CS528 Project 2: PCA and K Means

J.R. Powers-Luhn

October 22nd, 2018

1 Objective

The goal of this project was to explore unsupervised clustering and means of evaluating the clusters generated by the K-Means algorithm. The ability of Principal Component Analysis to improve the performance of this algorithm was also explored.

2 Preprocessing the data

A dataset was obtained with several summary statistics for fifty seven U.S. universities, mostly obtained from the Department of Education's Integrated Postsecondary Education Data System (IPEDS).

The dataset was processed in several ways before numerical analysis. Two missing values for Clemson University were obtained: the IPEDS ID number (from the IPEDS website) and the 2017 endowment balance[1]. Some information (medical and agricultural school research funding) was absent for many schools and was therefore removed from the data. None of the schools included in the dataset were historically black colleges, so the "HBC" column had zero variance and was therefore removed. Only some of the schools had Wall Street Journal rankings; this column was removed. Some columns were determined to be categorical–these were "one-hot" encoded.

For all processing in this project, the numerical attributes were mean-centered and scaled to unit variance. This was accomplished using the scikit-learn `StandardScaler` class [1]. Scaling is necessary due to the different units of the various measurements. Without this correction the variance would be skewed by the units and mean value of the individual measurements.

3 PCA

Principal component analysis (PCA) was applied to the data. PCA is a linear transformation of a dataset into a new basis set. The principal components have the following properties:

[1]Source: `https://www.clemson.edu/giving/cufoundations/documents/allocations.pdf`

- the components are orthogonal to each other, and
- the components are ordered by the amount of variance they exhibit.

This means that the dimensionality of the data can be reduced by transforming the data into the new space (where the transformed values are referred to as "scores") and throwing away all but `k` columns of the transformed data.

The transformation into the new space is performed by taking the singular value decomposition of the original data as indicated in equation 1.

$$\mathbf{U}, \mathbf{S}, \mathbf{V}^T = svd(\mathbf{X}) \tag{1}$$

In equation 1, **US** is the transformed data, **S** are the singular values, and **V** are the principal component loadings used to transform **X** into the PC space. **S** is a diagonal matrix with the property that the diagonal values are proportional to the amount of variance captured by the associated principal component. The number of principal components to capture some fraction f of the variance is shown in equation 2.

$$\sum_{i=0}^{k} \vec{s^2} = f \tag{2}$$

For the university dataset, the data in its pre-processed form had sixty two numeric dimensions. However, the vast majority of the variance was contained in the first few principal components, as shown in figure 1.

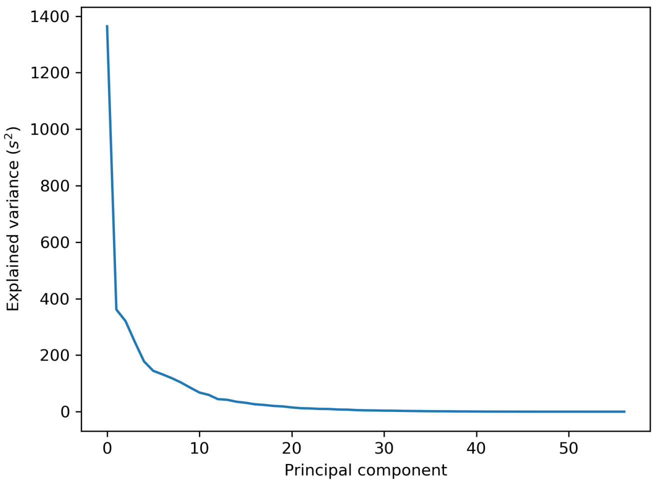

Figure 1: Variance explained by each principal component (unscaled)

In order to simplify the dataset while retaining 95% of the variance (and, therefore, the original information), the cumulative sum of the singular values squared (normalized to the sum of the square of all singular values) was calculated, and a threshold set at the first value that exceeded 0.95.

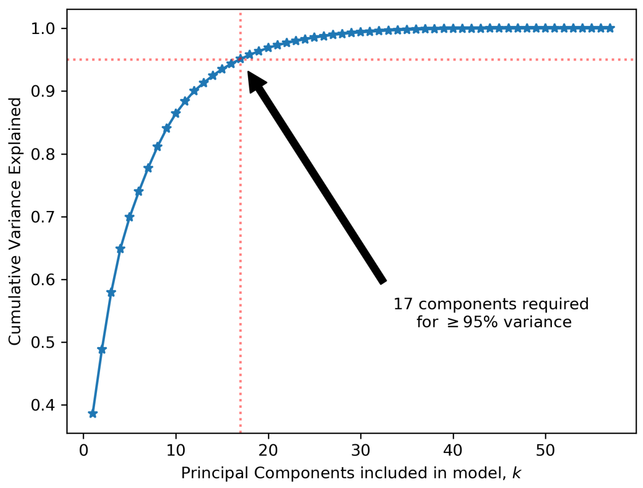

Figure 2: Cumulative variance explained by the first k principal components

As shown in figure 2, this corresponded to the first 17 components. A scatter graph of the first two components plotted against each other is shown in figure 3

4 K-means

In order to determine whether the schools could be grouped into categories, the `k-means` algorithm was employed.

1. Select k initial vectors from the data as seeds, $s_i \in \{s_0, s_1, \ldots, s_k\}$
2. Calculate the distance from each s_i to each vector
3. Assign to each vector the label associated with the seed with the minimum distance to that vector
4. Recalculate each $s_i = \frac{\sum_j x_j}{j}$ for all x_j with label i
5. Repeat 2-4 until the values of s converge

This algorithm was found to converge in less than ten iterations.

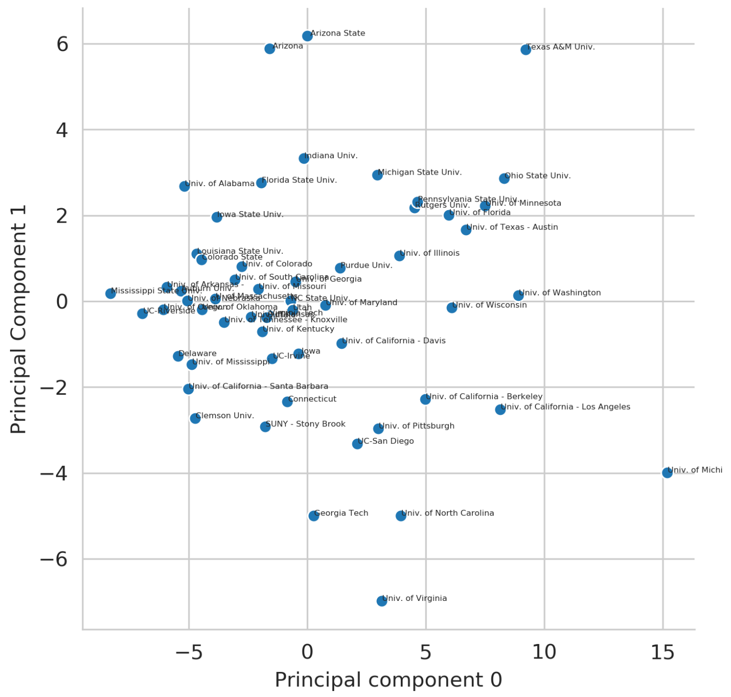

Figure 3: The first two principal components plotted against each other. Since the PC's are orthogonal to each other, no linear relationship is apparent. Note the scales of the two different axes. Since PC1 captures more variance than PC2, this axis convers a larger range.

This algorithm is not guaranteed to converge to the optimal solution since it is sensitive to the selection of the initial seeds. Because of this, the algorithm was repeated several times in order to improve the likelihood of optimal selection.

4.1 K-means++

While the `k-means` algorithm is guaranteed to converge, it is not guaranteed to converge to an optimum value. Specifically, it is subject to the selection of the initial seed vectors. In order to improve upon this, the `k-means++` algorithm was employed[2]. In this the first seed s_1 is selected at random from the data. Subsequent seeds $s_2, \dots, s_k$ are selected using a weighted probability proportional to the distance of each vector from the closest seed. In this way the initial selection of seeds avoids selecting vectors that are too close to each other. The `k-means++` algorithm therefore converges much more quickly than `k-means`, usually in a single iteration.

4.2 Cluster selection

Ideally, data clusters should exhibit both tight grouping (minimal intra-cluster distance) and wide separation (maximal inter-cluster distance). The ratio of these two values (the minimum inter-cluster distance to the maximum intra-cluster distance) is referred to as the "Dunn Index" [3]. Because the `k-means` algorithm is not guaranteed to converge to the optimal solution, the Dunn Index is subject to the random selection of the initial vector. Still, it provides a tool for evaluating the "true" number of clusters in the data.

The `k-means` algorithm was applied to the data with k ranging from 2 to 56. In each case the clusters were allowed to converge and the Dunn index was calculated. The results are shown in figure 4.

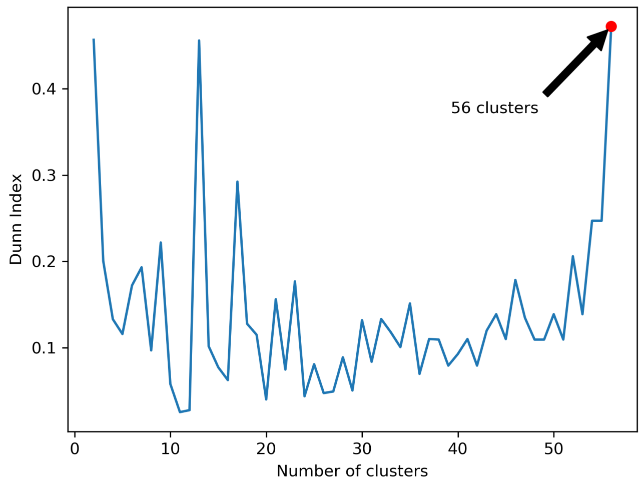

Figure 4: Dunn index for k clusters. The value oscillates but has a clear maximum.

5 PCA plus K-Means

Lorem ipsum dolor simet [4]

6 Conclusions

Lorem ipsum dolor simet

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[2] David Arthur and Sergei Vassilvitskii. “K-Means++: the Advantages of Careful Seeding”. In: *Proc ACM-SIAM symposium on discrete algorithms.* 8.3 (2007), pp. 1027–35. ISSN: 978-0-898716-24-5. DOI: `10.1145/1283383.1283494`. arXiv: `1212.1121`.

[3] J. C. Dunn†. “Well-Separated Clusters and Optimal Fuzzy Partitions”. In: *Journal of Cybernetics* 4.1 (1974), pp. 95–104. DOI: `10.1080/01969727408546059`. eprint: `https://doi.org/10.1080/01969727408546059`. URL: `https://doi.org/10.1080/01969727408546059`.

[4] K.Y. Yeung and W.L. Ruzzo. “An empirical study on principal component analysis for clustering gene expression data”. In: *Department of Computer Science and Engineering, University of Washington.[Links]* (2000). DOI: `http://dx.doi.org/10.1108/13527600610683390`. URL: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.5.8091%7B%5C&%7Drep=rep1%7B%5C&%7Dtype=pdf`.