Unsupervised Learning and Dimensionality Reduction

Peter Lucia

November 3, 2019

1 Abstract

In this work, I explore various unsupervised learning and dimensionality reduction methods for the task of finding patterns within handwritten digits and the chemical attributes of wine. I explore the effectiveness of two clustering algorithms – k-means and expectation maximization – as well as four dimensionality reduction algorithms – PCA, ICA, Randomized Projections, and t-SNE – in various combinations to reveal patterns in two datasets with classification labels removed. I also take newly projected data from the t-SNE algorithm and pass it through to a neural network learner to compare performance with previous experiments.

2 Datasets

The wine dataset comes from the UCI machine learning repository in [5]. Consisting of 13 continuous attributes whose composition makes up three different types of wines derived from the same region, the wine dataset is a good fit for this problem because it contains continuous rather than categorical attributes. If the dataset only contains categorical variables, which was found to be a problem with other candidate datasets, then finding distances between the features is non-trivial, especially for algorithms that depend on euclidean distances between features, such as k-means. Therefore, it is expected that the wine dataset will provide a for a more interesting exploration into this clustering and dimensionality reduction experiment than the other candidate datasets used in previous experiments. Comprised of the handwriting of 43 different people, the handwritten digit dataset is provided in normalized bitmap form, and consists of flattened 32x32 image matrices for every written digit where each element in the flattened array is an integer between 0 and 16 [6, 4]. The entire dataframe is numeric, and it is expected to yield interesting results for the clustering and dimensionality reduction experiments because of strong interdependence and mutual information shared between the digit datasets features, whose arrangement can be informed by ordinary distance functions. As part of preprocessing, both datasets are standardized such that each feature is independently normalized to have a mean of 0 and unit standard deviation of 1. The digit dataset was also used in previous supervised learning and randomized optimization experiments, so the neural network learner will train on this dataset. During preprocessing, all features of both datasets are independently standardized to have a mean of 0 and unit variance [3].

3 Clustering

3.1 K-Means Clustering

To select k, I first attempt to confirm the optimal number of clusters using the elbow method before and after PCA. The optimal number of clusters should be the same as the total number of unique classification labels available. For the wine dataset this is expected to be three, since all chemical components are derived from the three different wines. For the handwritten digit dataset, it should be ten so each digit from zero to nine is represented. Visualization of the clusters with the silhouette method will not be useful without bringing the high dimensional data down to three or two dimensions. This is especially true if no two features are fully independent of one another, where there is an absolute need for a clustering algorithm and possibly dimensionality reduction to expose any meaningful visualizations. With handwritten digits, and because we're working with raw pixel data, I expect that a comparison of only two features (i.e. two columns of pixels) will not yield much information because the clustering algorithm should need most columns of pixels that form the written character to glean any meaningful separation. Similarly, for the wine dataset, I only expect to find visible clustering between sets of two features prior to performing dimensionality reduction if there are two features with high enough importance.

The importance of the problem of choosing the best value for k is highest for an unsupervised learning problem where the possible categories of labels are unknown. For this experiment, only one of the two datasets that were used in the supervised learning experiment are reused, so with this partial a priori domain knowledge, an appropriate value for k is known ahead of time for only the digit dataset. For the digit dataset there are ten possible digits, zero through nine, so we would expect the results to show that the best value for k for this dataset is 10. For the wine dataset and given that only three types of wines were sampled from, 3 is chosen for k as a starting point from which to try increasingly larger or smaller values.

There are various methods for determining the appropriate number of k centers that will claim its closest points in the k-means algorithm. In this experiment, the elbow and the silhouette methods are used. The elbow method is a visual method where we plot the sum of squared errors versus increasing k, and find the location on the graph where the squared error abruptly stops decreasing as rapidly. In Figure 3 the selected metrics for kmeans clustering are shown as k varies from 2

to 14. In Figure 3a there is an inflection point at k = 10, where afterward the squared error decreases more rapidly before beginning to taper off. With a priori domain knowledge of the dataset, the inflection point is more compelling, but it would not be sufficient evidence that a k of 10 would be best for this dataset.

If we consider the results of the silhouette method for the digit dataset as k varies from 8 to 10 in Figures 2a and 1, a comparison of the clustered data visualizations is more compelling for a k of 10. The coefficients are plotted for each sample, and range between [-1, +1]. Each score indicates a sample's distance to neighboring clusters. For example, a silhouette coefficient score of +1 indicates that the sample is further away from neighboring clusters, a score of 0 indicates that the sample is very near the decision boundary between two neighboring clusters, and a score of -1 indicates that the sample was assigned to the wrong cluster [10]. Examining 1, with k set to 10, the silhouette clusters are all of reasonably similar size, and the overwhelming majority of coefficient scores for each sample shows greater distances to neighboring clusters.

In summary, for the digit dataset, a k of 10 should be chosen without a priori knowledge of the labels because the silhouette coefficients indicate that it provides for the least error and ambiguity of sample-cluster placement, and when the clusters are visualized, they are organized reasonably, such that logical groups are formed with lowest variance in the population size of each cluster.

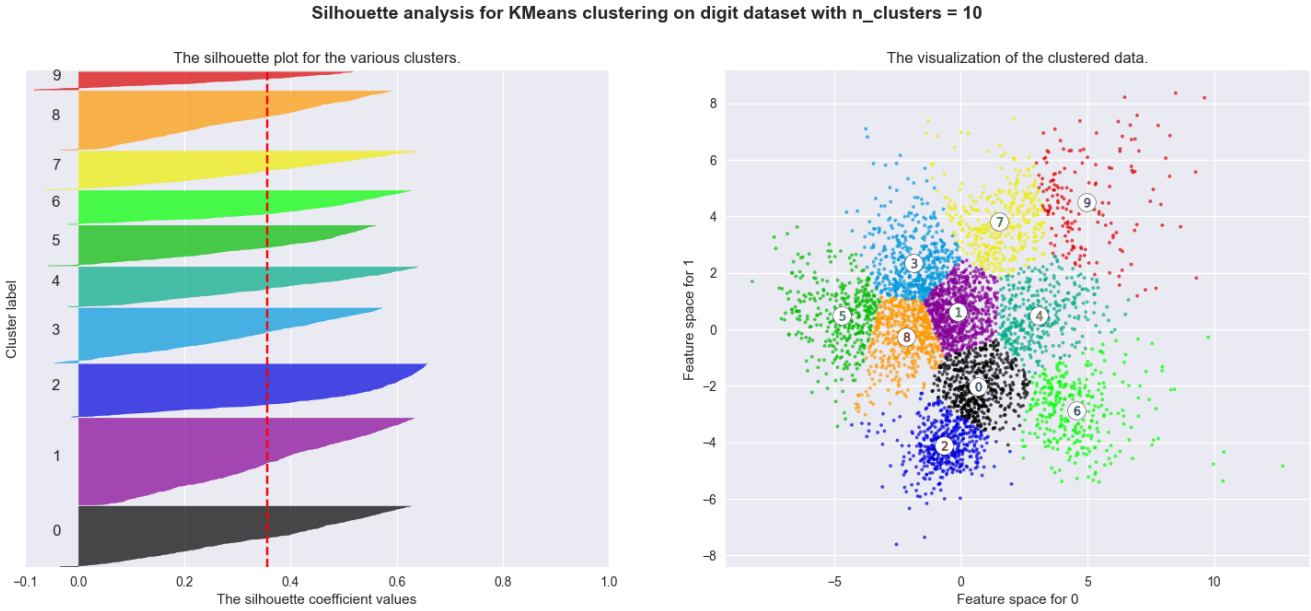
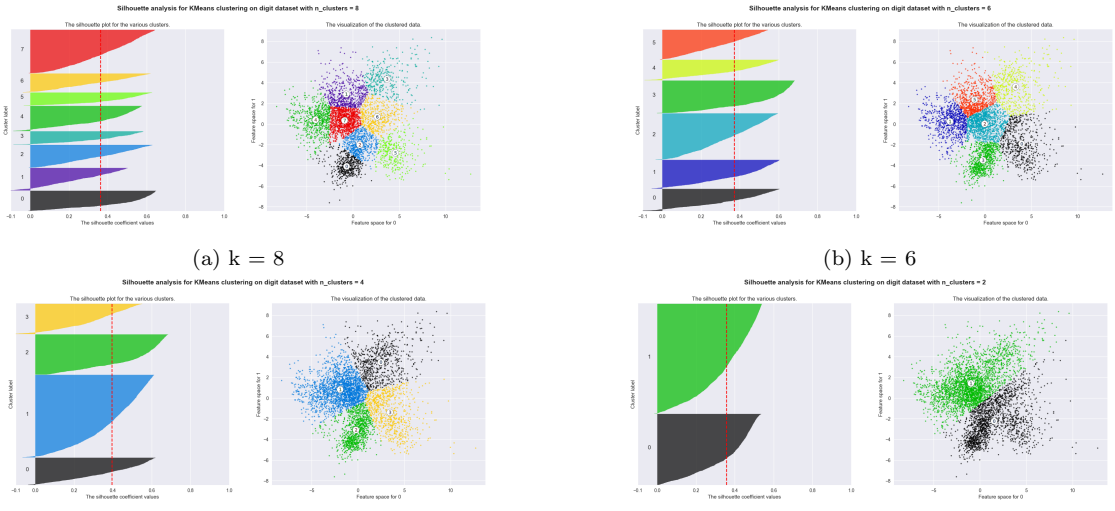


Figure 1: Silhouette Analysis for KMeans on digit dataset with k=10, Cluster visualization with PCA



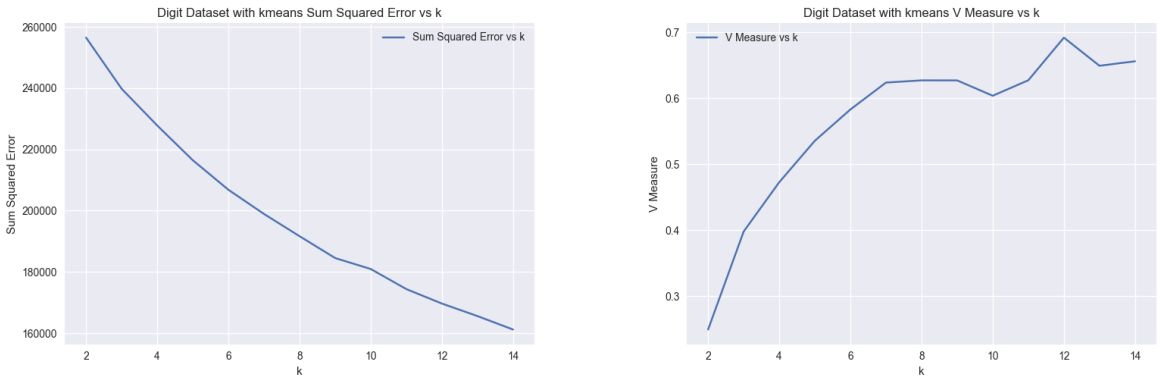
(a) k = 8

(b) k = 6

(c) k = 4

(d) k = 2

Figure 2: Silhouette Analysis for KMeans on digit dataset, Cluster visualization with PCA



(a) Sum of Squared Error vs. k (b) V Measure vs. k

Figure 3: Choosing k for Digit Recognition

To choose k for the wine dataset, after examining Figure 4 in comparison with other tested values for k in Figures 5d, 5c, 5b, and 5a, we can see the average coefficient score is highest when k = 3. After visually inspecting the clustered data, we can see it has the best visual separation and grouping between all three clusters.

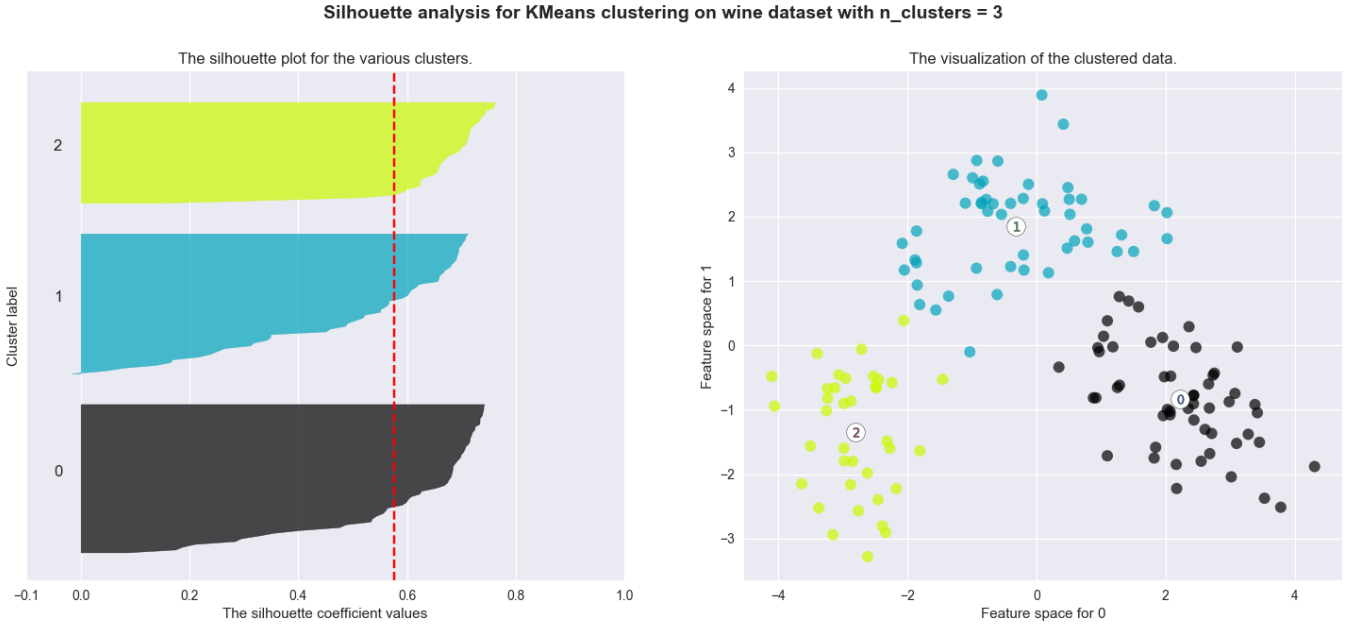
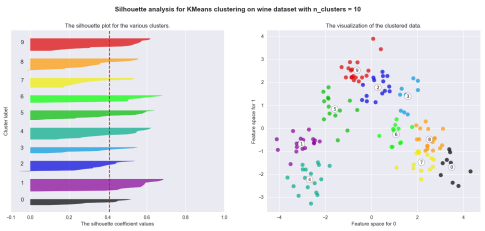
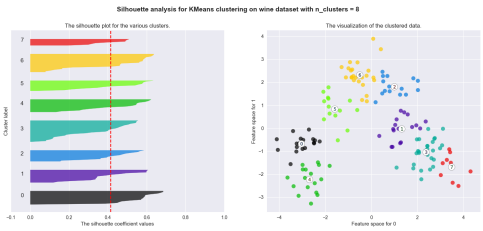


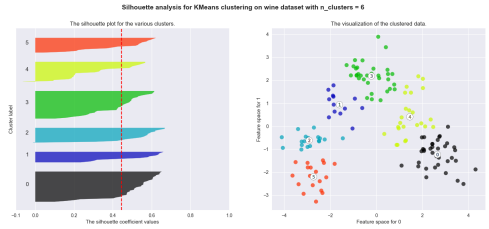
Figure 4: Silhouette Analysis for KMeans on wine dataset with k=3, Cluster visualization with PCA



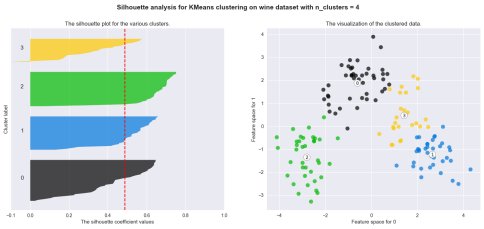
(a) k = 10



(b) k = 8



(c) k = 6

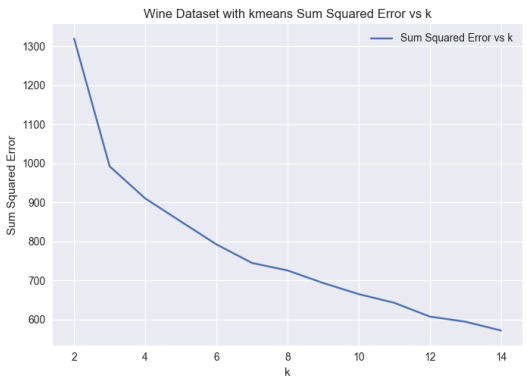


(d) k = 4

Figure 5: Silhouette Analysis for KMeans on wine dataset, Cluster visualization with PCA

Compared to the silhouette analysis and cluster visualization, the elbow method was not as useful for determining the best value for k for the k-means algorithm operating on the digit dataset. The elbow method cannot always be unambiguously identified in cases where there is no elbow or where there is perhaps more than one elbow [9]. Additionally, as k increases, we expect the average silhouette coefficient score to fluctuate based on each sample's distance to a neighboring cluster. Interestingly, the silhouette analysis did not penalize the complexity of the extra clusters against the wine dataset when comparing Figures 4 and 5, as the max silhouette score difference across all tested values for k was less than 0.2.

Against the wine dataset, the elbow method proved far more useful. As is shown in Figure 6a, there is a clear elbow point at k = 3. Additionally, both AMI [1] and V Measure peak at k = 3 in Figure 6b, providing further validation for choosing 3 for k. Considering the silhouette analysis, the highest average silhouette coefficient values are obtained with k = 3. In Figure 4, the high silhouette score for the points indicates that the majority of them are further away from neighboring clusters. The fact that there are very few negative scores indicates that there is low probability any of the points might be in the wrong cluster when setting k to 3.



(a) Sum of Squared Error vs. k

(b) V Measure vs. k

Figure 6: Choosing k for Wine Dataset

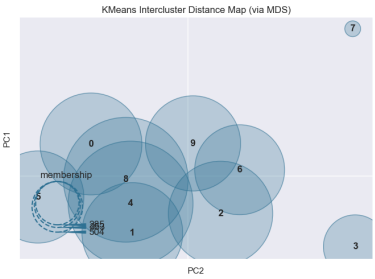
3.2 Expectation Maximization

Figure 7 shows intercluster distance maps and cluster visualization of the k-means and expectation maximization (EM) algorithms against both datasets. Each cluster is sized by membership in the distance maps to show the relative importance of each cluster. This also means that any overlaps do not imply the actual clusters of points overlap in the original feature space [11]. Interestingly, for the wine dataset, both algorithms show relatively equal membership across all three clusters ranging between 40 and 54 samples per cluster. Against the digit dataset, there is agreement among the number of the clusters between the two algorithms, but the two algorithms differ in relative cluster center placement. There is significant overlap between the gray and green clusters generated by the EM algorithm. In this particular case, the samples had high

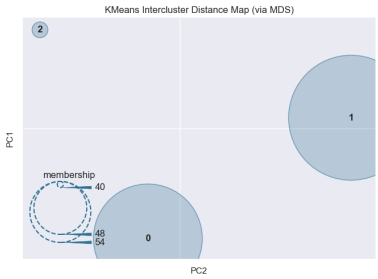
[1] To view this graph, please run the accompanying code

probability of being members of either cluster, but by assigning a sample to the cluster of marginally higher probability, we lose information about shared membership.

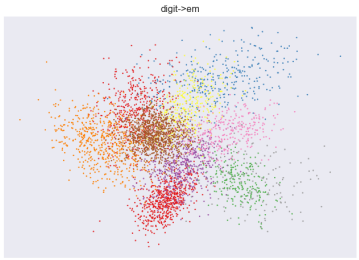
EM is susceptible to certain risks. For example, it may not converge, as there are infinite possible probability configurations, while k-means has a finite number of iterations and converges in finite time. EM can also get stuck in suboptimal local maxima, or it may never converge, while k-means takes a finite (but exponential) number of iterations $O(K^n)$, with error guaranteed to continuously decrease unless the tie-breaking rule is needed [7].



(a) digit → kmeans



(b) wine → kmeans



(c) digit → em

(d) wine → em

Figure 7: K-Means and EM Intercluster Distance Maps

The maximum number of iterations of EM was limited to 100, each component was given its own covariance matrix (as opposed to sharing the covariance matrix with other components), and the EM iterations were set to also stop if the lower bound average gain fell below 0.001 [3]. To improve performance, future experiments should more exhaustively test different parameters for the EM algorithm and add handling for joint cluster membership of samples rather than force points into a cluster regardless of its membership probability with another cluster.

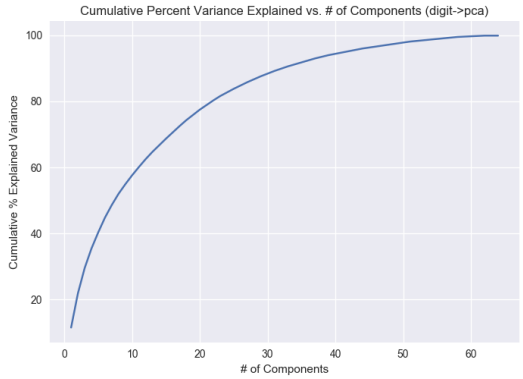
4 Dimensionality Reduction

4.1 PCA

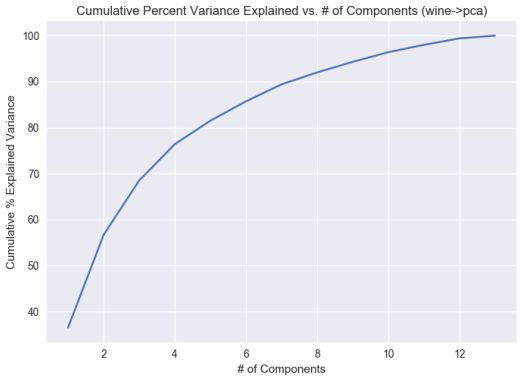
In order to reduce the dimensionality of the data, one technique is Principal Components Analysis (PCA), PCA is a particular kind of eigenproblem that finds two components: the first is the direction of maximal variance of the data and the second is a direction orthogonal to the first. [7] The eigenvalues corresponding to the eigenvectors indicate what percentage of the variance in the data can be attributed to a particular PCA component.

Since our goal with PCA is to project the data onto a smaller subspace, examining the distribution of eigenvalues for each component is useful to determine how many features are actually needed so that we can eliminate features that capture similar information or that don't explain variance in the data very well [12, 13]. Figure 8 shows a plot of the cumulative percent explained variance versus an increasing number of components limited by the number of features for both datasets. The results of Figure 8 reveal that over 80% of explained variance is contained within the first 25 and 5 components when running PCA against the digit and wine datasets, respectively. In other words, fewer components are needed to explain more variance in the wine dataset than the digit dataset. Intuitively, this makes sense because we expect each row of pixel data in the digit dataset to provide a roughly equal level of importance, with groups of pixels where edges are present to have slightly higher importance. The euclidean distance between pixels in each feature is also more easily quantifiable than some arbitrary distance function between two chemicals present in a particular type of wine. To offset a lack of geometrical distance function, the wine dataset still yields meaningful interpretation here with fewer features because they each have a greater statistical significance. However, if we perform a two-dimensional projection, for example, we should expect to lose a statistically significant amount of information with either of these datasets because, as is suggested by both graphs in Figure

8, at least 40% and 75% of explained variance in the wine and digit datasets will not be explainable without adding the rest of the components.



(a) digit → pca

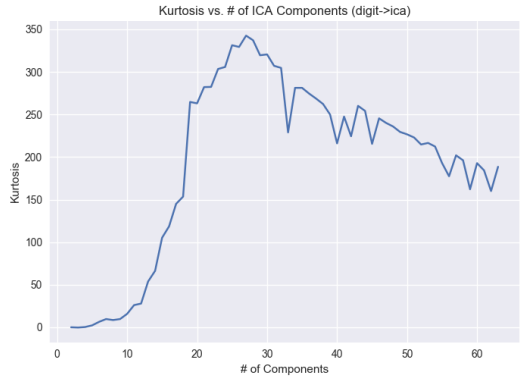


(b) wine → pca

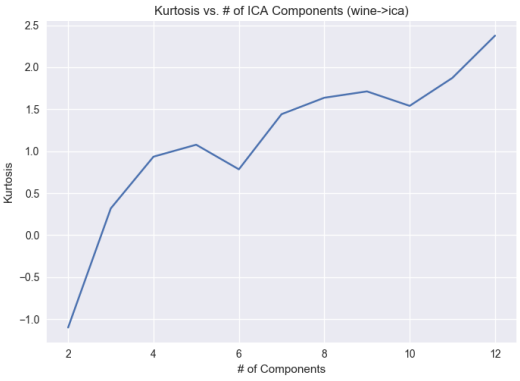
Figure 8: PCA % Explained Variance vs. # of Components

4.2 ICA

The goal of independent component analysis (ICA) is to find a linear representation of non-gaussian data in order to best capture its structure so that the resulting components are as independent as possible [16]. To measure the gaussianity of the resultant components, I employ kurtosis, which measures shape of a distribution relative to the normal distribution [14]. Kurtosis is a normalized version of the fourth moment Ey^4: $kurt(y) = Ey^4 - 3(Ey^2)^2$ [16]. If the kurtosis is found to be zero for a random variable, then it is gaussian or mesokurtic. Otherwise, if it is below zero, it will be subgaussian (platykurtic), and if it is above zero, it is considered supergaussian (leptokurtic) [16, 17]. Figure 9 shows how kurtotic the distributions are for varying number of FastICA components. In Figure 9a the distribution trends toward supergaussian while components increases to 28, but afterward trends downward, where it remains supergaussian before stopping when the number of components matches the number of features. The FastICA algorithm exhibits strong supergaussian behavior against the digit dataset, while is comparatively normal against the wine dataset. For example, the kurtosis spans between 0 and 345 against the digit dataset, while against the wine dataset, it shows much less variance by spanning between -1 and 2.5, with a trend toward being supergaussian for increasing # of components.

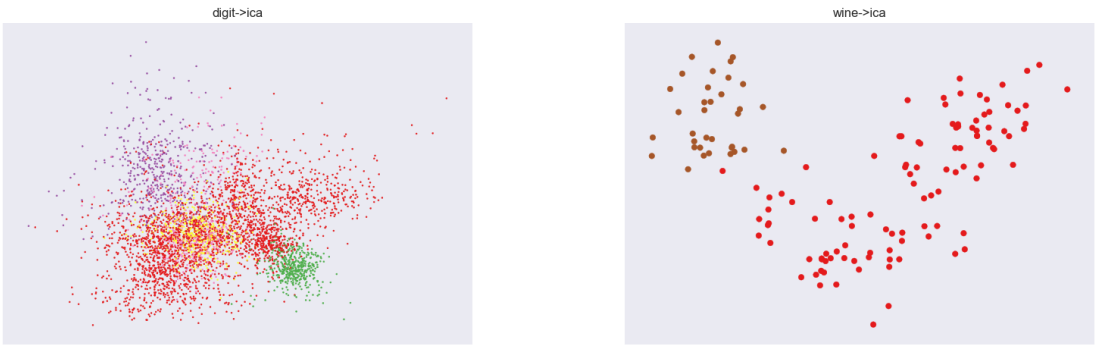


(a) digit → ica



(b) wine → ica

Figure 9: Kurtosis vs. # of Components

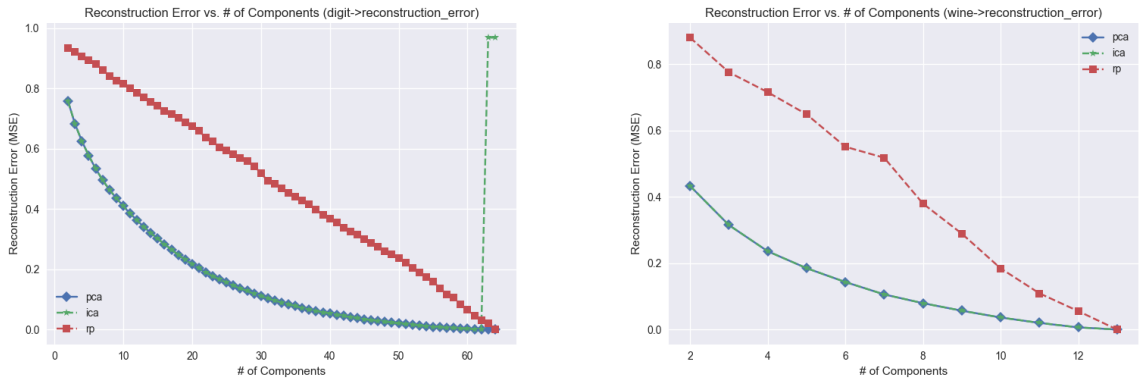


(a) digit → ica (b) wine → ica

Figure 10: ICA Results

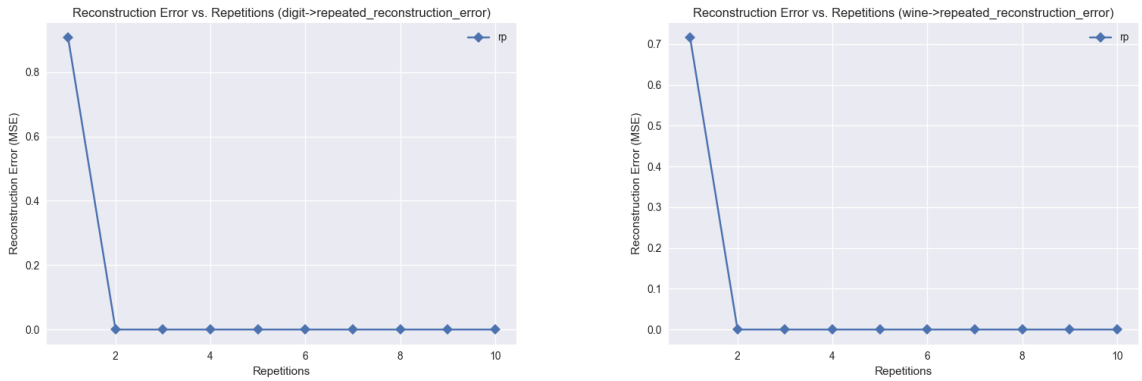
4.3 Randomized Projections

Figure 11 compares the reconstruction error of PCA, ICA, and Randomized Projection (RP) for varying number of components. t-SNE is excluded since at at the time of development, it was not feasible to inverse the t-SNE transformation efficiently with the libraries available. It is suggested for future experiments to develop or use a computationally cheaper method to invert the t-SNE transformation for the purpose of analyzing its reconstruction error. As one might expect, for all three of the tested algorithms, the MSE generally decreased as the number of components increased. The one exception to this is shown in Figure 11a, where the ICA algorithm's MSE spiked to almost 1.0 as the number of components approached 64. Since the ICA algorithm attempts to construct its transformed features to be mutually independent, it may have had difficulty transforming all features to be mutually independent of one another. Ignoring the ordering of the features, which is one difference it has with PCA, may have also contributed to this outlier [7]. Also, the randomized projection algorithms generally gave a higher reconstruction error than ICA and PCA for both datasets. Compared to PCA and ICA, the advantage of the randomized projection lies in its lower computational complexity requirement [7]. So, it will be faster, but this comes at a cost at having a higher reconstruction error. Now, the results of Figure 12 show that the MSE improves significantly if the randomized projection algorithm is retrained several times on the transformed data. Additionally, the variation is remarkably low and stable with more than two repetitions against both datasets. For the purpose of dimensionality reduction, the approach of RP is to generate and then pick a random direction onto which the data will be projected. Therefore, these results provide no evidence to the contrary that the advantage of RP is that it is faster than PCA or ICA, but not as accurate unless repeatedly run [7].



(a) digit → reconstruction error (b) wine → reconstruction error

Figure 11: PCA, ICA, RP: % Explained Variance vs. # of Components



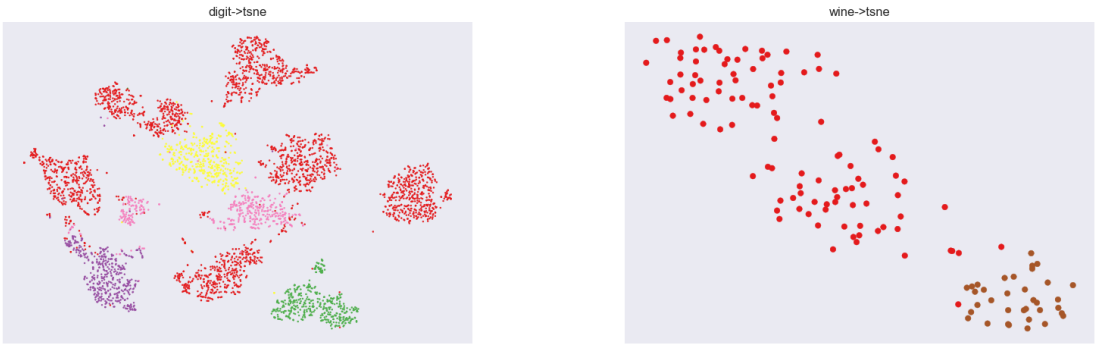
(a) digit → reconstruction error (repeated) (b) wine → reconstruction error (repeated)

Figure 12: Randomized Projection repeated iterations

4.4 t-SNE

Figures 13a and 13b show the results of the t-SNE algorithm against both datasets. After a visual inspection, it is clear there are eleven small cluster forming in the digit dataset plot and three clusters forming against the wine dataset. Figure 13b shows three clusters that are warped into a more eliptical shape than we saw previously with standalone ICA in Figure 10a. When compared with standalone ICA or PCA, these results suggest that t-SNE exhibits more well defined clusters with larger intercluster distances for the larger digit dataset. Against the wine dataset, the t-SNE algorithm does not perform dimensionality reduction any better than PCA or ICA do in Figures 4 and 10, but this may be attributed to the simplicity of the wine dataset, which may not require a dimensionality reduction algorithm like t-SNE that can determine highly complex polynomial relationships between features [21].

t-SNE is particularly useful for the visualization of high dimensional data down to two or three dimensions and has demonstrated superior performance against many other visualization techniques because of its ability to retain both local and global structure of the data simultaneously [19, 20]. Compared to a linear algorithm such as PCA, the better visualization and separation of the digit dataset can be attributed to the superiority of the t-SNE method, as it creates probability distributions during random walks to determine complex relationships between features [21]. To improve the performance of the t-SNE algorithm, it is suggested that future improvements to this experiment explore additional tuning of the t-SNE algorithm so that, for example, the number of clusters found by t-SNE might more accurately match the number of possible digits in the digit dataset which is ten, not eleven.



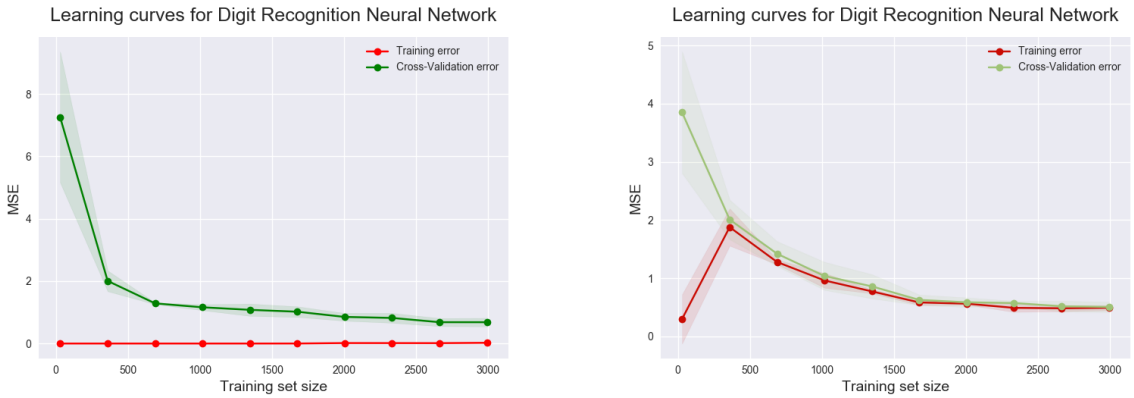
(a) digit → tsne (b) wine → tsne

Figure 13: t-SNE Clusters

5 Neural Networks

The digit dataset was previously used for the first assignment, and so its dimensions are first reduced with t-SNE, and then the newly projected data is passed through to the neural network learner. In order to replicate the experiment of assignment 1, the MSE of the neural network is measured with increasing training set size so that the bias and variance of the model can be compared to the performance on non-reduced data from assignment 1. Figure 14 shows the results of the learning curves with and without t-SNE dimensionality reduction. Interestingly, these results suggest there is less variance and less bias in the model with t-SNE based dimensionality reduction versus the model trained against the original dataset.

While the neural network trained on the dimension reduced dataset converged to a lower MSE with less variance, it also needed approx. 39% less time to train, as is shown in Figure 15. The query times for both the training and testing sets were unaffected, however speedup in training with *lower* bias and *less* variance is an interesting result. One possible explanation for the reduced training time could be that the digit dataset used in this experiment with results shown in Figure 14b was standardized to have feature-wise zero mean and unit variance, while the dataset the neural network trained on in Figure 14a was not standardized. Network training converges faster if its inputs are decorrelated and linearly transformed to have zero means and unit variances [22]. Therefore, while the results of this experiment may encourage performing dimensionality reduction on a dataset prior to training, more experiments are necessary to fully validate this claim. For example, it would be a useful exercise for future experiments to see if the speedup and bias advantages still hold if the original dataset is standardized in the same fashion as the reduced dataset.



(a) Without t-SNE dimensionality reduction (b) With t-SNE dimensionality reduction

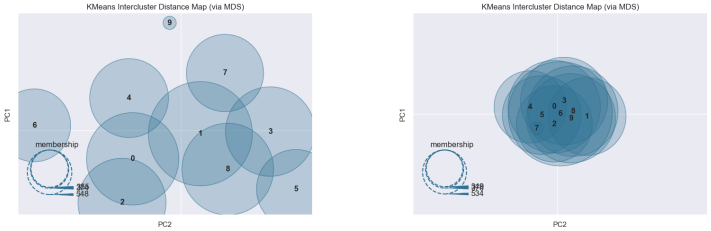
Figure 14: Learning Curves with and without t-SNE Dimensionality Reduction

classifier	training time (s)	train query (s)	test query (s)
NN Trained on dim-reduced data	3.607	0.012	0.003
NN Trained on original data	5.924	0.013	0.004

Figure 15: NN Speed Comparison for dim. reduced and non dim. reduced data

6 Clustering After Dimensionality Reduction

After reproducing the clustering experiments on the same datasets projected onto the new spaces created by ICA, PCA, and RP, I did not get the same clusters as before. When PCA was used, the intercluster distances expanded for both datasets. With ICA, the intercluster distances decreased so much that there were no longer any visible boundaries between the clusters, so the result was a single multicolored blob for both datasets. Figure 16 shows one example of how running PCA against the digit dataset improved the clustering results of the k-means algorithm, while running ICA caused it to have trouble separating the data. With randomized projections, the intercluster distances decreased for nearby clusters and expanded between clusters that were already farther apart. Now, the impossibility theorem requires that no clustering scheme can achieve all three of richness, scale invariance, and consistency [18]. Therefore, if any dimensionality reduction technique creates a new dataset that k-means or em cannot adapt for, we should not expect to see the same clusters as before. K-means does not satisfy the richness algorithm, so if any of the dimensionality reduction algorithms caused the value of k to change, it would no longer be able to properly classify the projected data [15]. Moreover, k-means is essentially a subset case of EM algorithm, and EM does not guarantee convergence since there are infinite number of configurations and it can get stuck in local optimum [7]. While k-means has k^n configurations and converges in finite time, against the digit dataset, Figure 16 suggests that kmeans may fall into suboptimal local maxima if the wrong dimensionality reduction algorithm is chosen, as it was able to improve the clustering with PCA, but had trouble separating data reduced by ICA and RP [2] [7].



(a) digit → pca → kmeans (b) digit → ica → kmeans

Figure 16: Clustering results after dimensionality reduction

[2]To view the RP graph, please run the accompanying code

7 Conclusion

After exploring the various clustering and dimensionality reduction algorithms for the purpose of revealing patterns in the digit and wine datasets, there are clearly benefits and drawbacks to each. For example, k-means tended to fall into suboptimal local maxima if matched with the wrong dimensionality reduction algorithm, while PCA, ICA, and t-SNE were equally effective at retaining the structure of the wine dataset when reducing it down to two dimensions. As is shown in Figure 13, the t-SNE separates the digit dataset well, with larger intercluster distances than any other dimensionality reduction algorithm tested, but its cost function is not guaranteed to converge to a global optimum and it identified eleven clusters instead of ten [19]. When choosing k for the k-means algorithm it was necessary to use both the elbow method and silhouette analysis with visualization of the clustered data to remove any uncertainty about the best value for k. The silhouette analysis was more reliable in determining the best value for k, if no a priori knowledge about the dataset is assumed.

As is shown in Figure 7, the expectation maximization and kmeans algorithms showed similar effectiveness in clustering non-dimensionality reduced data. However, both algorithms had a much greater average intercluster distance against the wine dataset than the digit dataset. One reason these algorithms had the better performance against the wine dataset is that the wine dataset is well structured and less challenging than the digit dataset. In a classification context, it only has three classes while the digit dataset has ten [5, 4]. Also, we can expect the algorithms to perform reasonably well against data that we already know have labels, even though the labels are removed from the dataset during preprocessing. When working with noisier data, it is expected that the performance obtained with the datasets of this experiment would be harder to match, especially if there are data that do not belong in any cluster, for example, or if the labelling of the data is not known beforehand.

Interestingly, the neural network trained on the dimension reduced dataset converged to a lower MSE faster with less variance. Considering that neural networks have been successfully applied to tasks such as handwriting recognition empirically, the result is not surprising but may be due to the standardization of the dataset to have zero mean and unit variance. The speedup may also be caused by the reduction in input layers, but it is suggested that additional experiments be conducted to fully explore these claims.

References

[1] Mitchell, Tom M. "Machine learning." (1997).

[2] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research 12.Oct (2011): 2825-2830.

[3] Buitinck, Lars, et al. "API design for machine learning software: experiences from the scikit-learn project." arXiv preprint arXiv:1309.0238 (2013).

[4] https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits

[5] https://archive.ics.uci.edu/ml/datasets/Wine

[6] Garris, Michael D., James L. Blue, and Gerald T. Candela. "NIST form-based handprint recognition system." Technical Report NISTIR 5469 and CD-ROM, National Institute of Standards and Technology. 1994.

[7] https://classroom.udacity.com/courses/ud262

[8] https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_digits.html#sphx-glr-auto-examples-cluster-plot-kmeans-digits-py

[9] Kodinariya, Trupti M., and Prashant R. Makwana. "Review on determining number of Cluster in K-Means Clustering." International Journal 1.6 (2013): 90-95.

[10] https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

[11] http://www.scikit-yb.org/en/latest/

[12] https://sebastianraschka.com/Articles/2015_pca_in_3_steps.html

[13] https://etav.github.io/python/scikit_pca.html

[14] http://labs.seas.wustl.edu/bme/raman/Lectures/Lecture14_ICA.pdf

[15] http://alexhwilliams.info/itsneuronalblog/2015/10/01/clustering2/

[16] Hyvärinen, Aapo, and Erkki Oja. "Independent component analysis: algorithms and applications." Neural networks 13.4-5 (2000): 411-430.

[17] https://en.wikipedia.org/wiki/Kurtosis

[18] Kleinberg, Jon M. "An impossibility theorem for clustering." Advances in neural information processing systems. 2003.

[19] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.Nov (2008): 2579-2605.

[20] https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/

[21] https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/

[22] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167 (2015).