Assignment 3: Unsupervised learning and dimensionality reduction

Name: Huan Wang
Username: hwang680

1. Datasets

The first data set I used is Wisconsin Diagnostic Breast Cancer (WDBC)[1]. Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. The dataset is predicting field 2, diagnosis: B = benign, M = malignant. Data sets are linearly separable using all 30 input features. And the best predictive accuracy obtained using one separating plane in the 3-D space of Worst Area, Worst Smoothness and Mean Texture. Estimated accuracy 97.5% using repeated 10-fold crossvalidations. Classifier has correctly diagnosed 176 consecutive new patients as of November 1995. The second data set I used is Optical Recognition of Handwritten Digits Data Set[2]. The dataset consists of images of hand-written digits. Each data is a 8*8 image of a digit. The feature space has a dimension of 64* 1797. The dataset is predicting hand-written digits to be one of 10 classes (digit 0 -9).

2. Clustering

Clustering methods

Sklearn's KMeans and BayesianGaussianMixture are used to carry out kmeans clustering and expectation maximization. Both of them by default use k-means++ (producing next center proportional to its distances to all current centers) for initialization, and KMeans uses LIoyd's method for iterating and improving. K-means is equivalent to the expectation-maximization algorithm with a small, all-equal, diagonal covariance matrix.

Clustering performance evaluation metrics

To evaluate unsupervised learning methods,silhouette coefficient, mutual information, completeness, homogeneity are calculated. Silhouette coefficient is a good indicator without true labels: it calculates the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample, and return coefficient (b - a) / max(a, b), for which values near 0 indicates overlapped clusters and 1 means perfect separation. But notice this metric is highly computationally demanding.

Having advantages of ground truth labels, I also used other metrics such as mutual information , completeness, homogeneity. Given the knowledge of the ground truth class assignments, Mutual Information is a function that measures the agreements of the two agreement of the two assignments. Adjusted Mutual Information is an adjustment of the Mutual Information score to account for chance. AMI scores a value of 1 when two partitions are identical. Random partitions have an expected AMI of ~0 on average hence can be negative. Completeness measures the degree of members of same class assigned to same cluster. A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster. Completeness score from 0.0 to 1.0 and 1.0 represents perfectly complete labeling. Homogeneity measures degree of a class dominates a cluster. A clustering result satisfies homogeneity if all of its clusters contain only data points which are members of a single class. Homogeneity scores from 0 and 1 and 1 represents perfectly homogeneous labeling.

Clustering without dimensionality reductions using KM and EM

As shown in Figure 1, KM perform smoother against number of clusters than EM on both datasets. This is possibly due to the assumption of Gaussian mixture distribution made by EM isn't a good approximation when most of features are not continuous value that subject to central limit theorem. Homogeneity score increases as number of clusters grow. However, this doesn't validate that increasing number of clusters should be increased. Since as number of clusters goes to extreme value when every data point forms a cluster of its own would score in homogeneity perfectly. As shown in Figure 1, for both dataset 1 and dataset 2, a spike of mutual information/rand index/ v-measure appears around some value of number of clusters and goes down afterwards. This spike indicates optimal value of number of clusters.

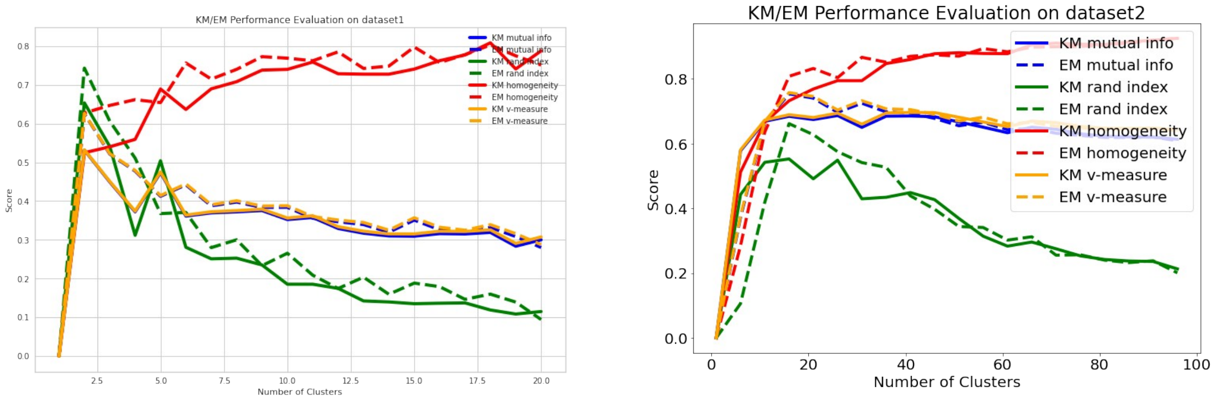
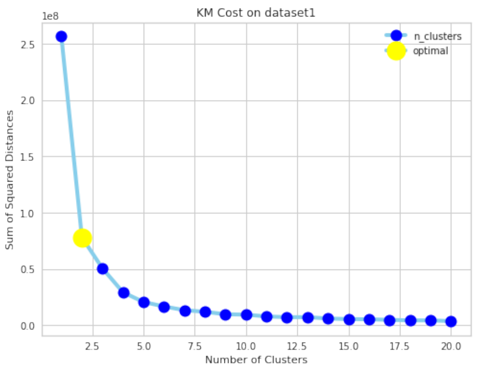
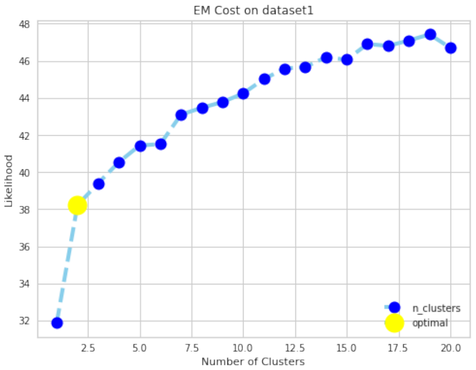


Figure 1. KM/EM performance evaluation on dataset 1 (left) and dataset 2(right).





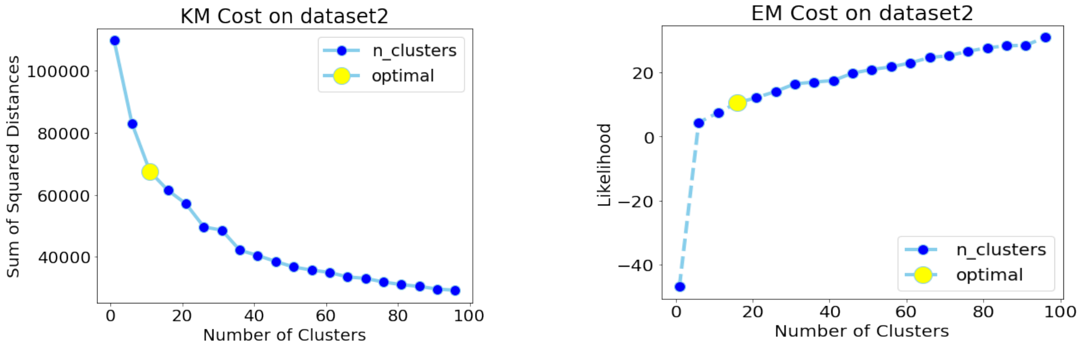


Figure 2. Sum of squared errors of KM and mean log-likelihood of EM method Using different number of clustersing for dataset 1 and dataset 2

Optimal number of clusters

Sum of squared errors of KM and mean log-likelihood of EM method are plotted against number of clusters in Figure 2 for data set 1 and dataset 2. As determined by the Elbow Method, optimal number of clusters could be determined using these plots. As shown in both figures (Figure 1 and Figure 2), I determined number of clusters of 2 to be the best value for KM and EM methods for data set 1. And since this data set 1 is labeled datasets for breast cancers(labeled B or M) , number of clusters here actually agrees well with its physical meanings. I determined number of clusters of 10 to be the best value for KM and EM methods for data set 2. And since this data set 2 is labeled datasets for hand-written digits , number of clusters here actually agrees well with its physical meanings.

Time cost:

With both max iterations set at 100 and number of initiations set at 1, EM are about 2 to 3 magnitude order slower than KM at the expenses of modeling Dirichlet distribution. Also as shown in Figure 3, Kmeans method is scalable with number of clusters while EM is not scalable.

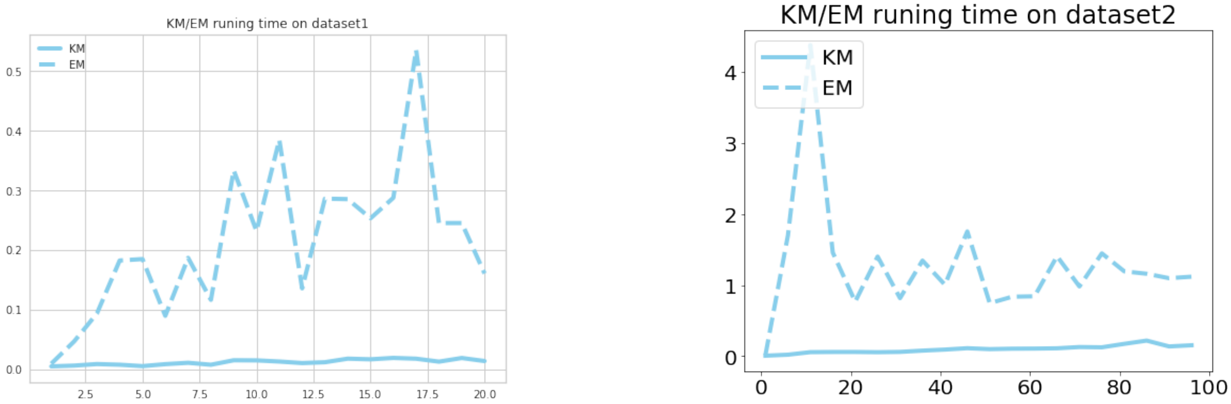


Figure 3. Time cost comparison of KM and EM for data set 1 (left) and data set 2(right)

3. Dimensionality reduction

In this section, Besides PCA, ICA and RP, the three dimensionality reduction methods, truncated singular value decomposition (SVD) is used for comparison. SVD does not center data and works well in practice with sparse data structure. This part is meant to explore the optimal dimensions which balance between loss of information and gain of time and space efficiency for each method.

3.1 Principle component analysis

Choosing number of features

To find the optimal number of components to be used for PCA, kneed Python package is used. It finds elbow point with maximum curvature. A sensitivity of 1.0 is adopted to find elbows. For dataset1, at number of feature at 7, an elbow is observed for variance/ cum variance plots. For dataset2, at number of feature at 13, an elbow is observed for variance/cum variance plots.

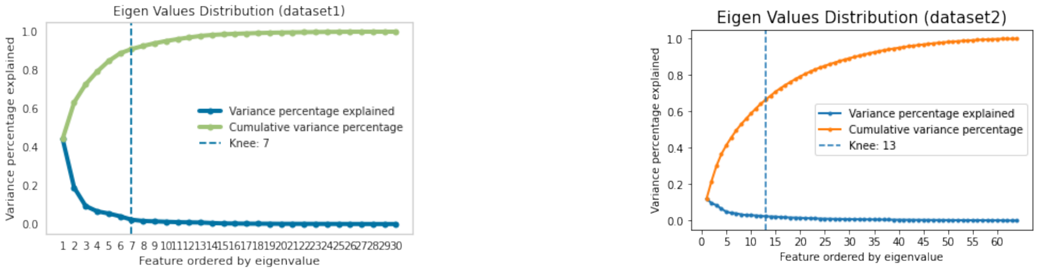
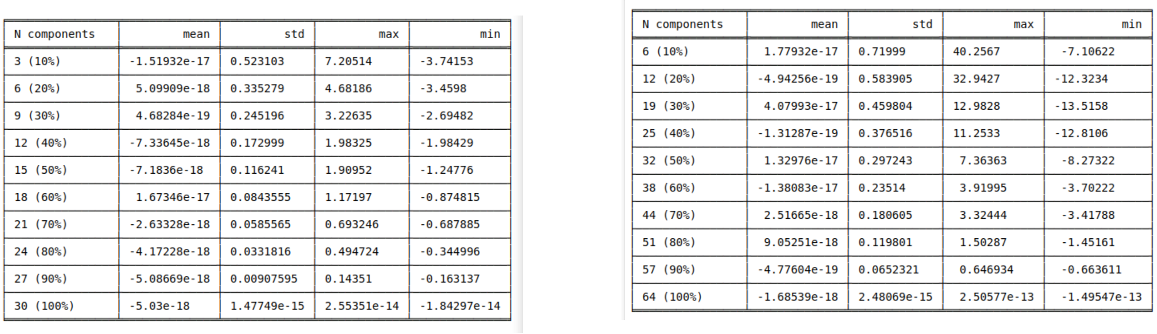


Figure 4. Eigenvalue distribution of PCA analysis for dataset 1(left) and dataset 2(right)



N components	mean	std	max	min
3 (10%)	-1.51932e-17	0.523103	7.20514	-3.74153
6 (20%)	5.09909e-18	0.335279	4.68186	-3.4598
9 (30%)	4.68284e-19	0.245196	3.22635	-2.69482
12 (40%)	-7.33645e-18	0.172999	1.98325	-1.98429
15 (50%)	-7.1836e-18	0.116241	1.90952	-1.24776
18 (60%)	1.67346e-17	0.0843555	1.17197	-0.874815
21 (70%)	-2.63328e-18	0.0585565	0.693246	-0.687885
24 (80%)	-4.17228e-18	0.0331816	0.494724	-0.344996
27 (90%)	-5.08669e-18	0.00907595	0.14351	-0.163137
30 (100%)	-5.03e-18	1.47749e-15	2.55351e-14	-1.84297e-14

N components	mean	std	max	min
6 (10%)	1.77932e-17	0.71999	40.2567	-7.10622
12 (20%)	-4.94256e-19	0.583905	32.9427	-12.3234
19 (30%)	4.07993e-17	0.459804	12.9828	-13.5158
25 (40%)	-1.31287e-19	0.376516	11.2533	-12.8106
32 (50%)	1.32976e-17	0.297243	7.36363	-8.27322
38 (60%)	-1.38083e-17	0.23514	3.91995	-3.70222
44 (70%)	2.51665e-18	0.180605	3.32444	-3.41788
51 (80%)	9.05251e-18	0.119801	1.50287	-1.45161
57 (90%)	-4.77604e-19	0.0652321	0.646934	-0.663611
64 (100%)	-1.68539e-18	2.48069e-15	2.50577e-13	-1.49547e-13

Table 1. Differences (mean, std, max and min) between reconstructed features and original features using PCA ranging from 10% to 100% of the features for dataset 1 (left table) and dataset 2 (right table)

Reconstruction using PCA

As shown in Table 1, with less percentage of features, range of differences between reconstructed and original and also standard deviation increase. This means that the more information is compressed, the more challenging the reconstruction is. Besides, reconstruction of PCA is high fidelity. While using 100% features, it is capable of completely reconstructed the original features.

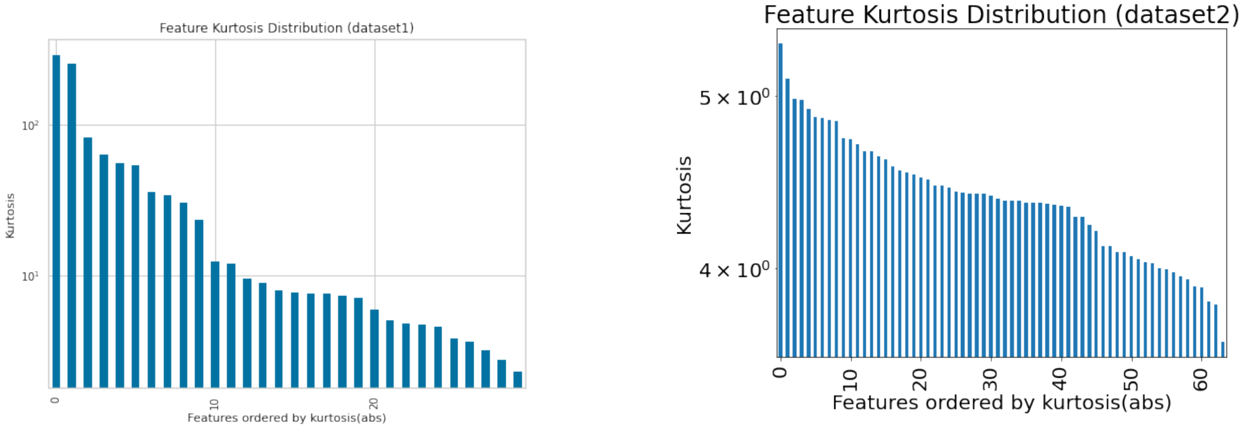


Figure 5. Kurtosis distribution of ICA analysis for dataset 1(left) and dataset 2(right)

3.2 Independent component analysis

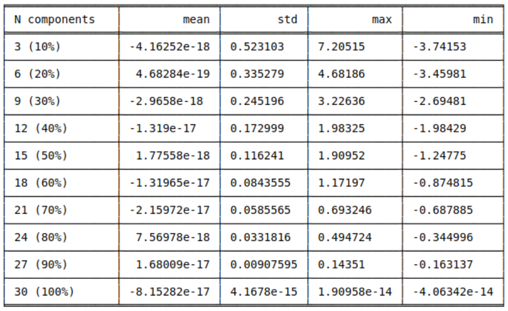
Choosing number of features

ICA[11] assumes that the observed features are actually blended from independent sources and it wants to sort out the "original" hidden variables guided by central limit theorem that: the more determinants in a process to generate a random variable, the closer it approaches Gaussian distribution. Thus, Kurtosis which measures tailedness of the probability distribution, can be used to rank features. The higher absolute value, the simpler the feature is and more likely to be the independent variables ICA seeks.

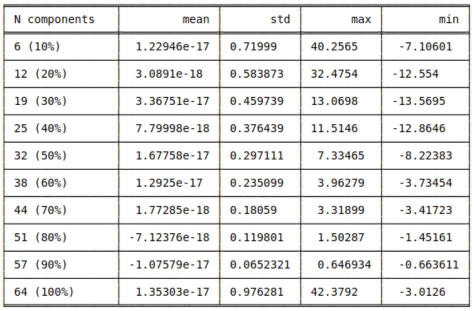
In Figure 5, Kurtosis of features for dataset 1 and dataset 2 are shown. Adopting similar methods, an elbow was found at 9 for dataset 1 and 8 for dataset 2.

Reconstruction

As shown in Table 2, ICA aren't as high fidelity as PCA. When using 100% features, there are big difference between original and reconstructed data (0.976 std and max of 42.3792 difference) even when no dimension is compressed.



N components	mean	std	max	min
3 (10%)	-4.16252e-18	0.523103	7.20515	-3.74153
6 (20%)	4.68284e-19	0.335279	4.68186	-3.45981
9 (30%)	-2.9658e-18	0.245196	3.22636	-2.69481
12 (40%)	-1.319e-17	0.172999	1.98325	-1.98429
15 (50%)	1.77558e-18	0.116241	1.90952	-1.24775
18 (60%)	-1.31965e-17	0.0843555	1.17197	-0.874815
21 (70%)	-2.15972e-17	0.0585565	0.693246	-0.687885
24 (80%)	7.56978e-18	0.0331816	0.494724	-0.344996
27 (90%)	1.68009e-17	0.00907595	0.14351	-0.163137
30 (100%)	-8.15282e-17	4.1678e-15	1.90958e-14	-4.06342e-14



N components	mean	std	max	min
6 (10%)	1.22946e-17	0.71999	40.2565	-7.10601
12 (20%)	3.0891e-18	0.583873	32.4754	-12.554
19 (30%)	3.36751e-17	0.459739	13.0698	-13.5695
25 (40%)	7.79998e-18	0.376439	11.5146	-12.8646
32 (50%)	1.67758e-17	0.297111	7.33465	-8.22383
38 (60%)	1.2925e-17	0.235099	3.96279	-3.73454
44 (70%)	1.77285e-18	0.18059	3.31899	-3.41723
51 (80%)	-7.12376e-18	0.119801	1.50287	-1.45161
57 (90%)	-1.07579e-17	0.0652321	0.646934	-0.663611
64 (100%)	1.35303e-17	0.976281	42.3792	-3.0126

Table 2. Differences (mean, std, max and min) between reconstructed features and original features using ICA ranging from 10% to 100% of the features for dataset 1 (left table) and dataset 2 (right table)

3.3 Gaussian Random Projection and Truncated Singular Value Decomposition

Choosing number of components

For Gaussian Random Projections, n_components can be automatically adjusted according to the number of samples in the dataset and the bound given by the Johnson-Lindenstrauss lemma which is controlled by eps parameter. However, target space goes larger than original even at eps = 0.99999, the number of components for PCA are thus applied on RP considering PCA as the gold standard of feature reduction.

SVD is also a feature reduction method based on eigen values however it doesn't center the data and uses efficient eigen solver. Similar elbow method is used to find optimal number of components: 7 for data set 1 and 13 for data set 2.

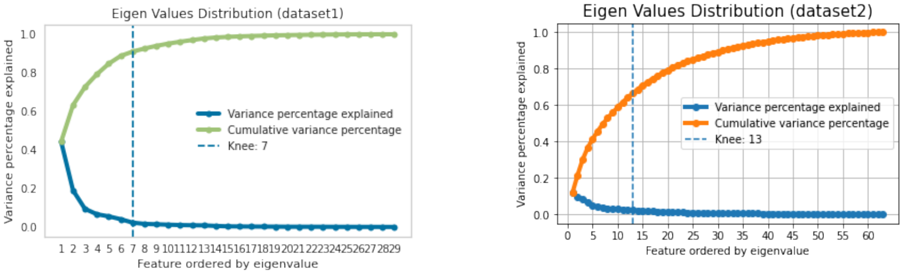
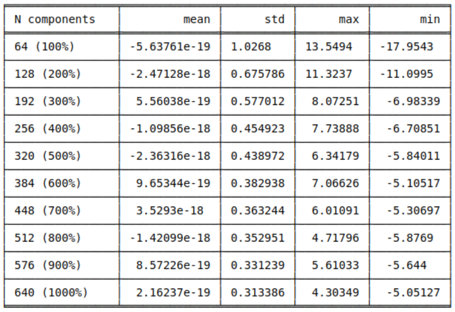


Figure 6. Eigenvalue distribution of SVD analysis for dataset1 (left) and dataset2 (right)

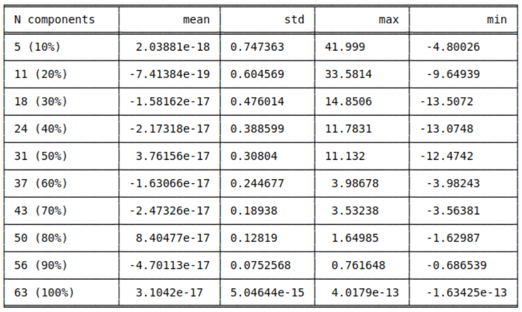
Reconstruction

Even using same n_components, transformed results of RP vary between runs due to the randomness of RP. Therefore RP was run 10 times using different random_state to reconstruct 10 folds of original data space to average out the randomness.

Even though the reconstruction of RP actually performs better with more dimensions, it's performed worse comparing with other 3 methods (STD, max - min). SVD are better than RP but worse than ICA on dataset2 reconstruction. The loss of information might be pseudo-inverse matrix it uses for calculating singular values. **Thus, the reconstruction fidelity rank is: PCA > ICA > SVD > RP**



N components	mean	std	max	min
64 (100%)	-5.63761e-19	1.0268	13.5494	-17.9543
128 (200%)	-2.47128e-18	0.675786	11.3237	-11.0995
192 (300%)	5.56038e-19	0.577012	8.07251	-6.98339
256 (400%)	-1.09856e-18	0.454923	7.73888	-6.70851
320 (500%)	-2.36316e-18	0.438972	6.34179	-5.84011
384 (600%)	9.65344e-19	0.382938	7.06626	-5.10517
448 (700%)	3.5293e-18	0.363244	6.01091	-5.30697
512 (800%)	-1.42099e-18	0.352951	4.71796	-5.8769
576 (900%)	8.57226e-19	0.331239	5.61033	-5.644
640 (1000%)	2.16237e-19	0.313386	4.30349	-5.05127



N components	mean	std	max	min
5 (10%)	2.03881e-18	0.747363	41.999	-4.80026
11 (20%)	-7.41384e-19	0.604569	33.5814	-9.64939
18 (30%)	-1.58162e-17	0.476014	14.8506	-13.5072
24 (40%)	-2.17318e-17	0.388599	11.7831	-13.0748
31 (50%)	3.76156e-17	0.30804	11.132	-12.4742
37 (60%)	-1.63066e-17	0.244677	3.98678	-3.98243
43 (70%)	-2.47326e-17	0.18938	3.53238	-3.56381
50 (80%)	8.40477e-17	0.12819	1.64985	-1.62987
56 (90%)	-4.70113e-17	0.0752568	0.761648	-0.686539
63 (100%)	3.1042e-17	5.04644e-15	4.0179e-13	-1.63425e-13

Table 2. Differences (mean, std, max and min) between reconstructed features and original features using ICA ranging from 10% to 100% of the features for dataset 1 using RP(left table) and SVD (right table)

4.Clustering on dimensionality reduced data

Visualization of Clustering on dimensionality reduced data

As shown in Figure 7, a few trend has been observed in visualization of clustering on dimensionality reduced data. 1)PCA and SVD tends scatter data points across the space (most variance); RP’s data points cluster around mean in new feature space (closet to Gaussian distribution); in contrast to PCA and SVD, data points in ICA space densely cluster to each other (most kurtosis), which can be supported by the smallest variation (a side effect of favoring simple form ‘source’ features). PCA and ICA re-center the data, while SVD and RP don’t, inferred by the the symmetry of sample distribution of PCA and ICA around origin. Noticeably, new features generated by ICA are very small in absolute value compared with others.

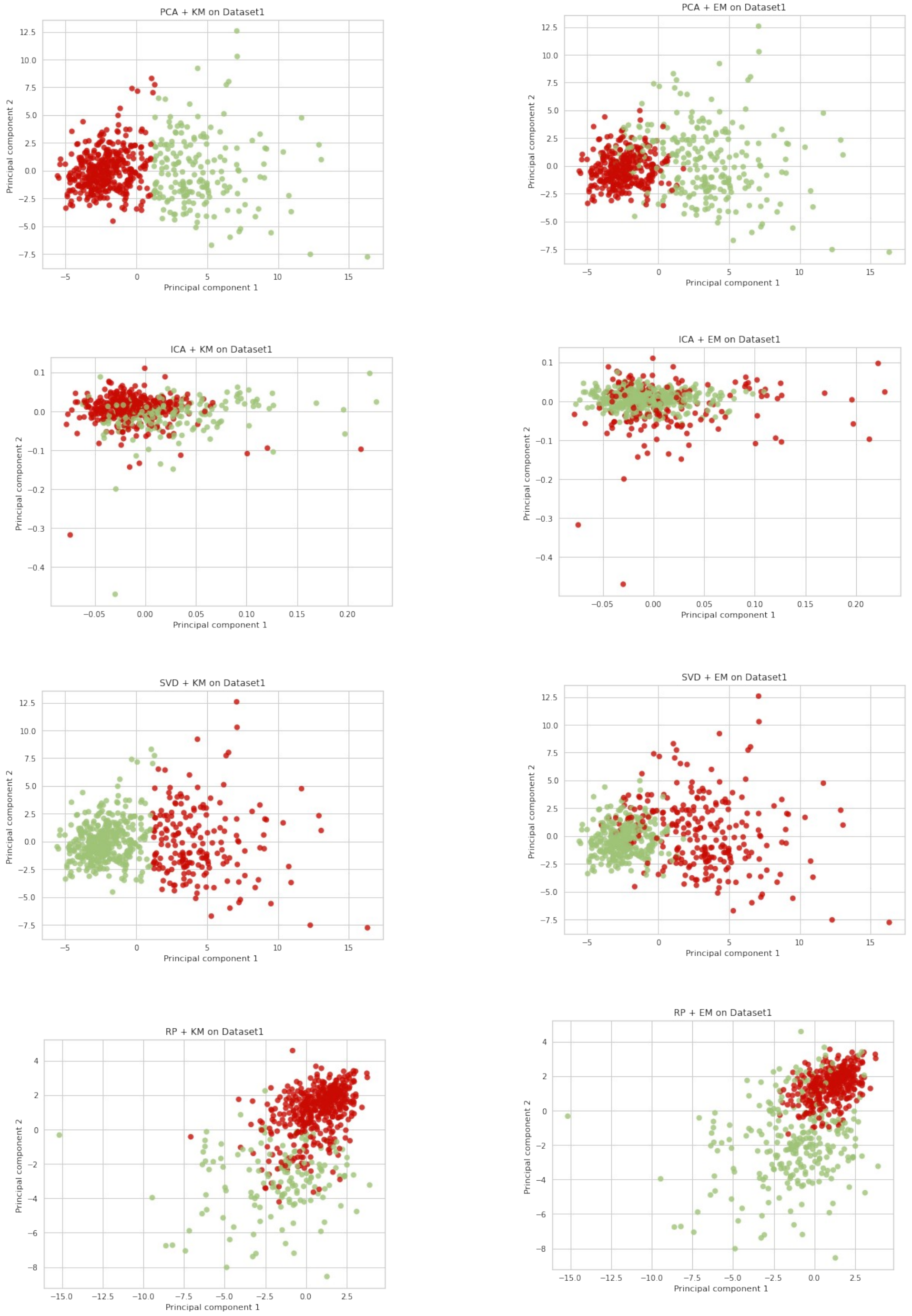


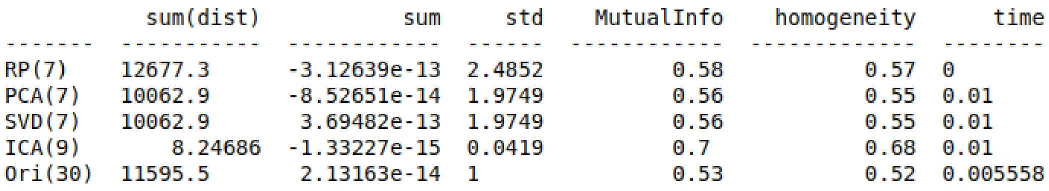
Figure 7. Visualization of clustering result of data set 1 using KNN or EM with dimentionality reduced data using PCA, ICA, RP and SVD

Comparison of clustering performances

As shown in Table 3 and 4, for dataset 1 using KNN clustering method, the best performer of dimensionality reductions is ICA with mutual information and homogeneity close to original data. The four dimensionality reduction method results in similar performances for EM clustering method, however, with a computation time ~ 7 times. For data set 2, SVD and PCA performs the best for dimension reduced clustering. If considering the computational time, SVD is the best performer out of the four methods.

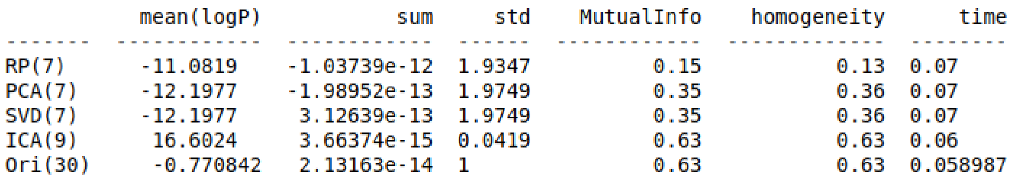
The reason for the best performer of dimension reduction is probably due to the nature of the feature space. Since the first data set is Wisconsin Diagnostic Breast Cancer, there could be convolved features in the data. Therefore ICA could serve as a better technique to separate out independent features. For dataset 2, the data are actual pixels of hand written figures. PCA or SVD could capture the largest variance new features.

KM



```
           sum(dist)           sum     std    MutualInfo    homogeneity      time
-------  -----------  ------------  ------  ------------  -------------  --------
RP(7)    12677.3      -3.12639e-13  2.4852          0.58           0.57  0
PCA(7)   10062.9      -8.52651e-14  1.9749          0.56           0.55  0.01
SVD(7)   10062.9       3.69482e-13  1.9749          0.56           0.55  0.01
ICA(9)       8.24686  -1.33227e-15  0.0419          0.7            0.68  0.01
Ori(30)  11595.5       2.13163e-14  1               0.53           0.52  0.005558
```

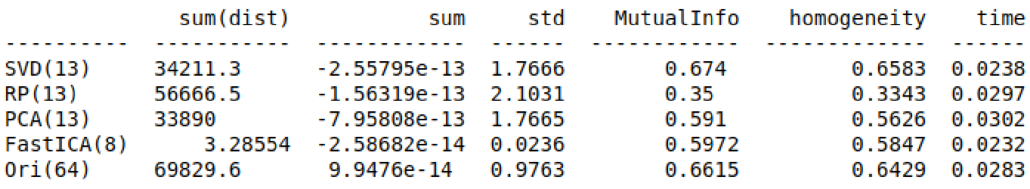
EM



```
          mean(logP)           sum     std    MutualInfo    homogeneity      time
-------  ------------  ------------  ------  ------------  -------------  --------
RP(7)      -11.0819    -1.03739e-12  1.9347          0.15           0.13  0.07
PCA(7)     -12.1977    -1.98952e-13  1.9749          0.35           0.36  0.07
SVD(7)     -12.1977     3.12639e-13  1.9749          0.35           0.36  0.07
ICA(9)      16.6024     3.66374e-15  0.0419          0.63           0.63  0.06
Ori(30)     -0.770842   2.13163e-14  1               0.63           0.63  0.058987
```

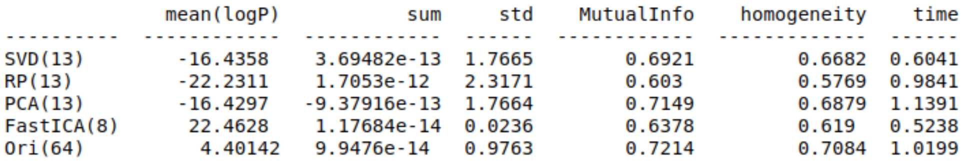
Table 3. Time and performance of on KM/EM on optimal sized new features (dataset1)

KM



```
              sum(dist)           sum     std    MutualInfo    homogeneity    time
----------  -----------  ------------  ------  ------------  -------------  ------
SVD(13)     34211.3      -2.55795e-13  1.7666         0.674          0.6583  0.0238
RP(13)      56666.5      -1.56319e-13  2.1031         0.35           0.3343  0.0297
PCA(13)     33890        -7.95808e-13  1.7665         0.591          0.5626  0.0302
FastICA(8)      3.28554  -2.58682e-14  0.0236         0.5972         0.5847  0.0232
Ori(64)     69829.6       9.9476e-14   0.9763         0.6615         0.6429  0.0283
```

EM

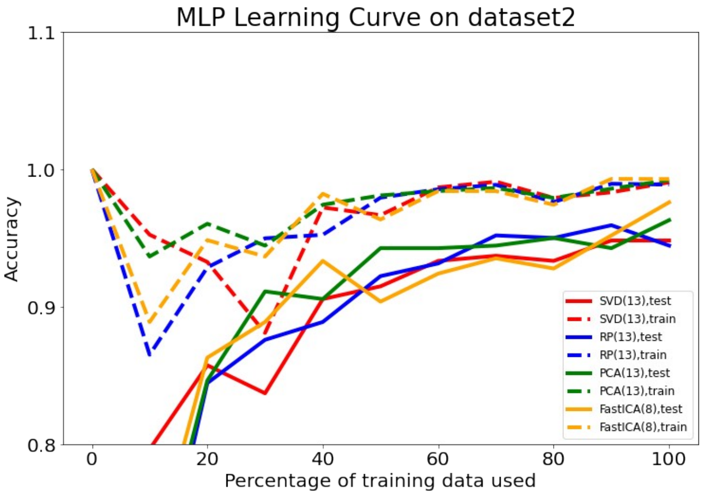


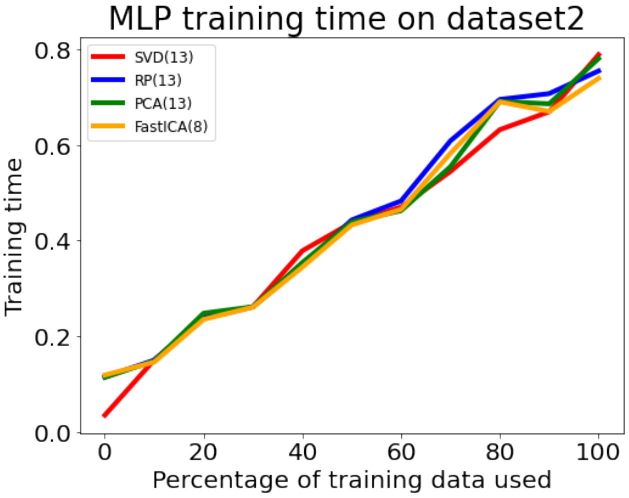
```
              mean(logP)           sum     std    MutualInfo    homogeneity    time
----------  ------------  ------------  ------  ------------  -------------  ------
SVD(13)         -16.4358   3.69482e-13  1.7665        0.6921         0.6682  0.6041
RP(13)          -22.2311   1.7053e-12   2.3171        0.603          0.5769  0.9841
PCA(13)         -16.4297  -9.37916e-13  1.7664        0.7149         0.6879  1.1391
FastICA(8)       22.4628   1.17684e-14  0.0236        0.6378         0.619   0.5238
Ori(64)           4.40142  9.9476e-14   0.9763        0.7214         0.7084  1.0199
```

Table 4. Time and performance of on KM/EM on optimal sized new features (dataset2)

5. Neuron Network on Re-projected Dataset2

The MultiLayerPerceptron with the structure optimized in assignment 1 is trained on re-projected dataset2. Compared to original data, transformed data didn't suffer much from information loss. Transformed data didn't improve much in MLP's training speed. But it might because the compression power isn't big enough to elaborate the difference here.

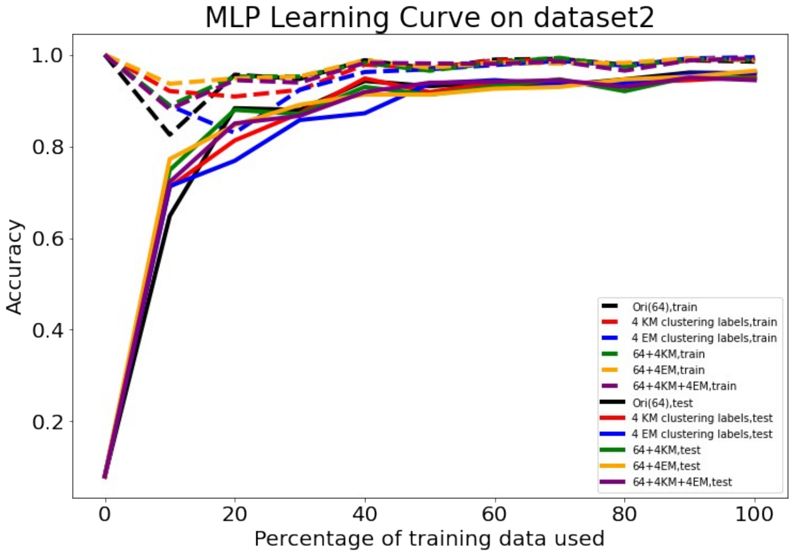




6. Neuron Network on Re-projected Dataset2 and Clustering Features

MLP built on 4KM/4EM(KM/EM clustering on PCA, ICA, RP and SVD), 64+4KM/4EM (original features + KM/EM clustering features) and 64+4KM+4EM (original features + KM + EM clustering features). Black curve (original features) is used as control.

The findings are only 4KM or 4EM could perform well as the orginal data, which justify the usage of dimensionality reduction algorithms in machine learning problems.


MLP Learning Curve on dataset2
Accuracy
Percentage of training data used
Ori(64),train
4 KM clustering labels,train
4 EM clustering labels,train
64+4KM,train
64+4EM,train
64+4KM+4EM,train
Ori(64),test
4 KM clustering labels,test
4 EM clustering labels,test
64+4KM,test
64+4EM,test
64+4KM+4EM,test