Machine Learning Homework 4 Report

Student ID: r05922063   Name: 陳啟中

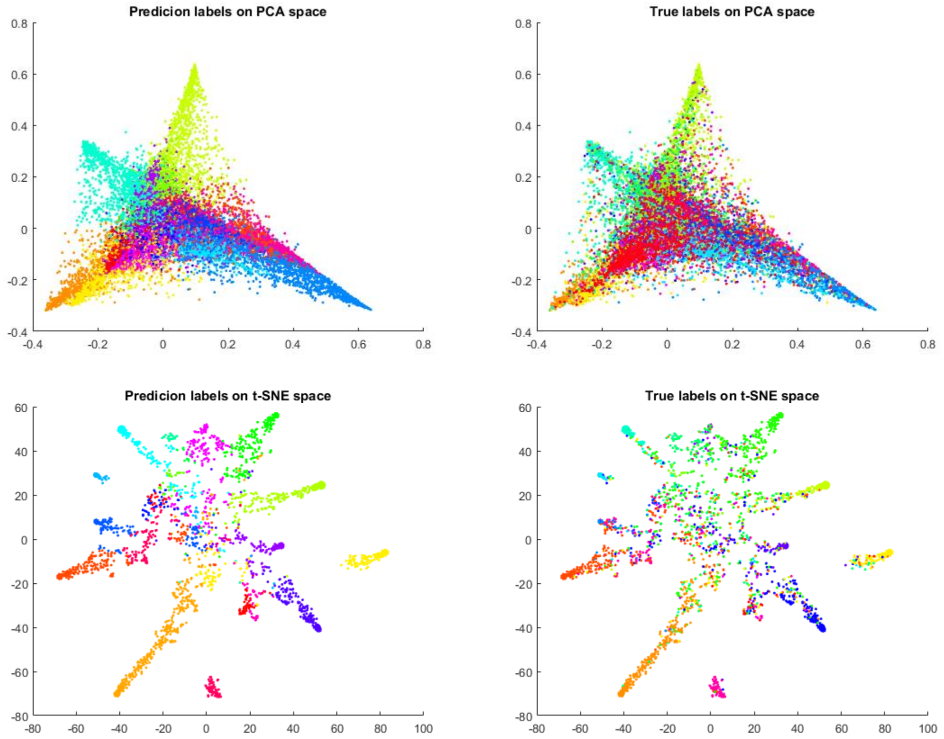
Analyze the most common words in the clusters. Use TF-IDF to remove irrelevant words such as “the”

The following chart shows the IDF of top-ten frequent words shown in the training data. Note that I have performed some preprocessing on the raw data, such as removing code segments, removing punctuations, Decapitalization and so on, and therefore it might be a little bit different from others’.

Rank	Word	IDF
1	the	0.640699326992
2	i	0.674152672291
3	to	0.712761759758
4	a	0.911954581738
5	is	1.1502931118
6	and	1.21862661839
7	in	1.24679601192
8	this	1.40515220165
9	of	1.40553057194
10	that	1.49422872066

Visualize the data by projecting onto 2-D space. Plot the results and color the data points using your cluster predictions. Comment on your plot. Now plot the results and color the data points using the truelabels. Comment on this plot.

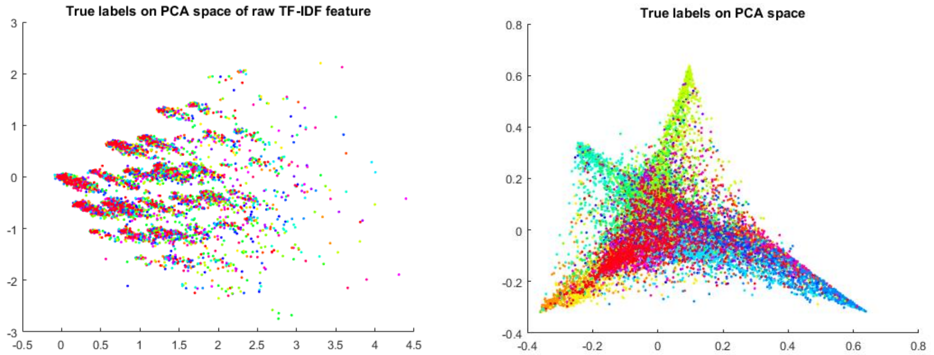
Here I run two dimensionality reduction algorithms, PCA and t-SNE, to project data to 2D space. The source space is the transformed space by my trained AutoEncoder, not raw TF-IDF feature space. The transformed space composes of 20 dimensions. PCA further transforms it to 2D space. t-SNE first samples 2000 points out of 20000 points to reduce computational cost and perform t-SNE on them. The following is the results:



First, we compare prediction labels and true labels. It's clear to see prediction labels show proximity on both dimension reductions. A trivial reason is that the predictions are directly based on the transformed space. Second, we compare PCA and t-SNE. t-SNE shows better sparsity among different labels. Mis-labeling is easily to be found on t-SNE space.

Compare different feature extraction methods.

I tried TF-IDF only and TF-IDF+AutoEncoder to perform feature extraction. The following is the comparison represented by PCA visualization.



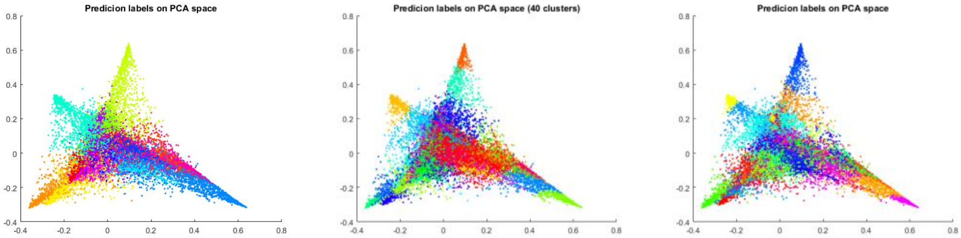
The left one is TF-IDF only and exposes poor proximity property in a cluster. In contrast, TF-IDF+AutoEncoder on the right hand side extracts latent features well. The following is the configuration of the AuthEncoder:

Input (9800 TF-IDF features)
FC-320 with BatchNorm, Dropout(0.3)
FC-80 with BatchNorm (code)
FC-320 with BatchNorm, Dropout(0.3)
Output (9800 decoded features)

Try different cluster numbers and compare them. You can compare the scores and also visualize the data.

Here I give 20 clusters(default), 40 clusters and 60 clusters on K-means algorithm. As the cluster number rises, the model tends to abandon low confident guess that two data belong to the same label. As a result, the precision rate goes up while the recall rate drops. Because the metric is F-0.25, precision and recall rates need to be balanced to get higher score. In my case, 20 clusters contribute to the best result.

The following is the visualization of three configurations. From left to right: 20 clusters, 40 clusters and 60 clusters.


Predicion labels on PCA space
Predicion labels on PCA space (40 clusters)
Predicion labels on PCA space