Dimension reduction for Clustering Algorithms

By: John Bestavros (U80097673), Shivangi (U35642613), Spandana Patil (U14329137)

Introduction

Dimensionality reduction aids in addressing these issues while attempting to retain the majority of the pertinent information present in the data required to develop accurate, predictive models. It becomes more difficult to visualize the training set and subsequently work on it as the number of features increases. Sometimes, the majority of these traits are redundant since they are connected. Algorithms for dimensionality reduction are useful in this situation.

Since having unnecessary features in the data might reduce the accuracy of the models and cause your model to train based on irrelevant features, dimensionality reduction removes useless features from the data. Additionally, it reduces the need for data storage and model training time.

Clustering is essentially an unsupervised learning technique. The process of drawing references from datasets of input data without labeled replies is known as unsupervised learning. It is typically used as a method to identify the groups, generative qualities, and significant structures that are inherent in a set of instances.

The objective of clustering is to divide the population or set of data points into a number of groups so that the data points within each group are more similar to one another and different from the data points within the other groups.

Project Objective

In this project, we aim to explore the application of dimension reduction for clustering and investigate how well different dimension reduction methods can preserve the cost of k-means clustering. Clustering is conducted over different scales of data as well as different types of data. The performance of different dimensionality reduction algorithms over these datasets was compared to each other and to their reactive performance over unreduced data.

Methodology

For this project, our main aim was to compare the cost of KMeans clustering with the original Data and the reduced Data. We implemented several Dimension Reduction algorithms for this project, like Factor Analysis, Truncated Singular Value Decomposition, Johnson Lindenstrauss Lemma, and Principal Component Analysis.

As the first step in the analysis, we researched a few Datasets online. We wanted to compare our results on Datasets with varying dimensions, to have a complete understanding of which algorithm works best for which kind of Dataset. Finally, we choose three different Datasets:

1. Credit Card Information Data with 17 dimensions
2. Digits dataset, a sklearn-library dataset that has a dimensionality of 64
3. A large real-world dataset analyzing the details of house prices, with more than 9000 dimensions

After preprocessing these datasets and removing any inconsistencies, we performed the above-mentioned dimension reduction algorithms on them.

Factor analytics is a unique method of managing which data should be present in a spreadsheet by factoring the data, which is the process of reducing a large number of variables into a small number of factors. In terms of a potentially smaller number of unseen variables termed factors, it is also used to describe oscillations among the observed and associated factors. Using the factor analysis technique, all the variables' highest common variance is extracted and combined into a single score. It is a theory applied when training a machine learning model, hence data mining is closely related to it.

Implementation of FA involved performing KMO and Bartlett's tests on the datasets. The KMO measure of sampling adequacy is a test to determine whether factor analysis should be used on the given data set. The variables in the population correlation matrix are tested for sphericity using Bartlett's test in order to rule out the null hypothesis that they are uncorrelated. The next step involved plotting the eigenvalues for each of the variables to assess the number of factors to be considered for the input to the clustering algorithm. This was estimated by considering the number of variables whose eigenvalues were seen to be over one. Then the dataset was reduced to the recommended dimension using matrix rotation techniques.

As a final step, K means clustering was performed on all the datasets mentioned above. The relative WCSS or the sum of the squared distances from the centroid was plotted against the number of dimensions.

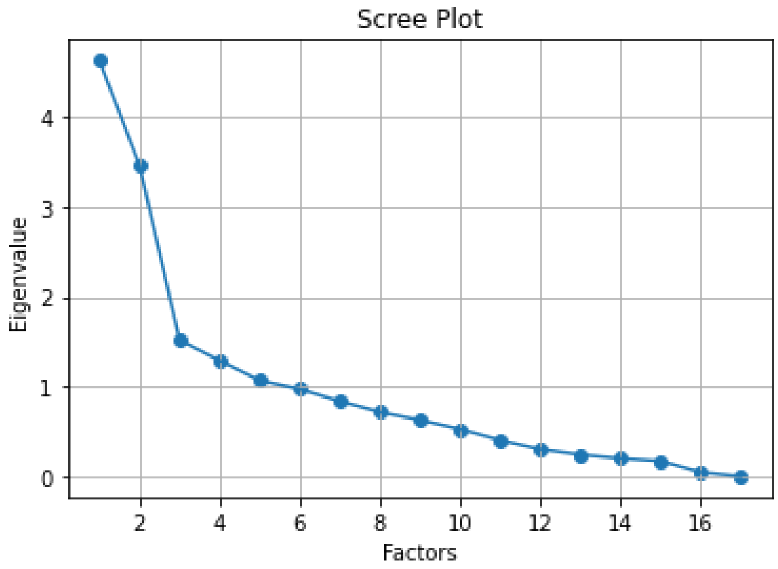
Another algorithm we implemented was Principal Component Analysis, arguably the most popular of the dimensionality reduction algorithms. Principal component analysis, or PCA, is a statistical procedure that allows one to summarize information and content from large datasets through a smaller dataset with "summary columns", making for easier visual analysis. Implementing this algorithm from scratch was not too hard, all 6 steps in PCA had libraries that made the step easier to do (mean-centering data, covariance matrix, eigenvectors/eigenvalues, rearranging, creating a subset, and transposing). From there, we applied the KMeans clustering algorithm w/o any further transformation. Perhaps one deficiency of this algorithm compared to the other ones we used was that there wasn't any real "method" to finding the ideal number of components to reduce to, leading to more trial and error than not. However, the general speed of computing PCA meant this was never a serious issue.

After PCA, we implemented Truncated SVD. It works by decomposing a matrix into three smaller matrices: the left and right matrices and the diagonal matrix of singular values. In truncated SVD, the singular values are sorted in descending order and only the top k singular values and corresponding left and right singular vectors are kept. This results in reduced-dimensional Data which captures the most important patterns between the variables. We used the elbow method, which is basically plotting the explained variance ratio against the number of components, to find the optimal number of components for each dataset. Then, truncated SVD was applied to datasets. We then performed KMeans clustering on both the original and the reduced datasets and the results were recorded.

Next, we implemented Johnson Lindenstrauss Lemma. This lemma is a mathematical theorem that states that a high-dimensional dataset can be projected into a lower-dimensional space while preserving the pairwise distances between the data points up to a small factor of $1 + \varepsilon$ with □ probability. To implement this lemma we used Python's scikit-learn's Johnson Lindenstrauss functionality to find the optimal number of components with ε = 0.5 and □ = 0.9. We then performed KMeans on both the original and the reduced data as we had done before and recorded the results.

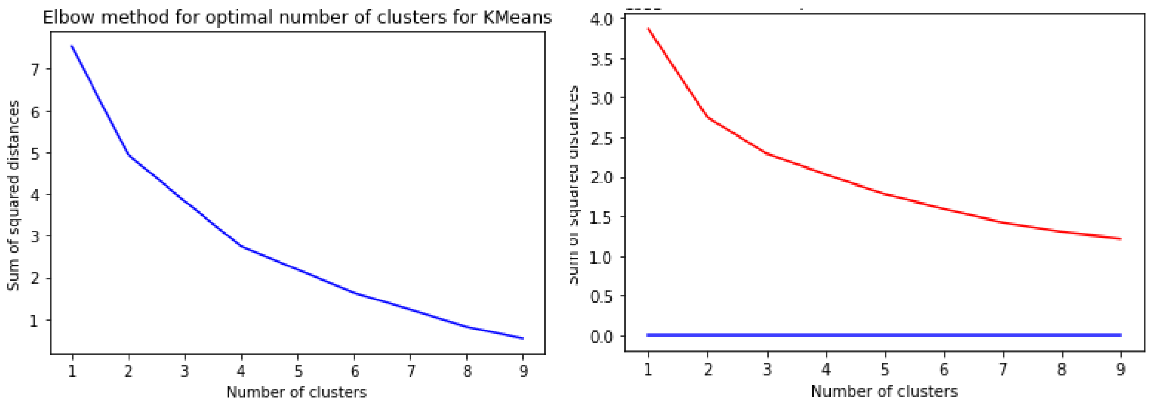
Results

Factor Analysis



Adequacy test for the small real-world dataset

The KMO model test performed over this dataset revealed that the KMO factor is approximately 0.64. This could be interpreted as a positive correlation between the performance of Factor analysis and the obtainable accuracy over k-means clustering. A value over 0.6 displays the suitability of factor analysis as a dimensionality reduction method for a specific dataset. The adequacy test performed for the small-scale dataset indicated that the optimal number of factors for the data is six.

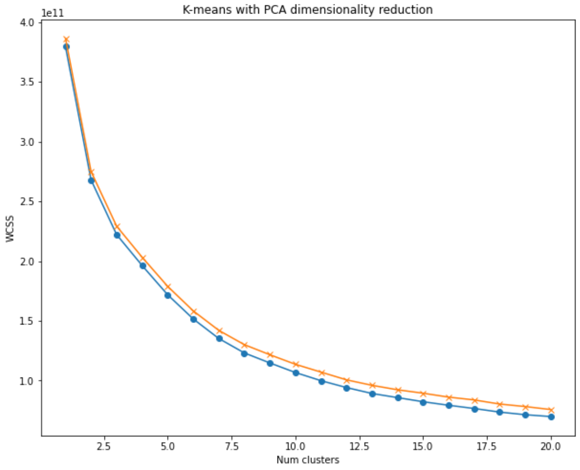


WCSS plots for K means clustering for the reduced dataset in relation to the non reduced

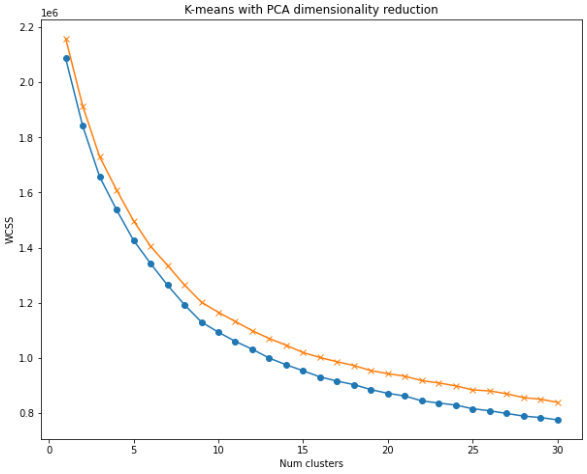
The performance of K means clustering to find the optimal clusters using WCSS shows us that that their values are extremes for the same result in clustering. The values of WCSS were observed to exponentially reduce from 159517814576 to 1.637 for the performance of clustering over the reduced dataset. This showed that factor analysis is successful in reducing the computational cost exceeding the satisfactory level of performance for the small real-world dataset used.

When the same adequacy tests were performed on the sklearn digits dataset and the real-world big dataset, we noticed that the KMO values were observed to be Nan. Upon further testing it was discovered that this was because the sparse nature of the matrix of these datasets in the process of factor analysis renders this method inefficient. This is why the methods below were employed for these datasets.

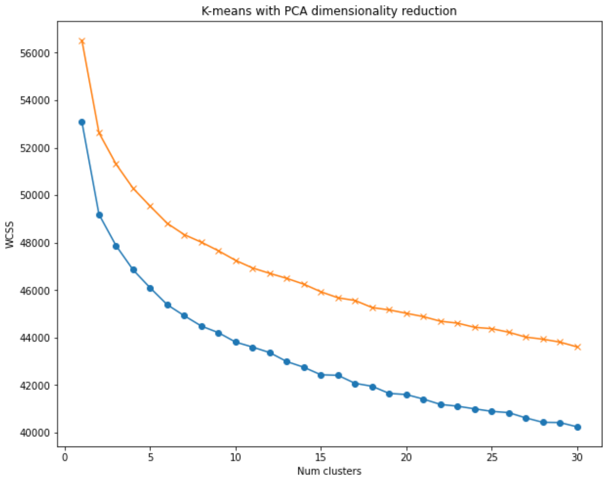
Principal Component Analysis



PCA for a small Dataset of 17 Dimensions



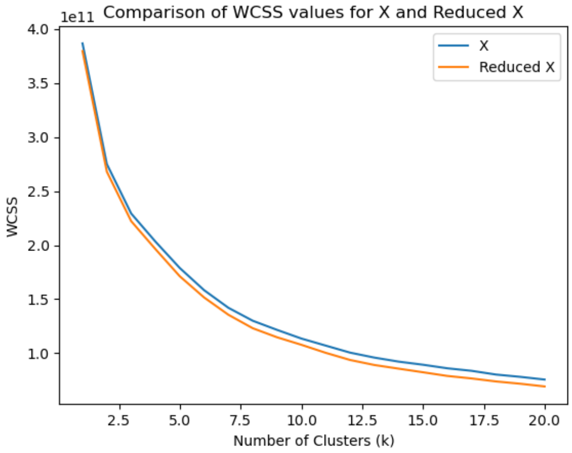
PCA for the Digits Dataset of 64 Dimensions



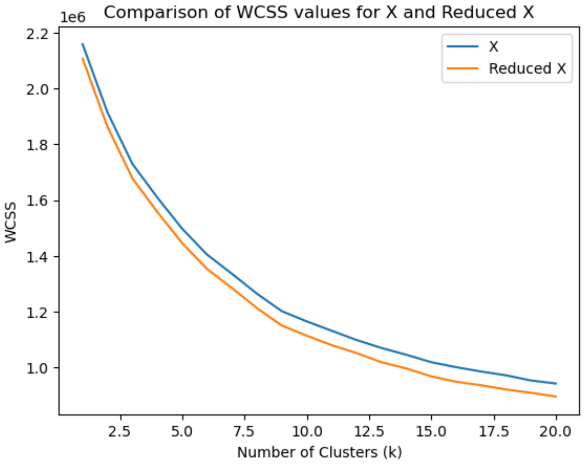
PCA for a huge Dataset of 9000+ Dimensions

As we can see with the lower-dimensionality graphs, PCA is quite strong at maintaining the original cost of the dataset it is reducing from. In the top left dataset, PCA reduces the dataset from 17 dimensions to 6, and in the top right, PCA reduces from 64 to 32 components. In both cases, the preservation of cost is very high, although with higher dimensionality it begins to taper off just a little bit. Then with the housing prices dataset of nearly 9k dimensions, we see a bigger struggle to preserve the cost. Although the behavior remains the same, far more is lost in WCSS with PCA, proving the initial inference of struggling with higher dimensionality.

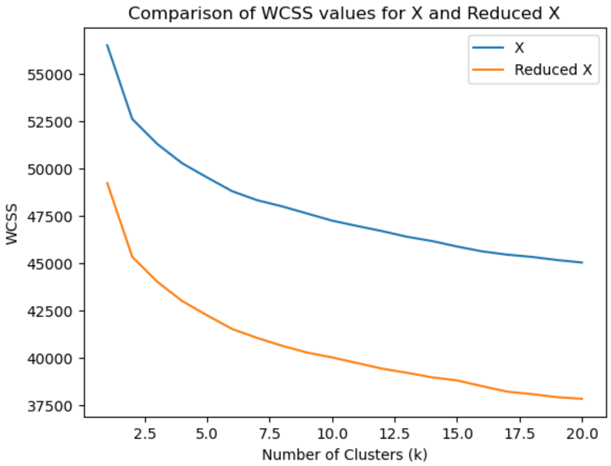
Truncated SVD



Truncated SVD for small Dataset of 17 Dimensions



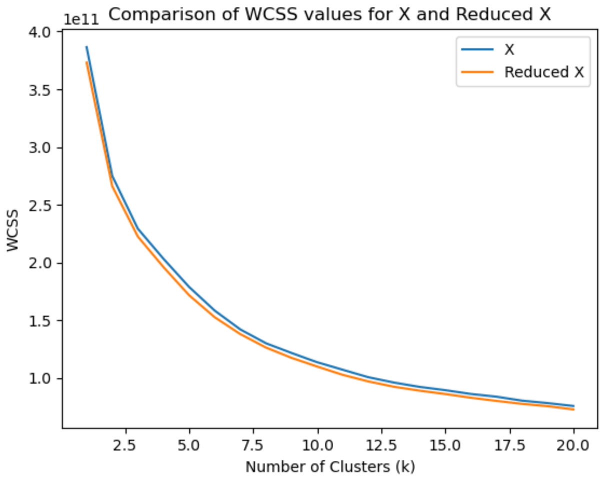
Truncated SVD for Digits Dataset of 64 Dimensions



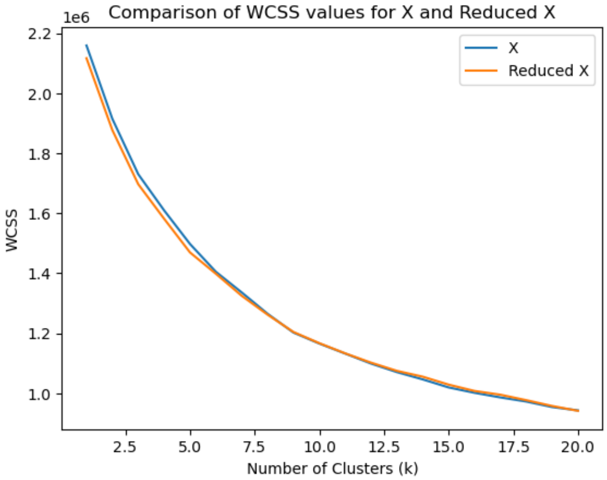
Truncated SVD for a Huge Dataset of 9000+ Dimensions

After applying Truncated SVD, we were able to reduce the dimensions of the small dataset from 17 to 6 dimensions. For the Digits Dataset, we reduced it from 64 to 35, and for the high dimensional dataset of 9000+, we were able to reduce them to 400 dimensions. From these graphs, we can see that Truncated SVD is able to preserve the cost of KMeans for the 17-dimensional and 64-dimensional Datasets, but for the high-dimensional dataset, it doesn't perform as well as it did for the lower-dimensional dataset.

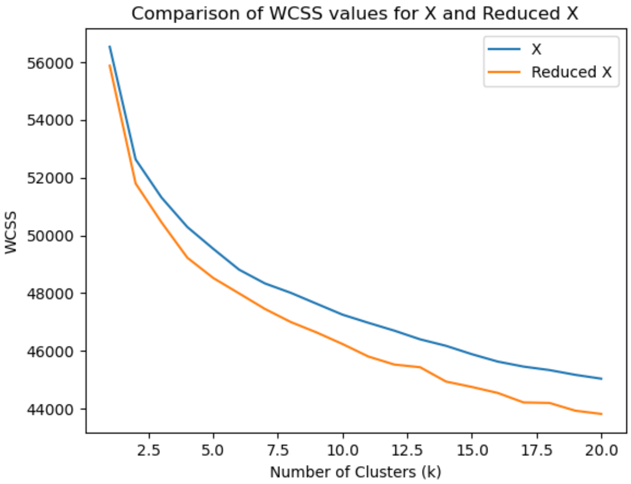
Johnson Lindenstrauss Lemma



JL Lemma for a small Dataset of 17 Dimensions



JL Lemma for the Digits Dataset of 64 Dimensions



JL Lemma for Huge Dataset of 9000+ Dimensions

We can see from the above graphs that JL Lemma is performing way better than others for higher dimensional data. For the high dimensional data, JL Lemma was able to reduce the dimensions from 9000+ to 650 dimensions.

Conclusion

We find that, in our comparison and study of different dimensionality reduction/projections algorithms and their relation to clustering, they vary in efficacy and usefulness. Factor Analysis, unfortunately, proved to be insufficient in providing enough meaningful data to show its behavior with more sparse datasets, which would pose a problem when applying it in the real world. With PCA and truncated SVD, we find that their ability to preserve the cost of WCSS is strong and efficient in lower dimensions, preferably less than 100, but they begin to struggle with large datasets. Overall PCA had a slightly better performance than SVD, but their behaviors were nearly similar enough to show us how "reduction" algorithms generally perform. Finally, we found that the Johnson-Lindenstrauss Lemma was able to address the problems that the reduction algorithms suffered from, and greatly improved the WCSS cost preservation at very high dimensionality. Although not quite as efficient there as PCA/SVD were at lower dimensions, Johnson-Lindenstrauss can still maintain very high-cost preservation and work with very large datasets much more stronger.

References

1. Zhang, Z., & Wang, J. (2018). A comparative study of dimensionality reduction techniques for clustering microarray gene expression data. BioMed research international, 2018.
2. Nguyen, H., & Lu, J. (2018). Clustering with dimension reduction: A comparative review. ACM Computing Surveys (CSUR), 51(6), 116.
3. Li, Q., & Yang, L. (2019). Comparison of dimensionality reduction methods for clustering analysis of high-dimensional data. PloS one, 14(5), e0217260.
4. Jia, J., & Zhang, C. (2018). A comparative study of dimensionality reduction techniques for clustering-based malware detection. International Journal of Advanced Computer Science and Applications, 9(3), 137-145.
5. Chen, S. H., Chen, M. H., Tsai, J. S., & Chang, Y. C. (2019). A comparative study of dimensionality reduction methods for clustering analysis of high-dimensional data. Journal of Industrial and Management Optimization, 15(2), 737-754.