AMATH 563 Homework 3

Christopher Liu

May 27, 2020

Abstract

Principal component analysis (PCA) and machine learning are two powerful tools that allow us to extract key features from data. In this assignment we will be applying the PCA to analyze 2 sets of face portraits. Using the SVD, we will determine the principal component modes and compute rank approximations of the faces. A comparison of the uncropped and cropped data will then highlight potential drawbacks when the data is not centered. By employing supervised and unsupervised learning, we will then attempt to classify and cluster the portraits. Classifying based on gender and individual face proved to be effective and we were able to successfully cluster data using unsupervised learning.

1 Introduction and Overview

In this assignment, we will be exploring the Yale Face data set which is a collection of gray scale portraits. For the first part, we will be applying PCA to 2 sets of faces. One set has centered and cropped portraits while portraits in the other set are not. Both sets contain photos of individuals under different lighting and facial expression configurations. By using the PCA, we will explore the concept of eigenfaces and how they can be used to determine the number of dominant modes needed to approximate a face. We will then compare the result of the PCA for the uncropped and cropped data and observe any differences.

For the second part, we will use methods for supervised and unsupervised learning to cluster and classify our portraits. We will compare a variety of supervised learning algorithms to classify the portraits based on their gender or identity and also vary the dimension of our feature-space. The methods used are K-nearest Neighbors, Linear Discriminant Analysis, Support Vector Machine, Naive Bayes and Classification and Regression Trees. Next we will employ unsupervised learning methods such as K-means Clustering attempt to find patterns in the faces that naturally cluster.

2 Theoretical Background

2.1 Singular Value Decomposition and Principal Component Analysis

In order to carry out PCA, we employ the SVD and the related covariance matrix to analyze the faces by taking the SVD of the mean-subtracted data [2]. Every matrix $A \in^{m\times n}$ has a singular value decomposition (SVD) given by Equation (1) [1].

$$A = U\Sigma V^* \tag{1}$$

where $U \in^{m\times m}$ is unitary, $V \in^{n\times n}$ is unitary and $\Sigma \in^{m\times n}$ is diagonal. Furthermore, σ_j, the components of Σ are uniquely determined for all A. The columns of U contain the PCA modes which constitute the orthonormal expansion basis of interest, Σ determines the strength of each projection onto the PCA modes and the columns of V^T show how X is projected onto the new basis. Additionally, once we calculate the SVD of A, we can compute low rank approximations (2).

$$A_n = \sum_{i=1}^{n} \sigma_i U_i V_i^T \tag{2}$$

where A_n is the nth rank approximation of A and $U_i,\ V_i$ are the ith columns of U and V respectively.

2.2 Machine Learning

The are 2 key goals of machine learning. The first is to obtain a low-rank feature space of the data set of interest. The second goal is to discover regression methods to both cluster and classify the data. We can attempt to obtain the low-rank feature space by using either an unsupervised learning method or by using pre-existing knowledge to explicitly to construct it [2]. In our case, the PCA modes can then be used as our low-rank feature space.

2.2.1 Supervised Learning

Supervised learning algorithms take in labeled data and attempt to find the best model that fits the input data to the output labels using regression methods. The best-fit model can then be used to predict and classify new data [2]. Below is a brief overview of the supervised learning algorithms used in this assignment,

1. K-nearest Neighbors: Given a new data point x_k, find the k nearest labeled points (or neighbors) in the feature space. The label for this new point is then determined by a majority vote of the k nearest neighbors.

2. Linear Discriminant Analysis: A decision threshold level is determined by finding suitable projection that maximizes distance between data of different classes while minimizing the distance between data of the same class. These thresholds are then used to classify new data. [1]

3. Support Vector Machine: The data is projected into a higher dimension and split into cluster using hyperplanes that optimize the largest margin between the clusters. Using the kernel trick, this method can be extended to non-linear classification [2].

4. Naive Bayes: This algorithm uses Bayes Theorem and the computation of conditional probabilities to classify a new data point based on prior probability distributions of the labeled data [2].

5. Classification and Regression Tree (CART): The CART is a hierarchical structure that attempts to create the most optimal way to split data in order to provide robust classification [2]. The algorithm scans the data over the feature-space in order to decide how to best split the data.

2.2.2 Unsupervised Learning

Unsupervised learning algorithms take in data without labels and attempt to cluster and classify the data by determining patterns in the data. Sometimes the goal itself is to uncover otherwise hidden patterns in the data [2]. We will be using the k-means clustering algorithm to carry out our unsupervised learning. The goal of the algorithm is to cluster the data by proximity to a set of k points. The algorithm determines the most optimal set of k points by continually updating the locations of the points according to the mean of the data closest to the point [2].

3 Algorithm Implementation and Development

3.1 Yale Faces: SVD Analysis

There are 2 main steps in performing the PCA on the faces. First, we need to load the data from the different sub-directories into a single matrix A which we will then decompose. It is important that any bad or corrupt data is omitted as it can skew our SVD. We then read in our images which are matrices where elements correspond to a pixel and the value of that element is its intensity in black and white. Each matrix corresponding to an image is reshaped into a column vector and A is then a matrix whose columns are each individual image. Finally, we subtract the row mean of our matrix A in order to 'center' our intensities and subtract the average face. For the second step, we compute the reduced SVD of A. Note that in order to explore the resulting data, we will have to reshape our columns of U to plot them.

3.2 Face Identification

In order to use our supervised, we first need to create our training/testing data and construct corresponding labels. We will use our principal components we found in the first section as our corresponding feature space using the SVD and only use the cropped data. For the labels, for individual classification we create a vector of values ranging from 1-38 that correspond to 1 individual. Similarly for gender classification, we use a label vector of 0s and 1s where 0 corresponds to male and 1 corresponds to female. The data is then split into training and testing sets by randomly selecting 20% (rounded to the nearest even number) of the photos for each individual for testing and using the remaining photos for training. Labels for the determined by manually inspecting the data set. 5 different train/test sets were created by varying the random seed. The overall accuracy of each supervised method was then determined by averaging the accuracy of each model in classifying the test data for all 5 sets. For unsupervised learning, the original pixel intensity data was used.

4 Computational Results

4.1 Yale Faces

4.1.1 Cropped Faces

Computing the the SVD of the yale faces, we observe that there are a few dominate modes (Figure 1). We can plot different rank approximations of the first cropped face and observe how many ranks we need to get a decent approximation (Figure 2). Note that this is the de-meaned data so it is darker than normal. Comparing to the original, we note that rank 50 gives us a decent approximation albeit a little blurry. The first 50 modes captures approximately 34.6% of the energy. We should also note that we have a few incredibly small singular vales that can be interpreted as 0 as they are smaller than machine precision 10^{-14}

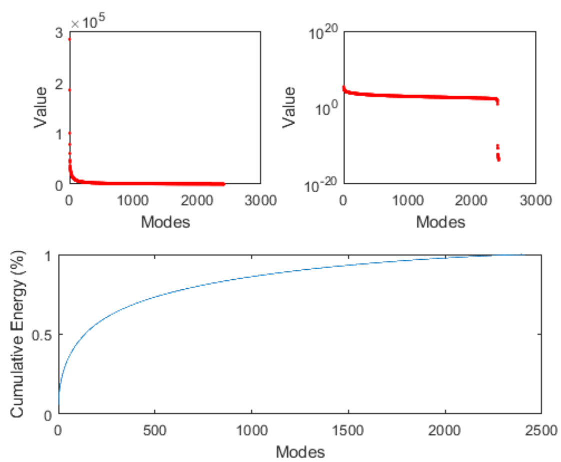


Figure 1: The singular values and cumulative energy for cropped faces

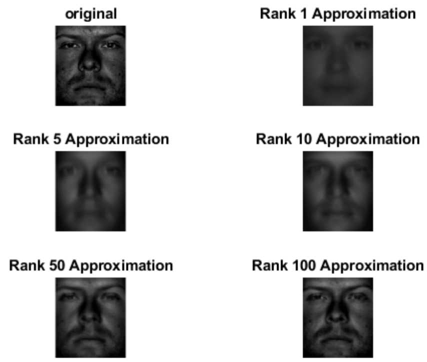


Figure 2: Rank approximations of the 1st cropped face

Plotting the first 4 columns of U, we find that this is the corresponding 'eigenface' basis of our cropped faces. So U is our new basis and Σ is the strength of the projection (or weights) of each face onto the PCA modes. Any given face, a column of A, can be represented as a linear combination of the weighted eigenfaces and the coefficients of the linear combination are determined by a column of V^T. That is, the first cropped face, A_1, is given by, $A_1 = U\Sigma V_1^T$, where V_1 is the first column of V^T

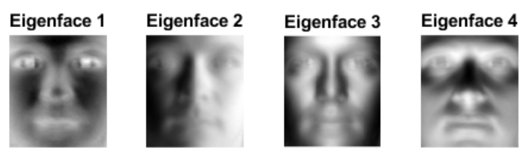


Figure 3: Eigenface basis for cropped faces

4.1.2 Uncropped Faces

Moving onto the uncropped faces, we notice that the SVD A produces similar results to the cropped case however we do not have have very large singular values that are an order larger anymore (Figure 4). Looking again at the rank approximations for the 1st uncropped faces, we note that we get a pretty good approximation at rank 50 which is about 69.5% of the energy (Figure 5). Again, we have a few very small singular values (close to 0).

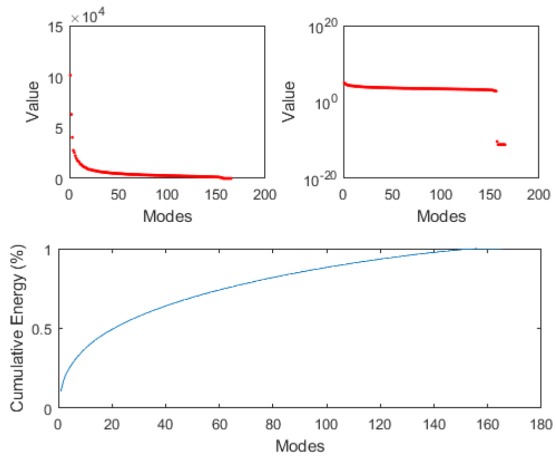


Figure 4: he singular values and cumulative energy for uncropped faces

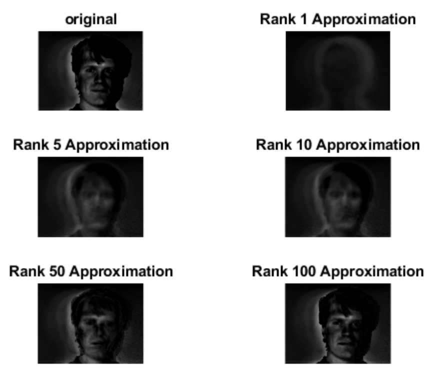


Figure 5: Rank approximations of the 1st uncropped face

Turning our attention to the columns of U (Figure 6), we notice a significant difference between the uncropped and cropped images. Because the face is not centered in the uncropped data set, we observe that our columns now show multiple faces unlike the cropped data set. This is because our basis now has to be able to re-create portraits where the face can be anywhere in the image.

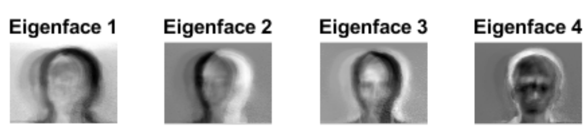


Figure 6: Eigenface basis for uncropped faces

4.2 Face Identification

For our supervised learning, we explore 5 different algorithms and compare them to see how they perform in identifying an individual face and gender. In the first section, we observed that we could get a reasonable approximation to a face by only using a small subset of the dominant PCA modes. In order to minimize compute time and avoid overfitting our model, we also explore the affect of the number of principal components on the accuracy.

4.2.1 Face Classification

Figure 7 shows the results of our face classification. We observe that LDA with 500 principal components gives use the best results with close to 100% accuracy. On the other hand, the decision tree classifier (CART) hovers around 50% accuracy and is significantly worse than the other methods. This suggests that the CART has a difficult time splitting the data up in this feature-space. Another interesting observation is that KNN (with k=5) exhibits characteristics of overfitting as we continue increasing the number of principal components. Naive bayes and SVM with an RBF kernel also have similar behavior but not to the same degree.

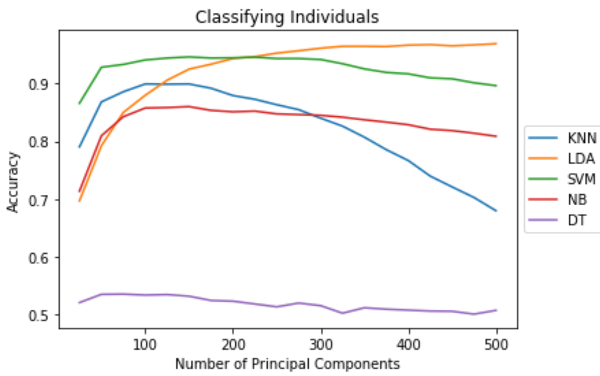


Figure 7: The singular values and cumulative energy for cropped faces

4.2.2 Gender Classification

Figure 8 shows the results of our gender classification. As expected, since there are only two categories, the overall accuracy of all the methods is significantly better for gender classification than individual classification. Once again, we observe that LDA with 500 principal components gives use the best results with close to 100% accuracy. CART gives us the worst overall performance if we consider all the entire range of principal components while Naive Bayes gives us the worst absolute performance. However, even the worst performance is still above 80% accuracy. A similar trend in overfitting can be observed for KNN with k=5, SVM with an RBF kernel and NB.

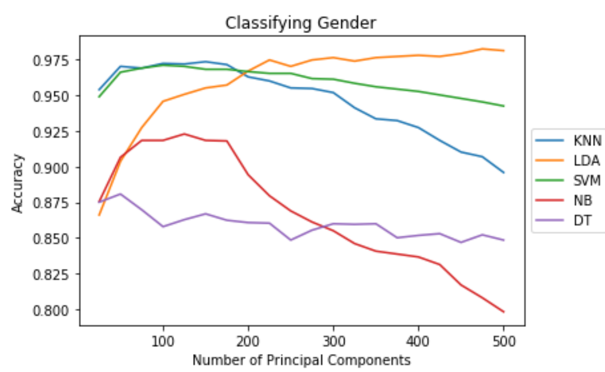
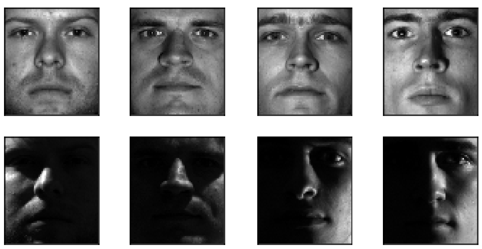


Figure 8: The singular values and cumulative energy for cropped faces

4.2.3 Unsupervised Algorithms

For our unsupervised learning, we use the k-means algorithm on our pixel intensity data as we assume that we have not chosen a feature-space for our data yet. Using $k = 2$ and 4, we observe how the algorithm decides to cluster the images. Figure 9 shows a few select portraits from the $k = 2$ and $k = 4$ case where each row of images corresponds to one cluster or label. Since our data set contains portraits of 38 individuals with 64 different lighting conditions and we are trying to cluster our portraits based on pixel intensity, we see that the algorithm has chosen to cluster them based on lighting. For the $k = 2$ case, the algorithm clusters images based on whether or not the image is light or dark. When we add additional clusters, as in the $k = 4$ case, the algorithm begins to cluster based on the lighting arrangement. The first (top) row groups portraits where the light source is on the right, the next is when the light source is on the left, the third is when there is little light and the last row is when the light source is not off-centered.



(a) k=2

(b) k=4

Figure 9: k-means clustering with 2 values of k. Each row in (a) and (b) corresponds to a different cluster or label

5 Summary and Conclusions

For the 1st part, we are able to demonstrate the power of PCA to extract the identifying characteristics of a data set (the eigenfaces). We were able to determine a new basis to project the data on and also which directions in this new basis are the dominant directions. As a result, we can use the SVD analysis to approximate images and represent the same information almost exactly but without having to store all the data. A comparison of the uncropped and cropped dataset highlights shortcomings of the SVD analysis when your data in the set isn't all centered.

By using the eigenfaces as our feature-space, we were able to successfully create classifiers for classifying an individual face and gender. We were also able to highlight the advantages of principal component analysis by determining low rank feature-space which improved accuracy. Furthermore, we explored how certain algorithms were better equipped to deal with our choice in feature-space. As for our unsupervised algorithms, we were able to highlight how the k-means algorithm can cluster the images based on lighting when a low value for k is specified.

References

[1] Jose Nathan Kutz. *Data-driven modeling & scientific computation: methods for complex systems big data.* Oxford University Press, 2013.

[2] Jose Nathan Kutz and Steven Brunton. *Data-driven science & engineering machine learning, dynamical systems, and control.* Cambridge University Press, 2019.

[3] Christopher Liu. *Homework 4: Eigenfaces Music Genre Identification*, AMATH 582, 2020

Appendix A Functions

A.1 MATLAB

- `dir` lists files and folders int he current folder
- `Y=reshape(A,X,1)` reshapes matrix A into a X by 1 vector
- `imshow(U)` displays the picture given by the intensity values in a figure
- `imread(filename)` reads in an imagine specified by the filename
- `[U,S,V]=svd(A,0)` returns the reduced singular value decomposition of A

A.2 Python

- `scipi.io.loadmat()` Loads a .mat file
- `plt.imshow(A)` Displays data as an image
- `numpy.linalg.svd(A)` Computes the SVD of a given matrix
- `np.asarray(A)` converts the input to a numpy array
- `np.random.permutation(x)` creates a random permutation from 0 to
- np.hstack([X,Y]) Stacks two arrays horizontally.
- `model.score(X,Y)` Calculates an accuracy percentage of a model given data and labels
- `KMeans(n_clusters=n, random_state=0).fit(X)` Computes the k-means clusters for n clusters and data, X.

Appendix B MATLAB and Python Code

https://github.com/chrismhl/AMATH563/blob/master/Homework3/