PCA, K-Means, and Hepatitis C Virus Data

COGS 118B Final Project

Cheng Chang, Nathan Fallahi, Yana Pyryalina,
Yuan Tang, Timothy Tran, Colin Wageman

Spring 2021

1 Introduction and Motivation

It is estimated that for every case of hepatitis C that is detected and reported in the United States, at least twelve cases go unreported (Klevens, Liu, Roberts, Jiles, & Holmberg, 2014). We find the possibility to help detect more cases of HCV infections to be very important. We explore whether we can successfully find clusters of hepatitis C virus positive and negative patients using K-means clustering and principal component analysis (PCA) on data collected from blood samples. We generate the result of K-means cluster using 2 and 3 principal components, and compare the result of the K-means analysis with a Gaussian Mixture Model (GMM) analysis.

2 Background and Related Works

Hepatitis C, caused by HCV, is a liver infection that is hard to detect, but with enough computational power, it is possible to use machine learning techniques to identify cases of Hepatitis C. For example, Hoffmann et.al. used machine learning algorithms such as decision trees in order to determine whether a patient has Hepatitis C (Hoffmann, Bietenbeck, Lichtinghagen, & Klawonn, 2018). Previous work shows that using some form of machine learning algorithm over a wide range of data of various types can generate good classification results for Hepatitis C (Khan, Soh, Maenner, Thompson, & Nelson, 2019). We think it is worth investigating whether an unsupervised machine learning algorithm can utilize blood test results to detect Hepatitis C.

3 Methods

3.1 Data

The HCV Data Set used was downloaded from the UCI Machine Learning Repository (Dua & Graff, 2017) (Lichtinghagen et al., 2013). Data was loaded using Python and pandas and null values were removed. Correlation between attributes were generated with pandas and plotted with matplotlib. We decide to abandon the ”Sex” feature in the use of our algorithms due to lack of correlation.

3.2 Principal Component Analysis (PCA)

PCA helps reduce the dimensionality of the dataset by capturing and ranking the variance of the data distribution. For data visualization purposes, we only use one, two, or three principal components (PCs) and search through combinations of these three to see which yields the best result after clustering using K-Means. To utilize PCA, we first create a covariance matrix of the zero-meaned data, say Z, and the covariance matrix is denoted by $Cov = (Z'Z)/N$. We then find and sort (based on the eigenvalues) the eigenvectors of this Cov matrix to get a matrix V. Computing $U = ZV$ gives a matrix of our PCs. To obtain transformed data using these PCs we compute $C = U'Z$. Our specific use of PCA was programmed in Python using a similar structure to the algorithm in Homework 4. The first plot in Figure 1 shows the correlation between each of the data attributes and the ranked principal components.

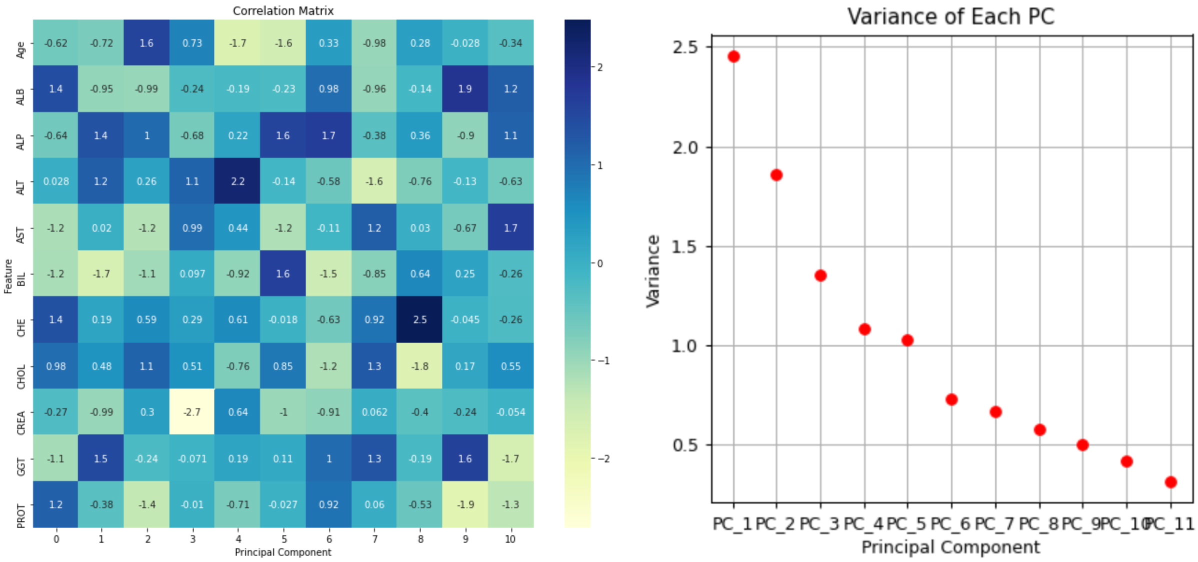

Figure 1: (a) Correlation Matrix of Attributes and PCs. (b) Variance of Each PC

3.3 K-Means Algorithm

The goal of the k-means algorithm is to identify clusters in a dataset and determine which cluster each data point belongs to. Essentially, the k-means algorithm works to minimize the objective function

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2$$

where r_{nk} is 1 if a data point x_n is closest to cluster k, and 0 otherwise. To minimize this function, the k-means algorithm iterates over the following two steps until the cluster means (μ_k) are at the center of all the points belonging to that cluster: (1) For a given μ_k, minimize J with respect to r_{nk}. (2) Minimize J with respect to μ_k, keeping r_{nk} fixed. Our specific use of the k-means algorithm was programmed in Python using a similar structure to the algorithm in Homework 2.

3.4 Supervised Learning before and after PCA

We used supervised learning to results of classification between three different datasets: raw data, data transformed into 2 principal components, and data transformed into 3 principal components. We first reformatted data to be binary - to represent healthy blood donors with 0s, and ill patients with hepatitis C with 1s. We then split the three types of datasets into 75% train set and 25% test set, making sure that the sets have both categories of 0 and 1. For this supervised comparison, we used the scikit-learn K-nearest-neighbor algorithm, and ran it for a list of n neighbors from 3 to 31. We also performed linear regression on raw data, 2 PCs, and 3 PCs. First we 75%-25% split the data using the train_test_split method from sklearn. Then we used LinearRegression() to fit the trained x and y data. Then we predicted y values using the x test data so we can compare it to the actual values. To verify our results, we found the coefficient of determination in order to find how accurate the model is.

3.5 EM after PCA

Expectation maximization (EM) for Gaussian mixtures was used to further evaluate the possibility of clustering diagnosis after PCA was performed. The algorithm was written to model EM for Gaussian mixtures in Bishop, p. 438-439. The process uses parameters μ_k the weighted mean for the cluster, Σ_k the weighted covariance for a cluster, and π_k the portion of the responsibility to the cluster. After initialization, a repetitive two step process is performed. Step one, the expectation step, $\gamma(z_{nk})$ is calculated, the responsibility of the data point to a cluster, equation 9.23 (Bishop, 2006, p. 438). In step two, the maximization step, parameters are updated, N_k, μ_k^{new}, Σ_k^{new}, and π_k^{new} from equations 9.27, 9.24, 9.25, and 9.26 respectively (Bishop, 2006, p. 439). This process is repeated until a max number of iterations has been completed. For simplicity the log likelihood convergence has been omitted. We passed the data through this algorithm with k set to 4 and the max iterations set to 120. Results were calculated using scikit-learn's (Pedregosa et al., 2011) confusion matrix, receiver operating characteristic (ROC) curve, area under curve (AUC), and accuracy metrics.

4 Results

4.1 Performance of K-Means with PCA

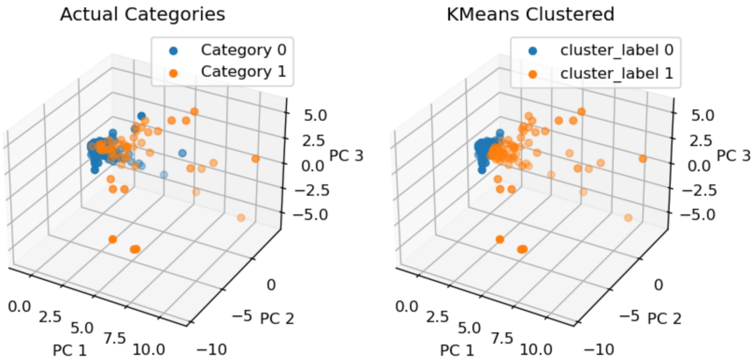

Figure 2: Actual classification with PCA vs k-means

An initial run of K-means was performed on data, prior to PCA analysis, which produced 92.87% accuracy with the parameter K set to 2, but only a sensitivity of 42.86%. After performing PCA, we can see that PC1 played the biggest role in accuracy for all three conditions, yielding the same maximum accuracy of 92.87%, with occasional exceptions. Therefore, PC1 combined with any other PC yielded the same maximum accuracy, followed by much lower performances around 66%. Another PC that made a big difference was PC4, yielding accuracy of 91.00%, also followed by much worse accuracies of around 51%. Therefore, PC1 and PC4 are usually the deciding factors to achieve best scores with K-means for this dataset. Verification of the PCA-K-means pipeline using the Scikit-learn library shows similar results.

We generate the graph of variance vs. PC. The y axis represents the variance that each PC contains. Figure 1 shows a relatively smooth downward curve, meaning that the PCs with the large index still have large amounts of information (variance). Additionally, the heatmap of the correlation matrix shows the contribution of each feature to each PC. Y-axis represents features, and X-axis stands for PCs. The absolute value of each entry is meaningful because the larger the absolute value, the more significant the contribution is made by the feature to the PC. For PC1, the features of ALB and CHE both made the most substantial and equal contribution. For PC4, CREA contributes the most.

4.2 Performance of EM

When performing EM on the PCA transformed data, we saw mixed results. With the number of clusters set to 4 we saw an accuracy of 74%, which was a greater accuracy than the K-means 43%. The area under the curve for the ROC was 0.76 which wasn't far from the best results seen in the experiment. When we cherry pick the attributes used for PCA to the 3 best attributes, we saw a large performance increase. The average accuracy of EM went to 92% and the average AUC to 0.83. Overall EM had an average false negative rate of 27.5%.

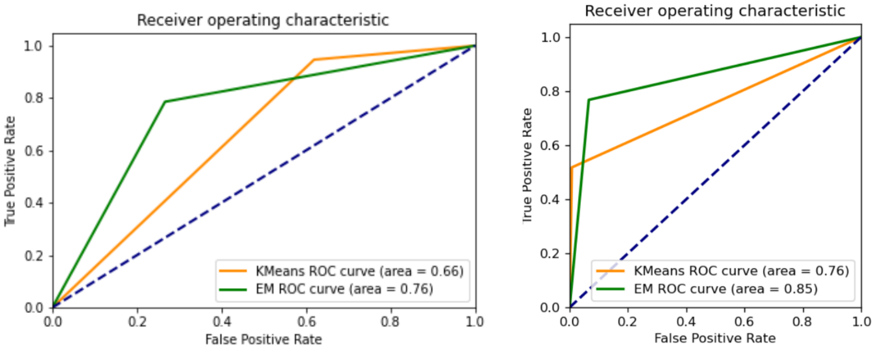

Figure 3: EM had more stability when clustering when compared with k-means.

Using the Mixture of Gaussian model from the Scikit-learn library, we were able to produce a 2-cluster classification for blood donors and HCV blood samples that achieves an average accuracy of 91.55%. From the confusion matrix generated, we also see that GMM increases the true positive and false negative categories of our results, which shows it is better at finding blood samples with HCV presence. However, GMM sometimes produces counter-intuitive results that classify less than 5 points into one cluster, showing that the results of a single trial is unstable.

4.3 Performance of Supervised Learning before and after PCA

As results of supervised learning show, PCA does not improve maximum accuracy across number of neighbors, keeping the accuracy at 97%. When comparing the mean accuracy however, using 3 PCA principal components produced the highest mean accuracy between numbers of neighbors, followed by raw data, and the post-PCA data with 2 principal components. Similarly to KNN, Linear Regression also produces higher accuracy on data with 3 PCs. We found that Linear Regression performs best on raw data (no use of PCA) with a coefficient of determination value of 61%, then after three components which is a 20% drop off, and then two components, which is an almost 25% drop off, making raw data the best model and two components the worst model.

5 Discussion

5.1 Discussion of PCA and K-Means

PCA helps us better illustrate the cluster shape of our dataset. We are able to visualize the data in two-dimensional and three-dimensional plots after 2 and 3 principal components are separately taken out to represent the data distribution. We also believe that PCA helps reduce the computation complexity of K-means and Gaussian Mixture Model, which means a reduction to both computation cost and time. Our work suggests that the first 2 and 3 principal components of PCA for this dataset are not able to provide a clear distinction between HCV and non-HCV presence blood samples for K-means and Gaussian Mixture Model. We think this shows that while PCA may provide the dimensions that best express the variance of the high dimensional data, it may still reduce the classification power of data with a relatively low number of observations.

We think that the unsatisfying performance result of our PCA and K-means pipeline is also partly due to the nature of K-means algorithm. The results of our PCA transformation generated two clusters that are irregular in their shapes, and we believe that K-means clustering does not perform well when dealing with non-spherical clusters that are close together.

5.2 Discussion of EM

Looking through the AUC numbers, we see that GMM consistently has a better balance of the two metrics. Looking at just the accuracy scores we see that both GMM and K-means have relatively consistent performance. GMM does typically outperform K-means when comparing AUC, which may be a better metric for medical diagnostic usage. Overall, the sensitivity doesn't seem high enough to adopt GMM as a final solution for clustering Hepatitis C possible patients.

The better GMM result seems to suggest that GMM is better at generating clusters with irregular shapes and close proximity. This is not surprising because, as discussed in lecture, GMM performs a graded classification that assigns possibilities of cluster assignment to each point, instead of a hard decision boundary. GMM also shapes the cluster as a Gaussian distribution, so that cluster with different variance along each dimension and those with high covariance between each dimension can be better classified by GMM.

5.3 Discussion of Supervised Learning results before and after PCA

When comparing results of KNN, it is not clear whether having the mean accuracy be higher across neighbors after PCA (with 3 PCs) is actually a considerable enough advantage in practice. On the other hand, the higher mean accuracy also allows for more flexibility between what number of neighbors is chosen (for example, in case of constraints) - because there are more numbers of neighbors as candidates for higher accuracy. We also believe that the reason KNN result is better than K-means and GMM results is the irregular decision boundary that KNN generates if an appropriate k-value is used. Overall, from both KNN and Linear Regression, we can conclude that using 3 components will lead to better accuracy in supervised learning in general, possibly because health factors would come down to three groups.

5.4 Suggested Follow-up

A larger sample with more balanced HCV-non-HCV ratio might help with clustering, because there would be a more similar number of samples in each cluster. Future work may also explore more attributes of potential patient data. For example, demographic, exercise, and medical visits may shed light on key factors for clustering HCV patients.

6 Contributions

Cheng Chang worked on verifying results of PCA, K-means, and EM algorithm made by the team with those generated using scikit-learn. Nathan Fallahi worked on our k-means and PCA implementations as well as the introduction and background sections. Yana Pyryalina worked on comparing results before and after PCA (including 2 versus 3 components) using supervised learning (KNN). Yuan Tang manually coded PCA, generated Variance and correlation maps, generated K-means Accuracy across all 1D, 2D, 3D PCA combinations, and did these in both ”Sex” and no ”Sex” conditions. Timothy Tran worked on the background as well as all the Linear Regression portions. Colin Wageman worked on data exploration and cleaning, K-means with and without PCA, EM comparisons, and follow-up work.

7 Code

`https://github.com/nashafa/cogs118bproject/releases/tag/v1.0.0`

References

Bishop, C. M. (2006). *Pattern recognition and machine learning.* springer.

Dua, D., & Graff, C. (2017). *UCI machine learning repository.* Retrieved from `http://archive.ics.uci.edu/ml`

Hoffmann, G., Bietenbeck, A., Lichtinghagen, R., & Klawonn, F. (2018). Using machine learning techniques to generate laboratory diagnostic pathways—a case study. *J Lab Precis Med*, *3*, 58.

Khan, M. A., Soh, J. E., Maenner, M., Thompson, W. W., & Nelson, N. P. (2019). A machine-learning algorithm to identify hepatitis c in health insurance claims data. *Online Journal of Public Health Informatics*, *11*(1).

Klevens, R. M., Liu, S., Roberts, H., Jiles, R. B., & Holmberg, S. D. (2014). Estimating acute viral hepatitis infections from nationally reported cases. *American journal of public health*, *104*(3), 482–487.

Lichtinghagen, R., Pietsch, D., Bantel, H., Manns, M. P., Brand, K., & Bahr, M. J. (2013). The enhanced liver fibrosis (elf) score: normal values, influence factors and proposed cut-off values. *Journal of hepatology*, *59*(2), 236–242.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.