Assignment 3: Unsupervised Learning and Dimensionality Reduction

CS7641 Machine Learning, 2019 Fall

Hui Xia (hxia40)
903459648
Georgia Institute of Technology

1. Introduction

Compared with supervised learning, unsupervised learning is closer to how humanoid react with and recognize the world – after all, nothing is labeled in the real world. Unsupervised Learning is utilized in exploratory data analysis and enables the investigation the (hidden) relationships within the dataset. Clustering algorithms such as k-Means (kM) and Gaussian Mixture Models (GMM) are used to 'cluster', and/or 'create' labels to, unlabeled data. Dimensionality Reduction (DR) algorithms focus on transform the data, reduce the dimension of the data, to avoid curse of dimensions. In this report, I will investigate the performance of these clustering and DR algorithms, i.e. kM, GMM, Principal component analysis (PCA), Independent component analysis (ICA), Gaussian Random Projection (GRP), and Factor Analysis (FA) over two datasets that have been used in both Assignments 1 and 2. As per required by the assignment, I will firstly run clustering algorithms and run DR algorithms on the datasets, respectively. Then, I will investigate how cluster methods perform on the dimensionality-reduced datasets. Finally, I will perform Neuron Network (NN) analysis on dimensionality-reduced and clustered dataset, and compare the performance of NN before and after clustering/DR.

2. Experiment design

2.1. Datasets

In this work, two datasets of similar size are analyzed. The details of these two datasets have been previously described in my report to Assignment 1. Briefly, one dataset is a slice from Fashion-MNIST. Each row of this dataset describes a 28 x 28 grayscale pixel image and thus have 784 features [1]. Each row is also associate with a label y (value of 0-9). The first 2000 rows in the train dataset, and the first 500 rows in the test dataset are used. The total size of the training and testing dataset is 5.5 MB. Another dataset is the Epileptic Seizure Recognition Data Set (ESR) from UCI Machine Learning Repository [2]. Each row of the dataset is derived by the electronic signal from brain waves, which has 178 features. Each row is associated with a label y of the value 0-4 (i.e. 5 different y-labels, which represent 5 different status: eyes open, eyes closed, healthy area, tumor area, and seizure, of the person tested). For this assignment, the first 6000 rows in the dataset are split into a 5000-row train and 1000-row test dataset. The total size of the training and testing dataset is 5.5 MB, equal to the size of dataset 1.

2.2. Experiment methods

2.2.1. Dataset clustering

In this section, kM and GMM are used to cluster the datasets under various number of clusters (i.e. `n_clusters` for kM and `n_components` for GMM) ranged from 3 to 100. The kM algorithm works by assigning k random means in the crowd of data points, and then try to minimize the distance between any given points to their mean, and maximizing distance between different clusters, iteratively, until convergence [3]. The GMM is a kind of expectation-maximization (EM) algorithm, which is similar to kM, except that GMM does not define midpoints between centers of clusters as the cluster-cluster interface. Rather, each cluster center defines a multidimensional gaussian model as their respective cluster. Thus, for any given points, there is a ratio composition on which cluster it belongs to (e.g. one point could be 98% belong to cluster A, 1.9% belong to cluster B, and 0.1% belong to other clusters). [4]

In theory, as the beginning assignment of cluster 'centers' is a random process, it is favorable to run the kM and GMM algorithms several times to find the best clustering result. However, in practice, after running the algorithm for 5 times, the results of clustering (evaluated by SS and DBS, which are described below) change in a 1% range. Thus, except in section 2.2.1, k-means and GMM are only run for once to simplify the experiments.

Both k-means and gaussian mixtures depend critically on the measure of distance between examples inside of a cluster, and between samples in different clusters. Two methods are used to evaluate the effectiveness of the clustering method: Silhouette Score and Davies–Bouldin Score (DBS). The Silhouette Score is calculated using the equation of:

$$Silhouette\ Score = \frac{b - a}{\max(a, b)}$$

Where *a* stands for the mean intra-cluster distance and *b* stands for the mean nearest-cluster distance for each sample [5]. DBS is defined as the average similarity measure of each cluster with its most similar cluster, where similarity is calculated as the ratio of within-cluster distances to between-cluster distances. [6] Thus, both SS and DBS measure and prefer high inter-cluster distance and low intra-cluster distance. For each size of cluster, their respective SS and DBS are evaluated and plotted against both the training and testing datasets. The number of the clusters that grant the maximum SS and minimum DBI will be adopted as the optimal.

2.2.2. Dataset dimensionality reduction

In this section, PCA, ICA, GRP, and FA are used to perform dimensionality-reduction on the datasets to various number of features [7-10] (i.e. `n_clusters` for FA and `n_components` for the rest), which ranged from 1 to the size of the dataset itself (i.e. no dimension reduction). Eigenvalue and kurtosis are used to evaluate the performance of PCA and ICA, respectively. To facilitate comparison among all DR algorithms, a Decision Tree (DT) classifier is also used to evaluate the performance of all algorithms: the datasets that have been dimensionally reduced (to various number of features) by the algorithms, are fitted on the training dataset, then predicted on both the training and the testing dataset using the DT. The performance of the DT is then plotted against the number of features. Generally, the lowest number of features that provide similar accuracy score (using DT classification) compared with the pre-dimension-reduction dataset is adopted as the optimal features. This step is taken to reduce the side of dimensions as much as possible, while keeping as much information as possible. The details on how to choose the optimal number of features will be discussed in the **Results and Discussion** section.

2.2.3. Dataset clustering after dimensionality reduction

In this section, kM and GMM are used to cluster the datasets that have been dimension-reduced in **Section 2.2.2.** Similar to what is performed in **Section 2.2.1**, the SS and DBS values derived from dataset clustering after dimensionality reduction are plotted against the size of the clusters. As descried in **Section 3.2,** dataset 1 are reduced to 20 features, and dataset 2 is reduced to 90 features, using all four DR algorithms.

2.2.4. Dataset dimensionality reduction and dataset clustering for neuron network

A neuron network (sklearn.neural_network.MLPClassifier) is used to predict both of the datasets (although the assignment only required to predict 1 dataset here), which have been clustered in **Section 2.2.1**, and the datasets that have been dimension-reduced in **Section 2.2.2.** The hyperparameter of NN have been optimized in Assignment 1: `hidden_layer_size = (50, ), alpha = 6.25` for dataset 1, and `hidden_layer_size = (50, ), alpha = 0.417` for dataset 2.

3. Results and Discussion

3.1. Dataset clustering

As discussed in **Section 2.2.1**, datasets 1 and 2 are clustered using kM and GMM under various number of clusters. The effectiveness of the clustering is evaluated using two scores: SS and DBS. **Figure 1** demonstrate the plotted curve along the increase of cluster size. While there lacks effective clustering on dataset 1, the SS score (which as a maximum of 1) show that under the cluster number of 3, both kM and GMM could cluster dataset 2 quite well, getting an SS score above 0.6 and 0.4, respectively. This result is sort of expected, as dataset 1 is about sorting different fashion of clothes in the form of pictures, which falls in the realm of computational vision, and is hard for the algorithm to 'get' and hidden logic on the sorting, which out providing any labels. In other words, picture makes more sense to human eyes that to a simple algorithm that is designed for clustering. On the other hand, although electronic signal from brain waves are not intuitively meaningful to human eyes, the kM algorithm could successfully sort the data (which actually have five labels) into three clusters.

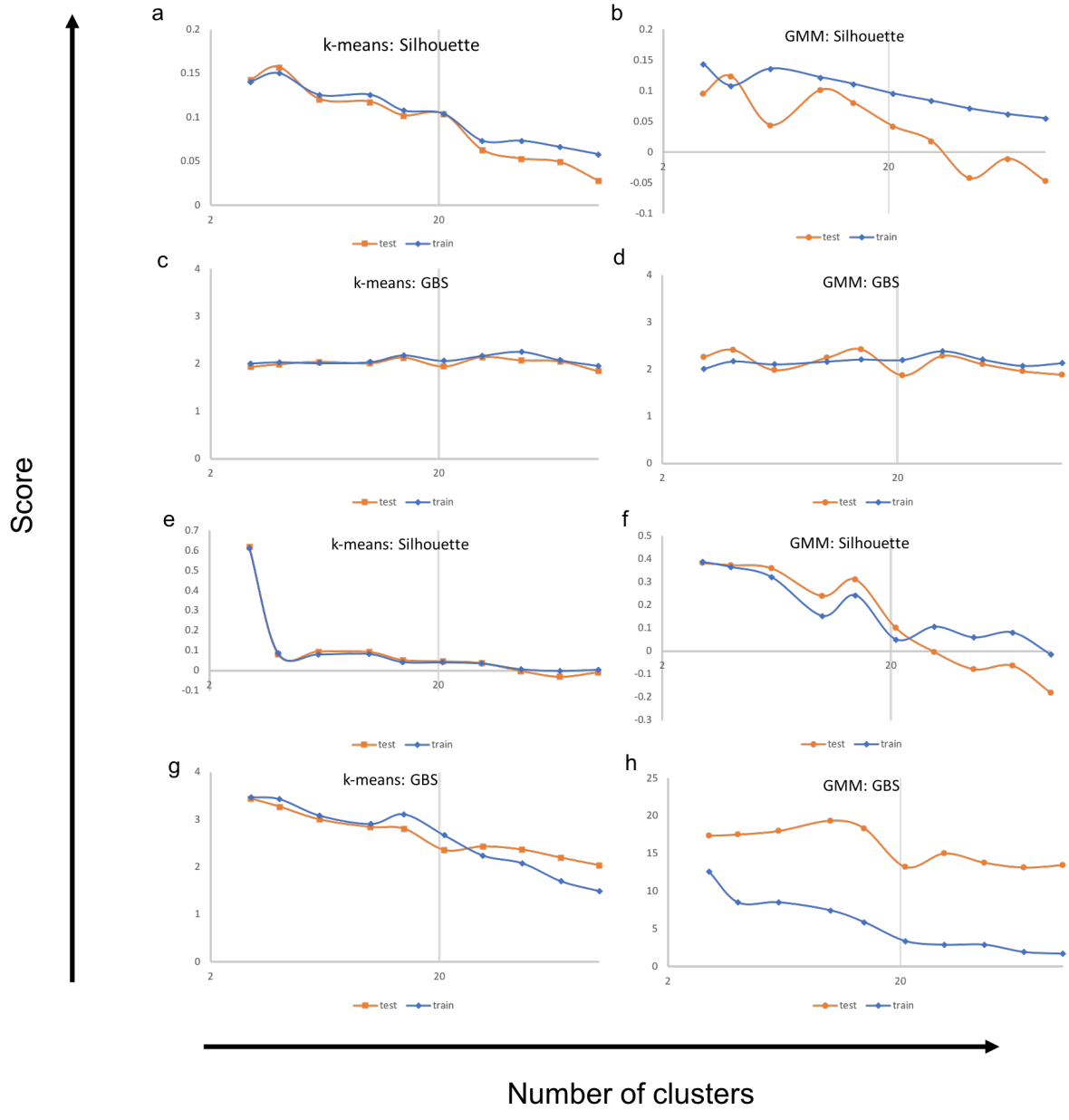

Figure 1: Dataset clustering using k-means and GMM, measured by SS and DBS . a-d) Dataset clustering for dataset 1. e-h) Dataset clustering for dataset 2.

Beyond the above-mentioned cases, neither kM, nor GMM performs well in other scenarios. However, the cluster number of 3 is still more favorable compared with other number of clusters, due to that when adopting 3 clusters, SS will consistently show a higher value.

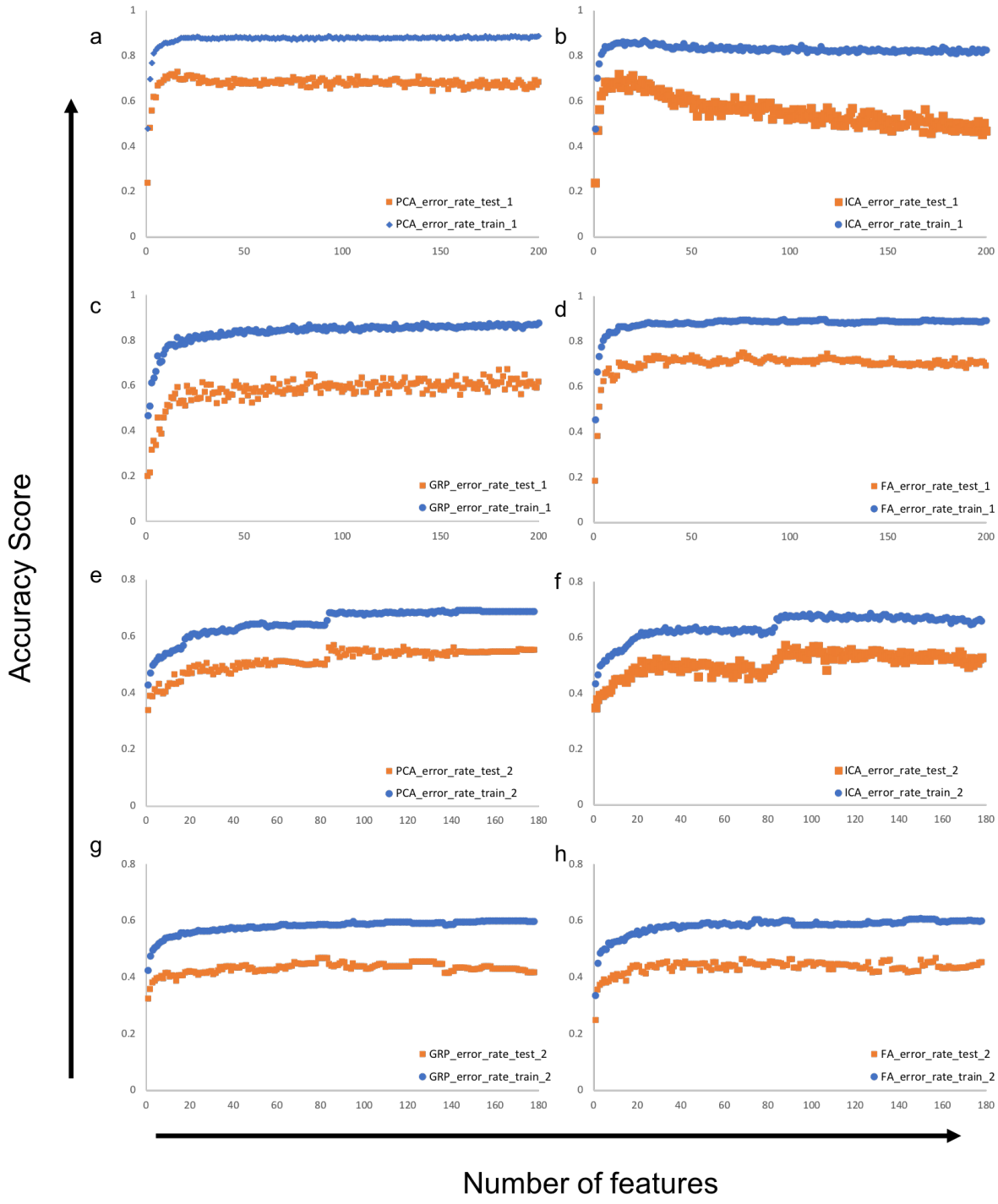

Figure 2: Dataset dimensionality reduction using PCA, ICA, GRP, and FA. a-d) **:** Dataset dimensionality reduction for dataset 1. e-h) **:** Dataset dimensionality reduction for dataset 2.

3.2. Dataset dimensionality reduction

As discussed in **Section 2.2.2**, to facilitate the comparison between the four different DR algorithms, a DT classifier is used to evaluate the DR. That is, if after dimension reduction, DT classifier can still sort the dataset with similar (if not higher) accuracy score compared with the dimension-reduction-free dataset, we would consider that this DR is favorable. As shown in **Figure 2**, all four algorithms perform similarly for both datasets. For dataset 1, which has 784 features, the accuracy score derived from the DT classifier will reach maximum value when number features equal to about 20. For dataset 2. For dataset 1, which has 179 features, the accuracy score derived from the DT classifier will reach maximum value when number features equal to about 90.

Interestingly, **Figure 2a-b** shows that for dataset 1, when the feature number is taken much larger than 20, the classifier accuracy will drop down. I reason that this is due to the curse of dimensions. If the dimension value is too high, the total real of all data become 'emptier', and thus the number of datapoints will become relatively 'less'. In another point of view, when there are too many features after DR, the accuracy score the test set of data gets more far-away from the training curve (for **Figure 2b**), or the training/test curve does not show the trait to converging (for **Figure 2a**), the model is suffering from a 'overfitting', i.e. high variance and low bias problem. Especially, for the ICA DR on dataset 1 (**Figure 2b**), this high variance and low bias problem is more pronounced, as increasing the feature number will increase the variance of the data.

On the other hand, as shown in **Figure 2e-f**, the DT accuracy score for both the training and the testing dataset 2 experience an obvious improvement. This observation is also interesting, as even though dataset 2 has less features (179 features) compared with dataset 1 (784 features), to maintain favorable result, for dataset 2, dimension reduction can only reduce the number of features to 90, but we can reduce the dimension of dataset 1 to 20. This observation indicates that the 'information value' of each common feature in dataset 1 is not as 'valuable' as that in dataset 2. The eigenvalue study for PCA and kurtosis study for ICA also supports such observation. As shown in **Figure 3a-d,** eigenvalues show that even dataset has 784 features, there are several features that has superior high eigenvalues compared with the rest of the points, making them less important. However, for dataset 2, there are more points that are of high eigenvalues - all of the features are somehow important. Similarly, for dataset 1, there are several features that has superior high kurtosis values, while in dataset 2, there are much more features that hold important information (i.e. high kurtosis than average) that are too important to be dimensionally reduced.

3.3. Dataset clustering after dimensionality reduction

Similar to what has happened in **Section 3.1**, datasets 1 and 2 that have gone through the DR process are clustered using kM and GMM under various number of clusters. Two score methods, SS and DBS, are used to evaluate the clustering. **Table 1** demonstrates the optimum number of cluster, and their respective SS/DBS score, separated with a comma. Most of the data shown are basically noise, which indicate that the dataset if unsuitable to clustering. However, there are some scenarios that the clustering is favorable (i.e. high in SS), which are marked in red.

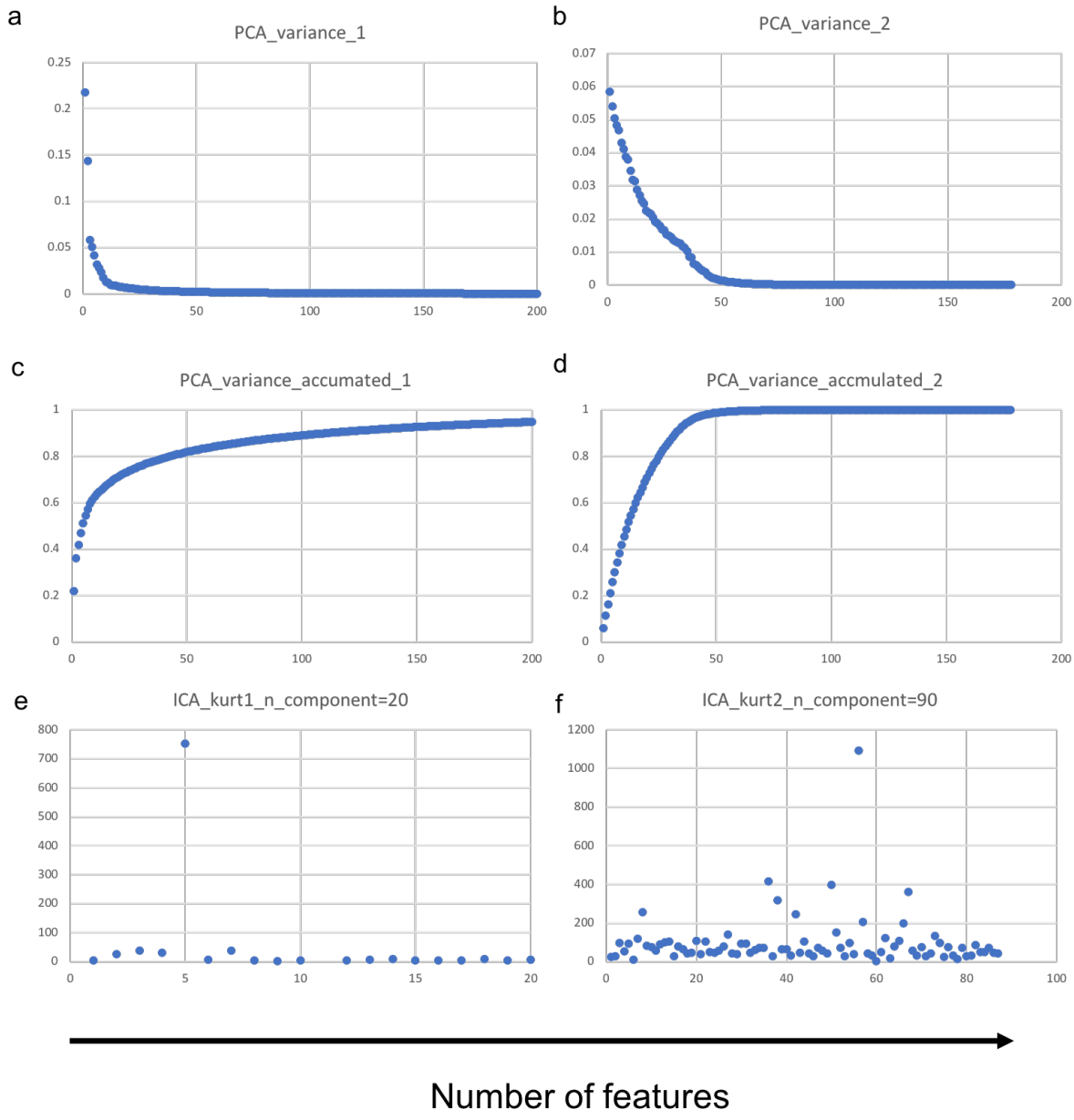

Figure 3: Dataset dimensionality reduction using PCA, and ICA. a-b) **:** Eigenvalues after PCA dimensionality reduction for datasets 1-2 to different number of features. c-d) **:** Accumulated eigenvalues after PCA dimensionality reduction for datasets 1-2 to different number of features. e-f) **:** Kurtosis values after ICA dimensionality reduction for datasets 1-2 to different number of features.

While the DBS failed to find any meaningful number clustering, as shown in **Table 1**, PCA is not able to preserve any meaningful clustering after they have dimension-reduced the datasets. However, ICA, GRP, and FA are able to preserve, to some extent, the ability of GMM or k-means to cluster in dataset 2. The poor performance on dataset 1 (MNIST-fashion) on clustering can be explained by that MNIST itself is hard to cluster anyways, as it is composed on pixels and the position information (that is 2-dimensional) of each pixel is hidden in a 1-dimensional array to the algorithm. Also, for dataset 2, in the process of PCA dimension reduction, only nearly half of the features (90 in 174) are saved. It is possible that the features that enabled clustering using k-means and GMM in the original dataset were lost in the transformation.

Table 1. Optimal number of clusters, and their respective score, on datasets before and after dimensionality reduction. The optimal number of clusters and their respective score are separated using comma. Optimal number of clusters with meaningfully high score are marked in red.

	dimension-reduction-free			
	dataset 1		dataset 2	
	k-means	GMM	k-means	GMM
SS	4, 0.15	3, 0.14	3, 0.61	3, 0.38
DBS	100, 1.96	3, 2.01	100, 1.49	100, 1.69
	PCA			
	dataset 1		dataset 2	
	k-means	GMM	k-means	GMM
SS	4, 0.24	3, 0.17	10, 0.09	4, -0.05
DBS	21, 1.29	21, 1.29	100, 1.53	100, 2.54
	ICA			
	dataset 1		dataset 2	
	k-means	GMM	k-means	GMM
SS	21, 0.19	14, 0.13	3, 0.01	4, 0.50
DBS	31, 1.28	31, 1.28	100, 1.28	100, 2.63
	GRP			
	dataset 1		dataset 2	
	k-means	GMM	k-means	GMM
SS	3, 0.17	3, 0.17	4, 0.62	3, -0.04
DBS	100, 1.73	100, 1.73	100, 1.50	100, 2.00
	FA			
	dataset 1		dataset 2	
	k-means	GMM	k-means	GMM
SS	14, 0.24	6, 0.17	3, 0.61	21, 0.06
DBS	14, 1.20	14, 1.20	100, 1.57	100, 2.39

3.4. Dataset dimensionality reduction and dataset clustering for neuron network

As shown in **Table 2**, even though dataset 1 have been reduced to only 20 features from the original of 784, dimension-reduction by PCA still managed to preserve nearly all the information – neuron network using PCA DR dataset can provide the accuracy score that is 95% as good as the original, not-dimension-reduced dataset. On the other hand, dataset 2 has been reduced to 90 features from the original of 178. Under such condition, nearly all DR algorithm can produce the dataset that looks nearly as good (>= 97%), or even better than, the original dataset when analyzed using a neuron network.

For dataset 1, it is not surprising to see that most of the feature reduction are not able to beat the original dataset. In the process of feature reducing from 784 to 20, it is reasonable that some useful information was discarded. However, reducing the feature number itself to such a small value is already quite meaningful, as the dimension-reduced dataset will be more computational cost-effective. As a more complex dataset (i.e. information is spread out in more data points, rather than concentrated), the DR on dataset 2 using ICA should be considered as more successful, as such process IMPROVED the prediction accuracy score when predicting the testing dataset.

Table 2. Accuracy score when predicting the testing datasets before and after dimension reduction.

	Dataset 1			Dataset 2	
Dataset	accuracy score	% to original	**Dataset**	accuracy score	% to original
Original	0.846	100.0%	**original**	0.638	100.0%
PCA	0.810	95.7%	**PCA**	0.621	97.4%
ICA	0.810	95.8%	**ICA**	0.648	101.6%
GRP	0.752	88.9%	**GRP**	0.618	97.0%
FA	0.707	83.6%	**FA**	0.623	97.8%

It is however worth to note that the clustering algorithms cannot perform similarly good when they are used for dimensionally reduction purposes. As shown in **Table 3**, under various number of clusters, the clustering algorithms can perform 80% as good as the original dataset (for dataset 1, when number of clusters equal to 50). Moreover, I can see an overfitting trend when a higher number of clusters are used. This observation is not surprising, as the clustering algorithms are not designed to preserve information in the dimension reduction process. It worth to note that when the number of clusters increases monotonically, dataset 1 will reach overfitting when dataset 2 still can be benefited from increase of number of clustering. This observation indicates that dataset 1 is simpler compared with dataset 2. That is, features in dataset 1 can be sorted into less clusters compared with the features form dataset 2.

Table 3. Accuracy score when predicting the testing datasets before and after dimension reduction using the clustering algorithms.

number pf clusters	3	5	10	20	50	100	200
	dataset 1						
Original	0.846						
k-means	0.269	0.359	0.545	0.635	0.683	0.467	0.289
k-means % to original	31.75%	42.42%	64.45%	75.12%	80.81%	55.21%	34.12%
GMM	0.206	0.345	0.419	0.601	0.655	0.479	0.222
gmm % to original	24.41%	40.76%	49.53%	71.09%	77.49%	56.64%	26.30%
	dataset 2						
Original	0.638						
k-means	0.238	0.264	0.314	0.301	0.358	0.365	0.387
k-means % to original	37.39%	41.44%	49.28%	47.19%	56.08%	57.25%	60.65%
GMM	0.207	0.368	0.373	0.263	0.268	0.278	0.287
gmm % to original	32.42%	57.78%	58.43%	41.31%	42.09%	43.66%	44.97%

4. Conclusion

In this report, I have compared four different dimension reduction algorithms and two clustering algorithms over two datasets. To me, the take-home message is, come datasets are more 'dimension-reduction worth' compared with others. E.g. in this report, dataset 1 is more 'dimension-reduction worth'. However, dimension-reduction should be considered as a

powerful too in general, as even for dataset 2, nearly half of the features can be reduced, and the dimension-reduced datasets can actually perform better when analyzed using a neuron network.

References

[1] Fashion MNIST: An MNIST-like dataset of 70,000 28x28 labeled fashion images (2017, August 25). Retrieved from https://www.kaggle.com/zalando-research/fashionmnist
[2] Epileptic Seizure Recognition Data Set (2017, May 24). Retrieved from https://archive.ics.uci.edu/ml/datasets/Epileptic+Seizure+Recognition
[3]. sklearn.cluster.KMeans (2019). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
[4]. sklearn.mixture.GaussianMixture (2019). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html#sklearn.mixture.GaussianMixture
[5]. sklearn.metrics.silhouette_score (2019). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html
[6]. sklearn.metrics.davies_bouldin_score (2019). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.davies_bouldin_score.html
[7]. sklearn.decomposition.PCA (2019). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
[8]. sklearn.decomposition.FastICA (2019). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.FastICA.html
[9]. sklearn.random_projection.GaussianRandomProjection (2019). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.random_projection.GaussianRandomProjection.html
[10]. sklearn.decomposition.FactorAnalysis (2019). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.FactorAnalysis.html