OMSCS\_CS7641 HW3

Unsupervised Learning and Dimensionality Reduction

Clement Li (cli620) Fall 2020

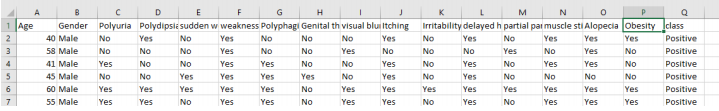
# Introduction:

The purpose of this project is to explore unsupervised learning algorithms and dimensionality reduction techniques. Unsupervised learning algorithms include k-mean clustering and expected maximization (EM). Dimensionality reduction algorithms include Principal Component Analysis (PCA), Independent Component Analysis (ICA), Randomized Projections (RP) and one feature selection algorithm of our choosing ( TO DO ) . The goal of this project is to compare and contrast these algorithms.

There are XX parts to this paper. 1) apply k-means clustering and expected maximization on two sets of data 2) apply the dimensionality algorithms 3) reapply the two clustering algorithms after the four dimensionality reduction algorithms 4) train a neural net on the dim-reduced data sets and 5) retrain the neural net with data post-processed with the cluster algorithm on top of the dim-reduce algorithms. The code will use the sk-learn libraries for GaussianMixture (EM), KMeans, PCA, ICA, RP and ( TO DO ). I modified ezerilli’s API for my own experiments and results. [1]

# About Data Sets

This section explores the differences between unsupervised learning algorithms such as k-mean clustering and expected maximization (EM). The datasets used is an early stage diabetes risk and prediction data set obtained from the UC Irving machine learning repository [2] and the default sk-learn library breast-cancer dataset [3].

This early diabetes detection dataset, pulled down from the UC Irving machine learning repository, has each row represents an individual patient. Each row describes whether they have any of the 16 characteristics of diabetes and a flag of if they were diagnosed with diabetes. Some example characteristics includes age, gender, obese, etc. There are 521 patients in this dataset. The data is mostly binary besides the age field; this would make the dataset not as interesting. To make the problem more interesting, as part of the preprocessing phase, the data is introduced to noise and compare the effects of the learning algorithms. To introduce noise, 25% of the data will be uniformly randomly replaced with ‘NA’. This is realistic because symptoms of patients may be unclear (i.e.: irritability, muscle stiffness, delayed healing) or the patient declines to disclose certain information (i.e.: genital thrush, obesity, gender). We used this in our previous analysis. 

The breast cancer data set is a binary classification dataset with 569 samples total and 30 features. This dataset is similar σto the diabetes dataset but with more features and non-binary features. The values are detailed in the following table:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Param | Min | Max | Param | Min | Max | Param | Min | Max | Param | Min | Max |
| radius (mean) | 6.981 | 28.11 | radius (std err): | 0.112 | 2.873 | radius (worst): | 7.93 | 36.04 | concave points (mean): | 0.0 | 0.201 |
| texture (mean) | 9.71 | 39.28 | texture (std err): | 0.36 | 4.885 | texture (worst): | 12.02 | 49.54 | symmetry (mean): | 0.106 | 0.304 |
| perimeter (mean) | 43.79 | 188.5 | perimeter (std err): | 0.757 | 21.98 | perimeter (worst): | 50.41 | 251.2 | fractal dim (mean): | 0.05 | 0.097 |
| area (mean): | 143.5 | 2501.0 | area (std err): | 6.802 | 542.2 | area (worst): | 185.2 | 4254.0 | concave points  (std err): | 0.0 | 0.053 |
| smoothness (mean): | 0.053 | 0.163 | smoothness (std err): | 0.002 | 0.031 | smoothness (worst): | 0.071 | 0.223 | symmetry (std err): | 0.008 | 0.079 |
| compactness (mean): | 0.019 | 0.345 | compactness (std err): | 0.002 | 0.135 | compactness (worst): | 0.027 | 1.058 | fractal dim (std err): | 0.001 | 0.03 |
| concavity (mean): | 0.0 | 0.427 | concavity  (std err): | 0.0 | 0.396 | concavity (worst): | 0.0 | 1.252 | concave points (worst): | 0.0 | 0.291 |
| fractal dim (worst): | 0.055 | 0.208 | symmetry (worst): | 0.156 | 0.664 |  |  |  |  |  |  |

# Analysis Techniques quick summary

## Akaike Information Criterion (AIC) / Bayesian Information Criterion (BIC)

Akaike and Bayesian Information Criterion are two ways of scoring a model based on its log-likelihood and complexity. Respectively, the formulas for these criterion are: AIC = -2/N \* LL + 2 \* k/N and BIC = -2 \* LL + log(N) \* k. (LL = log-likelihood, N = number of examples, k = number of parameters). These values are to be minimized for both criterions. [4] We will use this mainly to explore Expected Maximization algorithm to find the parameters (components and covariance type) with the highest log likelihood.

## Silhouettes Analysis

Silhouettes analysis can be used to study the separation distance between the resulting clusters. This measure has a range of [-1, 1] measuring how close each point in one cluster is to points of neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary. Negative values indicate those samples are assigned to the wrong cluster. The thickness shows the cluster sizes. [5] We will use this mostly to assess K-mean clustering to see how well the clusters inter-fit.

## Variance / Cumulative Variance

Variance refers to the sensitivity of the learning algorithm to the specifics of the training data, e.g. the noise and specific observations. [6] Variance is mostly used for PCA and KPCA to measure the variance in each principal component. The sum of the sample variances of all individual variables is called the *cumulative variance*. PCA is designed to maximize the first k components and minimize the variance of the last p-k components. We try to choose k big enough to make the lost information sufficiently small. [7] Hence, the k at which the cumulative variance converges, is the k we want.

## Kurtosis

|  |  |
| --- | --- |
| Kurtosis is the measure of peakedness or flatness of a distribution. See image on the right. We want K to be greater than 0, particularly as high as possible. This indicates that the values in that component is more centralized. If the values are Platykurtic (k<0) then it is more likely to spill into its neighbors. [8] |  |

## Mean Square Error (MSE) for the Reconstruction

To evaluate randomized projections, we will compare the MSE between the original data and the reconstructed data, which is the reduce feature data space weights cross the labels added to the new average data. This error should be low.

## t-SNE and PCA

To view the cluster, we visualized it in two ways: (1) via PCA where it plots out the first vs second principal component and color each of the different types of classes with a different color (2) via T-Distributed Stochastic Neighboring Entities (t-SNE) where it plots out the first and second components of the algorithm specifically reduced down to two components. [9] This ends up with two good visuals on how well the respective dimension reduction algorithms separated the data. We will also compare the results to the true classes of the data to see how well it did.

# Part 3: Clustering

## Results

### K Means

|  |  |
| --- | --- |
| *Figure 1: K-means algorithms analysis of Diabetes dataset using a Silhouette technique.  Here 2 clusters seem to have the least amount of errors indicated by the negative values. This makes sense because the dataset should only classify to positive or negative diabetes predictions.* | Figure 2: K-means algorithms analysis of Breast Cancer dataset using a Silhouette technique. It seems that there aren’t any good cluster sizes. Cluster of 2 has the least amount of errors in the negatives. All other clusters only spread out the error into the new cluster. |
|  | *Figure 3: K-means clustering with 2 clusters on the Diabetes dataset. Using the clusters found via silhouetting, we see some errors in the clusters compared to the true clusters mostly in the dividers when viewing with PCA. When viewing with TSNE, we see that most of the values are correctly classified. However, outlier samples that are embedded in the opposite clusters are easily mislabeled. It is also interesting to note that the classes are swapped.* |
|  | *Figure 4: K-means clustering with 2 clusters on the Breast Cancer dataset. These values show that the clusters are well grouped. The only errors are a grouping located at around (-10, -10) in the true TSNE plot. The PCA has good alignment but is hard to classify on the boundary between the two classes.* |

### Expected Maximization (EM)

|  |  |
| --- | --- |
| Figure 5: AIC/BIC evaluation for diabetes dataset clusters. Here the lowest criterion values for both AIC and BIC is using 10 components and diag covariance. | Figure 6: AIC/BIC evaluation for breast cancer dataset clusters. The lowest criterion value for AIC is 10 component and full covariance. However, 2 component and full has the lowest values for using BIC. Since this is a smaller dataset with fewer features, BIC is more likely to choose models that are too simple. So, we will be using 10 components and full cov. [4] |

|  |  |
| --- | --- |
|  | Figure 7: EM clustering with 2 clusters on the diabetes dataset. The clustering is much more mixed up here due to the characteristics of the algorithm. There are not many clear features in the PCA plot and classifications performed poorly. In the TSNE plot, the more grouped samples (bottom left clusters) have been classified well. Everything else did not have good classification. |
|  | Figure 8: EM clustering with 2 clusters on the Breast Cancer dataset. In both plots, the clusters are well grouped. There are still errors in the boundaries for PCA. In the TSNE plot, it was able to pick up some data points that the Kmeans clustering algorithm would had missed such as some red points at (-10, -10). |

## Analysis

In addition to the observations listed in the Figure captions, here is a table of summarized pros/cons of each algorithm.

|  |  |  |
| --- | --- | --- |
|  | Pros | Cons |
| Kmeans | Able to make good clear clusters | Can miss outliers hidden in the clusters  Clusters doesn’t directly reflect classes  Boundaries can be unclear |
| EM | Able to pick out some outliers | Performs worst than Kmeans algorithm  Can miss obvious clusters  Boundaries can be unclear |

# Dimension Reduction on Datasets

## Results

### PCA

|  |  |
| --- | --- |
| Figure 9: Variance analysis of PCA components for diabetes dataset. We want to choose the component k where cumulative variance (%) converges or reaches the peak. The best component is k = 16 (also the max). This shows that the distribution of the eigenvalues mostly in the first few components. But we cannot disregard the last few components because they still hold significant data. | Figure 10: Variance analysis of PCA components for diabetes dataset. Here the cumulative variance converges earlier than the max components at 21 components. This shows that all the data is captured within the first 21 components. More than 99% of the data is captured with the first 15 components (where it begins to converge) |

### ICA

|  |  |
| --- | --- |
| Figure 11: Analysis of kurtosis of with varying components for diabetes data. Here we show that at components of 11, we have the highest average kurtosis and where the curve starts to flatten out. | Figure 12: Analysis of kurtosis of with varying components for cancer data. The average kurtosis has two peaks, at 12 and at 30. I chose 12 because a reason why kurtosis at 30 is highest is because it indicates that there are many compacted peaks. This seems indicates overfitting. |

### RP

|  |  |
| --- | --- |
| *Figure 13: Reconstruction vs original MSE per # of component used for diabetes data. The MSE went down to 0 when we use all 16 components. There were very little variations over all the times it was re-ran. This is likely because the dimensions are small.* | *Figure 14: Reconstruction vs original MSE per # of component used for cancer data. Same as the diabetes dataset, the reconstructed data had 0 MSE at 30 components. The variations per run is higher due to having twice the dimensions.* |

### KPCA

|  |  |
| --- | --- |
| Figure 15: AIC/BIC evaluation for diabetes dataset clusters. In these variances, none of the parameters converges. However, it shows that 16 components with sigmoid or cosine ends at the lowest variance on the last component. Similar to PCA, this shows that the distribution of the eigenvalues mostly in the first few components. But we cannot disregard the last few components because they still hold significant data. Therefore, we use 16 components and sigmoid for this data. | Figure 16: AIC/BIC evaluation for breast cancer dataset clusters. Here we see some convergences while using the cosine kernel. The cumulative variance for the cosine plot converges at 1.0 at around 21 components. This shows that all the data is captured within the first 21 components. More than 99% of the data is captured with the first 15 components (where it begins to converge) We will use 21 components and cosine kernel. |

## Analysis

In addition to the observations in the figure captions, it is important to note that dimension reduction is optimized in high feature/class dimension data sets. It seems evident from the PCA and ICA plots, that the 16 features and 2 classes diabetes dataset needs the full set of components to not lose any data. For example, in contrast a dataset like breast cancer (30 features 2 classes) only need 21/30 components to retain its data. It is also noted that RP is great for even high dimension data, and both the diabetes and breast cancer does not have enough dimensions to optimize the use of RP. This is shown when RP requires all 16/30 components to have 0 loss in the reconstruction.

# Clustering on Dimension Reduced Datasets

## Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | PCA | ICA | RP | KPCA |
| Diabetes  KMeans (Complexity) |  |  |  |  |
| Diabetes KMeans (Clustering) |  |  |  |  |
| Diabetes EM (Complexity) |  |  |  |  |
| Diabetes EM (Clustering) |  |  |  |  |
| Cancer  KMeans (Complexity) |  |  |  |  |
| Cancer KMeans (Clustering) |  |  |  |  |
| Cancer EM (Complexity) |  |  |  |  |
| Cancer EM (Clustering) |  |  |  |  |

## Analysis

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Complexity | | | Clustering | | We want to analyze the effects of reducing the dimensionality prior to clustering. I will only look at only diabetes dataset because we used it in the previous assignments. We want to see if it has as good of an effect on lower dimension datasets. To be consistent for a fair comparison, a cluster of 2 is used. | |
| KMeans |  | | |  | |
| EM |  | | |  | |
| Cluster Comparison | | PCA | ICA | | RP | | KPCA |
| KMeans | | Roughly same results as KMeans and EM. Possibly because we use PCA to view the clusters for both Kmeans and EM | Performance went down compared to normal KMeans. The boundaries are less defined and not as evenly split between clusters. The output with lowered dimensions that ICA provides is more sparsely spaced in the PCA/TSNE domain and Kmeans is better at differentiating data points that better clustered. | | The RP dimension reduced dataset did not have very good clear clusters that KMean can do well on. The performance did not decrease mainly because the original KMeans also did poorly. As mentioned before, the dimensions of these data sets may not be large enough for RP to be optimally used. | | The results are roughly the same as PCA and like the regular KMeans/EM clustering. My reasoning is the same as while explaining PCA, that PCA was used to view both KMeans and EM. So, the visualization is the same. |
| EM | | Performance went up compared to normal EM classification. This is likely because EM is a method that measures max likehood which is optimal for an ICA reduced dimensions. The remaining data points are statistically best suited in two gaussian peaks. Since we chose a k component that provides the highest average kurtosis, these peaks are as statistically related as possible. | | Like KMeans, the RP output missed some obvious clusters. As mentioned above, this is likely due to this dataset not being large enough for RP to be optimally used. | |

# Neural Net with Dimensionality Reduction

## Results /Analysis

|  |
| --- |
| The first thing to note is that using the same projected data from the previous section, the reduced data proved to be harder to train for the same neural net. The f1-score for all reduced inputs haven’t exceeded directly inputting the original data. Here RP and PCA were able to maintain the f1-score meaning it is just as reliable as the original dataset results. It is interesting to note that RP and PCA has a better record of precision while giving up the same amount of recall. This indicates that more of the selected items are relevant and less of the relevant items are selected. This makes sense as we have decreased the dimensions and possibly omitting some samples that depended on that omitted feature to be selected correctly. I believe RP also uses the full 16 components to retain 0 data loss. That is the most likely reason why RP appeared to no effect on the Neural Net training. |

# Neural Net with Clustering on Dimensionality Reduction

### Results /Analysis

|  |
| --- |
| The performances for all the combinations show that KMeans and EM are optimal without including any dimension reducing. To understand the performances, we need to analyze the true clustering of these algorithm outputs. The true clustering holds the structure that is going into the neural nets, in addition we will look at the type of errors that the clustering from reduced data produces.  From the first experiment, the resulting clusters for KMeans had errors defining the boundaries, but the true clusters have well-defined shapes. The same is said about the EM clusters. This gives us the hypothesis that if the resulting true clusters has well-defined shapes then the neural net will be able to correct the belief in training.  To show this this hypothesis, we look at the other algorithm true clusters and error types. The best example is KMeans on ICA, there are very little well-defined features in the true clusters. This will insinuate that it will lead to less of the correct positives being selected and more false positives because the labels are so spread out. This is seen throughout the experiment, no matter if we are using KMeans or EM as the clustering method. The exception is RP and this was explained in the previous section; we used the max component to retain 0 data loss.  This brings us to the conclusion that neural net can correct the fuzzy borders of the KMeans and EM outputs. However, if the clustering boundaries becomes less clear the neural net performance decreases.    Time analysis (shown in the table snippet to the left) shows that ICA tends to take the longest and RP is, on average, the fastest algorithm when paired with NN and a clustering algorithm. |

# References:

[1] <https://github.com/ezerilli/Machine_Learning/tree/master/Unsupervised_Learning>

[2] <https://archive.ics.uci.edu/ml/datasets/Early+stage+diabetes+risk+prediction+dataset>

[3] <https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html>

[4] <https://machinelearningmastery.com/probabilistic-model-selection-measures/>

[5] <https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html>

[6] <https://machinelearningmastery.com/how-to-reduce-model-variance/>

[7] <https://ro-che.info/articles/2017-12-11-pca-explained-variance>

[8] <http://labs.seas.wustl.edu/bme/raman/Lectures/Lecture14_ICA.pdf>

[9] <https://towardsdatascience.com/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b>