CS 7641 - Assignment 3

Unsupervised Learning and Dimensionality Reduction

Author: Swapnil Ralhan(sralhan3@gatech.edu)
Nov 9, 2014

Abstract

In the following paper, we first explore the effects of 2 clustering algorithms - k-means clustering and expectation and maximization on letter image recognition and waveform data. Then we investigate 4 dimensionality reduction algorithms - Principal Components Analysis, Independent Components Analysis, Randomized Projects and Information Gain based feature selection and their effects on our clustering experiments, as well as previously conducted Neural Network classification. Last we examine how we can take our clustering results, and use the output clusters as features for our classification and dimensionality reduction experiments.

Data

Letter Image Recognition Data(letter.arff)

Description

The objective is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15.

Number of Instances

20000

Number of Attributes

16 (numeric features)

Number of Classes

26 capital letter (26 values from A to Z)
More details in the letter.arff header.

Waveform Generated Data(waveform-5000.arff)

Number of Instances

5000

Number of Attributes

21 with continuous values between 0 and 6, all of which include noise

Number of Classes

3

More details in the waveform-5000.arff header.

These 2 datasets were explored in our classification and randomized optimization experiments. They are interesting because they are non-trivial, moderate sized, allowing fast iteration while testing various algorithms and tuning parameters, and flexible enough to allow comparison between various different kinds of algorithms, as demonstrated in this document. Additionally these were good to compare against each other, as the first dataset has a larger number of instances but a smaller number of instance per class. The second dataset has a very small number of classes but a larger number of attributes. Also they are such different datasets(one looking at waveforms while the other looking at letter recognition), that it provides an interesting comparison of how the different algorithms perform. One is a generated dataset(taking base waves and adding noise to the attributes), while the other involves a real image recognition problem.

Clustering

For our clustering experiments, we pick the number of clusters as the number of labels that are present in the original classification dataset. This seems like a natural number, and makes evaluation of the clustering experiment much easier. We use 2 methods for evaluation:

1. Point to Point Comparison - We compare pairs of points, and note for each pair with the same classification/label assigned to them, whether they are assigned the same cluster by the clustering algorithm. We will call this P. For a given point i, let c_i be the classification assigned to i and $\hat{c}_i$ be the cluster assigned to i. Let $x_{ij} = 1$ if $c_i = c_j$ and 0 otherwise. Also let $y_{ij} = 1$ if $\hat{c}_i = \hat{c}_j$ and 0 otherwise. Thus given d points we define
$$P = \frac{\sum_{i=0}^{d}\sum_{j=0}^{i} x_{ij} \cdot y_{ij}}{\sum_{i=0}^{d}\sum_{j=0}^{i} x_{ij}}$$
2. Cluster to Label Comparison - We also try to gauge which label is most accurately represented by a given cluster. We further do this in 2 ways
    a. Given a cluster, we check the mode with regards to the label assigned to the points. That is, which label is most frequently present in the given cluster. Thus we check the average percentage of points in a given cluster belonging to this label. We will call this C_1. Given a cluster, let's call the label that is the mode l_m. For a given (label, cluster), let's call the number of points with that label p_l^c
    $$C_1 = \frac{\sum_{c \in clusters} \frac{p_{l_m}^c}{\sum_{l \in labels} p_l^c}}{Number\ of\ clusters}$$
    b. Second, we check the mode for a given label. That is, given all points of a given label, we pick the cluster with the maximum number of points, and we get the average percentage of points in this cluster across all labels. We will call

this C_2.Given a label, let's call the cluster that is the mode c_m.

$$C_2 = \frac{\sum_{l \in labels} \frac{p_l^{cm}}{\sum_{c \in clusters} p_l^c}}{Number\ of\ labels}$$

k-means clustering

Letter

Running K means clustering on the letter image recognition dataset, with number of labels = 26, it took 7.37 seconds to estimate the data. We got the values of
$P = 0.17263;\ C1 = 0.3037;\ C2 = 0.27843$.

Waveform

Running K means clustering on the waveform dataset, with number of labels = 3, it took 0.785 seconds to estimate the data. We got the values of
$P = 0.50387;\ C1 = 0.51888;\ C2 = 0.54783$.

Expectation Maximization

Letter

Running EM clustering on the letter image recognition dataset, with number of labels = 26, it took 94.12 seconds to estimate the data. We got the values of
$P = 0.19019;\ C1 = 0.37293;\ C2 = 0.31556$.

Waveform

Running EM clustering on the waveform dataset, with number of labels = 3, it took 8.34 seconds to estimate the data. We got the values of
$P = 0.51057;\ C1 = 0.52834;\ C2 = 0.56778$.

For the letter image recognition data, expectation minimization takes a lot long(>10 times as long, but we see that the clustering is marginally better).
A similar trend is seen for the waveform data, where EM takes about 10X the time, resulting in only marginally better clustering.
Clustering overall is better for waveform over letter image recognition, however this may very well be a side-effect of the fact that we have much fewer labels/clusters for the waveform dataset(3 as opposed to 26 for letter image recognition).

Dimensionality Reduction

Neural Network

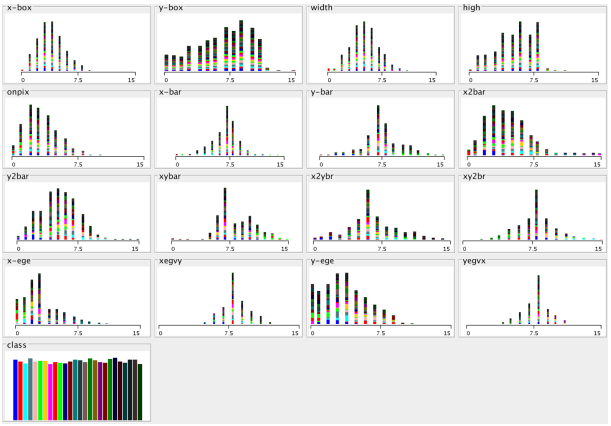
Before we apply dimensionality reduction, we run the neural network on our letter image recognition data. We get a training error of 17.785% with an RMSE of 0.1077, taking 36.75 seconds to build the model.
With a 66% training and 33% test split, we get a test error of 17.7794% and a RMSE of 0.1065.

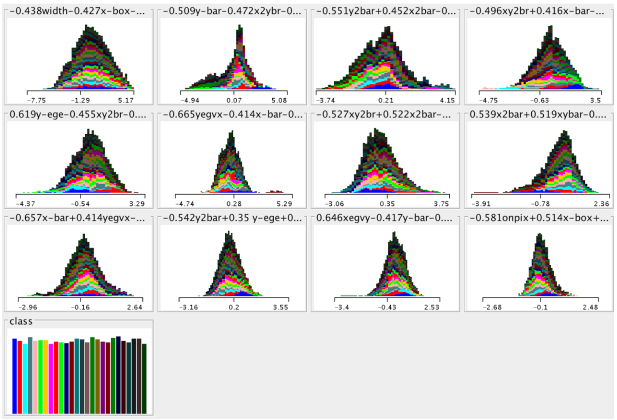
PCA

Basic

Letter

Running PCA, trying to retain 95% of the variance, we get 12 features(reduced from the original 16. We can see that each feature is a linear combination of the existing features. Looking at the existing data, we have integer values for each feature. This produces a disjointed graph, which has a mean that is offset from 0, and is far from normal. The transformed features all have a mean of 0, and look very close to a normal distribution:





Waveform

We see a very similar trend with with the waveform data as well, where we get 34 features from the original 40. The feature distribution was initially close to normal, so we see not much of a difference in the value distribution except that the distributions are now centered at 0.

Clustering

Now we re-run the clustering experiments on the data after we ran it through PCA.

Letter

K Means Clustering took 14.11 seconds with $P = 0.23707,\ C1 = 0.39426,\ C2 = 0.37390$

EM Clustering took 143.069 seconds with $P = 0.27601,\ C1 = 0.48683,\ C2 = 0.40494$

Waveform

K Means Clustering took 1.6245 seconds with $P = 0.50514,\ C1 = 0.52318,\ C2 = 0.55647$

EM Clustering took 18.25 seconds $P = 0.50284,\ C1 = 0.51385,\ C2 = 0.53479$

Clustering is much better for the letter image recognition dataset, taking about 2X the time. For the waveform data, we see about 2X time, but the clustering is the same as before PCA.

Neural Network

The neural network training achieved a 20.955% training error rate with an RMSE of 0.1161 taking 31.49 seconds to build the model.

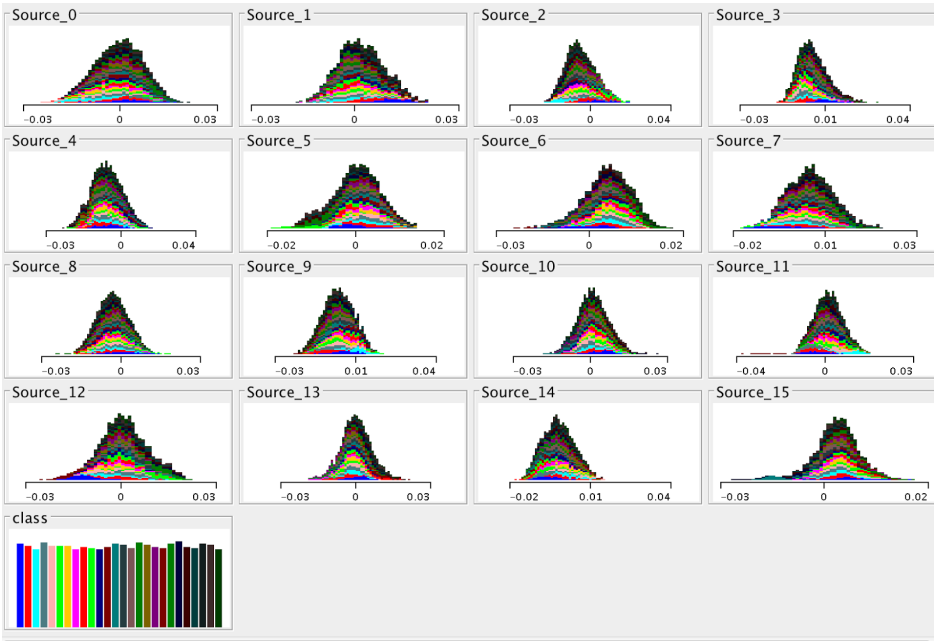
For a 66% training-testing split, we achieved a 21.3382% test error rate with an RMSE of 0.1165.
The neural network classification performance is worse post-PCA, taking about 10% less time.

ICA

Basic

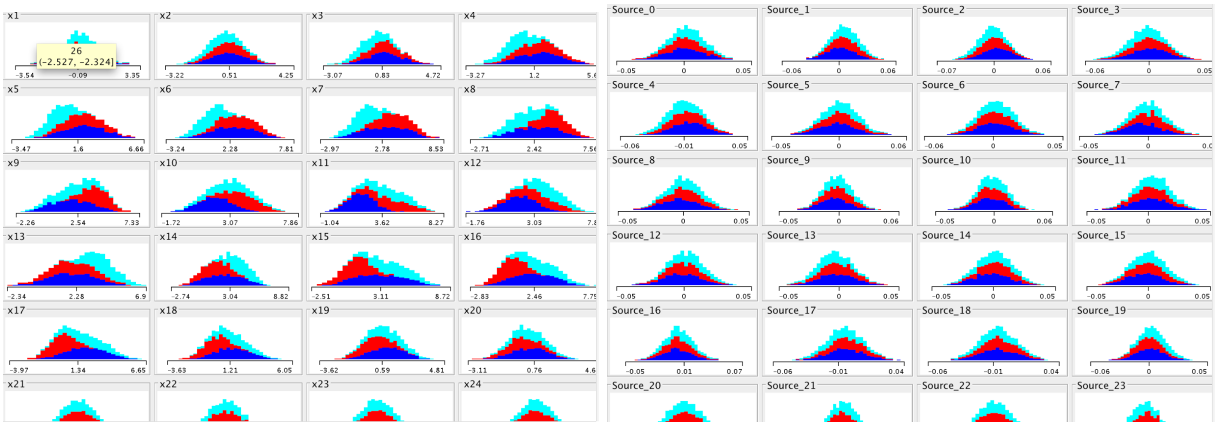
Letter

Running ICA, we get 16 features, from the original 16. Thus we can conclude that the provided set of attributes were all independent. They are a lot more normalized, similar to the post PCA distributions. Along the different directions, we get the following kurtosis:
[-0.414794 -0.6972617 5.75549335 8.95088324 2.92572178 3.51823215 2.08740675 2.28270808 18.91843297 5.24307508 2.62959125 2.11453828 0.90189806 2.60748849 1.94074201 4.39716531]
The distribution looks as follows:



Waveform

For the waveform dataset, we get 40 independent components, similar to the original 40. The distribution is more normalized as well:



The kurtosis values are as follows:

[-0.20297789 -0.9793939 0.54865834 0.03080034 -0.0499617 -0.40326408
  0.20205966 -0.02433173 0.23189471 0.00792537 0.04175526 0.50483225
  0.06589002 -0.06182549 0.39388137 0.51928326 -0.01473098 0.06318701
  0.12890761 -0.12627153 -0.08380964 0.2787905 -0.17281121 0.6750236
 -0.00732009 0.52624601 -0.01836432 0.60920617 -0.30375796 0.39163319
 -1.07631424 0.04079524 0.04841228 0.54768854 -0.06543964 -0.01557014
  0.60041205 0.70376672 0.49878708 0.56740589]

Clustering

Now we re-run the clustering experiments on the data after we ran it through ICA.

Letter

K Means Clustering took 9.79 seconds with $P = 0.22869,\ C1 = 0.42265,\ C2 = 0.35791$

EM Clustering took 183.18 seconds with $P = 0.27236,\ C1 = 0.47340,\ C2 = 0.41277$

Waveform

K Means Clustering took 2.63 seconds with $P = 0.39742,\ C1 = 0.47524,\ C2 = 0.44191$

EM Clustering took 22.51 seconds $P = 0.44881,\ C1 = 0.49156,\ C2 = 0.56985$

ICA causes the clustering time to increase, with little to no improvement for both datasets.

Neural Network

We get a training error of 15.65% with an RMSE of 0.0999 taking 36.97 seconds to build the model.

With a 66% split we get a 18.3088% test error and an RMSE of 0.1074.

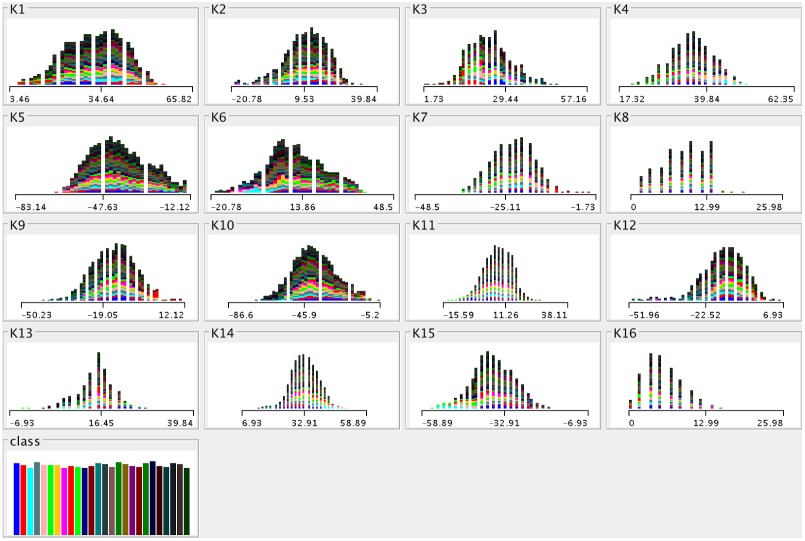
The neural network performance is marginally better post PCA, taking about the same amount of time.

Randomized Projection

Basic

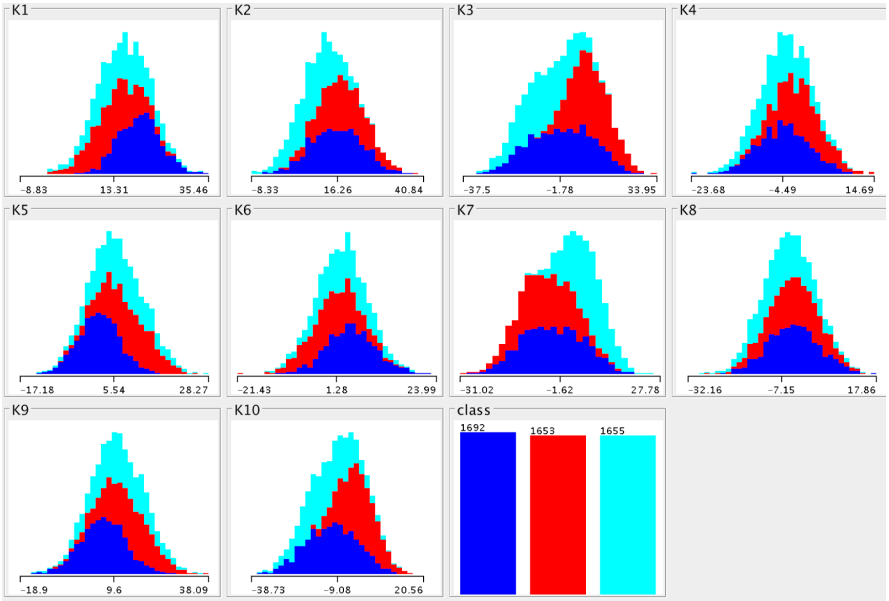
Letter

With Randomized Projection, we see a disjointed “near-normal” distribution. We stuck to 16 features, as predicted by ICA to be independent components. Here we see a distribution:



Waveform

Similarly, in the waveform data, we see distributions such as these:



Here we can see that for a given feature, the distribution of the labels may be a little skewed. Here we’ve shown a random projection with 10 features, we tried a number of different values, and concluded that 32 gives the optimal training error.

Clustering

Now we re-run the clustering experiments on the data after we ran it through the Randomized Projection. We see a lot of variance after running randomized project multiple times, and here we show the average results from 10 runs.

Letter

K Means Clustering took 6.90 seconds with $P = 0.11425,\ C1 = 0.21977,\ C2 = 0.20481$
EM Clustering took 143.08 seconds with $P = 0.10385,\ C1 = 0.20511,\ C2 = 0.18708$

Waveform

K Means Clustering took 0.258 seconds with $P = 0.49183,\ C1 = 0.51910,\ C2 = 0.52384$
EM Clustering took 2.76 seconds $P = 0.49787,\ C1 = 0.52560,\ C2 = 0.54841$

Randomized projection takes longer for the letter image recognition data(for EM; K-means takes the same amount of time), but with significantly worse performance. For waveform however, the time taken is less, but with very similar results. The greater number of features for waveform probably makes Randomized Projection's job easier.

Neural Network

We get a training error of 23.245 % and an RMSE of 0.1221, taking 29.63 seconds to train.
We get a test error of 25.4559 % and an RMSE of 0.1265.
We see worse performance for the neural network training after Randomized project, taking about 5% less time for training over the non-reduced dataset.

Information Gain based Feature Selection

Basic

Lastly, we use Feature selection based on Information Gain. We rank the features by information gain, and then choose the top N. We pick an information gain threshold of 0.001.

Letter

We get the following 15 attributes, by rank and information gain threshold of 0.001: x-ege, x2ybr, y-bar, xegvy, xy2br, y-ege, y2bar, x2bar, xybar, x-bar, yegvx, width, onpix, x-box, high.

Waveform

We select 19 attributes in order of rank, based on an information gain threshold of 0.001: x7, x15, x16, x6, x8, x14, x5, x13, x17, x9, x11, x12, x10, x18, x4, x3, x19, x20, x2.

Clustering

Now we re-run the clustering experiments on the data after we ran it through the Information Gain filter.

Letter

K Means Clustering took 7.02 seconds with $P = 0.27404,\ C1 = 0.47366,\ C2 = 0.39737$
EM Clustering took 40.81 seconds with $P = 0.28577,\ C1 = 0.48967,\ C2 = 0.41768$

Waveform

K Means Clustering took 0.292 seconds with $P = 0.46212,\ C1 = 0.53241,\ C2 = 0.53635$

EM Clustering took 2.52 seconds $P = 0.45718,\ C1 = 0.54046,\ C2 = 0.52631$

This method performs impressively, giving slight time advantages for both datasets. For letter recognition, we see much better performance as well, while for the waveform dataset we see only slight improvements.

Neural Network

We get a 22.73 % training error and an RMSE 0.1193.The test error with a 66% training set was 23.8235% with an RMSE of 0.1215.

For neural network training however, we see slight drop in the performance of the classification after running through the above dimensionality reduction algorithm.

Clustering

Last we look at clustering as a dimensionality reduction algorithm. We do this in 3 different ways:

Learning with clusters as an additional feature

First we simply compute the cluster of each instance, and add that as an additional feature. We do this with 26 clusters:

With K-means clustering, the training error we get is 7.575 % with an RMSE of 0.0704. The test error is 8.1765 % with an RMSE of 0.0733.

With EM clustering, we get 18.37 % training error and 0.1096 RMSE. We also get a 19.1176% test error and an RMSE of 0.1116.

Dimensionality reduction with clusters as an additional feature

Second, we take the new instances with the additional cluster feature, and run them through different dimensionality reduction algorithms, before doing a Neural Network training:

PCA

Using either clustering method, we get 12 features after running PCA. These clustering were again with 26 clusters.

Using K-means, we get a 8.935 % training error and a 0.0765 RMSE. We get a 9.9706 % test error and an RMSE of 0.0799.

Using EM clustering, we get a 20.78 % training error and an RMSE of 0.1159. The test error we get is 21.3088%, with an RMSE of 0.1168.

ICA

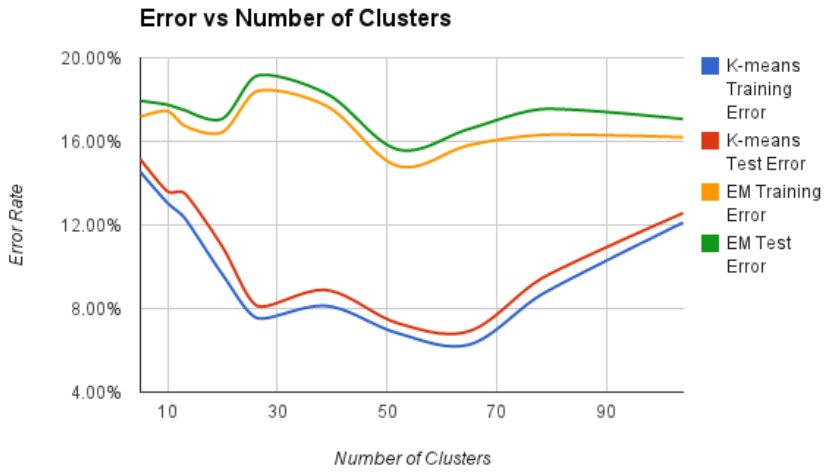
Using either clustering method, we get the 17 independent features. We run the 2 clsutering algorithms with 26 clusters:

With K-means clustering, we get a training error of 7.945 % and an RMSE of 0.0728. We see 8.7206% test error and an RMSE of 0.0763.

With EM clustering, we get a training error of 15.63 % and an RMSE of 0.1004, 16.6618% test error with and RMSE 0.103.

Learning with clusters as the only feature

Lastly we try learning with clusters as the only feature. That is, after running clustering, we drop the other features, and try to learn the classification using just the cluster label. We run this experiment with a number of different values for number of clusters



We see that the error rate is minimized around 65 clusters for k-means, and 52 clusters for expectation maximization.

Conclusion

As we can see, there are distinct advantages of running clustering algorithms before we apply classification. In some cases, it can help us clean up our data, such as in the case of ICA, getting rid of features that might be linear combinations of other features. Features like PCA, help us give a subset of the features, allowing us to avoid the curse of dimensionality. Different algorithms work better with different datasets, and often there is a cost, where faster classification or clustering is achieved at the cost of worse performance.

We performed and analyzed clustering algorithms and dimensionality reduction algorithms on our 2 datasets, as well as tried clustering after dimensionality reduction on both the datasets. For the letter recognition dataset, 3 of our dimensionality reduction algorithms give improved clustering, but the information gain algorithm gives that without the added cost of slow clustering. For the waveform dataset, we do not see much improvement in the clustering. However, we can achieve similar clustering in a much shorter time after applying any of the dimensionality reduction algorithms.

Last we performed neural network classification after running dimensionality reduction on the letter image recognition dataset. We also tried using clustering as dimensionality reduction in various ways. For the letter recognition dataset, we can perform better than the original neural network in many cases. The best performance can be achieved by using clustering, and the clusters can be used as the sole features, or we can use clusters in addition to the existing features, and apply ICA, to get improved classification(in terms of error rate, while keeping the time relatively constant).