CS7641 Report 3: Unsupervised Learning and Dimensionality Reduction

Yiwei Yan

November 8, 2015

Abstract

Unlike supervised learning algorithms we implemented in the first project (e.g. Decision Tree, K-Nearest Neighbors, SVMs, etc.), unsupervised learning algorithms (e.g. K-means, EM, PCA, etc.) do not train the model with labels. This report aims to explore the performance of unsupervised learning on datasets with clustering and dimensionality reduction performance. The clustering algorithms include K-means and Expectation Maximization(EM), while the dimension reduction algorithms include Principal Component Analysis (PCA), Independent Component Analysis (ICA), Randomized Projections (RP), and Linear Discriminant Analysis (LDA). In order to highlight the difference of those algorithms, we perform comparative experiments. We perform K-means and EM on iris and USPS datasets firstly, then apply dimensionality reduction by PCA, ICA, RP, and LDA on those two datasets. Furthermore, we apply clustering on the reduced new iris and USPS datasets. Lastly, we test the performance of neural networks on both the dimension-reduced USPS datasets and clustered USPS datasets and discuss the results.

1 Introduction

Unsupervised learning is a method that aims to find hidden structure in unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This is the key difference among unsupervised learning and supervised learning.

1.1 K-means and EM Algorithms

In particular, clustering can be seen as a way of choosing a small number of exemplars to compress the information.

(1) K-means:

K-means algorithm is one of the popular clustering algorithms that clusters data by separating samples into groups of equal variance, minimizing some in-cluster distance metrics. A popular choice of these distance metrics is the sum-of-squares. This algorithm requires the number of clusters to be specified by the modeler. Specifically, it divides a set of samples X into K disjoint clusters C, each described by the mean of the samples in the cluster named “centroids”. It chooses the centroid that minimize the in-cluster sum of squared criterion: $\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_j - \mu_i||^2)$.

It consists two main steps in the looping. The first step assigns each sample to its nearest centroid. The second step creates new centroids by taking the mean value of all of the samples assigned to each previous centroid. The difference between the old and the new centroids are computed and it repeats until the centroids do not move significantly.

The limitation of Kmeans algorithms are: firstly, it assume that clusters are convex and isotropic, which responds poorly to elongated clusters; secondly, in very high-dimensional spaces the euclidean distances tend to become inflated.

(2) EM:

EM algorithm is a method that iteratively look for maximum likelihood estimates of parameters of the model that depends on unobserved latent variables. There are two steps of EM: The first one is expectation step that creates a function for the expectation of log-likelihood evaluated using the current estimate for the parameters; the second one is maximization step that aims to compute parameters maximizing the expected log-likelihood found on the E step.

EM is frequently used for data clustering in machine learning. Because it usually converges to a local optimization instead of the global optimization and there is no bound on the convergence rate, it is possible very poor in high-dimension situations and there can be an exponential number of local optimization.

1.2 PCA, ICA, RP, and LDA Algorithms

There are many data analysis cases in which it is necessary to perform some form of dimensionality reduction. In general, reducing the number of data dimensions is about stripping out the less important parts of your data and exposing the more important ones to make the datasets easier to process with better insight quality.

(1) PCA:

PCA is used to decompose a multivariate dataset in a set of successive orthogonal components that explain a maximum amount of the variance. Suppose that you observe the n-dimensional random variable $X = (x_1, x_2, ..., x_n)^T$, PCA is going to find a "direction" $W = (w_1, ..., w_n)^T$, which is an n-dimensional vector, that maximize some feature of the linear combination $W^T X$. The number of principal components is usually less than or equal to the number of original variables. This transformation is defined that the best (first) principal component should have the maximum possible variance. PCA can supply the user with a lower-dimensional picture, a projection or "shadow" of this object when viewed from its most informative viewpoint by using only the first few principal components.

(2) ICA:

ICA is a computational method for separating a multivariate data into additive subcomponents.It assumes that the subcomponents are non-Gaussian and they are statistically independent from each other. Typically, ICA is not usually used for reducing dimensions but for separating superimposed signals.A common example is the "cocktail party problem" of listening in on one person's speech in a noisy room.

Most ICA algorithms need the "data pre-processing", so that sometimes ICA cooperates with PCA: for example, you can utilize PCA to get y firstly, and then standardize each component of y to be z. It uses z to estimate A and s in the model $x = As$.

(3) RP:

Random Projections are a cheap and fast method in computationally to reduce data dimensionality by trading a controlled amount of accuracy for more efficient processing time and smaller model sizes.

The dimensions and distribution of random projections matrices are controlled so as to preserve the pairwise distances between any two samples of the dataset. Thus random projection is a suitable approximation technique for distance based method.

Table 1: Selected Dataset Descriptions

Dataset name	# samples	Feature dim.	Feature attributes	#classes	Usage
Iris	150	4	continuous	3	illustration
USPS	9298	256	continuous	10	evaluation

(4) LDA:

LDA is a classifier that has a linear decision boundary generating by fitting class conditional densities to the data and using Bayes rule. LDA model assumes all classes share the same covariance matrix and fit Gaussian density to each class. It tries to identify attributes that account for the most variance between classes.

LDA can also be used to reduce the dimensionality of input by projecting it to the most discriminative directions. However, LDA is a **supervised** method using known class labels in contrast to PCA,ICA and RP.

1.3 Two selected datasets

Table 1 provides some details of the two datasets we selected to do clustering and dimensionality reduction. Those datasets are interesting to us because of their similarity and strong contrast (i.e. sample size, feature dimensions, number of classes). The types of similarity and contraction might be easy for us to test different algorithms and compare their performance.

Iris flower dataset is always a good benchmark dataset. It has a small sample size of 150, as well as a small feature dimension for three classes. Compared to this, USPS digit dataset has high/middle-size feature dimension (256, from 16 × 16 image patches) and larger sample size (9298). Each feature dimension of USPS is continuous and well scaled with all values between 0 and 1.

2 Algorithms Implementation and Evaluation

I wrote the code in Python for all algorithms mainly based on `Scikit-learn` packages and `Pylearn2` for Neural Network build.

2.1 Performance of k-means and EM on iris and USPS datasets

There are a lot of ways to evaluate performances of clustering. In order to evaluate k-means and EM, we choose two evaluation metrics given the knowledge of ground truth class assignments. Since the two evaluations we selected have no assumption made on the cluster structure, they will be good to evaluate clustering.

(1) Adjusted Rand index (ARI): A metric that measures the similarity of two assignments ignoring permutations and with chance normalization. Its bounded range is $[-1, 1]$. Negative values means independent labeling, 1 represents the perfect match score.

(2) Adjusted Mutual Information scores (AMI): Mutual Information measures the agreement of two assignments ignoring permutations. Its bounded range is $[0, 1]$. Values close to zero indicate two label assignments are largely independent, while values close to 1 indicate significant agreement.

We test the clustering number from 2 to 6 for iris datasets. Figure 1 gives the performance evaluations of K-means and EM on Iris datasets based on ARI (left) and AMI (right) scores. We utilize Euclidean distance for those two clustering algorithms. It means assigning points to the closest centroid using Euclidean distance from centroid.

Figure 1: ARI and AMI for K-means and EM on Iris Data by changing the number of clusters.

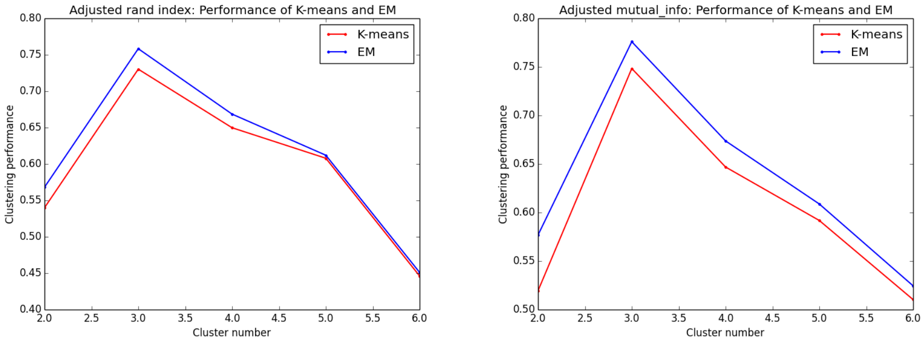
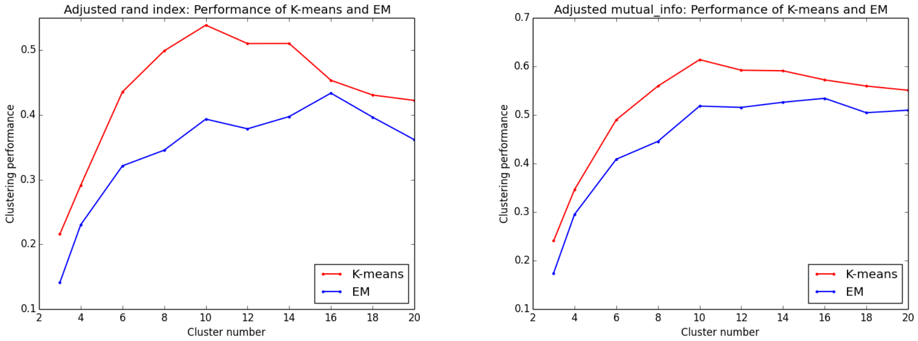


Figure 2: ARI and AMI for K-means and EM on USPS Data



We can see that both ARI and AMI give the highest scores at cluster 3, which means that our best suggestion for clustering on iris data should be 3.

According to the performances of K-means and EM, since both ARI and AMI gives around 0.25 higher overall scores for EM than K-means, we could say EM is more accurate than k-means in this case. It is because that in every iteration, k-means performs "hard-assignment", which assigns every sample to one certain cluster without further considering the probability of other possible clusters for those samples. Unlike K-means, EM performs "soft-assignment", which learns the mixture likelihood to the possible clusters. It also indicates that EM can capture more information and model more complicated underlying distributions ("mixture distributions") for clustering than K-means, which makes the result more accurate.

For USPS dataset, we tested the number of clusters from 2 to 20. Figure 2 gives the performance evaluations of K-means and EM on USPS datasets based on ARI (left) and AMI (right) scores.

In terms of training and testing time of clustering, EM and K-means achieve very comparable timing results with our implementation. Figure 3 shows the time change comparison example of USPS data (left side is for K-means and right side is for EM). It can be observed that training time increases for both algorithms as the number of clusters goes up. However, the testing time of K-meansdecreases as the number of clusters increases, while the testing time of EM goes up.

2.2 Applying PCA, ICA, RP, LDA on iris and USPS datasets

First of all, we apply PCA to perform dimensionality reduction on iris and USPS dataset. We utilize "explained variance ratio", which means the percentage of variance explained by each of the selected components, to determine how many components should keep in order to store some percentage of information from the original data. We used 95% for iris and USPS as the percentage of information that we want to keep after reduction.

Figure 3: Training and Testing Time of K-means and EM on USPS. LEFT: K-means timing. RIGHT: EM timing.

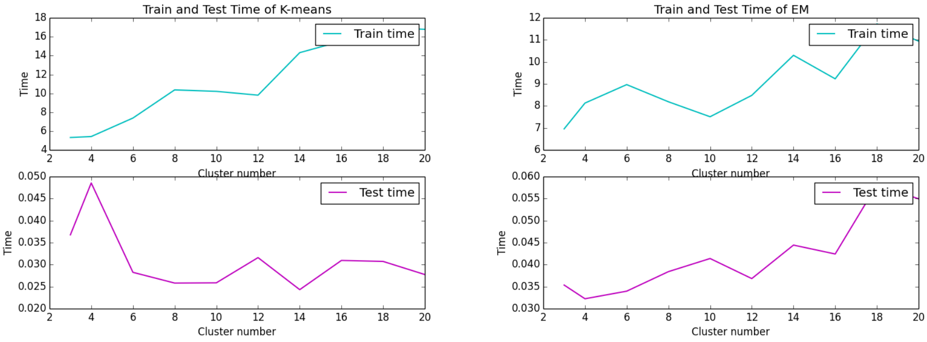
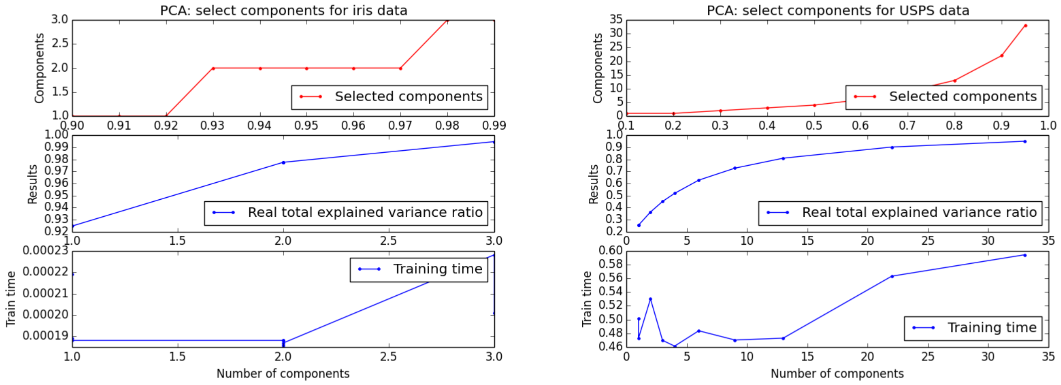


Figure 4: PCA performance on iris and USPS Dataset.



As shown in Figure 4, for iris dataset, if we want to keep 95%, the components we need is at least 2 out of 4; while for USPS, the components we need is at least 35 out of 256.

Then, we apply ICA, RP and LDA on iris and USPS datasets respectively. Specifically, we iterate the components from 2 to 4 for iris data and 10 to 230 for USPS dataset to see the runtime changes. For RP algorithm, RPG means reduce dimensionality through Gaussian random projection, and RPS means reduce dimensionality through sparse random projection.

In figure 5, the training time increases as components increase for ICA transformed data. For RP and LDA algorithms, however, the training time on iris data decreases when the components number goes up after transforming the dataset by LDA and RP-Gaussian and RP-sparse. On the other hand, Figure 6 and figure 8 visualizes how the three kinds of Iris flowers (Setosa, Versicolour and Virginica) spread after the dimensional reduction by PCA, ICA, LDA respectively.

In terms of USPS data, as shown in figure 7, the train time increases when the components number increase by ICA, RP and LDA algorithms, and RP-Gaussian makes train time faster than that of RP-sparse. In figure 8, it visualizes what the USPS data sample looks like after reducing its dimensionality from 256 to 36 by PCA, ICA, RP respectively.

In order to deeper understand how dimensionality-reduction algorithms impact the accuracy of learning, in the later section, we will use Neural Networks of USPS as an example, to show how the accuracy of the classifier results changes with reduced data.

2.3 Applying clustering on dimension-reduced datasets

Moreover, we again perform clustering algorithms K-means and EM on the reduced datasets of iris and USPS, to investigate how clustering performance scores will change by the components change. We only use AMI score to do cluster performing comparison in this case.

Figure 5: Train time change of Iris: ICA (LEFT), RP (MIDDLE), LDA (RIGHT) components-transformed data

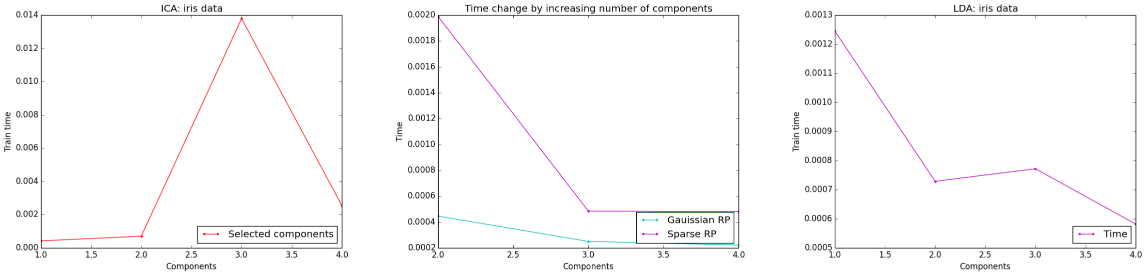
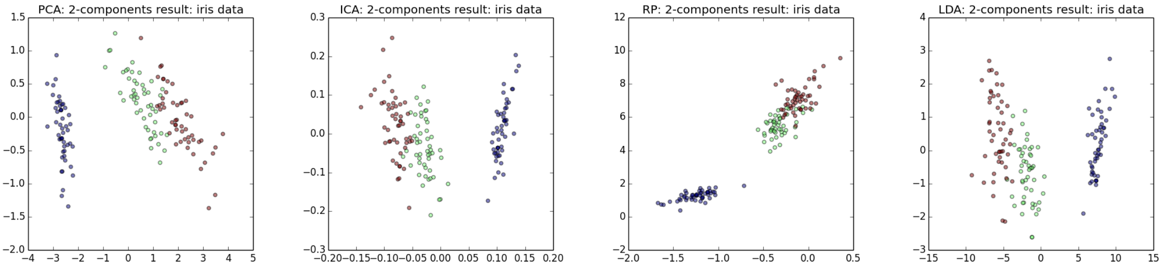


Figure 6: Visualization: Iris data reduce to 2 components by PCA, ICA, RP, LDA.



For iris datasets, we test clustering by using number of components $[2, 3, 4]$. figure 9 and Figure 10 give the performance details of K-means and EM respectively. For RP algorithm, RPG means reduce dimensionality through Gaussian random projection, and RPS means reduce dimensionality through sparse random projection.

For K-means, in figure 9 from left to right represents $components = 2, components = 3, components = 4$ separately. Overall, LDA performs the best over other algorithms for the dimension-reduced dataset of iris, while the accuracy of PCA and ICA are similarly lower. For EM, in figure 10 the components are $components = 2, components = 3, components = 4$ from left to right as well. Generally, LDA is still has the highest performance score, and the suggested cluster is 3. Notice that this is expected because LDA is **supervised**. It is a little strange that for EM at components =2, RPG gives cluster=3 the lowest performance score.

For USPS datasets with dimensional reduction, we test clustering by using the number of reduced components of USPS as $[40, 100, 150, 200]$. Figure 11 and figure 12 gives the performance details of K-means and EM on reduced USPS. Overall, the cluster number with the best performance score is 10.

In figure 11 and figure 12 from left to right represents $components = 40, components = 100, components = 150$, and $components = 200$. For K-means, LDA still performs the best over other algorithms in the dimension-reduction datasets of USPS. When components = 40, the lowest performance score belongs to RP. When the components becomes 100, 150, and 200, the the lowest performance score appears at PCA and ICA. Additionally, for EM, LDA still has the highest performance score on dimension-reduced USPS data, and the lowest score mostly appears at PCA and ICA.

Overall, we also notice that the standards of scores are always stay at a similar level no matter what the components change for both iris or USPS data. It does not decrease or increase by changing the number of components, which means that the dimension reduction of datasets may not affect the overall performance of clustering algorithms K-means and EM.

Figure 7: Train time change of USPS: ICA, RP, LDA components-transformed data

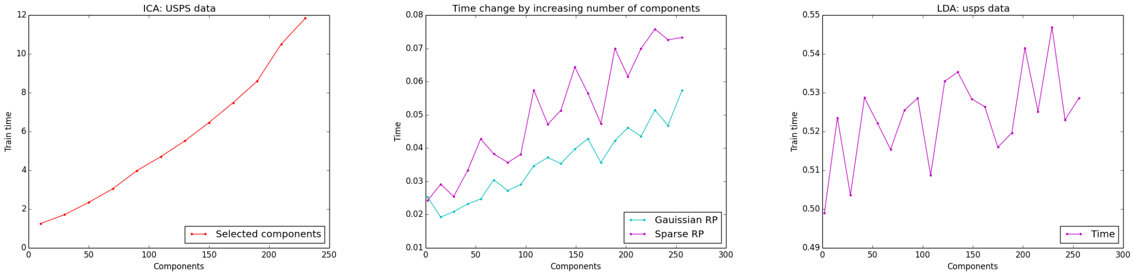
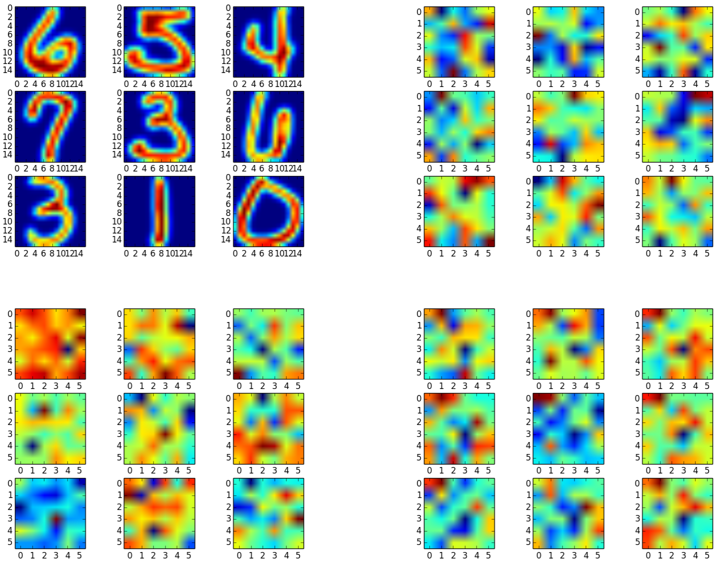


Figure 8: Visualization: USPS data reduce to 36 components by PCA, ICA, RP



3 Evaluating NN on dimension-reduced USPS dataset

A fundamental problem in neural network research, as well as in many other disciplines, is finding a suitable representation of multivariate data, i.e. random vectors. In this section, we are going to analyze if dimension-reduction datasets will affect the accuracy and train/test time of NN for USPS data.

For the Neural Network models, sigmoid functions are used as the activation function for two hidden layers (with 15 nodes) and one output layer. We split 30% of the 9298 samples in USPS to be test data that is 2790 samples.

The experiments are conducted as follows:

(1) The accuracy of NN on original USPS data is about 94% and the test time is about 0.0069 seconds. We use these two valuse as the "baseline" in our test.

(2) We loops over our number of components in the range :
$[2, 23, 44, 65, 86, 107, 129, 150, 171, 192, 213, 234, 256]$.

(3) Then we instantiate the PCA, ICA, RP (including RP-Gaussian and RP-sparse) and LDA algorithms using the current number of components to gain the dimension-reduction USPS data, and train the NN model using those transformed new datasets.

(4) The testing time of the iteration processes on the transformed dataset is further recorded. We visualized the iteratively changes of accuracy and testing time of NN model on transformed USPS datasets to compare with our baseline values.

Figure 9: K-means Performance on Dimension-Reduction Iris data. LEFT: two components; MIDDLE: three components; RIGHT: four components.

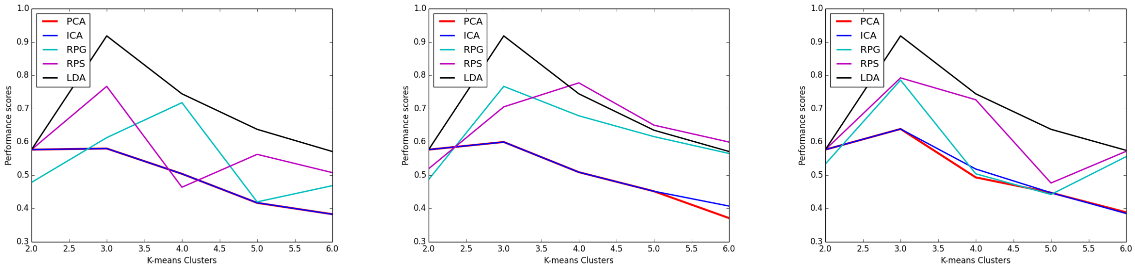
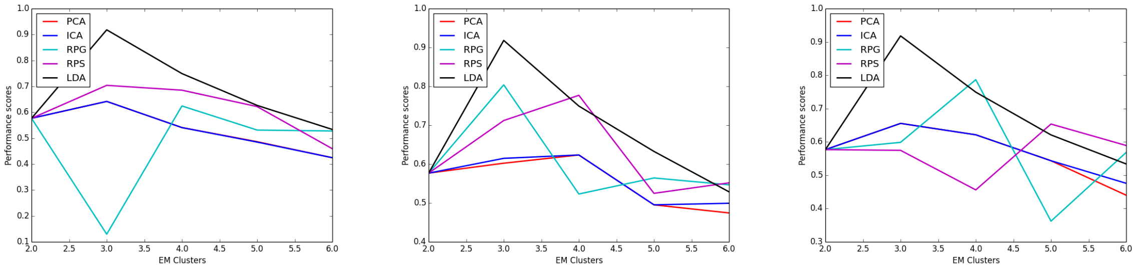


Figure 10: EM Performance on Dimension-Reduction Iris data. LEFT: two components; MIDDLE: three components; RIGHT: four components.



In figure 13, it is clear to see that except for ICA-transformed data, all other transformed data points have baseline-similar accuracies on training NN model. For RP, it starts to have a baseline-similar accuracy after $components >= 40$. ON the other hand, ICA shows a zigzag-shape downward trend of accuracy training NN, which below 0.6 mostly.

In terms of testing time, the overall amount of testing time are lower than the baseline even those test time changes up and down by different components. The fastest data on training NN is RP-Gaussian and RP-sparse transformed USPS data.

4 Evaluating NN on clustered USPS dataset

Lastly, we also want to discuss what the clustered-transformed USPS data will affect on training NN model.

The experiments are conducted as follows:

(1) The accuracy of NN on original USPS data is about 95% and the test time is about 0.0045 seconds. We use these two value as the "baseline" in our test.

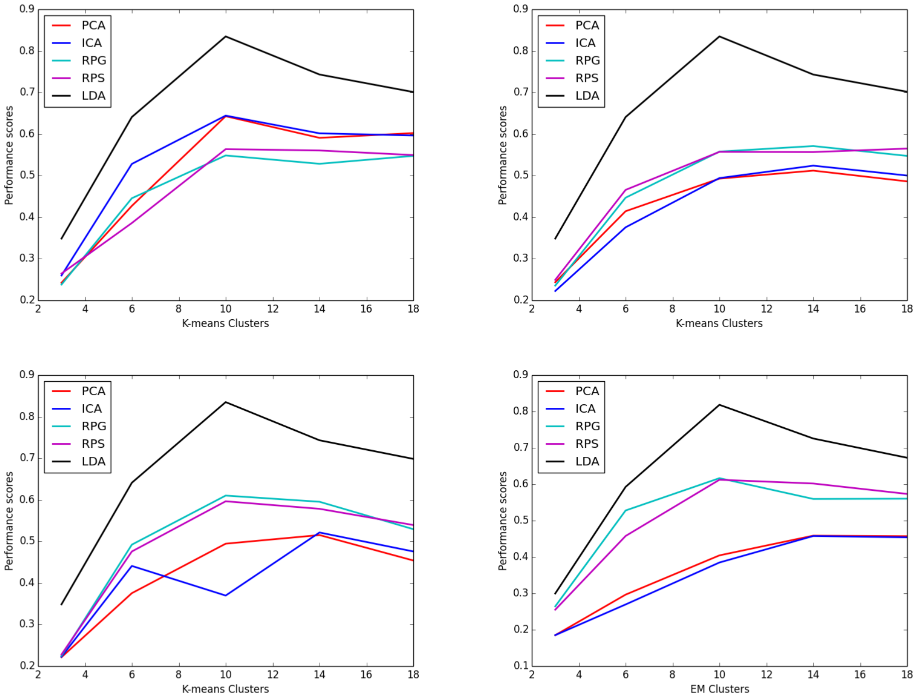
(2) Similarly, we loops over the number of clusters in the range $[3, 4, 6, 8, 10, 12, 14, 16, 18, 20]$.

(3) Then we instantiate K-means and EM algorithms using the current number of components to gain the "clustered-transformed" USPS data, and train the NN model using those new datasets.

(4) At the same time, we record the testing time of the iteration processes of transformed datasets. Plot the iteratively changes of accuracy and test time of NN model on transformed USPS datasets to compare with our baseline values.

In figure 14, unlike in figure 13 in which many transformed data have baseline-similar accuracies on training NN model, the training performance for both k-means and EM transformed USPS data are with bad accuracies.

Figure 11: K-means Performance on Dimension-Reduction USPS data. LEFT: 40 components; MIDDLE: 100 components; RIGHT: 200 components.



For K-means, the average accuracy is only around 0.35 to 0.4 instead of 0.95 in the clusters loops. The accuracies increase firstly before reach at a good clustering answer (i.e. cluster=8 to 10). Then it dropped exponentially after cluster = 10 and reach the lowest accuracy at cluster = 14. In terms of EM, the accuracy performance are always at a very low level that around 0.15 (15%).

According to the test time graph, K-means utilize more test time than the baseline before cluster = 8 (except that at cluster =4 there is a big drop at test time), while EM use more test time than the baseline time before cluster = 12. In general, EM is slower than k-means.

5 Conclusion

In this report, we investigated into unsupervised learning algorithms k-means, EM, PCA, ICA, RP (and the supervised dimension-transformed algorithm LDA) on Iris and USPS datasets. The results demonstrate that in terms of clustering, EM is more accurate but less efficient than K-means algorithms in our experiment; The performance of dimension-reduction algorithms depends on different models and the feature of datasets. In our case of NN model, ICA give the poorest accuracy performance than other algorithms. We also notice that except for ICA the dimension-transformed data has little effects on the NN model accuracy. On the other hand, clustered-transformed data changes the NN accuracy to a low level.

Figure 12: EM Performance on Dimension-Reduction USPS data. LEFT: 40 components; MIDDLE: 100 components; RIGHT: 200 components.

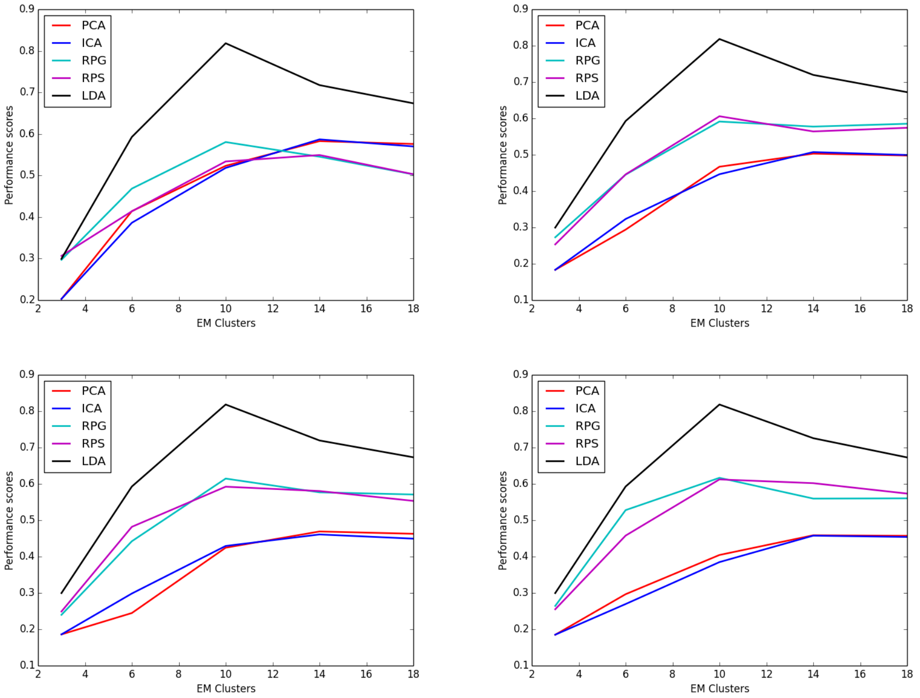


Figure 13: NN accuracy on Dimension-Reduction USPS data

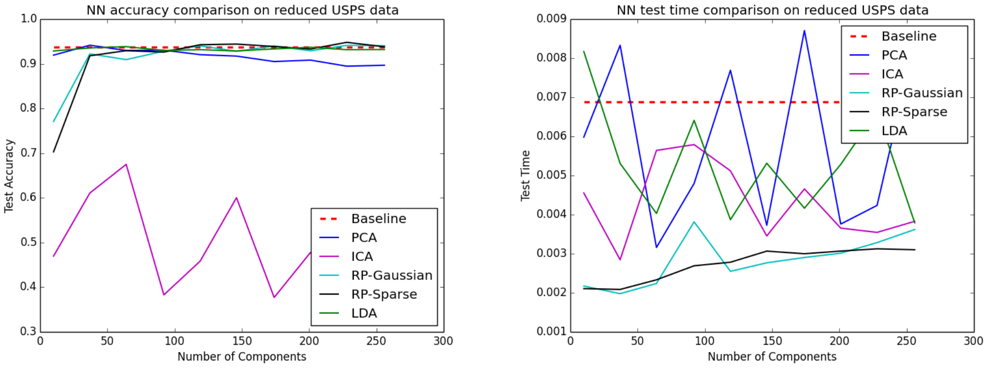
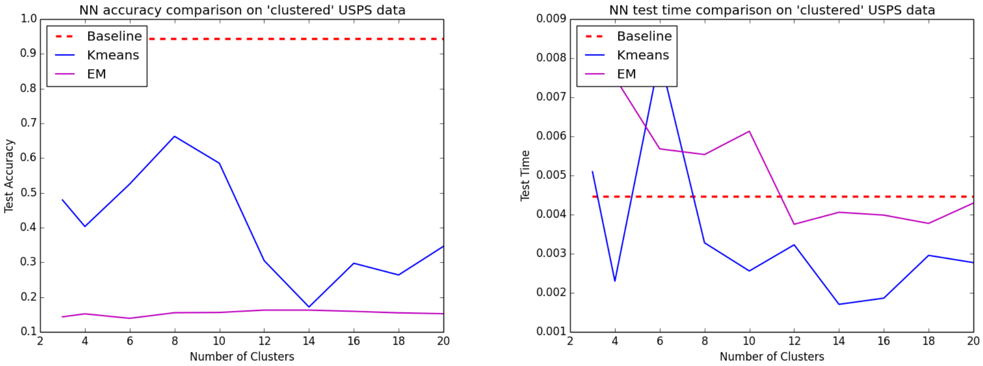


Figure 14: NN accuracy on Cluster-transformed USPS data


NN accuracy comparison on 'clustered' USPS data
Test Accuracy
Number of Clusters
NN test time comparison on 'clustered' USPS data
Test Time
Baseline
Kmeans
EM