches TPR=1 at about FPR=0.55. We should note here that for a particular application, the worth of an ROC curve depends on the cost of different types of errors. But since we do not know these costs in advance while designing our classifier, we therefore consider only the gross information revealed by the AUCH. Thus in this scenario OCC2 is performing better than OCC3.

Fig. 8. (here)

**7. Conclusion**

Different classifiers are good in learning different aspects of data and hence by combining them, a more generalized classifier is formed. Genetic programming can automatically develop optimum combined classifier (OCC) that has comparable performance to that of original classifier using only 2 features instead of 20. Using the idea of bonus fitness, maximum ROC curves can be generated instead of maximum realizable ROC curves. Using only those feature that are most important for gender classification instead of the first high-energy feature vectors could further improve the performance of a classifier. This might be carried out by following Zehang [16] for feature selection relevant to gender classification. Combining different classifiers trained on different features can further improve ROC curves, because different features have different discrimination capabilities.

In this paper, the proposed AUCHS curve and its AUCH summarizes the performance of a classifier corresponding to different feature subset and thresholds. It gives high importance to the ordinate values corresponding to initial abscissa values, which at first glance seems to be unnatural. But implicitly it is not a drawback, because in KLT

transformation, only the first few eigenvectors (features) contain most of the information about the data, and consequently should be given more importance.

**Acknowledgements**

This work is sponsored by the Higher Education Commission/Ministry of Science and Technology Government of Pakistan.

**References**

[1] C. Lincoln, Pose independent face recognition, PhD thesis, Dec. 2002, Department of Electronic System Engineering, University of Essex.
[2] D. Valentin, H. Abdi, B. E. Elderman and A. J. O'Tooli, Principal component and neural net analysis of face images: What can be generalized in gender classification?, Journal of Mathematical Psychology, 41,398-412. 1997.
[3] 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.
[4] H. Abdi. (1988), A general approach for connectionist auto-associative memory: interpretation, implication an illustration for face processing, In J. Demongeot (Ed). Artificial intelligence and cognitive sciences. Manchester University Press.
[5] http://white.stanford.edu/~diclaro/ee368a/code/male .zip
[6] J. A. Swets, R. M. Dawes, and J. Monahan, Better decisions through science, Scientic American, pp. 70-75, October 2000.
[7] L Sirovich and M. Kirby, Low dimensional procedure for the characterization of human face, J, Optical 4:519-524, 1987.
[8] M. Burton, V. Bruce and N. Dench, What is difference between men and women? Evidence from facial Measurement, Perception, 22,153-176, 1993.
[9] M. A. Turk and A. Pentland, Eigenface for recognition, Journal of Cognitive Neuro Science, 3, (1991),71-86.
[10] M. Kirby and L. Sirvich, Application of Karhunen Louve procedure for the application of human IEEE, 12:103-108, 1990
[11] 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, pages 304-315,University of Southampton, UK, 14-17 September 1998.
[12] 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.
[13] T. K. Ho, Data complexity analysis for classifier combination, Multiple Classifier Systems, Proc. Of 2nd International Workshop, MCS2001, Cambridge, UK, 2-4 July 2001, pp. 53-67, 2001.
[14] W. B. Langdon and B. F Buxton, Genetic programming for combining classifiers, In GECCO'2001, Morgan Kaufmann.
[15] Y. Freund and R. E Schapire, Experiments with a new boosting algorithm, In Machine Learning: Proc. 13th International Conference 1996, pp 148-156. Morgan Kaufmann.

[16] 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.

Table 1: GP Parameters

| ***Objective:*** | ***To evolve a classifier with maximum convex hull area*** |
|---|---|
| **Function Set:** | +, -, *, protected division, IFGT, IFLT, and ABS |
| **Special Function:** | Classifier (MDB, LDA) |
| **Terminal Set:** | Threshold T plus 200 constants randomly chosen between -1…+1 |
| **Fitness :** | Area under the Convex Hull of 11 ROC points. |
| **Bonus Fitness:** | Add 1.0 to fitness, if SumTPR > 8.5 and SumFPR < 3.8 |
| **Selection:** | Generational |
| **Wrapper:** | Positive if >= 0, else Negative. |
| **Population Size:** | 200 |
| **Initial Tree Depth Limit:** | 7 |
| **Tree generation Method:** | Ramped half and half |
| **Reproduction Prb:** | 20% |
| **Mate Selection Prb:** | 80% |

**List of Figure Captions**

Fig. 1. (a) Example sets of woman frontal face database. (b) Example sets of man frontal face database.

Fig. 2. GP trees. In (a) only one classifier is used as a special Unary Function, while in (b) two classifiers are used as special Unary Functions.

Fig. 3. ROC curves of LDA for feature subset range of 2-7. FS denotes the number of features being used (note: only points on the convex hull are shown).

Fig. 4. Comparison AUCHS curves of different classifiers for first 20 macro features with 1:3 train to testing ratio (note that to evaluate AUCH, x-axis is normalized).

Fig. 5. Comparison of AUCHS curves of different classifiers for first 20 macro features with 1:9 train to testing ratio.

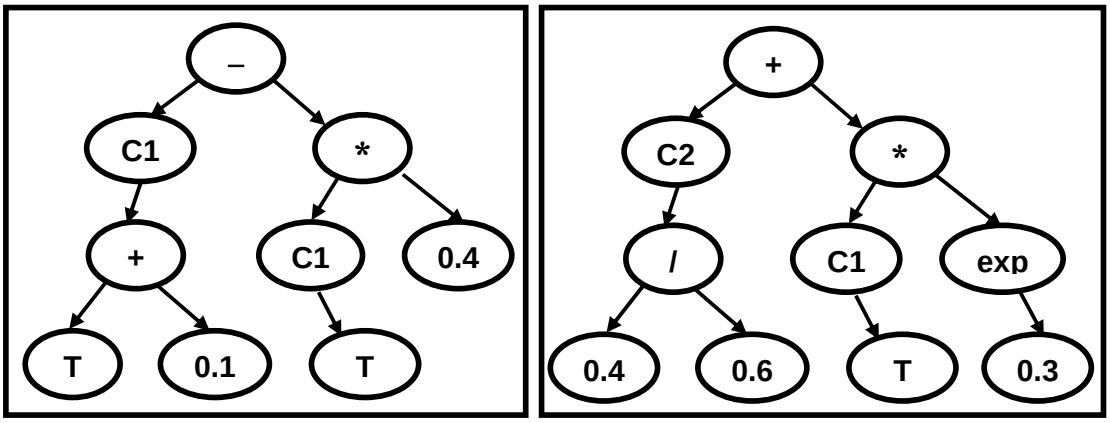
Fig. 6. Comparison of AUCHS curves of different classifiers for first 50 macro features with 1:3 train to testing ratio.

Fig. 7. Bar Chart showing the performance of different classifiers in terms of AUCH of AUCHS

Fig. 8. Comparison of LDA (using first 2 features for LDA1 and first 20 features for LDA2) and OCC (using first 2 features only). Note that only points on the convex hull are shown for LDA1 and LDA2.

Fig. 1. (a) Example sets of woman frontal face database.

Fig. 1. (b) Example sets of man frontal face database.




(a) (b)

Fig. 2. GP trees. In (a) only one classifier is used as a special Unary Function, while in (b) two classifiers are used as special Unary Functions.

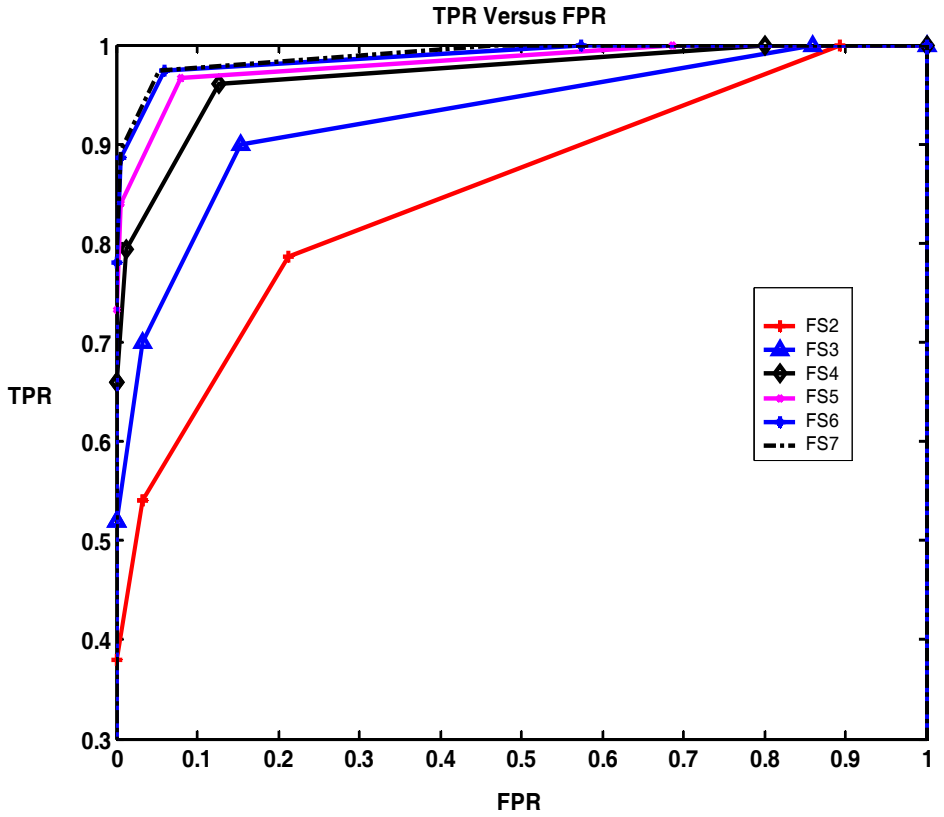



Fig. 3. ROC curves of LDA for feature subset range of 2-7. FS denotes the number of features being used (note: only points on the convex hull are shown).

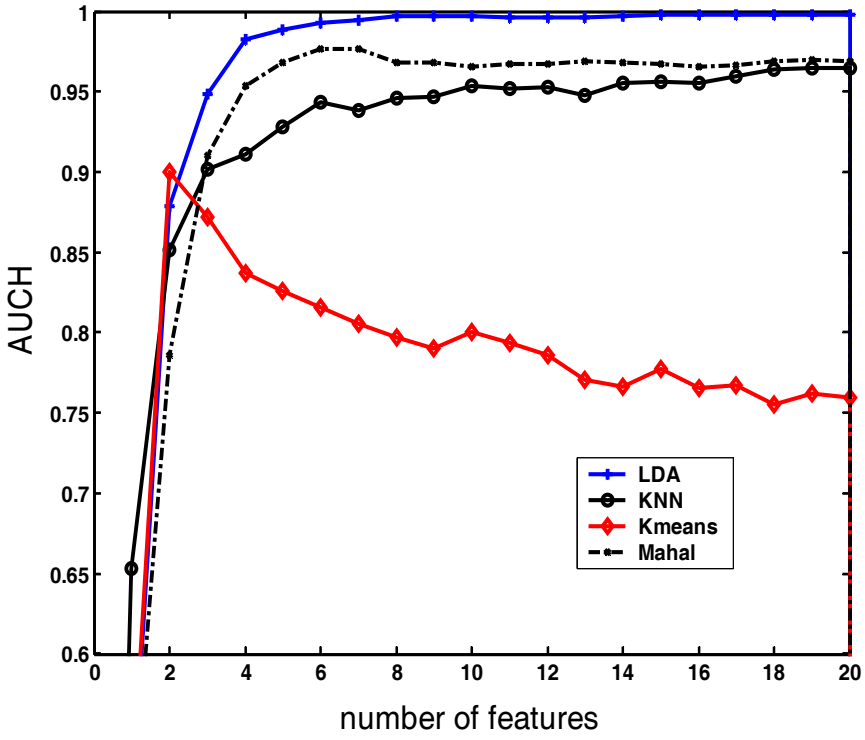



Fig. 4. Comparison AUCHS curves of different classifiers for first 20 macro features with 1:3 train to testing ratio (note that to evaluate AUCH, x-axis is normalized).

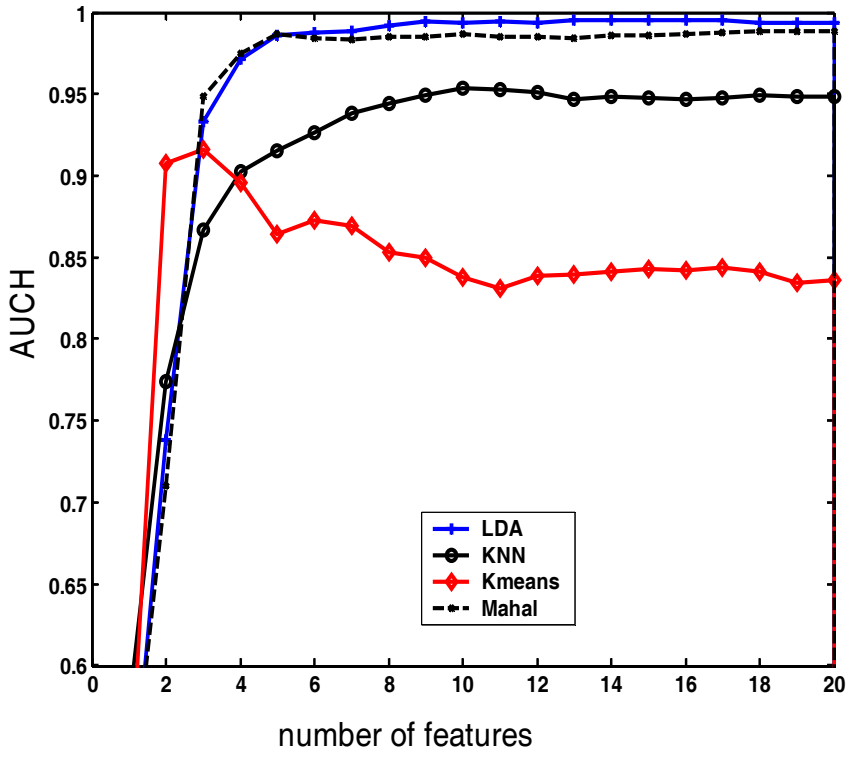



Fig. 5. Comparison of AUCHS curves of different classifiers for first 20 macro features with 1:9 train to testing ratio.

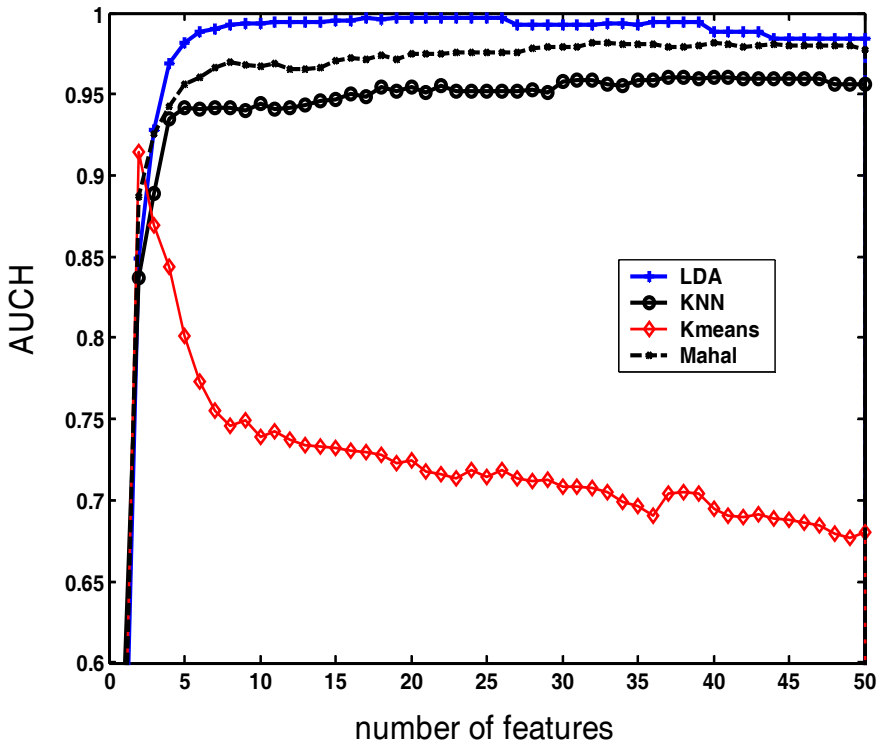



Fig. 6. Comparison of AUCHS curves of different classifiers for first 50 macro features with 1:3 train to testing ratio.

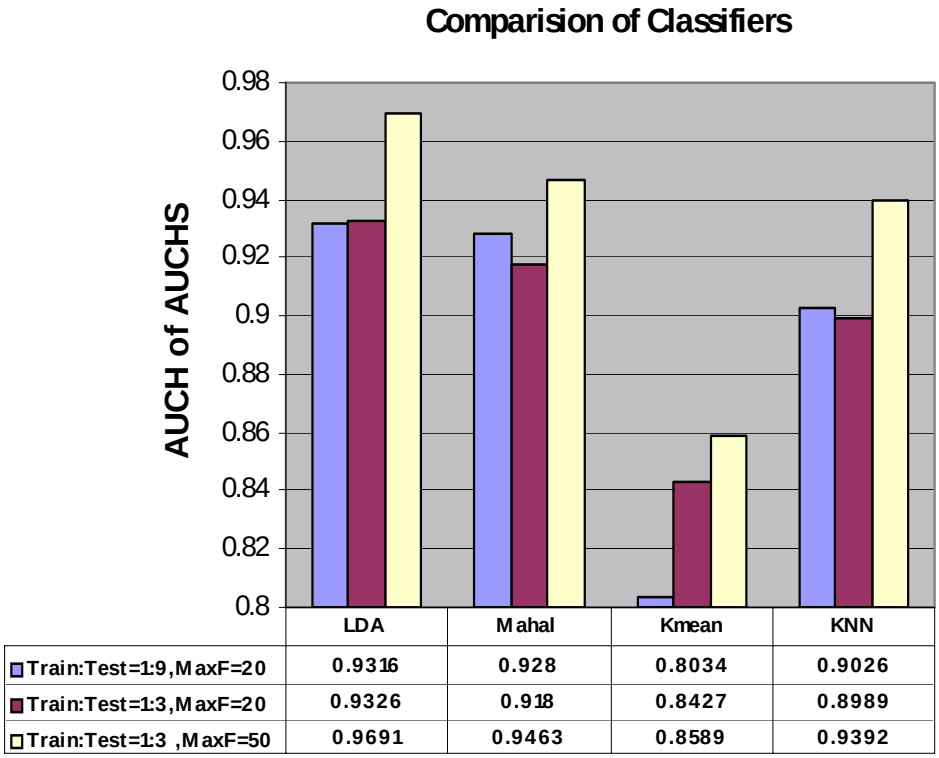



Fig. 7. Bar Chart showing the performance of different classifiers in terms of AUCH of AUCHS

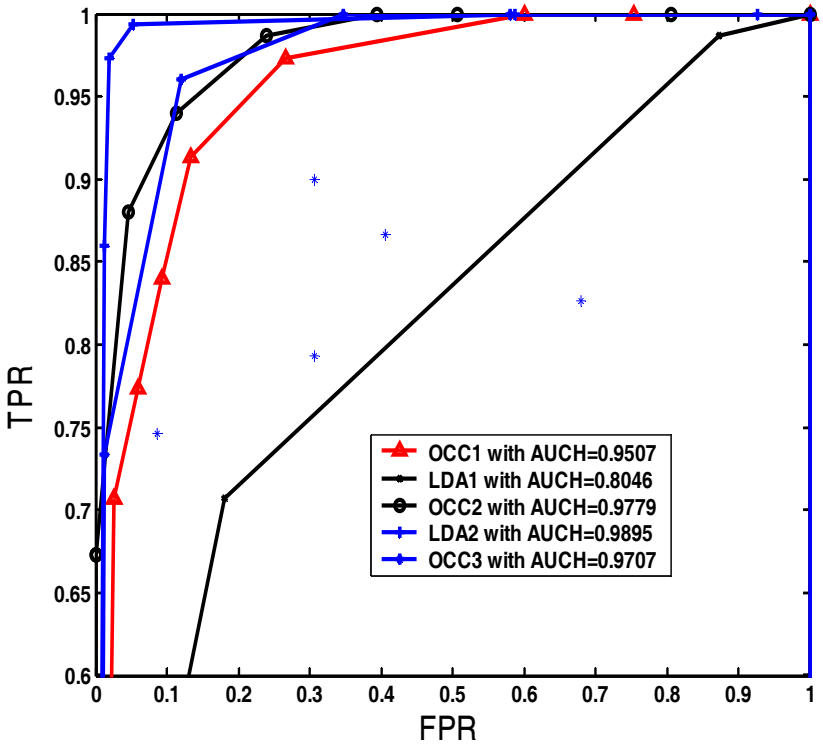



Fig. 8. Comparison of LDA (using first 2 features for LDA1 and first 20 features for LDA2) and OCC (using first 2 features only). Note that only points on the convex hull are shown for LDA1 and LDA2.