Facial Recognition of Labeled Faces in the Wild using Support Vector Machine and Principal Component Analysis

Aldrin C. Racusa
Department of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Manila
Manila, Philippines
acracusa@up.edu.ph

Lorenz Timothy Barco Ranera
Department of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Manila
Manila, Philippines
lbranera@up.edu.ph

***Index Terms*—Support Vector Machine, Facial Recognition, Principal Component Analysis, Eigenfaces, Support Vector, Machine Learning**

I. Introduction

Facial recognition is a biometric software application capable of uniquely identifying or verifying a person by comparing and analyzing patterns based on the persons facial contours. [1]

Support Vector Machine (SVM) is a popular supervised machine learning technique which is used to classification of different categories by determining the plane that would divide the labels of the different data in a space. Principal Component Analysis (PCA) is another well-known machine learning technique which is used to reduce the number of features of a data.

This paper aims to present a model using Principal Component Analysis for feature reduction and Support Vector Machine as the classification technique that would classify faces.

II. Dataset

The dataset that is used in this paper is *Labeled Faces in the Wild* containing 1127 images.

III. Principal Component Analysis

Principal Component Analysis is a machine learning technique where it is used to reduce the dimensionality of the features of the dataset. It generates n number of components called principal components which are variables that are linear combination of the original features. [2]

This technique is commonly used in order to easily visualize the data and train it faster since it has a lower number of features. [2]

IV. Support Vector Machine

Support Vector Machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. [4]

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N being the number of features) that distinctly classifies the data points. [5]

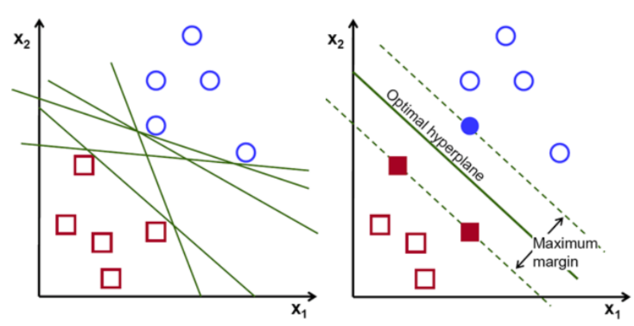

Fig. 1. Hyperplane

Some datasets are difficult to be seperated by a plane hence sometimes it is necessary that they are translated in another dimension. This technique is called *kernel trick*. There are four common kernel types used in SVM: *Linear*, *Radial Basis Function (RBF)*, *Polynomial*, and *Sigmoid* [6].

The most common parameters of SVM are *gamma* and *cost*. *Gamma* is the coefficient used in kernel types: *RBF, Polynomial* and *Sigmoid*. The gamma value influences correctness and fitness of the hyperplane in its dataset.

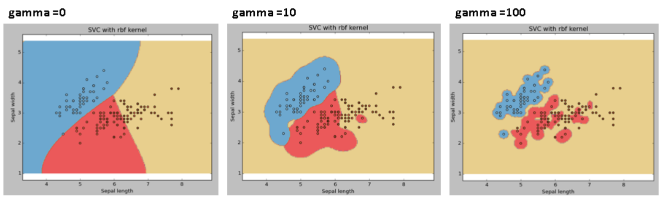

Fig. 2. Gamma parameter

Cost is the parameter that is used for the error term. It both influences the smoothness of the hyperplane and its correctness in classifying the labels.

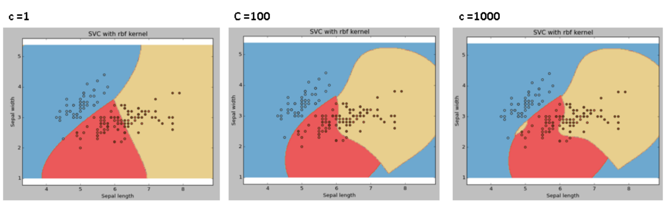

Fig. 3. Cost parameter

V. Methodology

The dataset are loaded as an array which are then scaled with mean 0 and variance of 1.

The python library *urllib* and *tarfile* was used in order to read the archived image files and the library *sci-kit learn* was used to implement SVM, and PCA

Since the dataset has 4096 features, the PCA can be used in order to reduce its number and dimensionality. Before implementing PCA in our dataset, we inspected the cumulative explained variance of our dataset. We will test later different principal components and check the effects on the accuracy of the classifier.

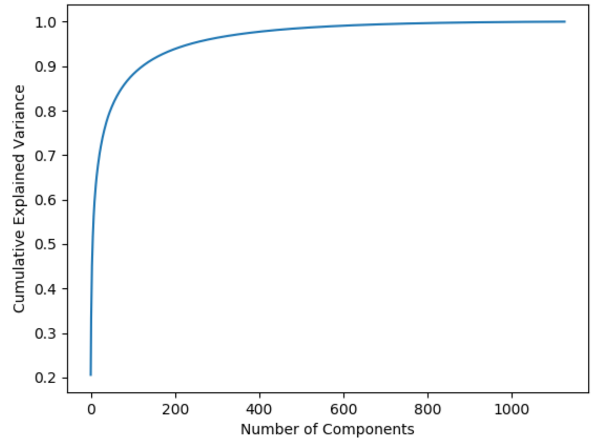

Fig. 4. Cumulative Explained Variance of our LFW Dataset

The data is normalized first using the function StandardScaler from scikit-learn before it is fitted and transformed using PCA. A new dataset was created using the values of the principal components and its features are the principal components itself. Afterwards, SVM was used in order to create model for facial recognition. The SVM implementation of sci-kit learn was also used in this study. Cross-validation was performed where 30% of the dataset are used for testing and 70% are used for training.

The parameters used where linear kernel with gamma value of 0.001.

VI. Results and Discussion

Here are the effects of changing the number of Principal Components in the accuracy of the model that we have created.

TABLE I
SVM WITH KERNEL=LINEAR, GAMMA=0.001

n_components	Accuracy	Precision	F1-score
100	0.823	0.82	0.82
200	0.837	0.84	0.84
300	0.855	0.86	0.85
400	0.879	0.88	0.88
500	0.873	0.88	0.87
600	0.864	0.87	0.86
700	0.853	0.85	0.85

VII. Conclusion

It is possible to create a Facial Recognition classifier using Machine Learning specifically Support Vector Machine and the dimension reduction Principal Component Analysis technique. In this study, we are able to generate a model with accuracy of 0.879% at best with the dataset having 400 components used.

References

[1] "Machine Learning on Facial Recognition ". Retrieved in https://medium.com/datadriveninvestor/machine-learning-on-facial-recognition-b3dfba5625a7

[2] "A step by step explanation of Principal Component Analysis". Retrieved in https://towardsdatascience.com/a-step-by-step-explanation-of-principal-component-analysis-b836fb9c97e2
[3] "Support Vector Machine". Retrieved in http://www.statsoft.com/textbook/support-vector-machines
[4] "SVM wiki". Retrieved in https://en.wikipedia.org/wiki/Support-vector_machine
[5] "Support Vector Machine Introduction". Retrieved in https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
[6] S. Ray. "Understanding Support Vector Machine algorithm from examples (along with code)," 2017. Retrieved in https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

VIII. APPENDIX