Course Project for Pattern Recognition

Fan JIN (2015011506)

July 5, 2018

Contents

1 Introduction and Preprocessing

Here is a glance at the datasets. Columns with only one unique value are removed, as they have no effect as observations. The training set is roughly 9 times as much size as the testing set.

File	sample size (nrow)	number of features (ncol)	number features in effect
2ctrainX.txt	2298	25133	17951
2ctestX.txt	256	25133	17951
10ctrainX.txt	16074	25133	24307
10ctestX.txt	1787	25133	24307

Table 1: Shape of datasets

- For two-category classification, the response “2ctrainY.txt” takes values 1 and 2.
- For ten-category one, the response “10ctrainY.txt” takes values ranging from 1 to 10.

2 Two-category classification

2.1 Feature selection

Feature selection is neccesary in this case, for we have much more features than data points, referred to as the “large p, small n” problem.

Methods of feature selection include:

- Non-wrapping: Fisher-based
- Non-wrapping: Correlation-based
- Non-wrapping: Entropy-based
- Non-wrapping: Statistical t-distribution
- Non-wrapping: Principal Component Analysis (PCA)
- Wrapping: Foreward-backward recursive
- Stochastic: Genetic algorithm

Here we use non-wrapping methods for feature selection. We chose Fisher-based and PCA.

2.1.1 Fisher criterion

The fisher criterion is defined as

$$F = \frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2},$$

where μ_1, μ_2 are the means of two categories, and s_1^2, s_2^2 are the varainces, in terms of some feature.

The larger the Fisher criterion, the more significant the feature is. We select 2000 features with the largest Fisher criteria.

2.1.2 the PCA

The Principal Component Analysis performs eigendecomposition to the correlation matrix of the data points, and retains the components with principal eigenvalues.

We retain 99% of total variance in raw data, which gives 2202 features in total.

2.2 Classifier

2.2.1 Linear SVM

The support vector machine, or SVM, is a powerful method for both classification and regression, particularly in case of “large p, small n”.

It is crucial to choose a kernel that balances between prediction accuracy and overfitting aversion. We tried linear, quadratic, and RBF kernels. The linear kernel turns out the best.

2.2.2 Fully connected NN

The neural network has advantages over the linear SVM in case of nonlinearity.

Since we have only limited data points, we cannot train a deep network. We took a simple perceptron network, which has one hidden layer with 1500 nodes. We tuned the hidden layer size, and 1000 proves the best hyperparameter.

2.3 Experiments

Dim. Reduction	Classifier	5-fold cross validation					Avg. Accuracy
Fisher	Linear SVM	0.9848	0.9934	0.9869	0.9869	0.9956	0.9896
PCA	Linear SVM	0.9544	0.9434	0.9629	0.9586	0.9498	0.9539
Fisher	FC NN	0.9891	0.9978	0.9891	1.0000	0.9956	0.9943
PCA	FC NN	0.8828	0.8652	0.8823	0.9019	0.8758	0.8816

Table 2: Comparison

The Fisher criterion has better performance in dimensionality reduction, compared with the PCA.

It is worth noting that the neural network does not always perform better than the SVM. It does when we use Fisher criterion for dimensionality reduction, while it does not when the PCA is used. This could be explained by the weak performance of the PCA, as it performs a linear transformation on the raw data. The data might not be normally distributed, and thus the correlation matrix fails to identify all the principal components.

We tuned the hyperparamters, and found the best network size is around 1000. This attains high accuracy while avoiding overfitting.

Size of hidden layer	5-fold cross validation					Avg. Accuracy
200	0.9934	0.9978	0.9934	0.9847	0.9934	0.9925
500	0.9891	0.9978	0.9891	1.0000	0.9934	0.9939
1000	0.9891	0.9978	0.9891	1.0000	0.9956	0.9943
1500	0.9891	0.9978	0.9891	1.0000	0.9956	0.9943

Table 3: Finding the proper network size

2.4 Prediction

Since it has the highest accuracy in 5-fold cross validation, the fully connected NN with Fisher-based dimensionality reduction is chosen for prediction. The prediction is stored in “2ctestY.txt” in the root directory.

3 Ten-category classification

3.1 Feature selection

The dataset is too large for SVD (singular value decomposition), so it would be diffucult to perform full PCA on it. We tried using "arpack" instead, but the script ran out of memory on a GPU server.

Considering this, only Fisher criterion was used in this task. We chose 2000 features out of 24307, and trained the reduced dataset using SVM and FCNN.

3.2 Experiments

Dim. Reduction	Classifier	5-fold cross validation					Avg. Accuracy
Fisher	Linear SVM	0.9916	0.9878	0.9850	0.9903	0.9875	0.9884
Fisher	FC NN	0.9891	0.9950	0.9937	0.9925	0.9934	0.9927

Table 4: Comparison

The fully connected NN attains higher accuracy as the linear SVM does.

Size of hidden layer	5-fold cross validation					Avg. Accuracy
200	0.9934	0.9925	0.9875	0.9940	0.9931	0.9921
500	0.9546	0.9928	0.9931	0.9934	0.9866	0.9841
1000	0.9953	0.9944	0.9944	0.9937	0.9937	0.9943
1500	0.9959	0.9940	0.9944	0.9947	0.9940	0.9946

Table 5: Finding the proper network size

The optimal hyperparameter for the hidden layer size is around 1000.

Compared with two-category classification, the ten-category one attains slightly lower accuracy in training and cross validation.

3.3 Prediction

Since it has the highest accuracy in 5-fold cross validation, the fully connected NN with Fisher-based dimensionality reduction is chosen for prediction. The prediction is stored in "10ctestY.txt" in the root directory.

4 Visualization and Clustering

4.1 PCA

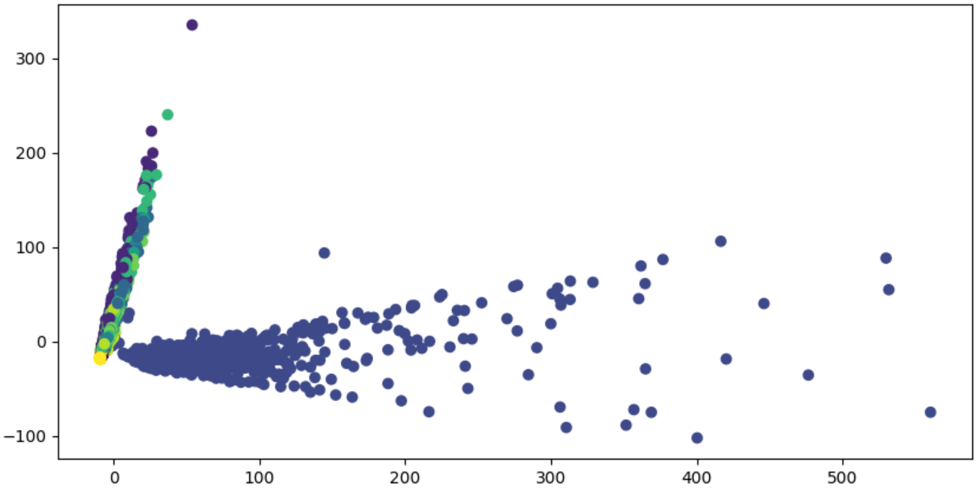

Figure 1: Dimensionality reduction using PCA

4.2 t-SNE

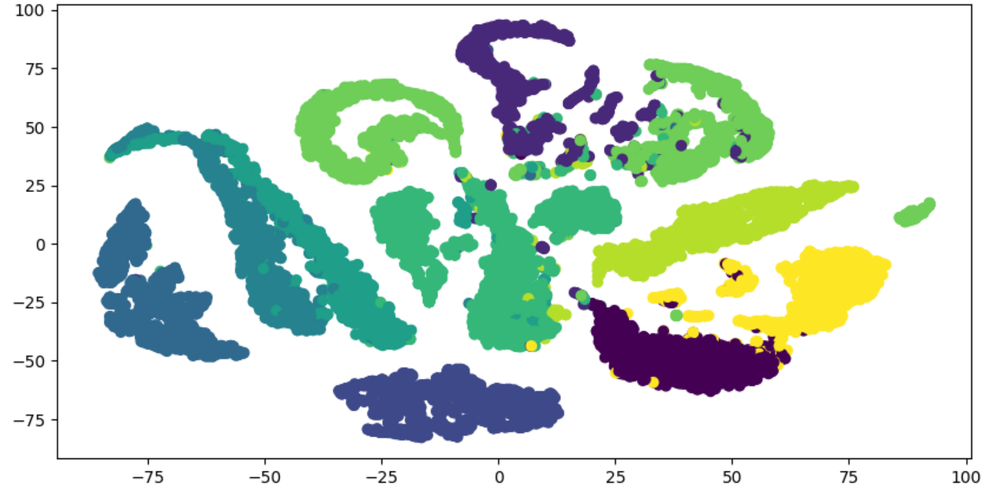

Figure 2: Dimensionality reduction using t-SNE

4.3 Clustering based on t-SNE

It is astouding to find the result of PCA, although with good linearity, does not split up all the 10 categories in the 2d space. Only one category is distributed away from others. In comparison, the t-SNE works fine. We will use t-SNE for clustering.

Since the number of categories is known, k-means seems a good choice. We fix $k = 10$. The birch clustering algorithm is also tested. The Birch algorithm turns out better than K-means.

Method	Accuracy
K-means	0.6926
Birch	0.7343

Table 6: Comparison

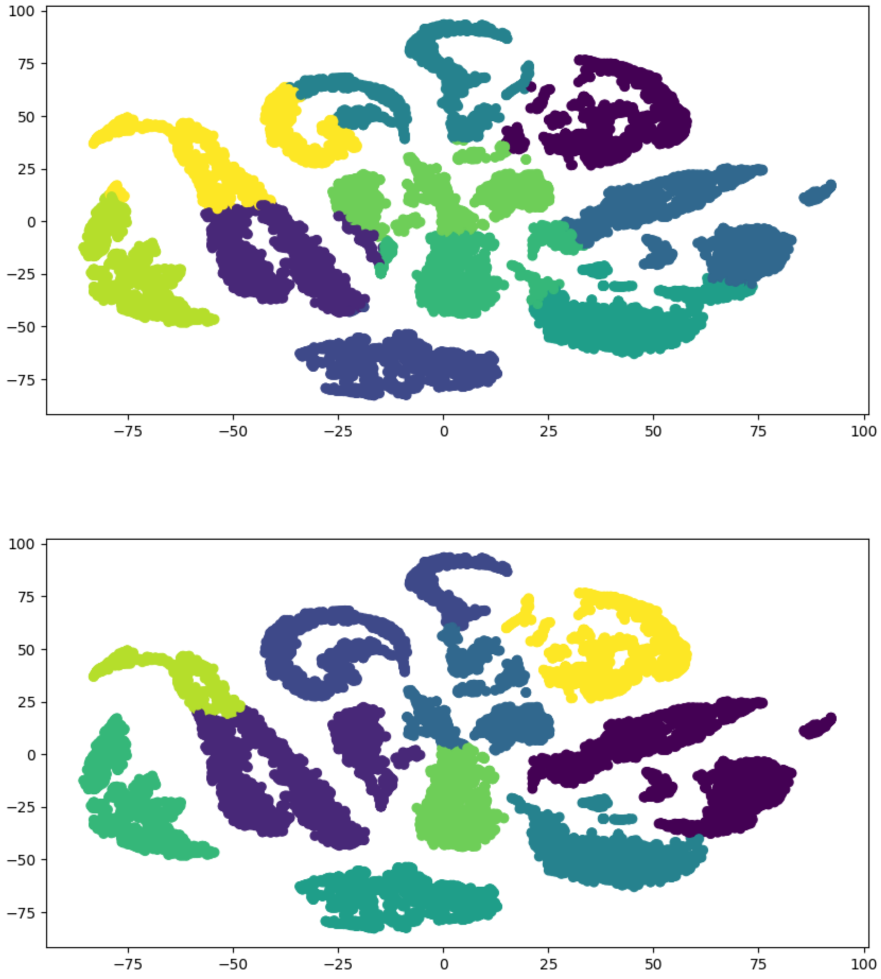

Figure 3: K-means (top); Birch (bottom)

5 Source Code

5.1 Root directory

- “2ctestY.txt”: Two-category prediction. Use this file for evaluation.
- “10ctestY.txt”: Ten-category prediction. Use this file for evaluation.
- “report.pdf”: Project report.

5.2 “data1” directory

- “2ctrainX.txt”: Raw dataset.
- “2ctrainY.txt”: Raw dataset.
- “2ctestY.txt”: Raw dataset.
- “data1.npz”: Preprocessed file. (invalid columns removed)

5.3 “data2” directory

- “10ctrainX.txt”: Raw dataset.
- “10ctrainY.txt”: Raw dataset.
- “10ctestY.txt”: Raw dataset.
- “data2.npz”: Preprocessed file. (invalid columns removed)

5.4 “src1” directory

- “preproc.py”: Preprocessing, removing identical columns.
- “feature_selection”: Feature selection, generating “fisher_2000.npz” and “pca_99.npz”.
- “SVM.py”: Training a SVM and use it for prediction.
- “FCNN”: Training a fully connected NN and use it for prediction.

5.5 “src2” directory

- “preproc.py”: Preprocessing, removing identical columns.
- “feature_selection”: Feature selection, generating “fisher_2000.npz”.
- “SVM.py”: Training a SVM and use it for prediction.
- “FCNN”: Training a fully connected NN and use it for prediction.

5.6 “src3” directory

- “pca.py”: Dimensionality reduction by PCA, generating “pca.npz”.
- “tsne.py”: Dimensionality reduction by t-SNE, generating “tsne.npz”.
- “visualization”: Visualization after loading “pca.pna” and “tsne.npz”.
- “kmeans.py”: Clustering by K-means.
- “birch.py”: Clustering by Birch.