**Facial Recognition Accuracy Improvement**

Matthew A. Owings, Julian R. Thomas

Auburn University at Montgomery

Machine Learning

Dr. Semih Dinc

April 21st 2021

Abstract

Facial Recognition, defined as a technology capable of identifying a person based on a dataset of images, is utilized in our problem to denote train-test split accuracies based on a variety of classifiers and parameters. These classifiers include K-Nearest-Neighbors, Random Forests, Naïve Bayes, Decision Trees, Multi-Layer Perceptron, and Support Vector Classifier. At default, the train-test split is an even 50% train 50% test split, and PCA is applied to classifiers that would benefit from having it. Other alterations to improve accuracy include removing test samples and editing the train-test split ratio. At default, the ***fetch\_lfw\_people*** dataset contains 62 people and 3023 pictures at default parameters. These pictures are not evenly split among people, for example, “George W Bush” is 530 of those 3023 pictures, while other people contain but a small fraction of pictures, for example, “Angelina Jolie” with only 20 pictures in the dataset, the default minimum for the dataset to be registered in the final results. There is no default maximum because samples with high quantity of pictures will yield higher results due to more samples to train and test with for that individual person.

How the Data is Classified and Scored

After importing the forementioned dataset fetch\_lfw\_people the data was tested through six key Machine Learning classifiers: K-Nearest-Neighbors (KNN), Random Forests, Naïve Bayes, Decision Trees, Multi Layer Perceptron (MLP), and Support Vector Classifier (SVC) alongside parameters such as PCA and classifier-specific parameters. After results are gathered between these six datasets, the classes in the dataset are then artificially removed in order to increase accuracy in order to gain a better look at the accuracy percentage and what it means.

K-Nearest-Neighbors Classifier

The KNN Classifier on the Facial Recognition dataset classifies the data by placing the samples it deems to be the same person into clusters, and based on the n\_neighbors number, will classify the sample based on how many samples it is closest to, and classify it as the same value as those closest to it.

|  |  |
| --- | --- |
| Default KNN | 23% |
| With PCA | 31% |
| With n\_neighbors=6 and weights=”distance” | 33% |

The highest KNN value is found at n\_neighbors=20 and using the parameter weight=”distance” and this is the case because we believe nearest neighbors of 20 allows the classifier to have a more than adequate mix between the samples that have a larger quantity of samples and a lower quantity of samples, 20 is a mix between underwhelming and overwhelming the number of nearest neighbors.

Random Forest Classifier

The Random Forest Classifier has a default accuracy of 34% and a max accuracy of 38% using the parameters max\_depth=20 and a class\_weight=”balanced” parameter to yield the most accurate results. The Max Depth parameters tells the classifier how many levels of Random Forest should occur. Having 20 levels yields the greatest results alongside using the parameter weight=”balanced”

|  |  |
| --- | --- |
| Default Random Forest | 34% |
| With max\_depth=20 | 30% |
| With weight=”balanced” | 38% |

Although adding a max\_depth=20 to the original classifier reduces the accuracy by 4%, adding weight=”balanced” in addition to that max\_depth value boosts the accuracy to 4% higher than the original.

Naïve Bayes Classifier

The Naïve Bayes Classifier yields an extremely low 24% accuracy.

|  |  |
| --- | --- |
| Default Gaussian Naïve Bayes | 24% |

Decision Trees Classifier

The Decision Trees classifier yields a pathetic accuracy of 11% with default parameters.

|  |  |
| --- | --- |
| Default Decision Trees | 11% |
| With max\_depth=15 | 12% |

Naïve Bayes and Decision Trees have limited applicable parameters for our specific dataset alongside very poor default accuracies, while attempts to increase the accuracy would yield only a one or two percent increase in accuracy, at best.

Multi Layer Perceptron (Neural Network)

The MLP Classifier yielded results of 23% with default Parameters. The Default MVP accuracy is extremely low, however is tripled after applying PCA. The accuracy was furthered increased after experimenting with various parameters such as ‘solver’, ‘hidden\_layer\_size’, and ‘alpha’ which boosted the accuracy to 36%.

|  |  |
| --- | --- |
| Default MLP | 5% |
| With PCA | 13% |
| With solver=”adam” and alpha=1e-5 and hidden\_layer\_size=(26,8) | 36% |

Support Vector Classifier

The LinearSVC yields the highest results from all other tested classifiers and is the classifier we spent the most time attempting to improve, due to the level of accuracy.

|  |  |
| --- | --- |
| Default LinearSVC | 49% |
| With class\_weight=”balanced” | 48% |
| With max\_iter=125 | 52% |

As LinearSVC yielded the highest default accuracy, it became important to spent time on this classifier in an attempt raise the accuracy even further than 49%. This was accomplished through a class\_weight=”balanced”, which lowered the default accuracy by 1%, however when used in addition to max\_iter=125 it boosted the accuracy up 3% from the default.

The Results

After experimenting with the various parameters between all six classifiers, SVC yielded the highest result at 52% accuracy when given the parameters of a class weight of balanced and a max iteration of 125. SVC is a strong classifier for this type of facial recognition as SVC is designed to have flexibility with a large amount of samples between multiple classes. In this example, classes being each individual person and samples being how many pictures that person has within the full dataset.

Artificially Raising the Accuracy

The main method of artificially raising the accuracy of the dataset ‘fetch\_lfw\_people’ was to remove classes with a small, limited amount of samples. For example, a setting a minimal sample variable to ‘30’ would remove any person that had less than 30 pictures of themselves in the dataset. For example, this would mean “Carlos Menem” who has 21 pictures of themselves would not be present in the calculations to determine accuracy, however “Serena Willaims” who has 52 pictures of themselves in the dataset would remain, as the minimum number to remain would be 30. The natural default for the dataset was 20 minimum samples, with all above values being tuned to that default, therefore the accuracies taken below begin at 30 minimum samples and then increase by ten.

|  |  |  |  |
| --- | --- | --- | --- |
| Minimum Samples | Classifier | Total Classes | Final Accuracy |
| 30 | KNN  Random Forest  Naïve Bayes  Decision Trees  MLP  SVC | 34  People | 31%  41%  30%  19%  38%  60% |
| 40 | KNN  Random Forest  Naïve Bayes  Decision Trees  MLP  SVC | 19  People | 42%  54%  37%  26%  14%  67% |
| 50 | KNN  Random Forest  Naïve Bayes  Decision Trees  **MLP\*\***  SVC | 12  People | 40%  56%  40%  25%  **63%**  71% |
| 60 | **KNN\*\***  Random Forest  **Naïve Bayes\*\***  Decision Trees  MLP  SVC | 8  People | **48%**  58%  **45%**  36%  13%  79% |
| 70 | KNN  Random Forest  Naïve Bayes  **Decision Trees\*\***  MLP  **SVC\*\*** | 7  People | 45%  53%  40%  **47%**  14%  **84%** |
| 90 | KNN  **Random Forest\*\***  Naïve Bayes  Decision Trees  MLP  SVC | 5  People | 44%  **62%**  38%  37%  20%  81% |

**\*\*** Bolded is the highest accuracy percentage for that classifier

Examining the Results

When altering the dataset by removing samples that do not contain a specific number of pictures for that person (minimum samples, above table) the accuracy begins to increase. This is because the fewer pictures an individual person has, the lower their overall accuracy will be due to a limited amount of training and testing data. You see the greatest values around the minimum sample values of 60, 70, and 80 because only the people with the greatest number of pictures remain at those numbers.

Train Test Split Manipulation

|  |  |  |  |
| --- | --- | --- | --- |
| Train/Test Split | Minimum Samples | Classifier | Accuracy |
| 25 / 75  25 / 75  25 / 75  25 / 75  25 / 75  25 / 75 | 60  70  60  70  50  70 | KNN  Random Forest  Naïve Bayes  Decision Trees  MLP  SVC | 39%  49%  34%  32%  72%  64% |

\*\* When only training 25% of the data, and testing 75%, the accuracy values go lower than the default data above which is all a 50% 50% split.

|  |  |  |  |
| --- | --- | --- | --- |
| Train/Test Split | Minimum Samples | Classifier | Accuracy |
| 75 / 25  75 / 25  75 / 25  75 / 25  75 / 25  75 / 25 | 60  80  60  70  50  70 | KNN  Random Forest  Naïve Bayes  Decision Trees  MLP  SVC | 48%  62%  45%  47%  63%  84% |

|  |  |  |  |
| --- | --- | --- | --- |
| Train/Test Split | Minimum Samples | Classifier | Accuracy |
| 90 / 10  90 / 10  90 / 10  90 / 10  90 / 10  90 / 10 | 70  60  50  80  60  70 | KNN  Random Forest  Naïve Bayes  Decision Trees  MLP  SVC | 52%  68%  53%  52%  51%  89% |

References

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011. <https://scikit-learn.org/stable/index.html>

Muller, Andreas, & Guido, Sarah. (2016). *Introduction to Machine Learning with Python: A guide for Data Scientists.* O’Reilly.