ISyE 6416 – Computational Statistics - Spring 2016
Project: “Big” Data Analytics
Final Report

Team Member Names: Simon Chow, Pravara Harati

Project Title: What Industry Are You?

Problem Statement

Our project is to build an image classification system. Given the headshot of a person, our classifier will determine to which industry that person is likely to belong. This could have numerous practical applications if a successful classifier is found. For example, based on photos a person posts on social media, it could be possible to determine which industries the person belongs to or is interested in. That person could then be targeted with papers or news articles concerning that industry or advertisements for products typically produced by that industry. Applications designed to match people together could reasonably guess a person’s industry from their profile picture, even if they do not choose to list it, and then use that information as part of the criteria for determining the best match.

To build this classifier, we will collect our own data using a subset of the list of the richest people in the world published by Forbes. We will then test various methods of classification, specifically categorical regression, k-means, and principle component analysis, and compare their results. Our goal is to determine which classification method is most successful in determining a person’s industry and to what extent it performs better than the others.

Data Source

Every year in March, the Forbes Magazine publishes a list of the world’s richest people. This list is an estimate of the net worth, in United States dollars, of a person by counting their assets and deducting debts. The list excludes royalty and government figures who acquire wealth from their positions. We scraped the list of the five hundred richest people from the Forbes website and also extracted the pictures of these five hundred richest people as displayed on the Forbes website. Each of these images is exactly 416 x 416 pixels in size and consists primarily of the person’s headshot. We removed people from the list with no picture shown on the website, amounting to 26 people, leaving us with a total of 474 people in our dataset. We further removed people whose images included their entire torso, had their face turned away, or were shared with additional people so that the images were more standardized. This left a final sample of 300 people in our dataset.

Forbes classifies each person into one of 18 industries based on how they obtained their wealth. These industries include construction and engineering, metals and mining, automotive, and telecom, among others. Because some of these industries have few people, we further grouped them, resulting in six total industry classifications: Energy/Manufacturing, Fashion/Media, Finance, Food/Healthcare, Real Estate/Diversified, and Technology. These groups were carefully chosen so that their component industries are generally associated with each other and so that each had about the same number of people. Specifically, each of these industries has at least 43 and at most 60 members in our sample, with Food/Healthcare having the least and Energy/Manufacturing having the most.

Table 1 shows the full set of industries along with their sample sizes and what was combined to form the larger groups.

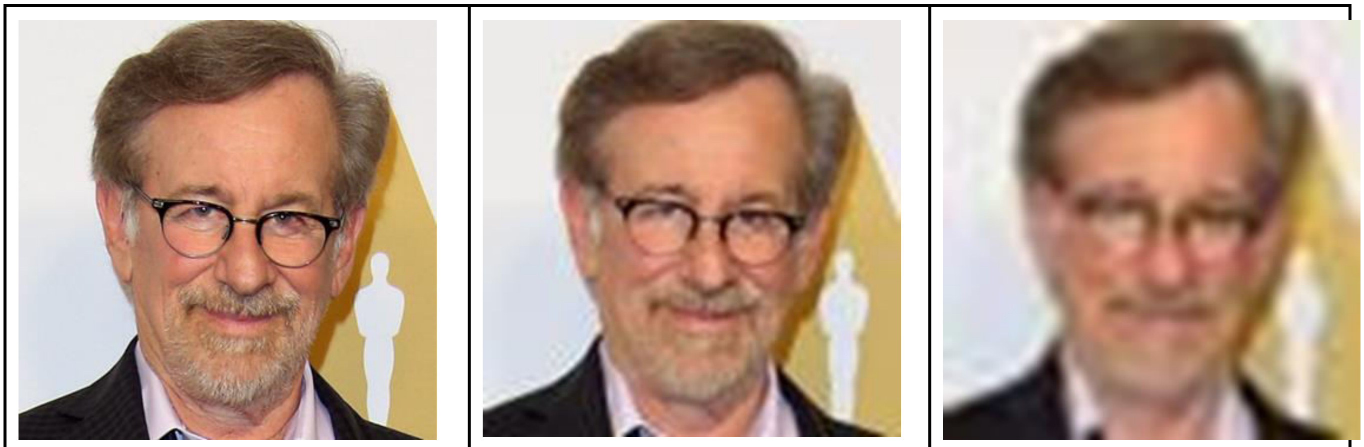
Table 1: Forbes Industry Groupings

Condensed Group	Forbes Industry	Sample Size
Food/Healthcare	Healthcare Service Food	43
Fashion/Media	Media and Entertainment Sports Fashion	50
Energy/Manufacturing	Energy Metals and Mining Construction and Engineering Manufacturing Automotive Logistics	60
Real Estate/Diversified	Diversified Real Estate	45
Technology	Technology Telecom	52
Finance	Finance and Investments Gambling	50

Methodology

Each of our classification methods followed the same general procedure. First all images were compressed to be 104x104 pixels. Images were later either downsampled or averaged further to be a reasonable size to undergo the analysis. Figure 1 shows an example of this downsampling. Images were reduced to 25% as pre-processing for multinomial regression and k-means, and were reduced to 10% for the eigenface analysis. Images were reduced to 10% because computing eigenvalues for large matrices was impossible with the larger images.

Figure 1: Downsampled Steven Spielberg



416x416 Pixel Image (100%)

104x104 Pixel Image (25%)

42x42 Pixel Image (10%)

Data was then partitioned into five groups. Each of these groups would be used once for testing during which the other groups would be used for training in a 5-fold cross-validation. This partitioning was done randomly but balanced so that a proportionate number of people in each industry would be in each partition. Each classification method would be used to fit a model on the training data. The resulting model would then be used to predict the industries for the testing data. The predicted and actual industries would be compared to calculate the error of each fold. The mean and variation of these errors were compared across the different classification methods.

The first approach we will use is to the common approach of computing eigenfaces. For each image in a given industry, we will subtract the mean face that we computed from the first step. We then aggregate all images per industry into one large matrix, patching in 4x4 grids, then vectorizing each image. Then, using this matrix, we calculate the eigenvectors/eigenvalues of the covariance of this matrix of images. After this is complete, we will sort the eigenvalues from largest to smallest, and use the top k eigenvalues to be the representative eigenfaces for the industry. In practice, people have used between 100 and 150 eigenfaces. Similar to the first approach for the classification, we divide the images into 5 groups and apply the 5-fold cross validation to select the best set of eigenfaces. To select the best eigenface we chose the industry closest to the test face in one of the three RGB dimensions, then checked that classified industry our known labels. At first we tried using the partially compressed (75% reduction) images, however encountered problems with the computer. When patching and vectorizing, this led to 10,816 eigenvalues that needed to be computed for each industry, quickly overloading the desktop computer we were using. Thus we tried using the images that had been reduced by 90% which only yielded 576 eigenvalues that needed to be computed, well within the limits of the desktop.

The multinomial regression methodology involved building a regression model that used red, green, and blue values as predictors and the industry as the response. Specifically, the input color information of the training data was used to estimate coefficients for a model that could predict the probability of being in each industry. The test data were classified as their most probable industry. Beforehand, each compressed image was divided into non-overlapping patches of size *n* pixels by *n* pixels, where *n* is a factor of 104, the original dimension. Different values of *n* were used to see how that would affect the accuracy results. For each patch, the average red, average green, and average blue value was computed, and these average values for each patch were used as the predictors, resulting in a total of $3*(104/n)^2$ predictors. Additionally, a second version of this method, with the images first converted into grayscale for a total of $(104/n)^2$ predictors, was used to see whether that would affect results.

Finally, for k-means, images were once again split into patches and the average red, green, and blue values were computed for each patch. Six random data points were selected from the training data to be used as the initial centroids. Training data were then classified into the closest clusters based on the L2 norm and cluster centroids were recalculated until 100 iterations were completed. The testing data were then classified into the cluster with the closest centroid. As k-means is an unsupervised algorithm, it is not known which industry each of the resulting clusters represents. As such, to determine the testing error, for each cluster, we counted the number of test data in each industry and declared that that cluster represented the industry which had the highest count. Test data within that cluster that did not belong to that industry was treated as misclassified. With this method, multiple clusters could be assigned the same industry.

Evaluation and Final Results

To get a baseline for what kind of results we should get with random chance, we simulated using 5-fold cross validation, what would happen if we randomly picked an industry for each image. This was done with the same testing and training data, with probabilities of picking each industry derived from the proportion of each industry that was represented in the training data. After repeating the 5-fold validation 1000 times, we had a maximum testing accuracy of 26.1% of and a minimum accuracy of 10.5%.

Using eigenfaces provided mixed results. Out of the 5-fold cross validation, the best accuracy seen was 18.87% images correctly classified, however the standard deviation was extremely high at 6.53% and the minimum accuracy was 1.54%. This indicates that our classifier was highly dependent on the data being used for testing as well as training. This is not surprising because of the 90% image loss when we compressed our images which was necessary for the computer to not run out of memory when computing the eigenvalues for the image matrices. Additionally, we were using full RGB images rather than black and white images that are commonly used for facial recognition, increasing the complexity of the problem.

For multinomial regression, the best results occurred when there were only four patches used. The overall accuracy was 23.8%, although it ranged from 20.0% to 28.3% across the five folds. Each testing group had a result that was better than what would happen by chance. Generally, increasing the number of patches resulted in a lower accuracy, counterintuitive to the idea that having more data to work with would result in better results. This may be because having too detailed data added extra noise and resulted in overfitting. Using black and white images instead of color also resulted in about the same or worse accuracy when number of patches was kept the same between the two methods. This suggests there is information conveyed about a person’s industry in the color of the image.

Interestingly, k-means had the most accurate and least variable results. The average accuracy was about 30%, although it tended to range from 28% to 35%. This may be because k-means really only output which images are most closely similar to each other and we assigned the industry ourselves based on which industry was most common, in a way taking into consideration the initial distribution of the industries. As with the multinomial regression, four patches tended to result in the best results, and increasing the number of patches typically resulted the about the same or worse results, probably for the same reasons.

We see a few areas for improvement with this project. First, as stated before, we believe our results were affected by the relatively small sample size. Thus, we could have collected more data from the internet, expanding number of data points for each industry. Second, because we used the provided pictures from Forbes, we did not have a chance to standardize the lighting, background, and facial angle that each picture was taken at. Given more time, it would be likely that the models would improve by sanitizing the input images, as such results have been seen in the literature. Another issue is that we are considering only the 500 richest people due to their industries being listed and their conveniently similar pictures being provided. Therefore the results we do have will likely not hold if we expand to consider the entire population since there will be more variety in people’s images. Finally, it is possible that our choice of condensed groups hid some variance within sub-industries. As stated before, a larger set of samples could help reveal some of the subtleties.

Work was divided as follows. Pravara collected the images from Forbes and determined industry groupings. Simon scraped the data about the billionaires and what industry they belonged to, as well as compressed the images. Pravara trained and evaluated the categorical regression and k-means algorithms and Simon trained and evaluated the eigenface classifier. Both team members worked on the proposal, final presentation, and final report.