Unsupervised Learning of Religious Group Facial Features

Christopher E. Mertin

Abstract

A paper published by N.O. Rule, *et. al*, explored the possibility of humans being able to discern if someone was part of a relgious group or not [2], and was able to achieve 55% accuracy. This paper explores the use of unsupervised learning techniques and eigenfaces to perform the same task, with the clustering algorithms obtaining up to 59.3% labeling accuracy on the clusters, and eigenfaces obtaining upwards of 80% accuracy on unseen data.

Introduction

A paper by N.O. Rule, *et. al*, explored the possibility of being able to identify someone by their religious affiliation based on facial features [2]. In the aforementioned study, a group of volunteers were given photos of people who had Mormon heritage and those who didn't, in an attempt to see if the volunteers could separate them into two groups. The results were that, on average, a person was able to identify who was Mormon or not with 55% accuracy, which was enough to be statistically significant, though not reliable.

For this project, we explored the use of unsupervised learning techniques to see if these methods can perform at the same level, or better than humans in the study.

In order to obtain photos of these two groups, the Tinder API package Pynder [5] was used. To make things simple, only female faces were looked at, though the same techniques can be used for their male counterparts. In order to differentiate from the groups, a users profile was downloaded if their profile said "LDS," or if they went to Bringham Young University. While this isn't 100% accurate, it does have an accuracy of more than 98% [3], and those false positives are considered noise in the data.

To get faces that were not considered Mormon, not only was the opposite used, but the location was changed to various major cities around the United States. These cities included Seattle, Miami, Los Angeles, New York, Denver, Dallas, Las Vegas, and Raleigh. This was chosen as Mormons are a small percentage of the population, and there is less likely to be a false positive outside of Utah in a city with a large population.

Transforming Data

The major difference between the data used in [2] and the data that was used with this project is that the images used in [2] were all taken at the same position. For accuracy, the algorithms used in this project also require the same, so we had to transform the data.

In order to do so, first a *similarity transform* had to be performed on the image. After cropping the faces into individual images, the faces had to be transformed so that they took up a similar coordinate system. The similarity transform can be defined by moving a point from the position (x, y) to the position $(x^\star, y^\star)$ by the equation

$$\begin{pmatrix} x^\star \\ y^\star \end{pmatrix} = \begin{pmatrix} \phi_x \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \phi_y \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \psi_x \\ \psi_y \end{pmatrix}$$

where ψ is the translational movement of each coordinate x and y, ϕ is the scaling factor to move the given square in the x and y, and θ is the angle of rotation. In order to keep this consistant without influencing the data, the two corners of the eyes were all mapped to the same pixels between the images to align the faces. This gave all of the faces being in the same position such that the algorithms would be learning with similar data sets.

Eigenfaces

Eigenfaces is a technique developed in 1991 which was one of the first identification softwares [4]. The idea behind it requires subtracting the "average face" from a group of images so that the outlying values can be used to identify a given face. This was done for the two groups of Mormons and non-Mormons. First, the average face needed to be calculated for each group. These can be seen in Figure 1 and Figure 2.

These average faces were obtained by averaging all of the pixels by the number of images in each group and then overlaying them one by one. As can be seen in the figures, there are slight differences between the two, but the difference themselves will be explored in the section on Clustering, as they're irrelevant for this algorithm. The algorithm is essentially Principal Component Analysis with the matrix being the images.

After subtracting the averages, Singlar Value Decomposition (SVD) can be performed to extract the top k features. In order to determine the top features to be extracted, we can look at the variance of the singular values.

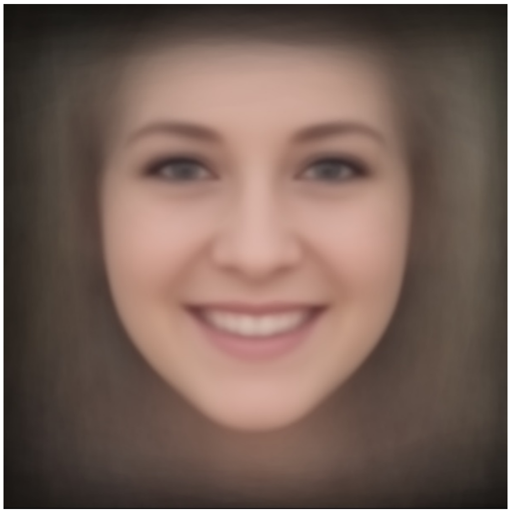

Figure 1: Average Mormon Face

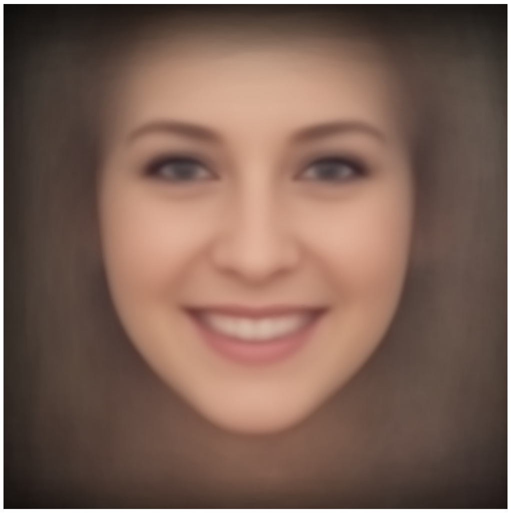

Figure 2: Average Non-Mormon Face

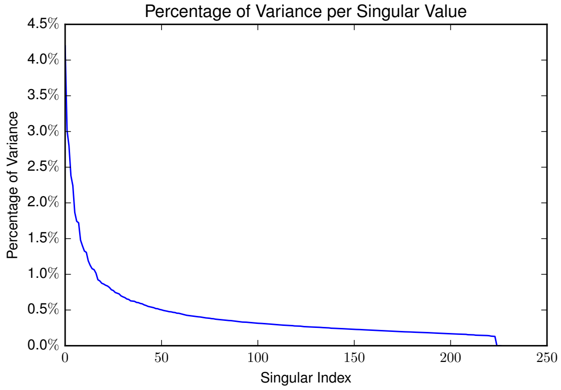

Figure 3: Variance of Singular Values from SVD

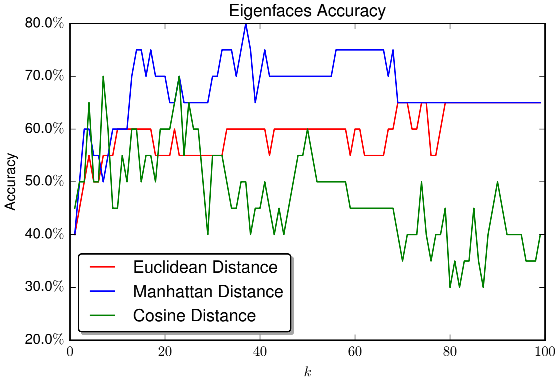

Figure 4: Accuracy of Eigenfaces with various metrics

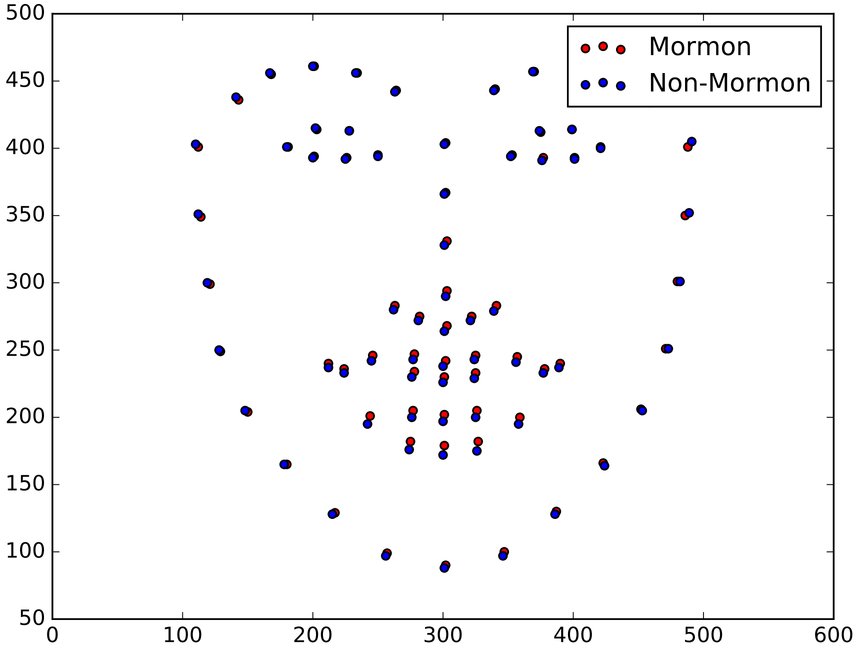

Figure 5: Average SIFT features for mormon and non-mormon

The variances can be seen in Figure 3, which shows the elbow occurs at around 25, which is a good indicator as a value of k.

This variance shows that the most important features are represented with a small value of k, relative to the entire features of 250. We can test over an unseen data set to see how the accuracy compares for various values of k, which can be seen in Figure 4. Note that the best accuracy comes about for the range $25 \leq k \leq 40$.

Clustering

With clustering, we cannot cluster the images directly, but we can cluster SIFT features contained in the images [1]. SIFT features are automatically detected features in the data. In fact, we can generate SIFT features on the average faces in Figures 1 and 2 to gain some insight on how to build our clustering models. These SIFT features can be seen in Figure 5. In this figure, each point represents 2 SIFT features in the above average images. It's important to note that there are large variances around the mouth and the nose between the two.

We can use the fact that there are these large difference to reduce the dimensionality. Instead of having 128 feature vectors, we can just reduce it down to the points that vary for the average images. On top of this, instead of looking at the x and y points, it's easy to see in Figure 5 that the x positions are essentially the same, and there's only a variance in the y, so we can drop the x values as well and just focus on the prominent y coordinates as the features.

Hierarchical Clustering

For Hierarchical Clustering, as the points are relatively close, we opted to use complete-link along with the euclidean distance. Complete-Link is useful when two groups are close together in order to separate them apart. The dendrogram of the final result can be seen in Figure 8. The percentage of labeling accuracy that this achieved is 58.4%.

KMeans

As KMeans is not a deterministic algorithm, it has to be ran multiple times. Therefore, the results of the labels can be seen in Figure 6. The best accuracy that was achieved was 59.3%.

After running KMeans, the accuracy of the labeling can be seen in Figure 7. This figure was achieved in the following way, where X is the points.

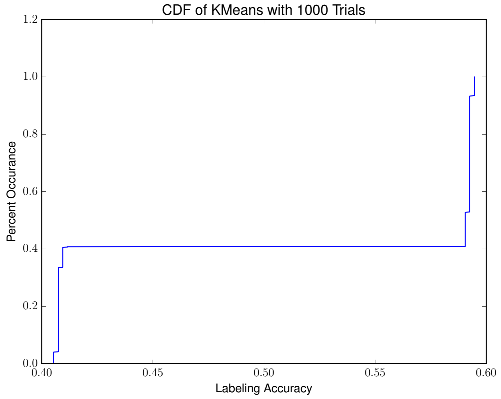

Figure 6: KMeans Cumulative Density Function

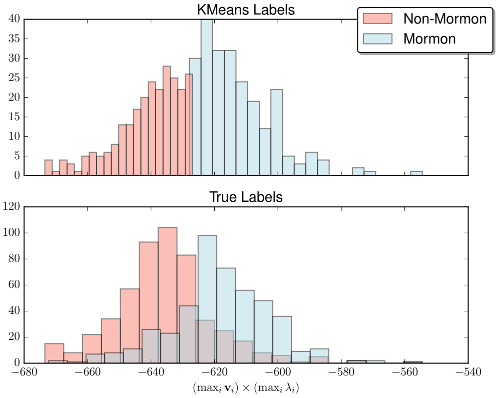

Figure 7: Labeling based on largest eigenvector and eigenvalue

$$U, \Sigma, V^T = \texttt{SVD}(X)$$
$$U^\dagger = U \cdot \Sigma$$

From here we can take the column corresponding to the largest eigenvalue and eigenvector in $U^\dagger$ to get the prominent values. These are the values that are plotted in the histogram. In the top figure, those labels are from the KMeans algorithm, while the bottom figure is from the true labels. As can be seen, the labels are quite accurate and work on splitting up the data even though there isn't such a "clean separation."

Conclusion

In conclusion, the unsupervised algorithms such as hierarchical clustering and KMeans did better than humans in [2], with hierarchical clustering obtaining 58.4% accuracy and KMeans 59.3%.

The Eigenfaces algorithm performed much better with various values of k and the metric. The best metric was the Manhattan Distance which was able to achieve up to 80% labeling accuracy for $k \sim 35$. On average, the manhattan distance performed much better than either of the aforementioned clustering algorithms.

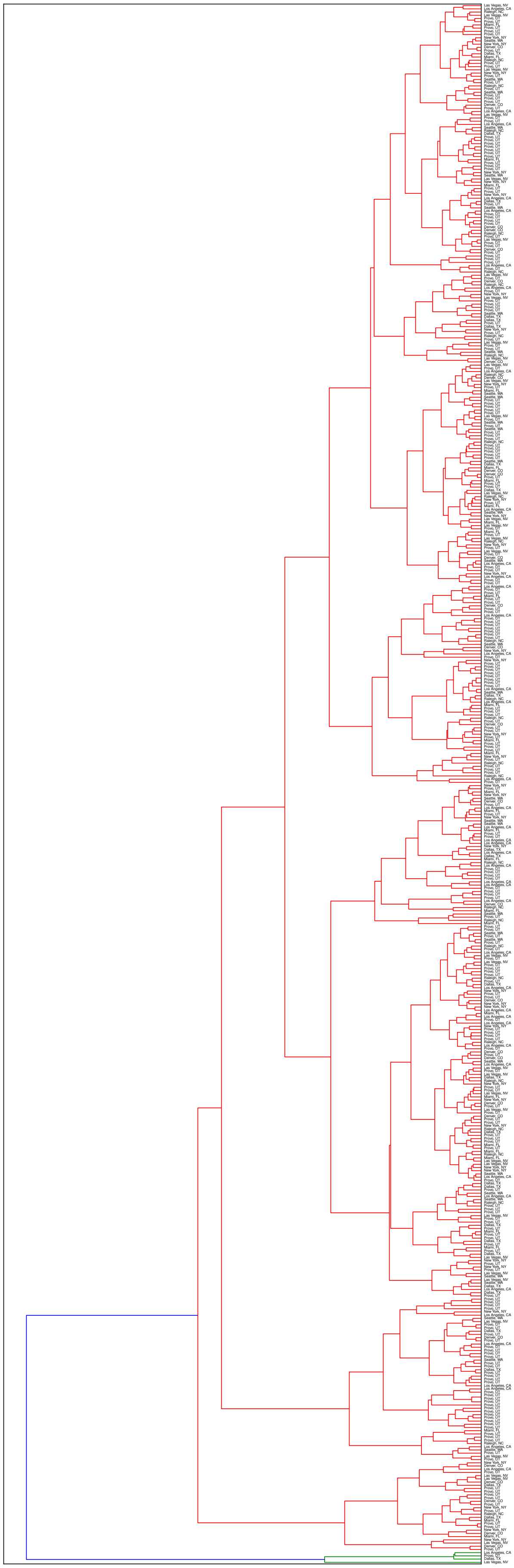

Figure 8: Dendrogram for Hierarchical Clustering

References

[1] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[2] Rule, Nicholas O. AND Garrett, James V. AND Ambady, Nalini. On the Perception of Religious Group Membership from Faces. *PLOS ONE*, 5(12):1–10, 12 2010.

[3] Tom Schad. Cody Hoffman stared as rare non-Mormon at BYU before joining Redskins. `http://www.washingtontimes.com/news/2014/aug/12/cody-hoffman-starred-rare-non-mormon-byu-joining-r/`.

[4] Turk, Matthew and Pentland, Alex. Eigenfaces for Recognition. *J. Cognitive Neuroscience*, 3(1):71–86, January 1991.

[5] Charlie Wolf. Pynder: Python Client for Tinder API. `https://github.com/charliewolf/pynder`.