Lab Report For INT104 Coursework 1

Xu Chen
2257453
TA: Changwei Li

***Abstract*—Data reduction aids in visualizing data directly, facilitating further analysis or classification. We demonstrate that by employing the greedy method in conjunction with information gain, we achieve a comprehensive understanding of data classification. With only four extracted features, computational resources are conserved, while achieving robust generalization performance in future training.**

I. Introduction

Data reduction is paramount in machine learning, as it significantly reduces computational complexity, mitigates dimensionality issues, minimizes data noise, and facilitates data visualization. Common data reduction techniques include Principal Component Analysis (PCA), Locally Linear Embedding (LLE), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Autoencoder. In this report, we aim to perform data reduction on a given dataset comprising 11 features: Index, Gender, Programme, Grade, Total, MCQ, Q1, Q2, Q3, Q4, and Q5, with a sample size of 619. Our objective is to achieve classification aligned with the Programme label through visualization.

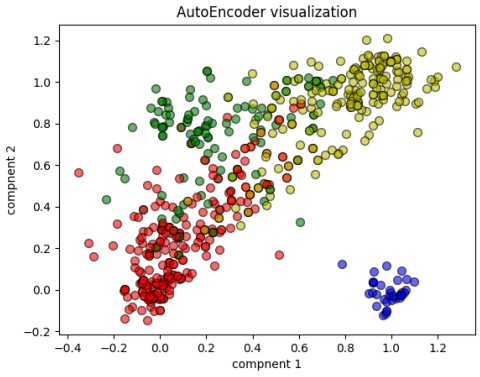

Fig. 1. The distribution of each class of data.

My Contributions:

- We use a greedy algorithm combined with information gain to get the extracted data not based on Experimental trial and error. We use experiments and correlation components with Programme data to prove it comes true.
- Successfully get a good classification based on Autoencoder with feature extracted in Figure 1.
- Demonstrate the outlier and variance order in the box plot have no influence on feature extraction and explain the benefits compared with step-by-step classify tags.

II. Data Observation Using Box Plot

Initially, we generated a box plot using the raw data. Box plots are effective tools for displaying the median, quartiles, and outliers within the data. In Figure 2, the orange line represents the median. Upon observation, we noted that the Index feature exhibits excessively high values, obstructing the visualization of other feature distributions. Given that it does not significantly impact the dataset, we opted to remove it.

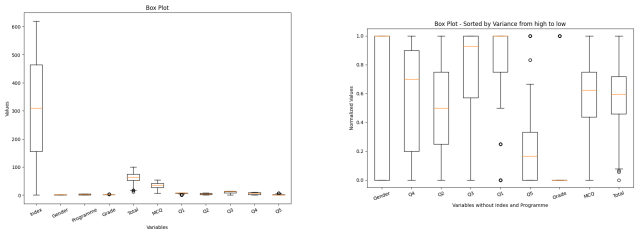

Fig. 2. Box plot in raw data and reprocessed data

Additionally, we set the column features to be sorted by variance (from highest to lowest) and normalized the data using the min-max method, which allows us to ensure that the value is between 0 and 1. From Figure 2, we can see that Q1, Q5, Grade, and Total have outliers according to the calculation of the box plot program (not means have outliers in real data). Whether this order has a connection with feature extraction and if the outliers in the feature will influence the feature extraction? We will explain this in section V.

III. Data Reduction Using PCA

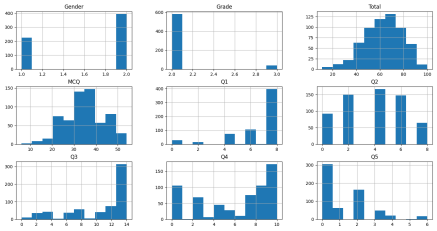

Fig. 3. The distribution of each feature of data.

After seeing all values are in the range in Figure 3, We normalize the data by calculating

$$x' = \frac{x - \text{mean}}{\text{std}} \tag{1}$$

where x represents each value of the features, mean is the average value of the specific feature of the sample, and std is the standard deviation of the specific feature of the sample.

From Figure 4, it becomes apparent that we are unable to plot at least 8 principal components due to the constraints imposed by the explained variance ratio, which must be larger or equal to 0.95. This observation underscores the challenge of achieving a comprehensive visualization of the data in both 2D and 3D using PCA without prior data extraction.

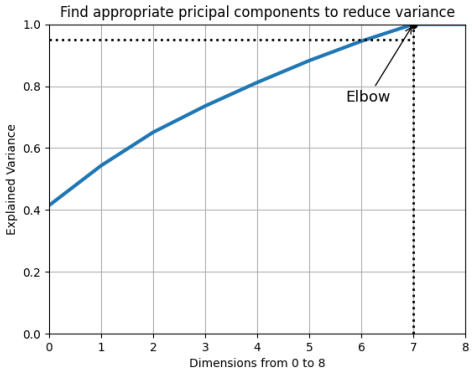

Fig. 4. The explained ratio of the normalized data.

Next, we draw the figure of 2D and 3D versions of PCA, which each have a 54 percent and 65 percent explained ratio. We can see the scatters have four colors from Figure 5: the red color represents Programme 1, the green color represents Programme 2, the blue color represents Programme 3 and the yellow color represents Programme 4.

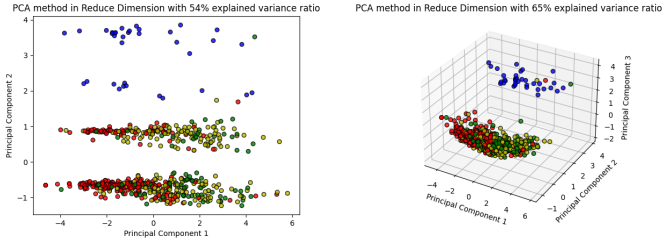

Fig. 5. PCA Projections in 2D and 3D

Although not all classes could be classified, we observed that both PCA and t-SNE successfully extracted the data related to Programme 3. Comparing the two figures, the 3D plot exhibited superior performance in visualizing the red class, benefiting from the expansive vector space that effectively showcased its discrete distribution. However, the 2D plot provided a clearer understanding of the data. Consequently, we will proceed with utilizing only two components in subsequent figures.

IV. Algorithm to Extract Feature

We do not rule out that this is caused by the PCA algorithm alone, but we will prove in Chapter 5 that traditional algorithms are indeed unable to perform effective classification. The algorithms we use here include PCA, t-SNE, LLE, and Autoencoder.

In this section, we will describe how we extract the feature and the result you can see in section V . Extracting features can reduce calculations, remove redundant information, and reduce noise. Common feature extraction methods include Filter Method, Wrapper Method, and Embedded Method. We calculate the information gain here. Information gain tells us how important a given attribute of the feature vectors is.

$$\text{Gain(S, f)} = H(S) - \sum_{i=1}^{n} \left(\frac{|S_i|}{|S|} H(S_i) \right) \tag{2}$$

$$H(S) = -\sum_{c \in C} p_c \log_2(p_c) \tag{3}$$

Where S is the whole dataset, $\{S_i\}$ are the datasets extracted from the feature f at each value. $H(S)$ is the information entropy of S. p_c is the probability of an element belonging to class c.

Algorithm 1 Feature Selection Using Information Gain

$X \leftarrow$ standardized features
$y \leftarrow$ class labels
$list \leftarrow []$
$remaining\_features \leftarrow$ features in X
while $remaining\_features \neq \emptyset$ **do**
  $gains \leftarrow []$
  for each $feature$ in $remaining\_features$ **do**
    $gain \leftarrow Equation(2)$
    $gains \leftarrow gains \cup \{(gain, feature)\}$
  end for
  $(best\_gain, best\_feature) \leftarrow$ feature with max $gain$
  if $best\_gain \leq 0.1$ **then**
    exit loop
  end if
  $list \leftarrow list \cup \{best\_feature\}$
  $remaining\_features \leftarrow remaining\_features \setminus \{best\_feature\}$

Information gain is used here because it most directly represents how much uncertainty is eliminated by the features used, effectively reducing the early dependence on labels. Here, our pseudo code first finds the optimal single feature through information gain, then retains the features selected in the first round, and continues to select and evaluate the remaining features and the first feature combination that is optimal and better than the first round and exceeds a certain threshold. will be selected, and so on, until no one can be found that meets the conditions. This method is reasonable as it compares all conditions and we just have 9 features that do not need large computation.

In all, we remove the features: 'Q2', 'Q5', 'Q3', 'Q1', and 'Gender' and we reserve the features: Grade, Total, MCQ, Q4.

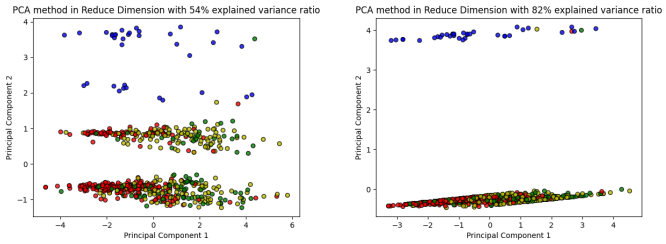

Fig. 6. PCA with raw feature and extracted feature

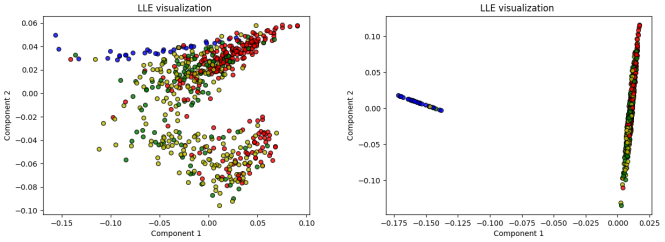

Fig. 7. LLE with raw feature and extracted feature

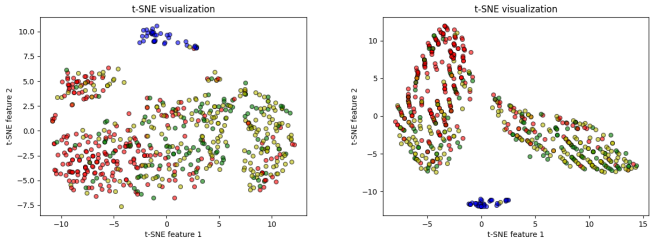

Fig. 8. t-SNE with raw feature and extracted feature

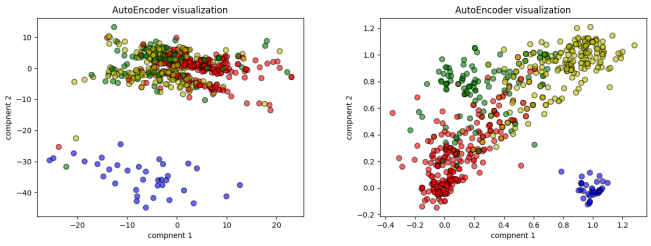

Fig. 9. AutoEncoder with raw feature and extracted feature

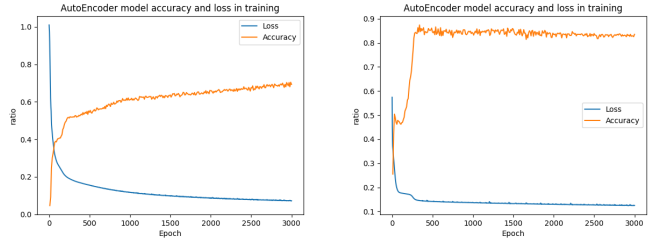

Fig. 10. Accuracy and Loss with raw feature and extracted feature

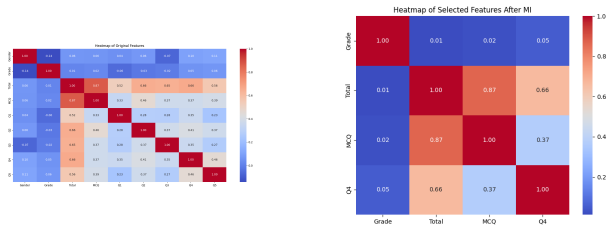

Fig. 11. Hotpot with raw feature and extracted feature

V. Compare and Visualize the Experiment Results

A. Whether the extracted feature helps the Data reduction ?

In PCA, the explained variance ratio has increased from 54 to 82, and the blue label appears farther from the others in Figure 6. With LLE, setting neighbors to 16 and the random state to 128, the red label is easily distinguishable, mainly located in the upper right corner, unlike the previous scenario where it couldn't be effectively separated. In t-SNE, with a random state of 60, perplexity of 100, learning rate of 50, iteration of 1000, and exaggeration of 70, compared to the previous setting, the red label tends to cluster more towards the upper left corner. In AutoEncoder, using networks with 3 linear layers and 2 ReLU layers for both the encoder and decoder, employing MSE loss function, and training for 3000 epochs, the extracted features can almost completely separate all labels, achieving an accuracy close to 90percent, as opposed to the previous 70. This indicates that the extracted features are easier to train and result in better classification performance.

B. Whether the extracted features are optimal ?

It can be seen from the heat map that most of the correlations of feature selection are extremely high. Compared with the past, gender is due to the lack of representation because there are only boys and girls, and the calculation shows that the correlation coefficient is low.

We use mutual information to calculate the value between each feature and the label Programme, and take the absolute value. Here we calculate 10 times, randomly select the seed, and then average it to get the results in the table. We can find that the four features I extracted are all among the top five, among which Q2 and Q4 are very similar and lead the next fault. It can be considered that the feature selected by the algorithm is the optimal solution.

C. Discussion and Conclusion

In this report ,we successfully extract the label with AutoEncoder and demonstrate the features that may be the optimal in classification : Grade, Total, MCQ, Q4.

#	Feature	Average Mutual Information
1	Grade	0.198751
2	Total	0.187396
3	MCQ	0.104358
5	Q2	0.087815
7	Q4	0.086718
8	Q5	0.058307
6	Q3	0.048980
4	Q1	0.040192
0	Gender	0.032749

TABLE I
Feature Ranking Based on Average Mutual Information