A comparative study of machine learning models for ethnicity classification

A comparative study of machine learning models for ethnicity classification

Advait Trivedi[1] and D Geraldine Bessie Amali[2]
[1]North Carolina state university, Raleigh, North Carolina 27695, USA
[2]School of computer science and engineering, VIT University, Vellore -632014, India

E-mail : geraldine.amali@vit.ac.in

Abstract. This paper endeavours to adopt a machine learning approach to solve the problem of ethnicity recognition. Ethnicity identification is an important vision problem with its use cases being extended to various domains. Despite the multitude of complexity involved, ethnicity identification comes naturally to humans. This meta information can be leveraged to make several decisions, be it in target marketing or security. With the recent development of intelligent systems a sub module to efficiently capture ethnicity would be useful in several use cases. Several attempts to identify an ideal learning model to represent a multi-ethnic dataset have been recorded. A comparative study of classifiers such as support vector machines, logistic regression has been documented. Experimental results indicate that the logical classifier provides a much accurate classification than the support vector machine.

1. Introduction

The field of machine perception has worked towards establishing a standard for vision inference and recognition. Despite the multitude of complexity involved, Ethnicity identification comes naturally to humans. This Meta information can be leveraged to make several decisions, be it in target marketing or security. With the recent development of intelligent systems a sub module to efficiently capture ethnicity would be useful in several use cases.

Several traditional image processing and vision techniques have been used to capture ethnicity. However, this paper attempts to view this problem as a statistical classification problem, where the ethnic symbols represent different classes. Traditional vision problems have found home in unsupervised machine learning techniques. However the bottleneck to such deep learning models is that they require an extensive data set per class to yield accurate results.

The research focussed on strong supervised classifiers such as logistic regression and support vector machines to train ethnicity per class model**.**

2. Survey of existing models

Facial recognition is a field that has lots of heavy undergoing research activity in it. A lot of work has been done in improving the accuracy of facial recognizers to recognize identities of people from their facial images. Matthew Turk and Alex Pentland's paper titled 'Eigenfaces for Recognition'[1], talks about the development of near-real-time computer system that can locate and track a subject's head, and then recognize the person by comparing the characteristics of the face to those of known individuals. They have gone forward with the eigenface approach.

These eigenvectors can be thought of as a set of features that together characterize the variation between face images. Each individual's face can be represented exactly in terms of a linear combination of the eigenfaces [1], [10]. What they thought of is that if a multitude of face images can be reconstructed by weighted sums of a little collection of eigen pictures, an efficient way to learn and recognize the ethnicity would be to build up the characteristic features by experience over time and recognize particular faces by comparing the feature weights needed to approximately reconstruct them with the weights associated with known individuals [11]. This approach to face recognition involves the following steps:

1. Acquire an initial set of face images (the training set).
2. Calculate the eigenfaces from the training set, keeping only the *M* images that correspond to the highest eigenvalues. These *M* images define the *face space*. As new faces are experienced, the eigenfaces can be updated or recalculated.
3. Calculate the corresponding distribution in *M*-dimensional weight space for each known individual, by projecting their face images onto the 'face space'. Apart from identifying the person's identity from the image of his face, there are also algorithms that do a great job at identifying the person's gender, expression (sad or happy, etc.) age among other things.

In the paper, "Recognizing Semantic Features in Faces using Deep Learning" by Amogh Gudi [2], a more sophisticated approach is explained that used deep learning on a large dataset of images. The deep network's final softmax output layer was again altered to fit the annotation by setting the number of neurons to 5. As usual, the network was training using stochastic gradient descent for batches of 100 samples with a linearly declining learning rate of 0.0025 to 0.001 over 50 epochs. This network required 124 epochs to converge, which suggests that the facial features required to identify a person's ethnicity are not quite obvious and common.

In their paper "Gender and Ethnicity Recognition using Silhouetted Face Profiles" [3] the authors have discussed ways to identify both gender and ethnicity. They suggested that silhouetted faces have several contouring information that reveal the difference between two subjects being studied. If this face context were to be studied and mapped on a common plane, then the "shape distance" of a train and test data set would help us classify both gender and ethnicity. This approach gave necessary inspiration to come up with more face contour related features such as chin lining and chin to mouth distance [7][8]. Work of [4] states that similar facial arrangements can be discriminated by the variation in shared configuration. Their work "Coding spatial variations in faces and simple shapes: a test of two model" helped in the conception of features that the model could pick up on after training, for example there are distinct variation in eyebrow outlines among people from different ethnicities.

Paper titled "Ethnicity Identification from face images" by [5] exposed us to different approaches to feature engineering and classification techniques. It uses eigen face values to compute various facial features and then trains it on a LDA classifier. Most of our features have been inspired from this paper. We then began looking at ways to extract the desired features from the images. "Feature Points Extraction from human face" by [5] helped divine image methods by which we would finally get our data set [6].

This paper will introduce the machine learning techniques: logistic regression and support vector machines as classifiers. Necessary theoretical understanding will be depicted using mathematical functions. The various models that was used for this study is presented in section 3. The methodology adopted is discussed in section 4. Experimental results and comparison is done in section 5.

3. Machine learning models

3.1 Logistic Regression

Let's say we have a series of variable feature for our problem denoted by $[x_1, x_2, \dots, x_n]$ and they are in some way correlated with decision y such that

$$y = F(x) \tag{1}$$

Let us assume our model hypothesis as H_θ that is used to predict the closest value to y. The logistic relation of this hypothesis with the available set of variables is given by

$$H_\theta(x) = G(w_1 + w_2 x_1 + \cdots + w_n x_n) \tag{2}$$

or

$$H_\theta(x) = G(\theta' x) \tag{3}$$

The sigmoid function is chosen to represent the value of function G. The sigmoid function behaves similar to the step function. It can be represented as:

$$g(z) = \frac{1}{1+e^{-z}} \tag{4}$$

When z is equal to 0 the value of the sigmoid is 0.5. The increase in the value of z will scale the value of the function to 1. Similarly on decreasing the value of z, the function will approach 0. This is depicted in Figure 1.

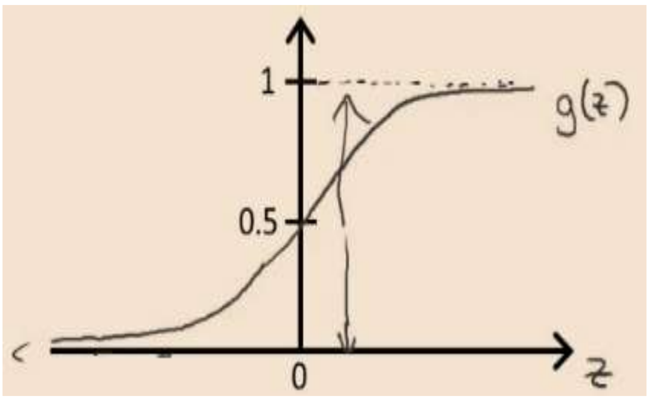

Figure 1. The Sigmoid Curve

Thus the sigmoid function assumes values 0 and 1 as asymptotes. The basic premise now would be to fit value θ to our data. Fitting the original equation, we get:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}} \tag{5}$$

Interpretation of the output of our hypothesis is questionable in respect to our classification as we can generate several values. However, a probabilistic estimate is considered on the number generated by the hypothesis on the case y=1 on input x. For example, if we have problem such as classifying if a cancer is malignant or not, If the feature vector generated in this case is $x_0 = 1$ and $x_1 = tumorSize$ then $h_\theta(x) = 0.7$ suggests that the possibility of tumor being malignant is 70 percent for the given patient. This can be represented as a conditional probability$h_\theta(x) = P(\, y = 1 \mid x;\, \theta)$. This means that the odds of y being 1 for a certain value of x and θ is equal to the value it generates. Since this is a binary classification task the probability of $y = 0$ will be the same as probability $1 - P(y = 1)$.

The probabilities computed can be compared against a threshold or a decision boundary that helps us classify our data. For the example above $h_\theta(x) = 0.5$ will give us the right decision on given predicted value of prediction. The give algorithm can work for both linear and non-linear decision boundaries.To fit the correct value of θ a cost function should be estimated. This basically helps the learning model realize how far away the parameterized hypothesis solution is from the actual solution. The most commonly used cost function is the mean squared error function.

Thus our learning algorithm will have to pay this cost while setting up / fitting the parameters. We can logically conclude that to form the best performing model we must minimize the cost function. Unfortunately, non-linearity bogs us down. The thing is the hypothesis $h_\theta(x)$ which is the sigmoidal function is non-linear. Thus when this is put in the cost function a complex function is formed which has multiple local optima. Thus the global best parameter is difficult to find using brute search. This problem is called non-convex function problem. A convex logistic cost function must be engineered to get the optimal parameter value. This is done by the use of gradient descent. To understand gradient descent geometrically we can picture a multi minima graph. To get to the minima we can differentiate the function to get the change in gradient. In other words, we can move in the opposite direction by a slope factor from a current prediction. This will help us converge to the utmost minima of the graph.

3.2 Support Vector Machines:
Support Vector Machines (SVMs) are discriminative classifiers that form a separating hyper-plane. During supervised learning, the algorithm spews out a hyperplane (optimal) that can categorize new examples. The way an SVM algorithm works is that it finds a hyperplane that maximizes the margin of the training data. The hyperplane that gives the largest minimum distance to all the training points – twice this distance is called the 'margin'.
A hyperplane can be formally defined as

$$f(x) = \beta_0 + \beta^T x \tag{6}$$

There are an infinite number of ways to represent an optimal hyperplane by scaling the β and β_0 parameters. For convention, the one chosen is

$$|\beta_0 + \beta^T x| = 1 \tag{7}$$

Where x symbolizes the closest training example to the hyperplane. The training examples that are closest are called support vectors.
Using the result of geometry, distance between x and a hyperplane (β, β_0):

$$distance = \frac{|\beta_0 + \beta^T x|}{||\beta||} \tag{8}$$

$$distance_{support\ vectors} = \frac{1}{||\beta||} \tag{9}$$

4. Methodology
Ethnicity Recognition is an application that involves multiple fields in the study of Computer Science [1]. The basic quanta of data or the basic input that the system takes in would be an image of some sort. At this stage comes the application of Image Processing. The input image has to be analyzed in the following steps:

- Ensure that the background has been removed.
- Ensure the image is of a front-facing face
- Normalize the image as required
    - Align the image and fix rotation
    - Crop the image to a standard size. In our case – 200 x 250 pixels.
- Extract all the required features from the images. Some of the many features that we extracted from the images are:

Following the image processing stage, we move on to the Machine Learning stage. Here we trained multiple models using facial image data of 3 different races / ethnicities – White, Hispanic, and Chinese. We first used Support Vector Machines (SVMs) to train on the data. Following that, we trained logistic classifiers for the 3 classes.

The final output presented to the user of the system would be the class (ethnicity prediction) that has the highest probability to be true. The user is shown which ethnicity the system predicts along with its confidence.

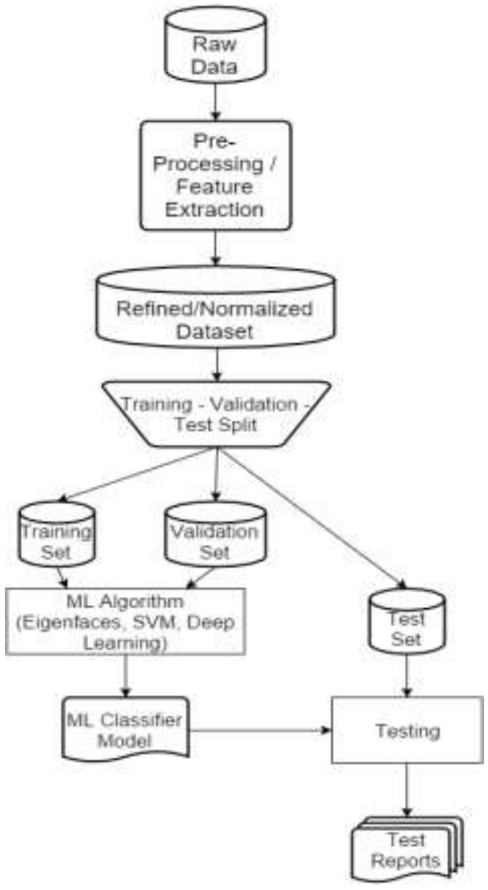
Raw Data
Pre-Processing / Feature Extraction
Refined/Normalized Dataset
Training - Validation - Test Split
Training Set
Validation Set
ML Algorithm (Eigenfaces, SVM, Deep Learning)
Test Set
ML Classifier Model
Testing
Test Reports

Figure 2. Model training flow diagram

Figure 2 shows the overall flow of the Ethnicity Recognition training phase. Following are the steps of action taken to train the various models that are to be used when classifying a given input image.

- Raw data is acquired from the Internet. For the Chinese faces, we used the CUHK Face Sketch Database [9] provided by The Chinese University of Hong Kong. For the White faces, we used the KDEF Database and for the Hispanic faces we used data from the FEI Face Database.
- The preprocessing state follows the data acquisition phase. Here the images are cropped to 200 by 250 pixels. The faces are detected and aligned so that they are straight and the center of the face coincides with the center of the image.
- Feature extraction is performed on the images. In total we extracted around 86 feature points in each image. Some of the feature points include the position of eye balls, position of nose, corners of mouth, center of mouth, etc.
- From the feature points extracted in the previous step, we calculate the necessary features required by our model. The features that we used in training all the models are:
  - Size of chin
  - Distance between chin and mouth
  - Distance between eye balls

- Distance between inner corner of eyes
- Distance between outer corner of eyes
- Distance between inner corner of eyebrows
- Distance between outer corner of eyebrows
- Size of mouth
- Size of nose
- Distance between nose and mouth.

- The resulting dataset of features is then split into training and test sets. We took an 80-20 split.
- Using the training set, we trained the various models. We tried three different classifiers:
  - Support Vector Machines (SVMs)
  - Logistic Classifiers
- The trained classifier is then evaluated on the test sets to give us a figure of its accuracy.

This trained classifier is what is used by our web application to predict the ethnicity of the input image.

5. Experiments, Observations and Discussion

Table 1 shows the total number of images for each class (ethnicity) that we used for this study. The dataset was divided into training and test sets.

Table 1. The ethnicity dataset statistics

	No. of Images	**Train Set**	**Test Set**
Chinese	188	159	29
White	110	95	15
Hispanic	48	41	7
TOTAL	346	295	51

Each of the 346 input images have been cropped aligned and normalized before extracting the feature points and training the models based on features derived from the feature points. The training set consists of a total of 295 images and the test set consists of a total of 51 images.

Table 2. Accuracy of various models trained to identify Chinese faces

Classifier / Data Sets	**SVM**	**Logistic Classifier**
Chinese Test Set	100%	96.55%
White Test Set	66.67%	86.67%
Hispanic Test Set	71.42%	85.71%

Table 2 clearly shows that the SVM classifier worked exceptionally well in identifying the test images of Chinese people. In fact, it correctly identified *all* the images in the Chinese test set. We attribute this to coincidence as there were just 29 images in the test set – a pretty small number to get a reasonable value of accuracy. Meanwhile, it only correctly predicted 67% of Whites as Non-Chinese and 71% of Hispanics as Non-Chinese.

In the case of the logistic classifier, it is observed that it is pretty balanced in its predictions. It gives a realistic 96% accuracy on the Chinese test set while at the same time correctly predicts 86% of Whites and 85% of Hispanics as Non-Chinese. The advantage of logistic regression is that it can be trained to give accurate results even when the training set consists of a relatively smaller number of examples than that required for SVMs.

Table 3. Accuracy of various models trained to identify Hispanic faces

Classifier / Data Sets	**SVM**	**Logistic Classifier**
Chinese Test Set	100%	57.14%
White Test Set	0.00%	100.00%
Hispanic Test Set	0.00%	86.67%

Table 3 shows the accuracy of various models for the Hispanic classifier. In the case of SVMs is exactly what we saw for the classifier trained on Chinese faces. The SVM classifier trained on Hispanic faces seems to correctly identify Chinese faces as Non-Hispanics. But it does not identify Hispanics as Hispanics and incorrectly identifies Whites as Hispanics. We attribute this odd result to having just 48 training examples for the SVM classifier. Like neural nets, SVMs also require a pretty decent amount of data to be able to train accurately.

The logistic classifier performs the best in this case too. It gives a decent accuracy of 86% against the Hispanic test set. It correctly predicted all White faces as Non-Hispanics but has a somewhat low accuracy of 57% against the Chinese test set.

Table 4. Accuracy of various models trained to identify Caucasian faces

Classifier / Data Sets	**SVM**	**Logistic Classifier**
Chinese Test Set	0%	100%
White Test Set	100.00%	66.67%
Hispanic Test Set	100.00%	57.14%

Table 4 tabulates the accuracy of various models trained to identify Caucasians. The SVM classifier trained on White faces seems to correctly identify Hispanic faces as Non-White. It also correctly identifies all White faces as Whites. But it did not perform well against the Chinese test set and predicted all the example images as Whites. We attribute this odd result to not having enough data. Like neural nets, SVMs also require a pretty decent amount of data to be able to train accurately.

The logistic classifier again outperforms the other two with an accuracy of 67% in correctly predicting Whites. It also predicted all the faces in the Chinese test set and 57% of faces in the Hispanic test set correctly as Non-Whites.

6. Conclusion

In this paper machine learning approach to solve the problem of ethnicity recognition was taken. Experimental results indicate that the logistic classifier is a good class classifier. Support vector machines either over fit or under fit the data. The major bottle neck of this experiment was the limited per class open source data set available. Building an extensive image data base particularly vision models for ethnicity and racial profiling will be done in the future. Further this experiment chose remote ethnicities which are relatively easy for identification. Classification among similar ethnicities (eg. Korean, Chinese) will be considered for future work.

References

[1] Xiaoguang Lu and Anil K Jain 2004 Ethnicity Identification from Face Images *Proceedings of SPIE* 5404

[2] Usman Tariq, Yuxiao Hu and Thomas S Huang 2009 Gender and Ethnicity Identification from Silhouetted face profiles, *16th IEEE International Conference* **7** 2441-2444

[3] Hua Gu, Guangda Su and Cheng Du 2003 Feature points extraction from faces, *Image and Vision Computing NZ.* **26** 154-158

[4] Serge Belongie, Jitendra Malik and Jan Puzicha 2002 Shape matching and object recognition using shape contexts , *IEEE transactions on pattern analysis and machine intelligence* **4** 509-522

[5] Gillian Rhodes and Susan Carrey 1998 Coding spatial variations in faces and simple shapes: a test of two model,*Vision Research* **38** 2307-2321

[6] Lanitis A, Taylor C J and Cootes T F 1997 Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and machine intelligence* **19** 743-756

[7] Rhodes G, Maloney L T, Turner J and Ewing L 2007 Adaptive face coding and discrimination around the average face. *Vision Research* **47** 974–989

[8] Zhao W, Chellappa R, Phillips P J and Rosenfeld A 2003 Face recognition: A literature Survey *ACM computing surveys (CSUR)* **35** 399-458

[9] The Chinese University ‘CUHK Face Sketch Database (CUFS), http://mmlab.ie. cuhk.edu.hk /datasets.html.[Online].Available:http://mmlab.ie.cuhk.edu.hk/archive/facesketch Html

[10] Ng M, Boynton G M and Fine I 2008 Face adaptation does not improve performance on search or discrimination tasks *J. Vis.* **8** 1–20

[11] Calder A J and Young A W 2005 Understanding the recognition of facial identity and facial expression *Nat. Rev. Neuroscience* **6** 641–651