Politecnico di Torino

Classification of Gender Identification dataset

Machine Learning and Pattern Recognition

Davide Ariotto Mohamed Khaled Hassan Aly Motrash

[S302489@studenti.polito.it](mailto:S302489@studenti.polito.it) [S295805@studenti.polito.it](mailto:S295805@studenti.polito.it)

July 03, 2023

Abstract

*In this paper we will illustrate the variable outcomes from different Machine Learning (ML) Algorithms using database extracted from high-level features representing face images. The results from several (ML) Algorithms are to be compared and analyzed to reach the best conclusion of the best model performance. The data base contains samples of male and female. It is imbalanced with significantly more female samples as will be explored and further analyze it. Afterwards we will apply different classification algorithms and finally calibrate the scores. Ultimately, the result will show which model is a stronger candidate for our problem.*

# 1. Introduction

The dataset consists of image embeddings which are low-dimensional representation of face images. As images dimensionality is reduces significantly to obtain a more tractable model and to keep the computational effort more reasonable. The embeddings have significantly lower dimensions than in real use-cases.

The embeddings are then reduced to a dimension of 12 of a continuous-valued vectors where the data is basically divided to (2 Classes) “Male” (label 0) and “Female” (label 1) classes.

The embeddings don’t have a physical interpretation as they represent reduced image dimensions.

The Database are imbalanced, with the training set contains more female samples than the males, however in the evaluation set we could see the other way around as it contains more male samples.

We will be analyzing a total of 2,400 samples on the training set separated between the 2 classes such that for the female class it has 1680 sample and 720 for the male class.

The evaluation set also is composed of the same number of embeddings (12) such as the training set.

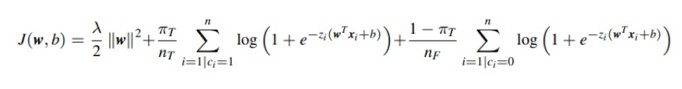
Each record in our training or testing set is composed of 12 features separated by a comma and in the last column the label of that record. Each sample could belong to 3 different age groups, however that age information is not available.

Several Machine Learning Classification Algorithms will be applied with different learning parameters to be optimized and further analyzed.

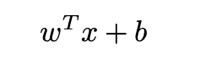
The data shall be analyzed without any further reduction in dimensionality and with the intervention of pre-processing techniques.

# 2. Logistic Regression

We start considering regularized linear logistic regression. Since classes are unbalanced, we change the objective function that we need to minimize so that costs of the different classes are re-balanced:



The model parameters are w and b. The decision rules are assumed to be linear hyperplanes orthogonal to w:



In the following figures we will attempt to make the model to learn the model parameters which are ‘w’ and ‘b’. we look for λ such that if λ is too small, we will get a solution with a higher norm but poor classification accuracy for test/unseen data.

On the other hand, if we have high value of λ then we will obtain a solution that has a small norm but won’t be able to separate well the classes.

We’ll look for the best value of the parameter with single-fold, k-fold, no PCA and PCA with m=8 and m=10.

*A picture containing text, line, diagram, screenshot

Description automatically generatedA picture containing text, screenshot, line, diagram

Description automatically generated*The results are summarized here:

*(A)* No PCA – Single Fold (b) No PCA – K-fold

*A picture containing text, line, diagram, screenshot

Description automatically generated*A picture containing text, screenshot, line, diagram

Description automatically generated

(C) PCA (m=10) – Single Fold (D) PCA (m=10)– K-fol*d*

*A picture containing text, diagram, line, plot

Description automatically generated*A picture containing text, line, diagram, plot

Description automatically generated

(E) PCA(m=8) – Single Fold (F) PCA (m=8)– K-fold

*Figure:* min DCF with different values of λ

From the previous figure, we can’t see a big difference in performance between the single fold and the k-fold analysis. However, we could notice a best value for λ that provides a min DCF.

Also, a point to be noticed is the performance differences between different applications such as in a no PCA application the performance is better either in single fold or in k-fold and by further reducing the dimensionality of the dataset it is found that the performance becomes poorer.

Based on the previous observations we could select the best value for the hyper-parameter is λ = 10-4. Now let’s compute again the min DCF:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Single Fold | | |  | 5-fold | | |
|  | A picture containing symbol  Description automatically generated = 0.5 | A picture containing symbol  Description automatically generated = 0.1 | A picture containing symbol  Description automatically generated = 0.9 |  | A picture containing symbol  Description automatically generated = 0.5 | A picture containing symbol  Description automatically generated = 0.1 | A picture containing symbol  Description automatically generated = 0.9 |
| Z-Normalized Features – no PCA | | | | | | | |
| Log Reg (λ = , ) | 0.093 | 0.233 | 0.233 |  | 0.112 | 0.285 | 0.348 |
| Log Reg (λ = , ) | 0.105 | 0.214 | 0.283 |  | 0.121 | 0.296 | 0.368 |
| Log Reg (λ = , ) | 0.099 | 0.256 | 0.231 |  | 0.111 | 0.317 | 0.343 |
| Z- Normalized Features – PCA (m = 8) | | | | | | | |
| Log Reg (λ = , ) | 0.244 | 0.461 | 0.666 |  | 0.259 | 0.511 | 0.660 |
| Log Reg (λ = , ) | 0.246 | 0.447 | 0.680 |  | 0.263 | 0.493 | 0.679 |
| Log Reg (λ = , ) | 0.243 | 0.453 | 0.660 |  | 0.256 | 0.551 | 0.648 |
| Z-Normalized Features – PCA (m = 10) | | | | | | | |
| Log Reg (λ = , ) | 0.150 | 0.312 | 0.498 |  | 0.183 | 0.419 | 0.535 |
| Log Reg (λ = , ) | 0.169 | 0.291 | 0.533 |  | 0.181 | 0.398 | 0.555 |
| Log Reg (λ = , ) | 0.166 | 0.321 | 0.475 |  | 0.185 | 0.470 | 0.514 |

*Figure 5:* min DCF computed using the estimated hyper-parameter.

From the previous results we could see a superiority in performance using application with a prior of 0.5 over the other applications. Also, It is noted that a no PCA version of the application gives better min-DCF value over all as we use a dimensionality reduction of m =10 the performance decreases slightly and when using further dimensionality reduction of m = 8 the performance is worse and that makes sense as we predicted. Also, rebalancing the cost of different classes( ) has no significant impact on the min-DCF.

3.4 Gaussian Mixture Models

In the GMM and as we know from the dataset analysis that the data consists of 3-different age groups and therefore we expect a better training phase with different components as this model will further analyze the training data and could perform slightly better. We will use GMM with full covariance, full diagonal, and tied covariance. In the tied covariance model, tying takes place at class level, so different classes have distinct covariance matrices.

We’ll use the single fold and cross-validation to evaluate how good the model is on the validation set and based on that we’ll select the number of components.

A picture containing text, diagram, line, plot

Description automatically generatedA picture containing text, diagram, line, plot

Description automatically generatedA picture containing text, diagram, line, plot

Description automatically generatedResults for Single-fold validation: -

(A) No PCA- Full Covariance (B) PCA (m=8)- Full Covariance (C) PCA (m =10)- Full Covariance

A picture containing text, diagram, line, plot

Description automatically generatedA picture containing text, diagram, line, plot

Description automatically generated

A picture containing text, diagram, line, screenshot

Description automatically generated

A picture containing text, screenshot, diagram, line

Description automatically generatedA picture containing text, diagram, line, plot

Description automatically generatedA picture containing text, diagram, line, screenshot

Description automatically generated (D) No PCA- Naïve Bayes (E) PCA (m=8)- Naïve Bayes (F) PCA (m =10)- Naïve Bayes

(J) No PCA- Tied Covariance (H) PCA (m=8)- Tied Covariance (I) PCA (m =10)- Tied Covariance

*Figure 12: min DCF for different priors, different models, and numbers of GMM components.*

A picture containing text, diagram, line, plot

Description automatically generatedA picture containing text, screenshot, diagram, line

Description automatically generatedA picture containing text, diagram, line, plot

Description automatically generatedApplying K-fold validation and compare the results:-

A picture containing text, screenshot, line, diagram

Description automatically generatedA picture containing text, diagram, screenshot, line

Description automatically generated (A) No PCA- Full Covariance (B) PCA (m=8)- Full Covariance (C) PCA (m =10)- Full Covariance

A picture containing text, diagram, line, screenshot

Description automatically generated

(D) No PCA- Naïve Bayes (E) PCA (m=8)- Naïve Bayes (F) PCA (m =10)- Naïve Bayes

A picture containing text, diagram, line, screenshot

Description automatically generatedA picture containing text, diagram, line, screenshot

Description automatically generatedA picture containing text, diagram, screenshot, line

Description automatically generated

(J) No PCA- Tied Covariance (H) PCA (m=8)- Tied Covariance (I) PCA (m =10)- Tied Covariance

*Figure 12: min DCF for different priors, different models, and numbers of GMM components.*

The K-fold results give us a better and more accurate perspective on the problem so we will focus on analyzing its graphs.

From the graphs we could see a future prediction on the number of components in general we could see that the best number of components is balancing between [2,16] and that is consistent with our previous prediction such that starting from nearly 3 components the performance is getting better.

And based on the previous results we selected the best number of components for the “Full-Covariance” type to be at (#Components =4)

And for the “Naïve Bayes” type to be (#Components =16) and for the “Tied Covariance” (#Components =8).

The following Table describes the min DCF with both single-fold and k-fold:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Single Fold | | |  | 5-fold | | |
|  | A picture containing symbol  Description automatically generated = 0.5 | A picture containing symbol  Description automatically generated = 0.1 | A picture containing symbol  Description automatically generated = 0.9 |  | A picture containing symbol  Description automatically generated = 0.5 | A picture containing symbol  Description automatically generated = 0.1 | A picture containing symbol  Description automatically generated = 0.9 |
| Z-Normalized Features – no PCA | | | | | | | |
| Full-Covariance (FC), 4-G | 0.075 | 0.162 | 0.165 |  | 0.071 | 0.201 | 0.204 |
| Diagonal-Covariance (NB), 16-G | 0.166 | 0.396 | 0.469 |  | 0.197 | 0.501 | 0.474 |
| Tied Full-Covariance (TC), 8-G | 0.061 | 0.189 | 0.152 |  | 0.067 | 0.232 | 0.193 |
| Z- Normalized Features – PCA (m = 8) | | | | | | | |
| Full-Covariance (FC), 4-G | 0.162 | 0.451 | 0.382 |  | 0.181 | 0.449 | 0.427 |
| Diagonal-Covariance (NB), 16-G | 0.228 | 0.497 | 0.638 |  | 0.216 | 0.581 | 0.597 |
| Tied Full-Covariance (TC), 8-G | 0.179 | 0.448 | 0.355 |  | 0.183 | 0.446 | 0.438 |
| Z-Normalized Features – PCA (m = 10) | | | | | | | |
| Full-Covariance (FC), 4-G | 0.108 | 0.241 | 0.226 |  | 0.112 | 0.257 | 0.273 |
| Diagonal-Covariance (NB), 16-G | 0.180 | 0.465 | 0.413 |  | 0.182 | 0.499 | 0.480 |
| Tied Full-Covariance (TC), 8-G | 0.109 | 0.300 | 0.258 |  | 0.121 | 0.312 | 0.294 |

From the previous table we could deduce some important information, the performance of the Naïve Bayes Type performed the worse in all the applications in every case or dimensionality reduction and that is expected as the Naïve Bayes depends on a diagonal Matrix and neglecting the correlation between the features which we know from feature analysis is highly correlated.

The Full-Covariance performed significantly better in most of the applications, and it came close in performance with the Tied-Covariance with a slight improvement for the Tied Covariance when not using Dimensionality reduction methods.

By analyzing different applications, it is clear that using an application with prior = 0.5 gives the best results at all.