Gender Classification using LDA, KNN, SVM, Naive Bayes and Decision Tree Learning

Mayank Wadhawan, Student, UFID - 59148122

***Abstract*—In this project, I have performed Gender Classification from static images using 5 different classifiers. They are Linear Discriminant Analysis, K-Nearest neighbors, Support Vector Machines, Naive Bayes and Decision tree learning. I was able to determine Gender of a person with accuracy of 98.75% (using Linear Discriminant Analysis). The main purpose of this project was to compare performance and accuracy of these classifiers for the purpose of Gender Classification. The comparisons of results these classifiers are mentioned in detail in results and analysis section.**

***Index Terms*— Linear Discriminant Analysis, K-Nearest neighbors, Support Vector Machines, Naive Bayes and Decision tree learning.**

I. Introduction

I have implemented the code for this project in Matlab. Firstly, during learning phase, features of a face are extracted using Histogram of Oriented Gradients and then fed into the 5 classification algorithms. Histogram of Oriented Gradients (HOG) is used to extract visual features in images. It is used in this project to determine shape of face in this image. This increases the accuracy of classification of data. In the next step, prediction of a new image is done. The predicted label is then compared to the actual label to determine the accuracy of the system. The 5 classification algorithms used are described briefly below.

In Linear Discriminant Analysis (LDA), we find the component axis to separate the two class labels. However, in Principal component analysis (PCA), we find the component axis which maximizes the variance between data points [1]. K-Nearest neighbors (KNN) algorithm makes use of k nearest neighbors to determine the label of a new data. The value of k is provided by the user. KNN would have disadvantage when number of data elements of one class label are a lot more than other labels. Since the number of instances of one label is more, therefore it will have more occurrences as closest neighbors. In this cases, data can be erroneously classified. Support Vector Machines (SVM) is a method which takes advantage of supervised learning. In supervised learning, we can determine a function from training data (labelled). We feed training data into supervised learning algorithm and this algorithm determines a function. This function can later be used for new instances to find the correct class/label. Naive Bayes classifier is an example of probabilistic classifier and it uses Bayes Theorem. Decision tree learning is a learning algorithm. It looks a lot like flow chart. The internal nodes in this tree contains tests. The branches if this tree are the outcome. The leaves of this tree have the label of class. Gender classification has applications in targeted advertisement for men and women. It can also be useful to gather customer statistical data and in video games.

II. DATASET

I have used the FEI Face database for training phase and testing phase. This database is allowed for research purposes.

Figure 1: Sample Dataset

I have used 156 images for men and 156 images for women respectively for training phase and 40 images each of men and women for testing phase. A total of 396 images were used. 80% images were used for training and 20% images were used for testing. This is shown in Table 1.

TABLE 1

	Male	**Female**
Training Images	156	156
Testing Images	40	40
Total Images	196	196

III. Explanation of classifiers

A. Linear Discriminant Analysis (LDA)

LDA is used to reduce the dimensions of a data set into lower dimensions. This new feature space will have better class label separation. LDA is a supervised algorithm because it takes in consideration of class labels of data. Whereas, PCA is unsupervised algorithm. In Figure 2, we can see that LDA has good class separation in feature space. Dimension reduction has advantages like reduction in computation cost

and less usage of memory. LDA can be performed using following steps.
1) In this step, we determine the mean vector for different class labels.
2) In this step, we determine within class (M_w) and in between class (M_B) scatter matrices.
3) In this step, we determine eigenvalue for matrix $M_{w^{-1}} M_B$.
4) In this step, we use the eigenvectors and eigenvalues from step 3. We sort the Eigenvectors by decreasing order of the Eigenvalues. Then we choose top k eigenvectors.
5) This is then transformed in new space. This new space will have good class separation. The new data can then easily be classified.

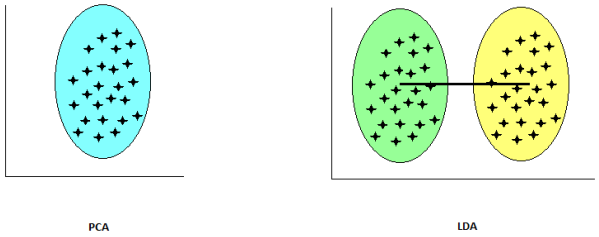

Figure 2: PCA vs LDA

B. K-Nearest neighbors (KNN)

K -Nearest neighbor algorithm has 2 steps. First step is training phase, in which we just store the data and class labels. In Second Step, the algorithm is given two inputs, the value of k and new data to be classified. In this step, we just find the k closest neighbors of the new data. The new data is then assigned a class label based on which class label had maximum occurrences in the k closest neighbors. This algorithm is one of the most easiest among classification algorithms [2]. Figure 3, explains how a new data is classified using KNN algorithm. In this example, we determine the 3 closest neighbors of new data (for k=3). Out of 3 three neighbors, label red has 2 occurrences and label blue has 1. Therefore, new data is classified as red.

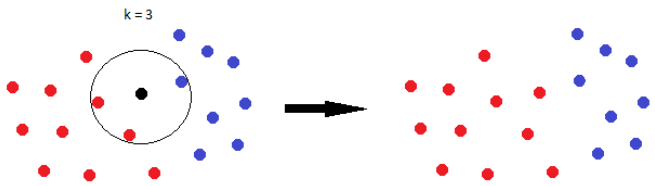

Figure 3: K-Nearest neighbors

C. Support Vector Machines (SVM)

A linear classifier divides different data points in their groups with a straight line. This generally does not provide high accuracy [2]. In real work situations, we would need better way to separate these data points. In Figure 4, you can see a linear classifier which separates data points labelled in green and red. In Figure 4, you can also see a more common situation. In this example, green and red data points would have to be separated using curve. This is an example of hyperplane classifier as many lines are required to separate these data points. Support vector machines are used for these situations. In SVM, input space is mapped to feature space using kernels (functions). We can see that in feature space, we can separate data groups using a straight line.

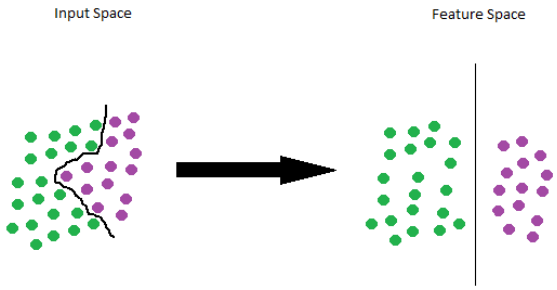

Figure 4: Support Vector Machines

D. Naive Bayes

Figure 5 shows how a Naive Bayes classifier can be used for finding class label of new data [4]. In this example, we determine the prior probability of Orange and Purple. Prior probability can be used to determine future outcomes.

Prior Probability (Orange) = Sum of Orange data points / Sum of all data points
Prior Probability (Purple) = Sum of Purple data points / Sum of all data points

In the next step, we find n number of neighbors of new data. The value of n is predefined.

Likelihood of new data being Orange= Orange neighbors / Sum of Orange data points
Likelihood of new data being Purple= Purple neighbors / Sum of Purple data points
Therefore, for the new data point, posterior probability is given as,

Posterior Probability (Orange) = Prior Probability (Orange) *Likelihood of new data being Orange
Posterior Probability (Purple) = Prior Probability (Purple) *Likelihood of new data being Purple

This can be understood easily by referencing Figure 5. In the example shown in Figure 5, even though number of data points of Purple label are more, since the number of Orange points in the vicinity of new data are more, therefore Posterior Probability(Orange) > Posterior Probability(Purple). Hence the new data will be labelled as Orange.

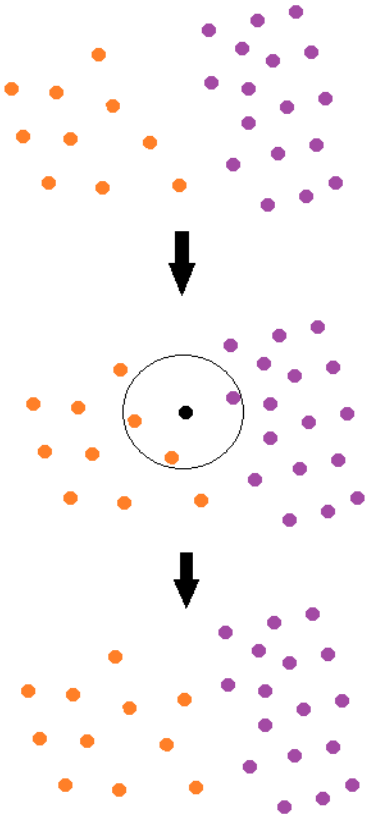

Figure 5: Naive Bayes

E. Decision Tree learning

During the learning phase, the decision tree is constructed based on training data. When we want to test a new data, the algorithm will start at root and tests (in internal nodes) will be conducted. Based on results of these tests, we follow the branches to the leaf nodes. The leaf nodes will have final result (class label) [5]. Advantage of Decision tree learning is that it is easy to understand and is well suited for large data.

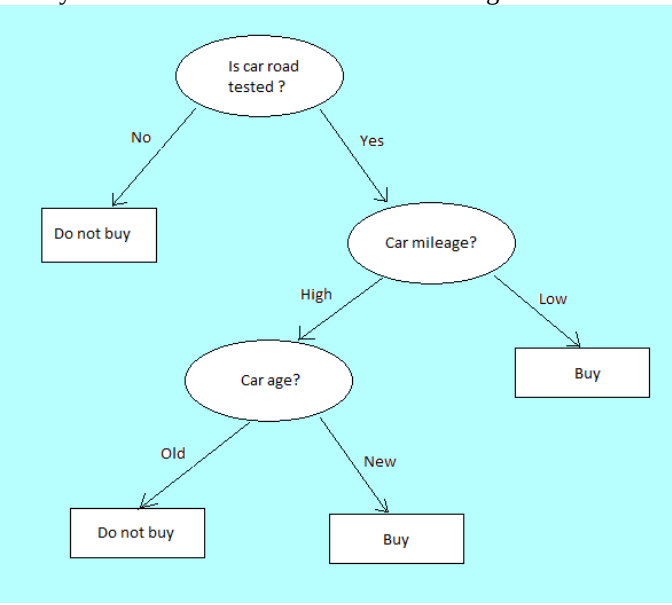

Figure 5: Decision Tree learning example

IV. Project Implementation

Gender classification was done using following steps.

1. Acquired dataset from FEI Face database for training phase and testing phase. I used 396 images and they were placed in separate folders in Training and Testing phase.
2. 80% images are used for training and 20% images are used for testing.
3. Images were read one by one from the folders and stored in a variable in Matlab. After that, they were converted to gray and then Gaussian filter is applied.
4. HOG features are extracted and then stored in a variable. Shown in figure 6.
5. Labels are then assigned to these features.
6. This learning data (features and labels) is then fed into 5 different classifiers for gender classification.
7. Test images are then read one by one sent to these classifiers for prediction.
8. Actual label and predicted label are then compared to determine of the predicted classification is correct.
9. Step 8 is performed for each of the classifiers.
10. Results are then captured and they are analyzed in Results and Conclusion section.

Figure 6: HOG features extracted

V. Code Instructions

Please refer to readme.txt file for the steps. The submitted version of code demonstrations gender classification for one image. Comments are mentioned in the code on how to use this file for all the test data and for different classifiers.

VI. Related Work

[10] proposes age and gender classification using convolutional neural networks. They were able to achieve an accuracy of 86.8 percent. Convolutional neural networks are easy to implement but are difficult to train. They are also not probabilistic. [11] proposes gender classification based on gait. They were able to achieve an accuracy of 97.9% [12] proposes gender classification using Local Directional Pattern (LDP). They were able to achieve accuracy of 95.05%. In this research, small features of face are extracted and then fed into SVM. [13] proposes gender classification based on speech. They were able to achieve accuracy of 98% and 95 for noisy speech. The disadvantage of this system is that it works best if microphone is close to the user. There is plenty of reading material available, I have listed a few here.

VII. RESULTS

A. Confusion Matrices

The performance of Gender Classification using different classification algorithm has been mentioned in this section.

The classification results for 5 classifiers have been compiled in Table 2 to 6.

TABLE 2: LDA Confusion Matrix

LDA		**Predicted Label**	
Actual Label		Male	Female
	Male	40	0
	Female	1	39

TABLE 3: KNN Confusion Matrix

KNN		**Predicted Label**	
Actual Label		Male	Female
	Male	40	0
	Female	2	38

TABLE 4: SVM Confusion Matrix

SVM		**Predicted Label**	
Actual Label		Male	Female
	Male	40	0
	Female	3	37

TABLE 5: Naïve Bayes Confusion Matrix

Naïve Bayes		**Predicted Label**	
Actual Label		Male	Female
	Male	38	2
	Female	6	34

TABLE 6: Decision Tree Confusion Matrix

Decision Tree		**Predicted Label**	
Actual Label		Male	Female
	Male	31	9
	Female	6	34

Since my project classifies images to Male or Female, therefore I have used confusion matrix to describe the performance of different algorithms.

1. True Positive - Image predicted as class X and belongs to class X.
2. True Negative - Image correctly predicted that it doesn't belong to class X.
3. False Positive - Image predicted that it belongs to X but it doesn't belonged to X.
4. False Negative - Image predicted that it doesn't belong to X but it belongs to X.

Accuracy = (True Positive + True Negative) / Total
Misclassification Rate/Error Rate = (False Positive + False Negative) / Total
These calculations are shown in Table in Table 7.

TABLE 7

Classifier	Accuracy	Misclassification Rate (Error Rate)
Linear Discriminant Analysis	98.75%	1.25 %
K-Nearest neighbors	97.5%	2.5%
Support Vector Machines	96.25%	3.75%
Naive Bayes	90%	10%
Decision Tree Learning	81.25%	18.75%

Precision = True Positive / Actual Positive
Specificity = True Negative / Actual Negatives
These calculations are shown in Table 8.

TABLE 8

Classifier	Precision	Specificity
Linear Discriminant Analysis	1	0.975
K-Nearest Neighbors	1	0.95
Support Vector Machines	1	0.925
Naive Bayes	0.95	0.85
Decision Tree Learning	0.775	0.85

B. Result analysis and Screenshots

The screenshots of the result have been shown in Figure 7 to 11. The image data for learning and testing are independent. There were 156 images each for men and women in learning phase. Additionally, 20 images each for men and women were used for testing phase. The images used for testing are of different people and still the prediction outcome of gender classification was very accurate. I have used the similar image for all classifiers for consistency in testing.

Figure 7: LDA Classification Result

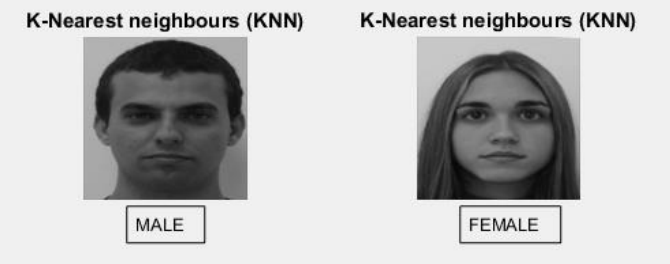

Figure 8: KNN Classification Result

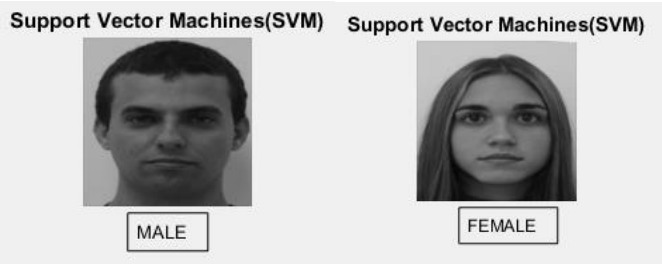

Figure 9: SVM Classification Result

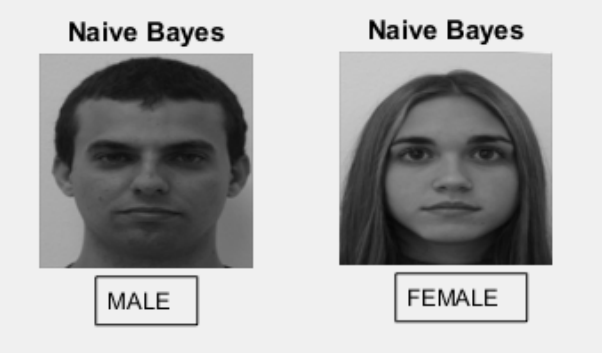

Figure 10: Naïve Bayes Classification Result

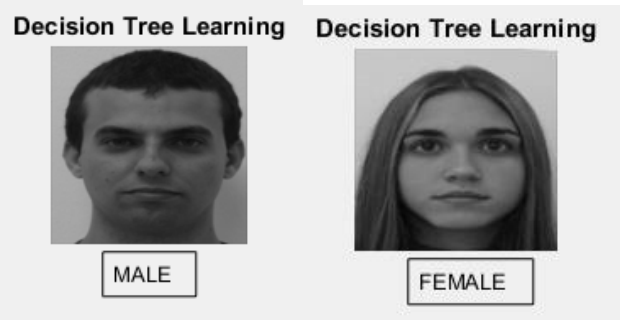

Figure 11: Decision Tree Learning Classification Result

C. Scope for Improvement

Gender classification works very accurately for images. This project can be extended for video stream. Since the time taken to predict is not that small, therefore output streaming would have some delay.

VIII. Conclusion

Gender classification in this project is done using Histogram of Oriented Gradients with the help of 5 different classifiers. Linear Discriminant Analysis, K-Nearest neighbors and Support Vector Machines have very good accuracy. LDA had highest accuracy among all the classifiers (Table 7). LDA had only one wrong classification, KNN had 2 wrong classifications and SVM had 3 wrong classifications. Therefore, LDA, KNN and SVM can be used accurately for gender classification.

Naive Bayes classification had 8 wrong classification with accuracy of 90%. Decision tree learning 15 wrong classifications with accuracy of 81.25%. Therefore these two algorithms are not suited for Gender Classification.

ACKNOWLEDGEMENT

I would like to thank Dr. Dapeng Wu for his help finding the right topic and his continuous guidance throughout the semester. Information taught in his lectures were very useful.

References

[1] PCA versus LDA. A. M. Martinez ; Robot Vision Lab., Purdue Univ., West Lafayette, IN, USA ; A. C. Kak
[2] Support vector machines. M. A. Hearst ; California Univ., Berkeley, CA ; S. T. Dumais ; E. Osman ; J. Platt more authors
[3] http://scholarpedia.org/article/K-nearest_neighbor. Leif E. Peterson, Center for Biostatistics, The Methodist Hospital Research Institute.
[4] Feature selection for multi-label naive Bayes classification. Min-Ling Zhanga, José M. Peñac, Victor Roblesc
[5] On the boosting ability of top-down decision tree learning algorithms. Michael Kearns AT&T Research Yishay Mansour Tel-Aviv University
[6] Probabilistic principal component analysis. Michael E. Tipping and Christopher M. Bishop
[7] Compressed Histogram of Gradients: A Low-Bitrate Descriptor. Vijay Chandrasekhar , Gabriel Takacs, David M. Chen, Sam S. Tsai, Yuriy Reznik, Radek Grzeszczuk, Bernd Girod
[8] Face recognition by using neural network classifiers based on PCA and LDA. Byung-Joo Oh ; Dept. of Electron. Eng., Hannam Univ., Daejeon, South Korea
[9] Face recognition using LDA mixture model. Hyun-Chul Kim, Daijin Kim, , Sung Yang Bang
[10] Age and Gender Classification using Convolutional Neural Networks. Gil Levi and Tal Hassner Department of Mathematics and Computer Science The Open University of Israel
[11] A Study on Gait-Based Gender Classification. Shiqi Yu ; Nat. Lab. of Pattern Recognition, Chinese Acad. of Sci., Beijing, China ; Tieniu Tan ; Kaiqi Huang ; Kui Jia more authors
[12] Gender Classification Using Local Directional Pattern (LDP). Taskeed Jabid ; Comput. Eng. Dept., Kyung Hee Univ., Yongin, South Korea ; Md. Hasanul Kabir ; Oksam Chae
[13] Robust GMM Based Gender Classification using Pitch and RASTA-PLP Parameters of Speech.Yu-min Zeng ; Department of Radio Engineering, Southeast University, Nanjing 210096, China; School of Physics Science and Technology, Nanjing Normal University, Nanjing 210097, China. E-MAIL: zengyumin@njnu.edu.cn ; Zhen-yang Wu ; Tiago Falk ; Wai-yip Chan
[14] The Distance-Weighted k-Nearest-Neighbor Rule. Sahibsingh A. Dudani ; Hughes Research Laboratories, Malibu, CA 90265.
[15] Optimization of k nearest neighbor density estimates. K. Fukunaga ; L. Hostetler