REPORT MACHINE LEARNING AND PATTERN RECOGNITION

Fingerprint spoofing detection

Alessia Intini
s309895
Politecnico di Torino
2023/24

Contents

MLPR Project:
Fingerprint spoofing detection

The goal of project is to perform a binary classification on fingerprint spoofing detection, that is to identify genuine vs counterfeit fingerprint images. The dataset consists of labeled samples corresponding to the genuine (True, label 1) class and the fake (False, label 0) class, the data is 6-dimensional.

1 Dataset Analysis

1.1 Training and evaluation sets

The datasets provided contain 6000 samples; the first 6 values in each row represent the features, while the last value is the label. Specifically they are:

- Training Set: 2990 samples beloging to the Fake class (label 0) and 3010 samples belonging to the Genuine class (label 1)
- Evaluation Set: 3010 samples beloging to the Fake class (label 0) and 2990 samples belonging to the Genuine class (label 1)

We will use the Training Set to perform all the analysis, during this phase the dataset is divided to use about 60% of it as the training dataset and the remaining 40% for validation. After this step, the most promising models were chosen and the evaluation dataset was used to evaluate them and make the final considerations.

1.2 Features analysis

We can start to analize all features, here is the graph of each feature, through these histograms it is shown how each feature is distributed:

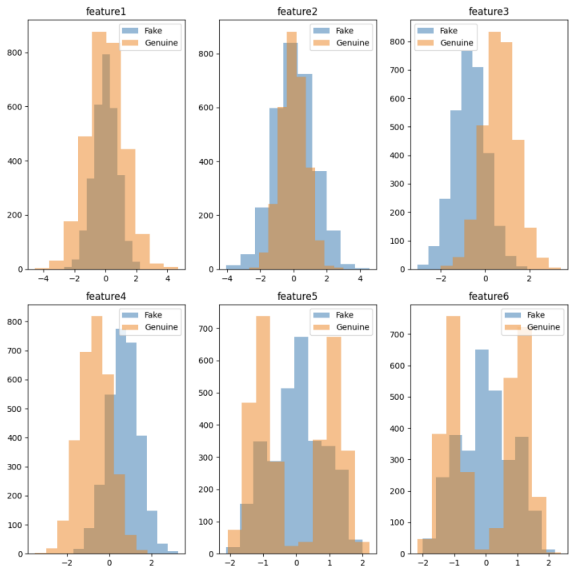

Figure 1: All features

By plotting a histogram for each feature, separately for the Fake and Genuine classes, it is possible

to show whether the samples follow a Gaussian distribution and to what extent. Thus, we can say that some features follow a Gaussian distribution. Furthermore, it can be seen that features 4 and 5 may be the most discriminating features based on their ability to separate the data of the two classes.

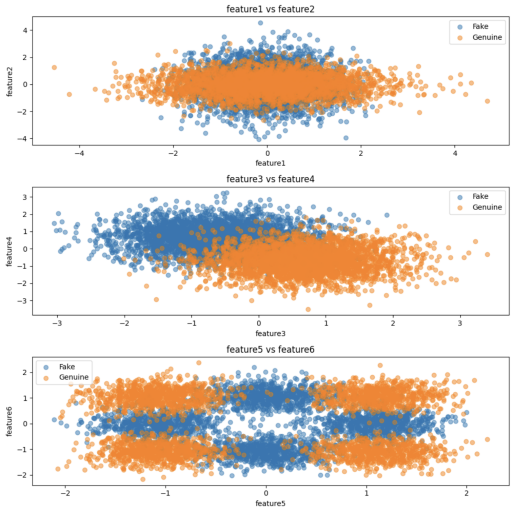

Figure 2: Distribution of feature pairs

Feature	Media	Varianza
1	0.00170711	1.00134304
2	0.00503903	0.9983527
3	-0.00560753	1.0024818
4	0.00109537	0.99029389
5	-0.00700025	1.00119747
6	0.00910515	0.99722374

Figure 3: Mean and variance for each features

After making these calculations we can conclude that the data are already centered since all features have mean close to zero.
Analyzing the pairwise averages and variances of the features, we can make some observations:

- Feature 1 and Feature 2: These two features have averages very close to zero, indicating that their values are similarly distributed around zero. However, the variance of Feature 2 is slightly lower than that of Feature 1, indicating that the values of Feature 2 are slightly less dispersed than those of Feature 1.
- Feature 3 and Feature 4: Feature 3 has a negative mean, while Feature 4 has a positive mean. This might indicate that the values of Feature 3 tend to be lower than those of Feature 4. Also, the variance of Feature 3 is slightly higher than that of Feature 4, indicating that the values of Feature 3 are more dispersed than those of Feature 4.
- Feature 5 and Feature 6: Feature 5 has a negative mean, while Feature 6 has a positive mean. This might indicate that the values of Feature 5 tend to be lower than those of Feature 6. Also, the variance of Feature 5 is slightly higher than that of Feature 6, indicating that the values of Feature 5 are more dispersed than those of Feature 6.

2 Dimensionality reduction

Dimensionality reduction is useful in this context to compute a mapping from an n-dimensional feature space to an m-dimensional space; it is applied because it can compress information, remove noise and simplify classification.

2.1 PCA

PCA is an unsupervised dimensionality reduction technique that, given a centered dataset $X = \{x_1, \ldots, x_k\}$, it aims to find the subspace of R^n that allows to preserve most of the information, i.e. the directions with the highest variance. We start by calculating the covariance matrix:

$$\mathbf{C} = \frac{1}{K} \sum_i (x_i - \bar{x})(x_i - \bar{x})^T \tag{1}$$

After that we can compute the eigen-decomposition of $\mathbf{C} = \mathbf{U}\Sigma\mathbf{U}^T$, where $\mathbf{U}$ is the matrix of eigenvectors and Σ is the diagonal matrix of eigenvalues. Now we can project the data into the new subspace, it is spanned by the m columns of U corresponding to the m values of the highest eigenvalues.

$$\mathbf{y_i} = \mathbf{P^T}(x_i - \bar{x}) \tag{2}$$

P is the matrix corresponding to the m columns of U associated with the m highest eigenvalues of C. At this point to choose which is the best value of m we need to check how much total variance in the data we are able to retain using different values of m.

To do this evaluation, it was decided to represent on a graph the variance of the data as m increases, i.e., we exploited that each eigenvalue corresponds to the variance along the corresponding axis, and the eigenvalues are the elements of the diagonal of the matrix Σ . The percentage will be calculated as the ratio of the sum of the first m eigenvalues to the sum of all eigenvalues.

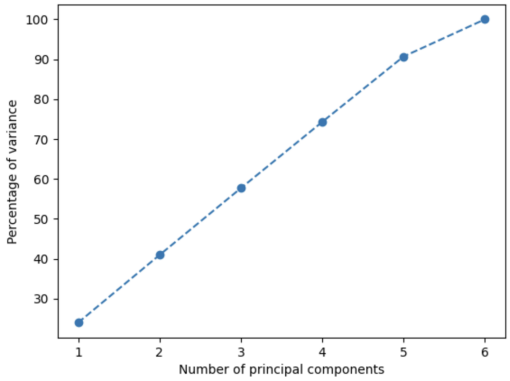

Figure 4: Variance of data for each m values