Unsupervised Learning and Dimensionality Reduction
CS 7641

Abel Aguilar
aaguilar61@gatech.edu

1 INTRODUCTION TO DATA

1.1 Pima Indians Diabetes Database

The Pima Indians Diabetes Database is an interesting choice for unsupervised learning because this would normally be used for classification, as I used in A1. However, this presents a very interesting challenge in that it would in theory only have two clusters, positive and negative to diabetes. One interesting aspect of this assignemnt is to see if other clear clusters emerge. Another interesting aspect of this data is the fact that it has eight features to play around with. This will give dimensionality reduction algorithms a larger challenge in deciding which features are relevant and which features are not. This dataset also has unbalanced data, as there are many more negative examples to positive examples, providing yet another interesting challenge to our algorithms.

1.2 Iris Species

Iris Species provides its own set of challenges to this problem. As it is also a classification problem initially it would be interesting to see how well our clustering algorithms group similar species together. In theory three clusters should emerge for each species. This dataset is also rather small and has four features. It might be difficult for our dimensionality reduction algorithm to find which features are the most important in such a small sample size. It will be interesting to examine the results given these sets of challenges.

2 EXPERIMENTAL METHODOLOGY

In this experiment, more than in other assignments, visualization will be very important in determining the performance of our algorithms. How well are our algorithms clustering? Visualizing this will be able to provide much more insight. And while using the clusters that are generated to predict the outcome of untested data is not entirely the goal of clustering, this will be able to tell us how

far off our clusters are from what should be expected. I will employ both of these methods, visualization and prediction to test our algorithms for effectiveness.

3 ALGORITHM RESULTS

3.1 Pima Indians Diabetes Database (PIDD)

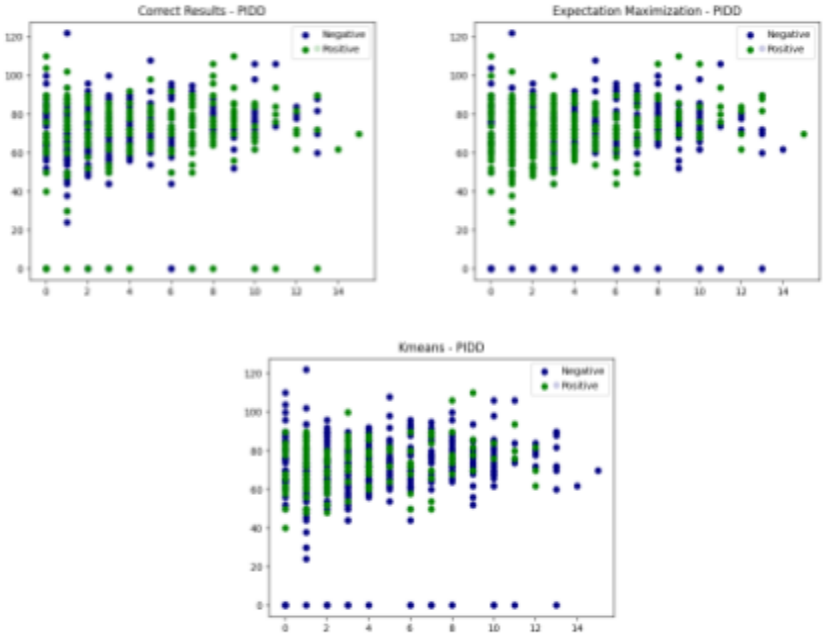

Figure 1- PIDD

The images above show Expectation Maximization and KMeans algorithm ran against our PIDD dataset. The resulting scatter plots are two features of PIDD plotted against one another. These clusters, most likely due to the large number of features, do not do very well when tested for accuracy. It does not seem the clustering was able to separate positive from negative very well and when I ran the training data through the cluster to have it give a prediction, it was only correct about 48% of the time for EM and 67% for KMeans. Also looking at the plots it seemed it was unable to find patterns in the data and the clusters don't really look like clusters at all. This could be due to the limited ability of scatter plots with datasets that have multiple features which would translate into more multidimensional charts. However it does seem that the number of features is difficult to make sense of for a clustering algorithm.

3.2 Iris Species (IR)

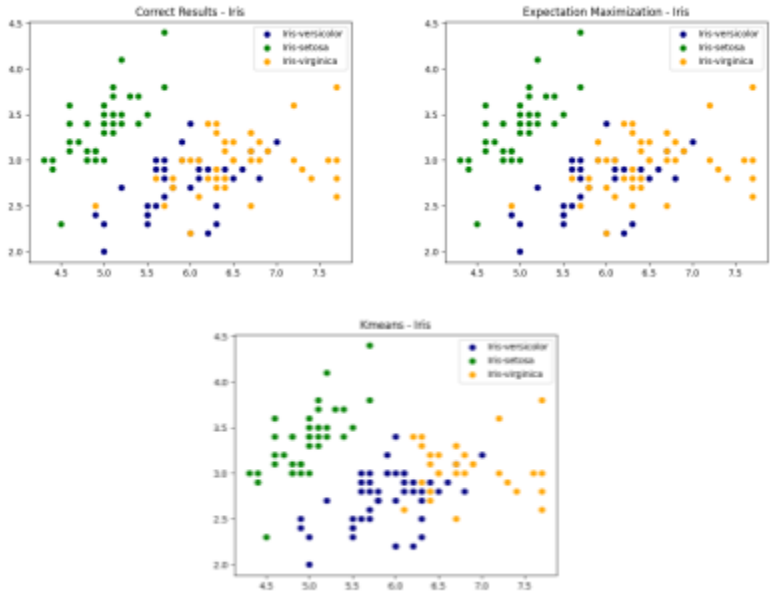

Figure 2- IR

Iris Species did a much better good of matching the clusters that would be ideal based on the data. As you can see the clusters are much more defined and isolated which makes it much easier for EM and KMeans to find and isolate clusters. In fact EM when tested against training data correctly predicted 97.5% of species and KMeans predicted 88.3%. Looking at the scatter plots they seem to align very well with natural clusters that form, and the ones that EM and KMeans misclassify look to be outliers that don't fall naturally within the cluster of their peers. IR seems to be much better suited for clustering problems than PIDD.

3.3 Dimensionality Reduction

	PCA	ICA	RP	ISOMAP
PIDD - EM	0.4869	0.4169	0.4869	0.51465
PIDD - KM	0.6726	0.4121	0.6726	0.6645
IR - EM	0.975	0.6	0.975	0.8833
IR - KM	0.8833	0.266	0.425	0.89166

The above table shows each of the Dimensionality Reduction algorithms results ran through a clustering algorithm(EM or KMeans) and then tested against training data. For PCA we notice that generally, IR is much easier to separate the features into different clusters then PIDD. When tested against test data we can see that IR samples do much better then PIDD. Looking at the charts, for each it looks like PIDD are separated into clusters but those clusters aren't great representations of the data. This shows that variance is much more important for IR than PIDD, this makes sense looking at the data, since different species have different lengths, and similar flowers within each species are more or less the same, this tends to do well with PCA. PIDD on the other hand depends more on different features interacting with each other, which makes it difficult for PCA to make sense of. ICA on the other hand struggles with both PIDD and IR. It looks like in general the features of both datasets are not independent of each other. They rely on the combination of features to make a distinction, this means ICA is not well suited for these problems.

3.3 Dimensionality Reduction and Clustering

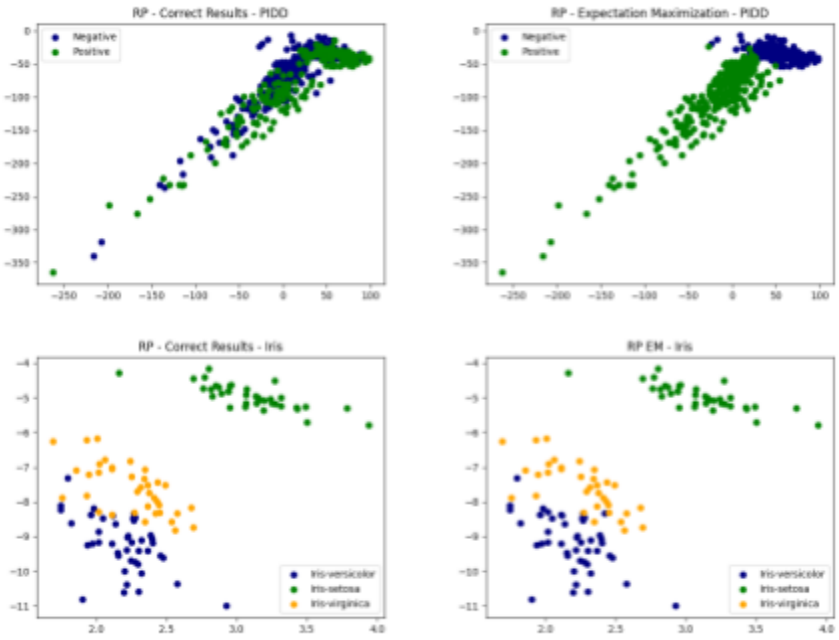

Figure 3- Random Projection

Looking at the results of my clustering algorithms after applying random project you can see that for PIDD, as has been the case throughout this project, random

projections did not do the best job at isolating the variables into well defined clusters, so when EM was run on the output it the clusters that were generated did not closely align with separation of the actual dataset. This could again have to do with the fact that features for PIDD are more closely related to each other and so separating those is a difficult task for our algorithm.

In contrast, you can see that IR the clusters were pretty easily separated and were naturally clustered to match the data. Three distinct clusters were created which our clustering algorithm EM was able to pick up on. This is most likely due to Iris having less features and those features being less intertwined with one another.

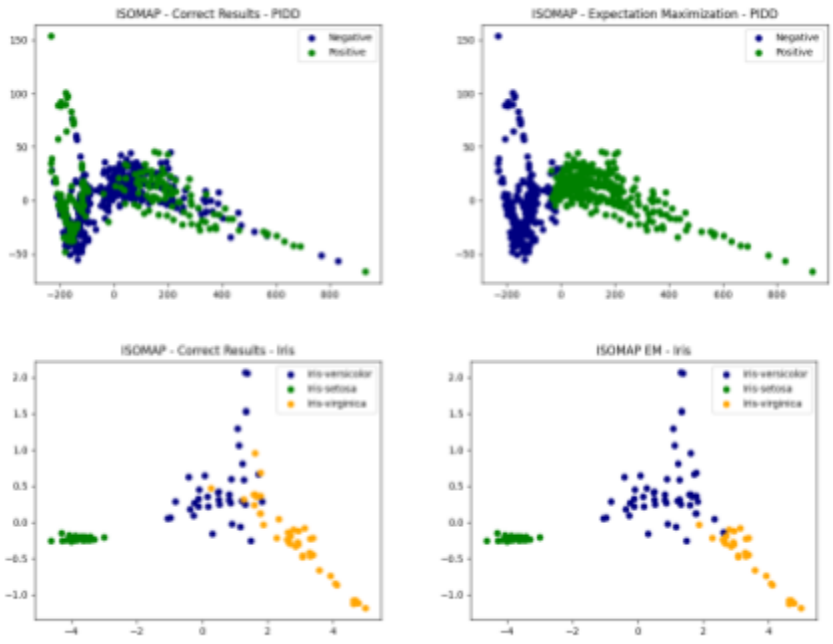

Figure 4- Isomap

With Isomap you can see a similar picture emerging , while Isomap and Random Projection each have their own way of separating out features, It seems that with this datasets that results were rather similar. One interesting aspect of Isomap that did however lead to a marginally better outcome then the other dimensionality and reduction algorithms was the fact that Isomap is non linear and given that PIDD has such a larger number of features, the linear methods did

seem quite limited in their approach. Isomap marginally outperformed almost all other dimensionality and reduction algorithms by anywhere from 3% to 10%. I again attribute this to the nonlinear nature of this method in PIDD.

For IR Isomap performed more consistently then other methods however it did not have best results of any test. I believe that Isomap generally works best with problems that have more than a few features so the problem can not be solved linearly. This seems to be why isomap was able to perform consistently high around 90% for all tests. The features as you can see from the charts above were pretty cleanly split in clusters, which EM was able to capitalize on.

3.4 Dimensionality Reduction and Neural Networks

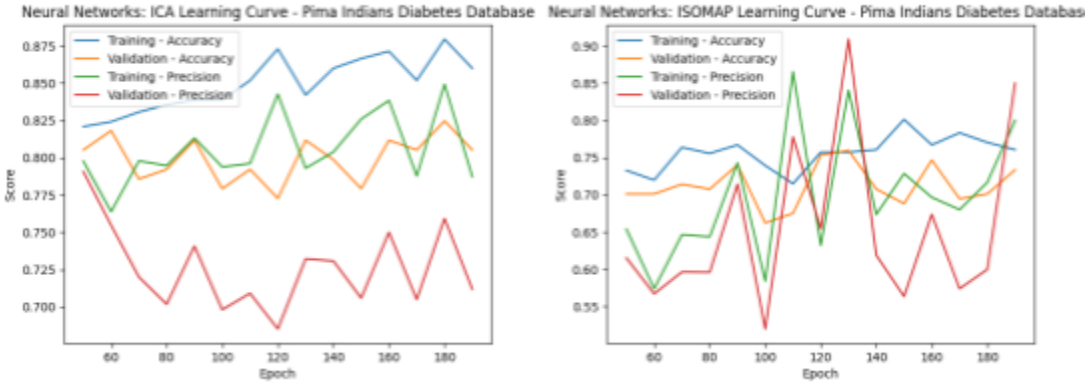

Figure 5- ICA and Isomap Neural Network

For this section I decided to use PIDD to test how much neural networks could improve if they were trained on models using data after running it through a ICA model. Compared to the normal training it does seem that ICA helps the neural network perform better. We saw an increase of roughly 5% to 7% across the board in accuray. These results make sense as ICA is refining the data and adding more weight to features that are more important via independence while still allowing features to rely on each other. This works well with Neural Networks as they are better at picking up these nuances than clustering algorithms are.

Isomap also saw an increase in performance but at a much smaller scale, only 1% to 2% in accuracy. This surprised me as I assumed because Isomap is nonlinear it would be better at isolating a large number of features then ICA could. However,

we saw a large increase in precision which stands to suggest Isomap would be more consistent at scale.

3.4 EM, KMeans and Neural Networks

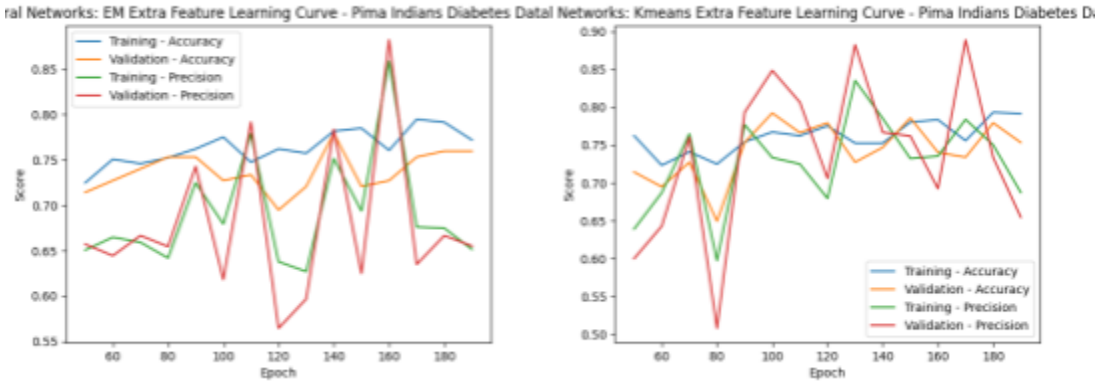

Figure 6- EM and KMeans Neural Network

In this demonstration, the extra feature that I added was the predictions from the clustering algorithms EM and KMeans after being run through the training data. And we can see that both EM and KMeans really did not produce better results than the original Neural Network. However, we did see an increase in precision for both. This could be that since neither of the predictions were very accurate it did not allow our Neural Network to overfit to the data at hand. Making the network much more scalable. However since it did not provide any new accurate information, that translates into similar performance on accuracy.

4 CONCLUSION

Clustering algorithms can be beneficial when data is unlabeled and we need to make sense of groupings. However, they have limitations with datasets with a large number of features. Dimensionality reduction can improve the focus on important features but just like all of ML, the algorithm you choose needs to be tailored to the problem you are trying to solve.

5 REFERENCES

1. "2.1. Gaussian Mixture Models." Scikit, scikit-learn.org/stable/modules/mixture.html. Accessed 5 Nov. 2023.
2. "2.2. Manifold Learning." Scikit, scikit-learn.org/stable/modules/manifold.html. Accessed 5 Nov. 2023.
3. "API Reference." Scikit, scikit-learn.org/stable/modules/classes.html#module-sklearn.random_projection. Accessed 5 Nov. 2023.
4. "Blind Source Separation Using Fastica in Scikit Learn." GeeksforGeeks, GeeksforGeeks, 10 Feb. 2023, www.geeksforgeeks.org/blind-source-separation-using-fastica-in-scikit-learn/.
5. "Blind Source Separation Using Fastica in Scikit Learn." GeeksforGeeks, GeeksforGeeks, 10 Feb. 2023, www.geeksforgeeks.org/blind-source-separation-using-fastica-in-scikit-learn/.
6. "Comparison of Manifold Learning Methods in Scikit Learn." GeeksforGeeks, GeeksforGeeks, 8 June 2023, www.geeksforgeeks.org/comparison-of-manifold-learning-methods-in-scikit-learn/.
7. Learning, UCI Machine. "Iris Species." Kaggle, 27 Sept. 2016, www.kaggle.com/datasets/uciml/iris.
8. Learning, UCI Machine. "Pima Indians Diabetes Database." Kaggle, 6 Oct. 2016, www.kaggle.com/datasets/uciml/pima-indians-diabetes-database.
9. "Machine Learning - K-Means." Python Machine Learning - K-Means, www.w3schools.com/python/python_ml_k-means.asp. Accessed 5 Nov. 2023.
10. "Principal Component Analysis with Python." GeeksforGeeks, GeeksforGeeks, 22 Apr. 2023, www.geeksforgeeks.org/principal-component-analysis-with-python/.
11. "Principal Component Analysis with Python." GeeksforGeeks, GeeksforGeeks, 22 Apr. 2023, www.geeksforgeeks.org/principal-component-analysis-with-python/.
12. "Sklearn.Cluster.Kmeans." Scikit, scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. Accessed 5 Nov. 2023.
13. "Sklearn.Decomposition.Fastica." Scikit, scikit-learn.org/stable/modules/generated/sklearn.decomposition.FastICA.html. Accessed 5 Nov. 2023.
14. "SKLEARN.DECOMPOSITION.PCA." Scikit, scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html. Accessed 5 Nov. 2023.
15. "Sklearn.Manifold.Isomap." Scikit, scikit-learn.org/stable/modules/generated/sklearn.manifold.Isomap.html. Accessed 5 Nov. 2023.
16. "Sklearn.Mixture.Gaussianmixture." Scikit, scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html. Accessed 5 Nov. 2023.
17. "Sklearn.Random_projection.Gaussianrandomprojection." Scikit, scikit-learn.org/stable/modules/generated/sklearn.random_projection.GaussianRandomProjection.html#sklearn.random_projection.GaussianRandomProjection. Accessed 5 Nov. 2023.
18. "W3cubDocs." Random_projection.GaussianRandomProjection() - Scikit-Learn - W3cubDocs, docs.w3cub.com/scikit_learn/modules/generated/sklearn.random_projection.gaussianrandomprojection. Accessed 5 Nov. 2023.