Discriminant analysis classifiers

Shiou-Shiou Deng & Guangjie He

Math 285: Classification with Handwritten Digits

Introduction

As one of the fundament problems in designing practical recognition systems, the recognition of handwritten digits is an active research field. One of the major challenges in the recognition of handwritten digits is the within class variance, because people do not always write the same digit in exactly the same way. Classification of hand written digits is the task of predicting the class to which a number belongs.
In this poster, we introduce three different classifications: Quadratic Discriminant Analysis (QDA), Linear Discriminant Analysis (LDA), and Naive Baye, and implement these classifiers to the MNIST data. The results not only give raw accuracy in each dimension but also the time requirement. Furthermore, comparing these three classifications' performances, we have better understanding the properties of data and which situations the classifiers usually failed. The last but not least, comparing methods which handling the singularity is another useful issue for recognition systems.

Analysis Method

Maximum A posterior classification

Definition: to assign the label based on the posterior probabilities

$$i = \arg\max_i \; p(x \in C_i \mid x)$$

According to the Bayes' rule

$$p(x \in C_i \mid x) = \frac{f(x \mid x \in C_i)\, p(x \in C_i)}{f(x)} \propto f_i(x)\pi_i$$

This is called MAP classification.

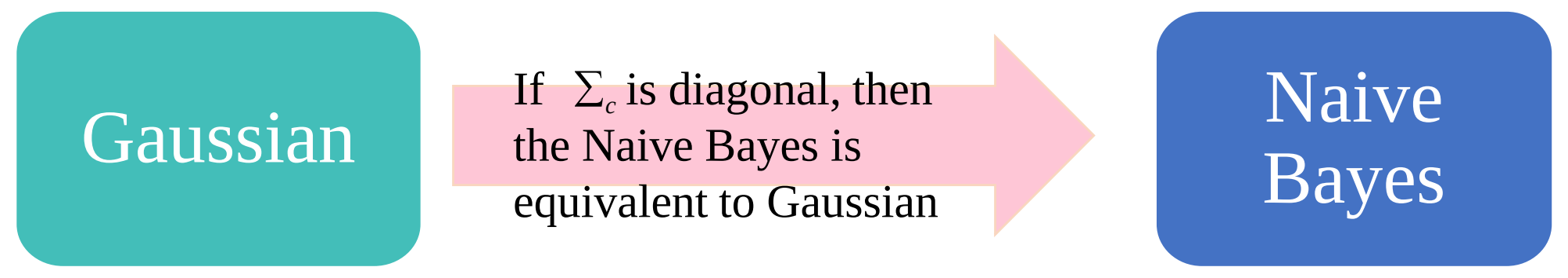
To estimate $f_i(x)$, we need to pick a model
- LDA/QDA: by using multivariate Gaussian distributions
- Naive Bayes: by assuming independent features in $x = (x_1, \ldots, x_d)$

Gaussian Discriminant Analysis

Definition: One important application of multivariate normal is to define the class conditional densities in a generative classifier

$$p(x \mid y = c, \theta) = N(x \mid \mu_c, \Sigma_c)$$

The result technique is called Gaussian discriminant.

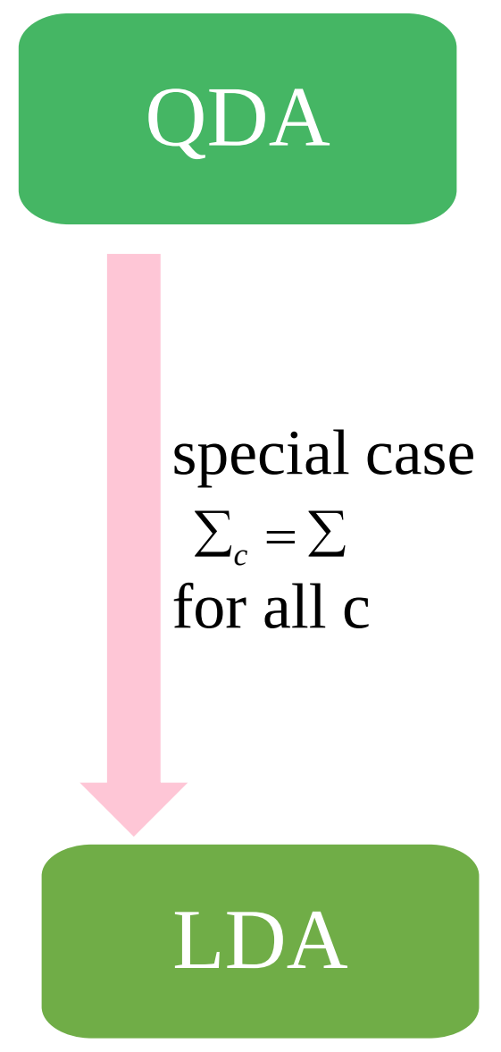


Gaussian

If Σ_c is diagonal, then the Naive Bayes is equivalent to Gaussian

Naive Bayes

The model is called "naive" since we do not expect the features to be independent, even conditional on the class label. However, even if the naive Bayes assumption is not true, it often results in classifiers that work well. One reason for this is that the model is quite simple.



QDA

special case $\Sigma_c = \Sigma$ for all c

LDA

- More specifically, for LDA and QDA, is modelled as a multivariate Gaussian $p(x \mid y)$ distribution
- In the case of LDA, the Gaussians for each class are assumed to share the same covariance matrix
- In the case of QDA, there are no assumptions on the covariance matrices $\Sigma_c$of the Gaussians, leading to quadratic decision surfaces.

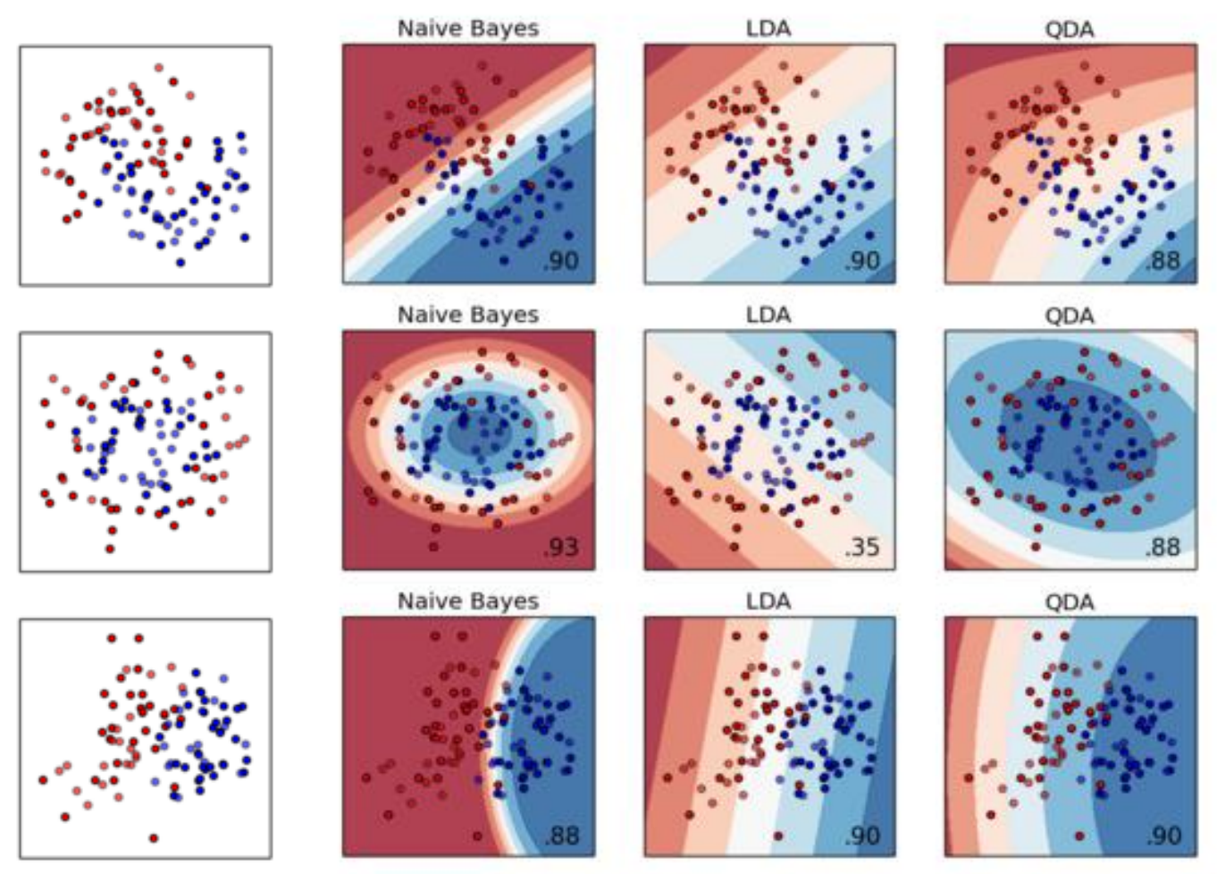


Figure 1: The appearances of distinguishing the different data using the three classifiers.

Figure 1 showed in the first row the LDA performs better result than the other two classifiers if the data look like linear patterns. In the second row, if the data look like circle patterns, the Naïve Bayes performs better than the other two classifiers. In the third row, QDA performs better than the other two classifiers if the data look like quadratic patterns.

Result

Figure 2 showed that after classifying the test images using from 1 to 154 dimensions through LDA, QDA, NB with normal distribution and NB with kernel distribution, the four curves of test error are consistent. The test errors appeared steady at each classifier with 20 or more dimensions. Among the four classifiers, quadratic discriminant analysis (QDA) performed the best result. Figure 3 showed that the curves of test errors using 2DLDA + LDA and using 2DLDA + QDA are inconsistent. With large size, 2DLDA+LDA performs better than 2DLDA+QDA.

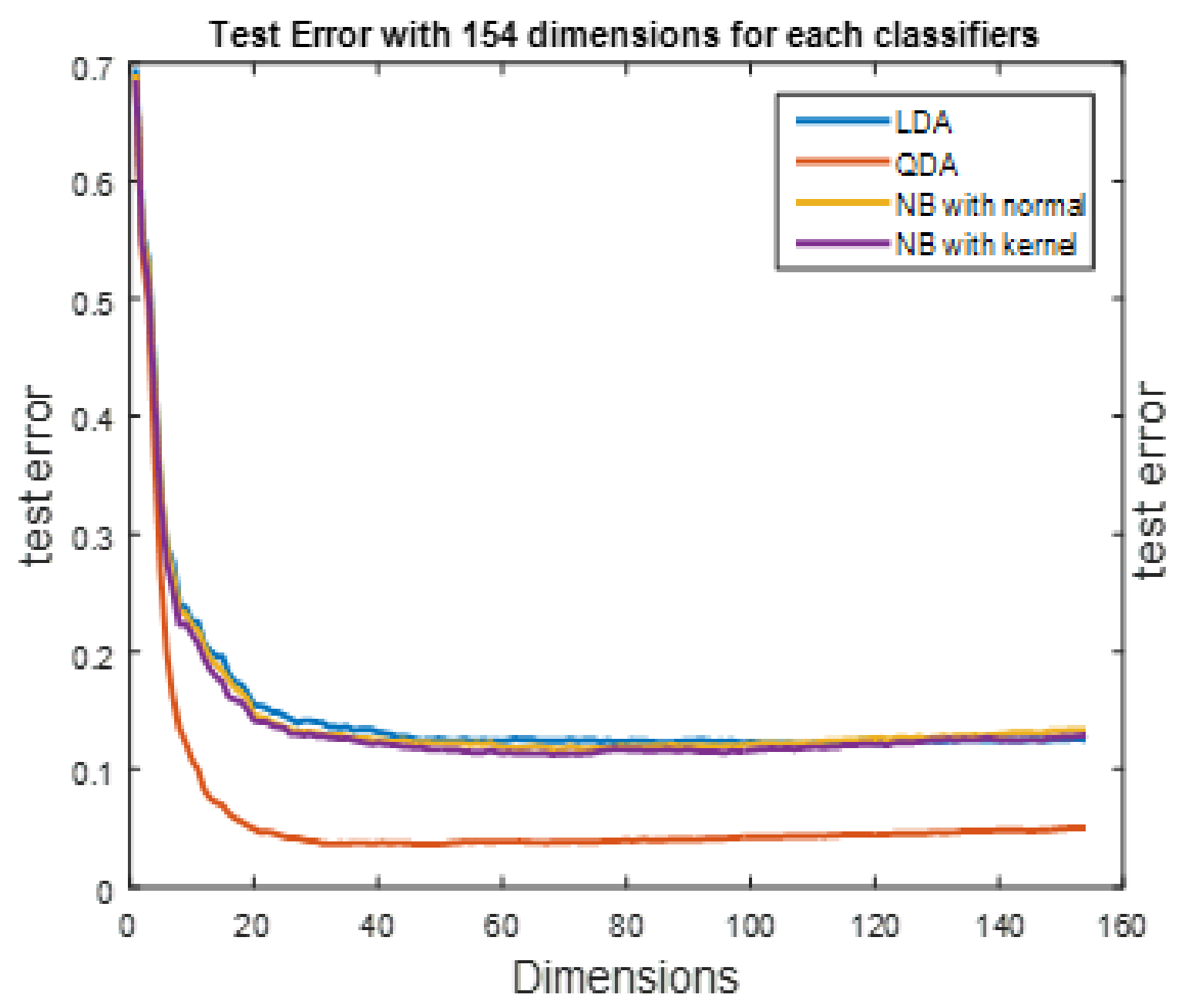


Figure 2: The curves of test error using four classifiers with from 1 to 154 dimensions

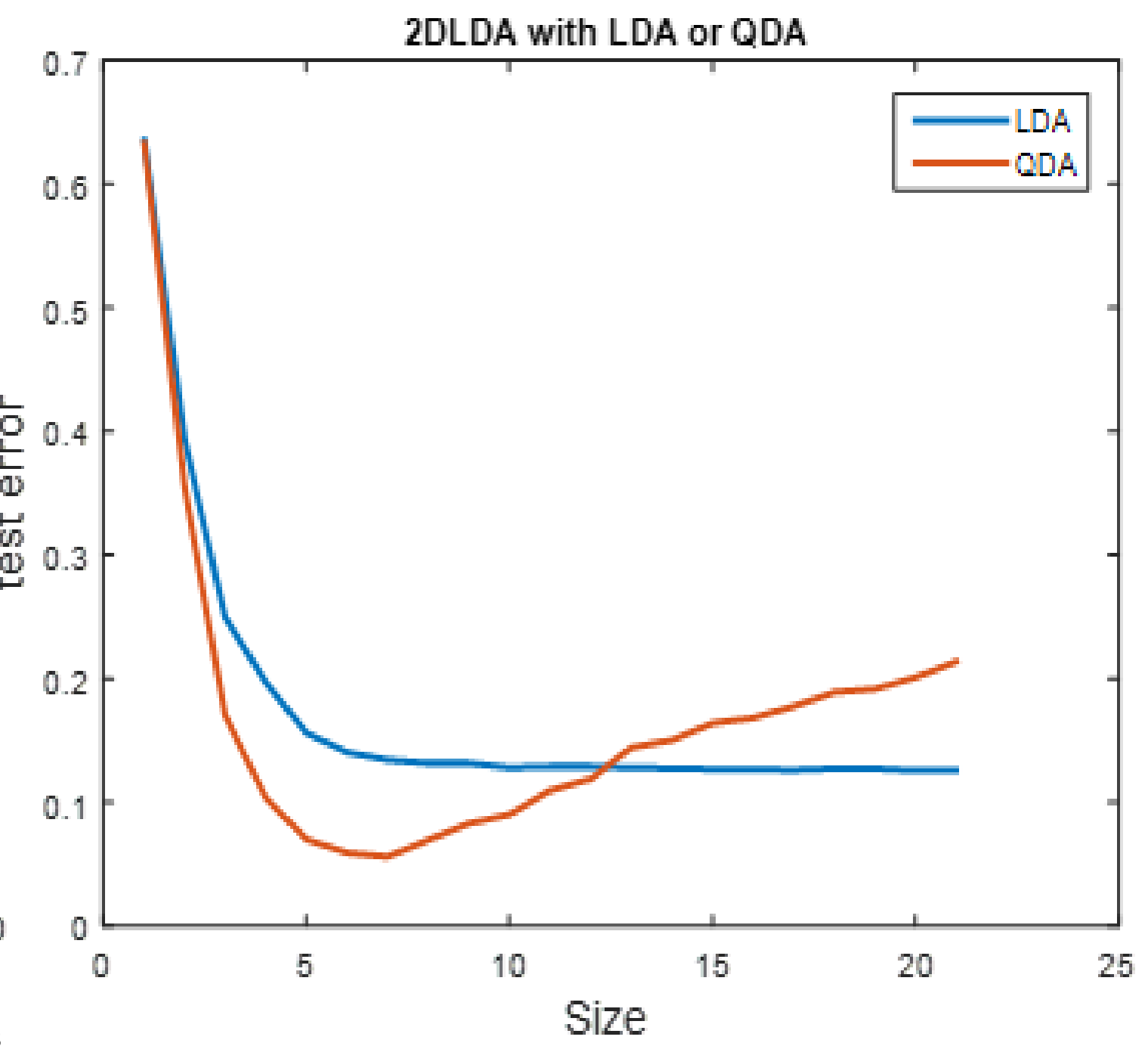


Figure 3: The curve of test error using 2DLDA + LDA with from 1 to 28 sizes

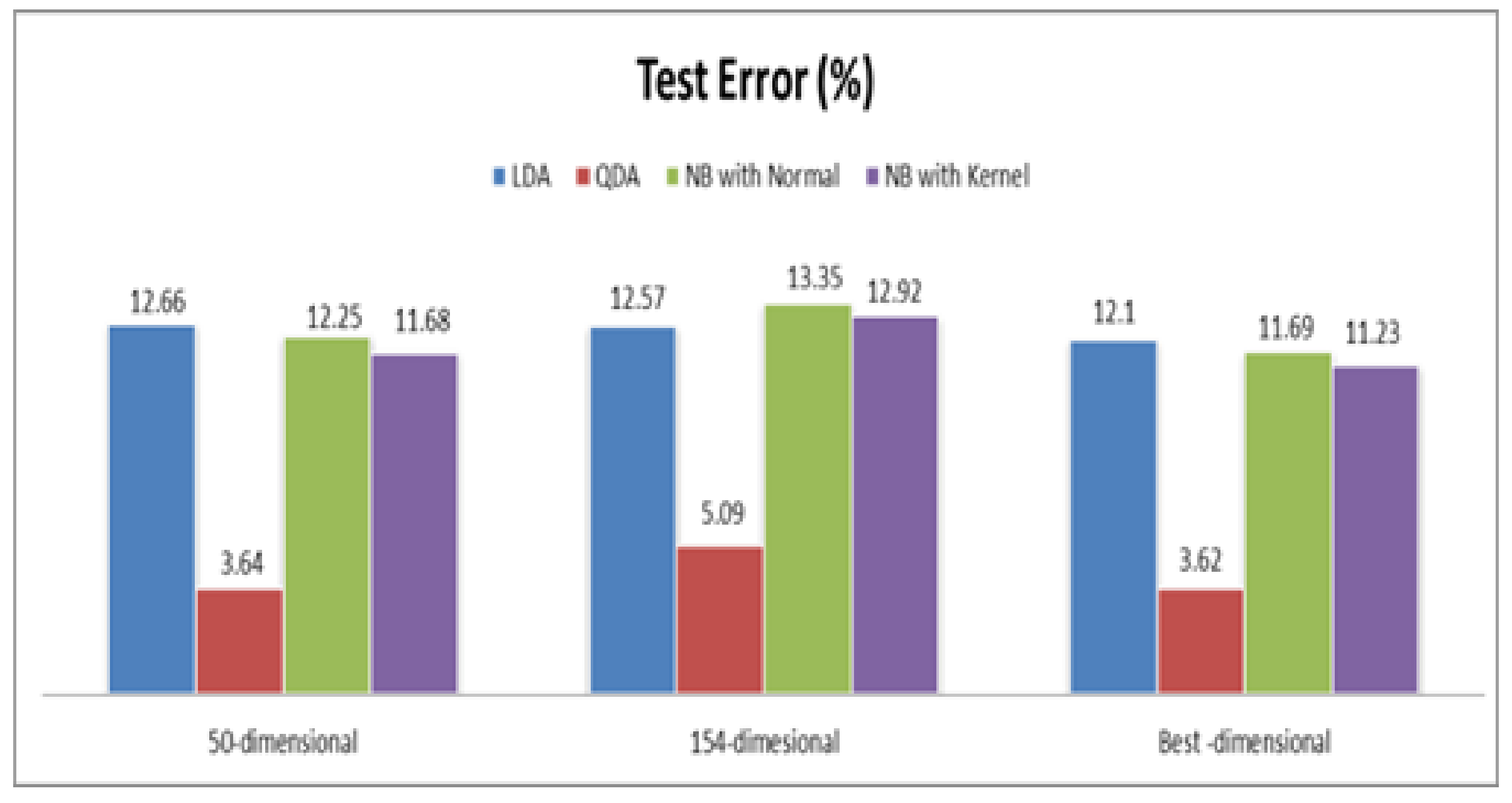


Figure 4: The bar charts of test error using four classifiers with 50, 154, and the dimension of performing best

Figure 4 showed the test error were produced by the four classifiers in the 50-dimensional feature vector, which represents 84% variance, the 154-dimensional feature vector, which represents 95% variance, and the best feature vector. LDA and NB with kernel distribution with 154 dimensions perform better than with 50 dimensions. However, QDA and NB with normal distribution with 50 dimensions perform better than with 154 dimensions. In the last bar chart, LDA performs best with 109 dimensions, QDA performs best with 46 dimensions, NB with normal performs best with 69 dimensions, and NB with kernel performs best with 68 dimensions. Also, considering the time-consuming issue, QDA is the fastest among the four classifiers.

Figure 5 showed that comparing the PCA to the 2DLDA, the test error with LDA classifiers decreases when the dimensions or the size increases, however, the test error with QDA classifiers increases when the dimensions or the sizes increases. In addition, PCA with LDA or QDA perform better than 2DLDA with LDA or QDA. Figure 6 showed that each classifier with different method to dealing the singularity performs the lowest test error. PCA + QDA performs the best result among those classifiers. Comparing QDA with the four methods to dealing the singularity, PCA performs better than the other three methods, 2DLDA, Direct QDA, and Psuedoinverse.

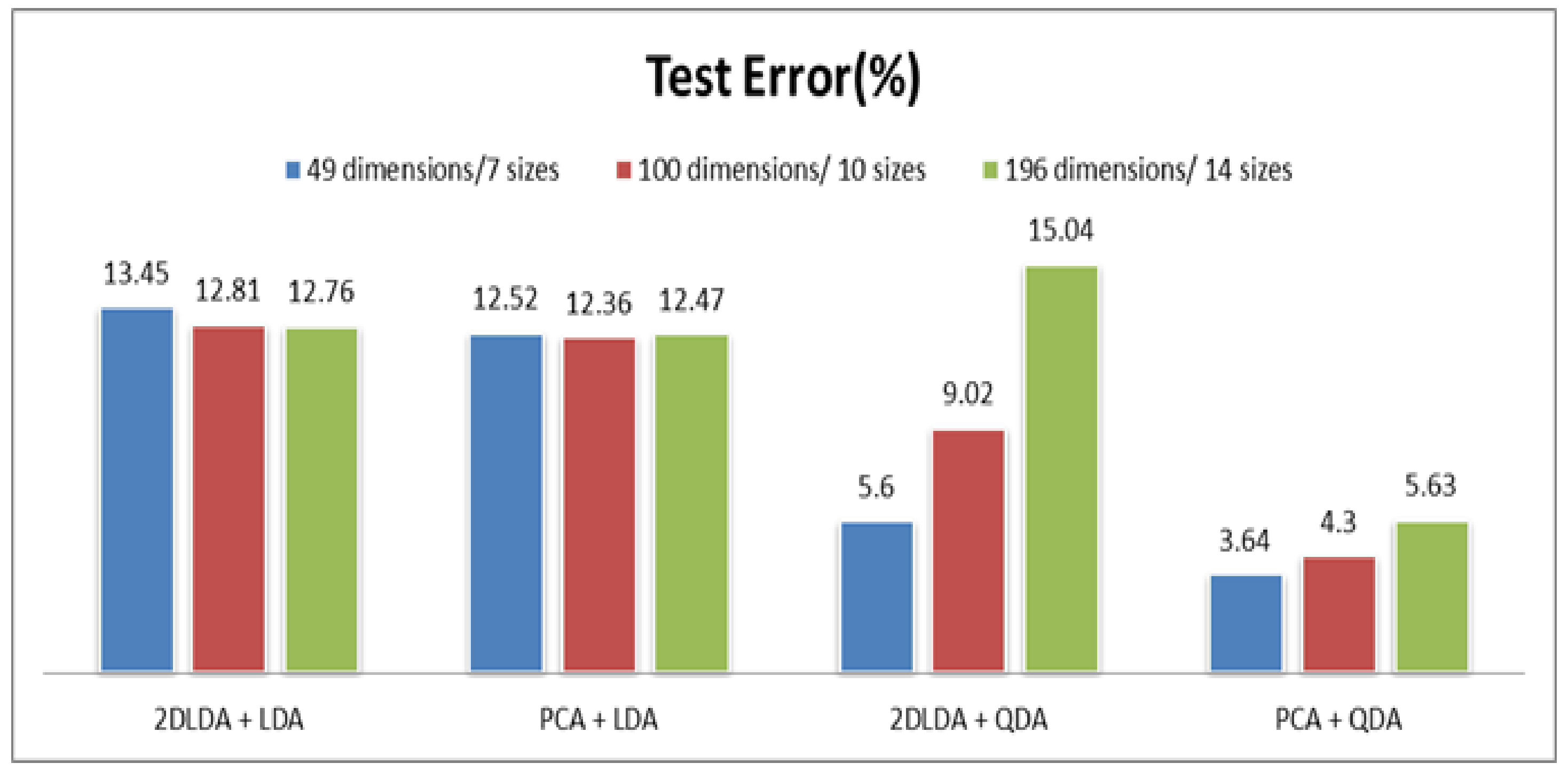


Figure 5: the test error of LDA and QDA using PCA or 2DLDA

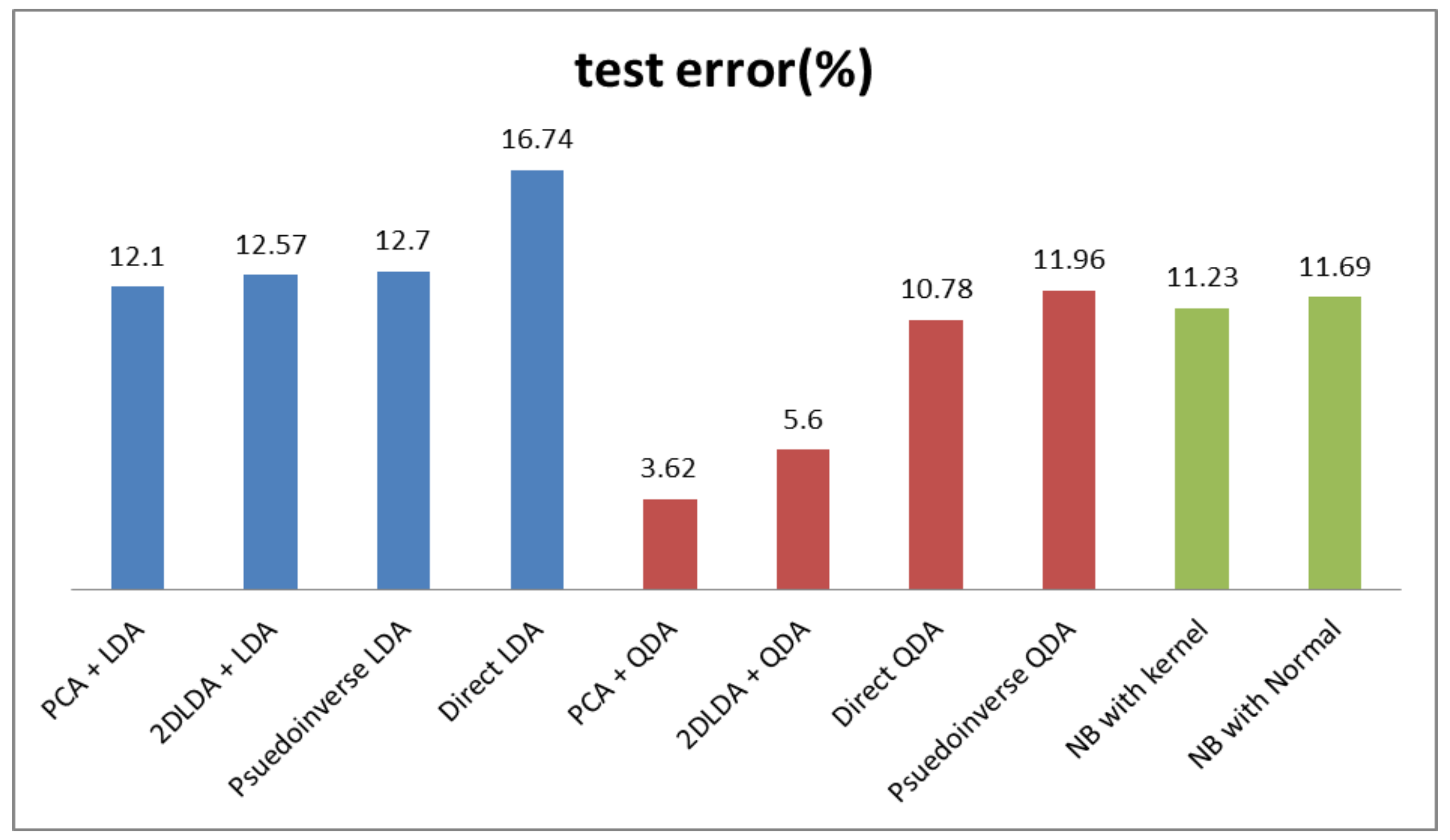
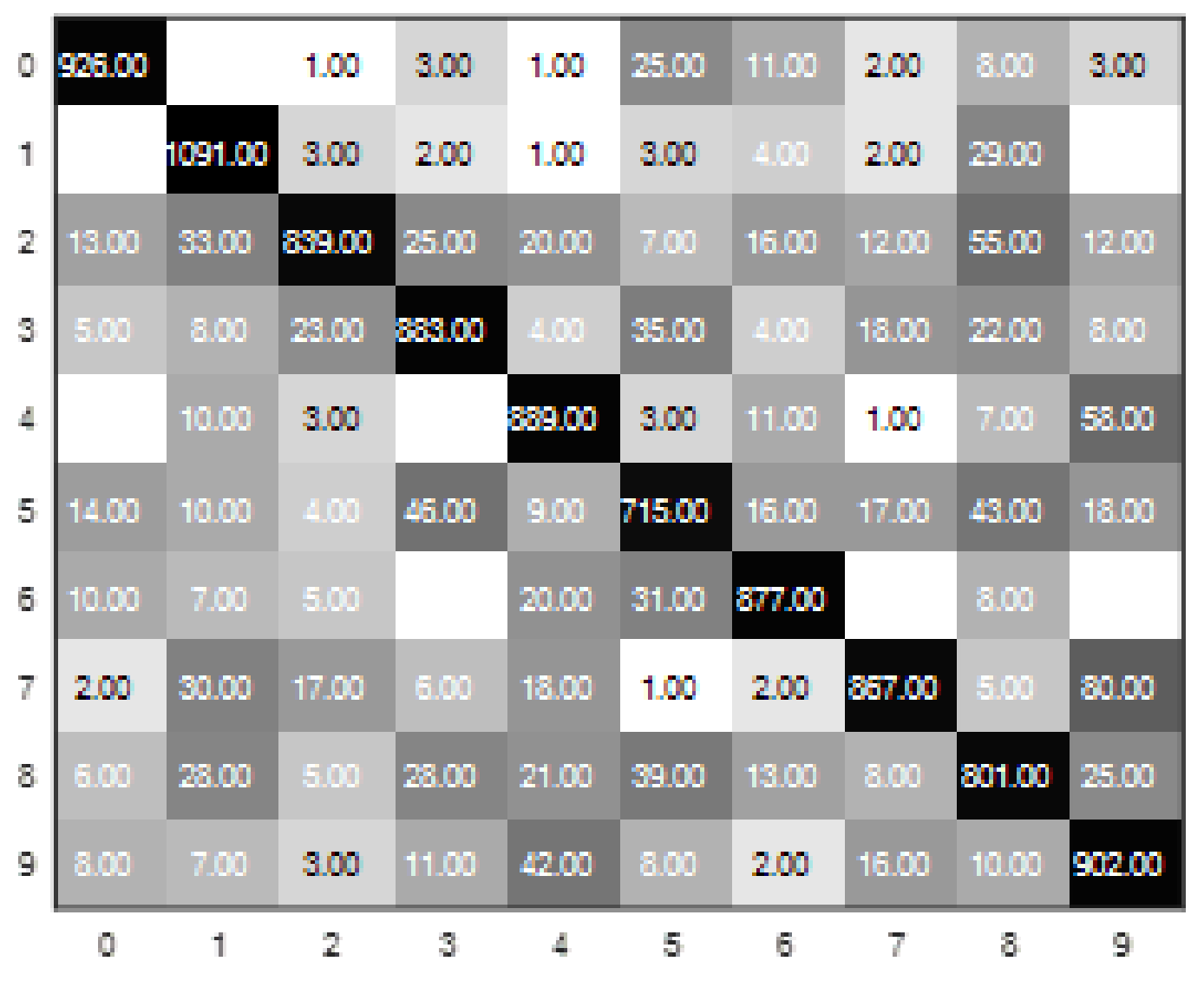


Figure 6: the lowest test error which each classifier produces

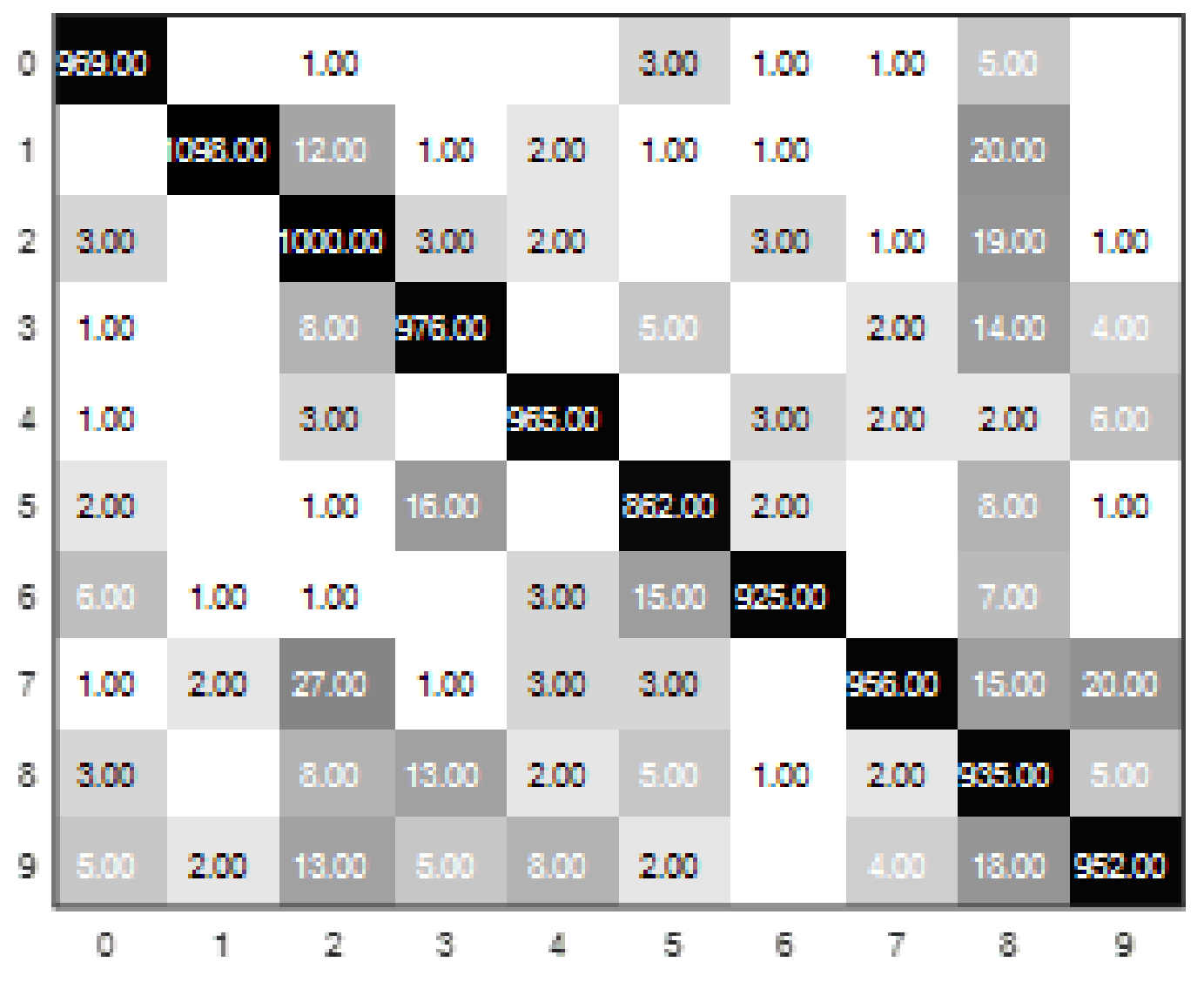
From the table 1, the test errors were higher when LDA classified 3 and 5, 4 and 9, 5 and 8, 2 and 8, 5 and 8. The table 2 showed that the test errors were higher when QDA classified 7 and 2. The table 3 showed that the test errors were higher when NB with normal distribution classified 3 and 5, 2 and 8. The table 4 showed that the test errors were higher when NB with kernel distribution classified 3 and 5, 2 and 8, 4 and 9.

Table 1: The confusion matrix of Linear Discriminant Analysis with PCA



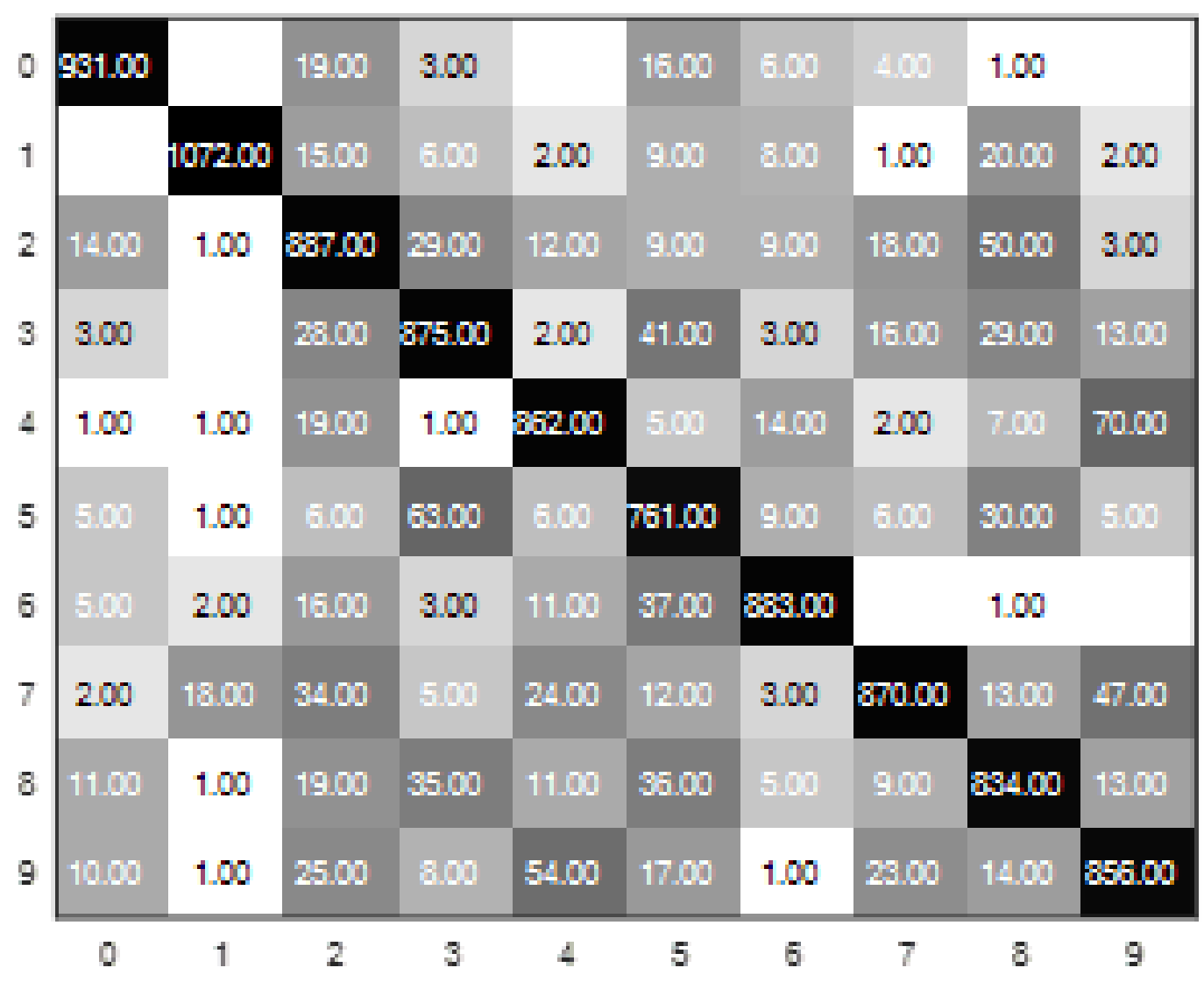
	0	1	2	3	4	5	6	7	8	9
0	926.00		1.00	3.00	1.00	25.00	11.00	2.00	8.00	3.00
1		1091.00	3.00	2.00	1.00	3.00	4.00	2.00	29.00	
2	13.00	33.00	839.00	25.00	20.00	7.00	16.00	12.00	55.00	12.00
3	5.00	8.00	23.00	883.00	4.00	35.00	4.00	18.00	22.00	8.00
4		10.00	3.00		889.00	3.00	11.00	1.00	7.00	58.00
5	14.00	10.00	4.00	45.00	9.00	715.00	16.00	17.00	43.00	18.00
6	10.00	7.00	5.00		20.00	31.00	877.00		8.00	
7	2.00	30.00	17.00	6.00	18.00	1.00	2.00	857.00	5.00	80.00
8	6.00	28.00	5.00	28.00	21.00	39.00	13.00	8.00	801.00	25.00
9	8.00	7.00	3.00	11.00	42.00	8.00	2.00	16.00	10.00	902.00

Table 2: The confusion matrix of Quadratic Discriminant Analysis with PCA



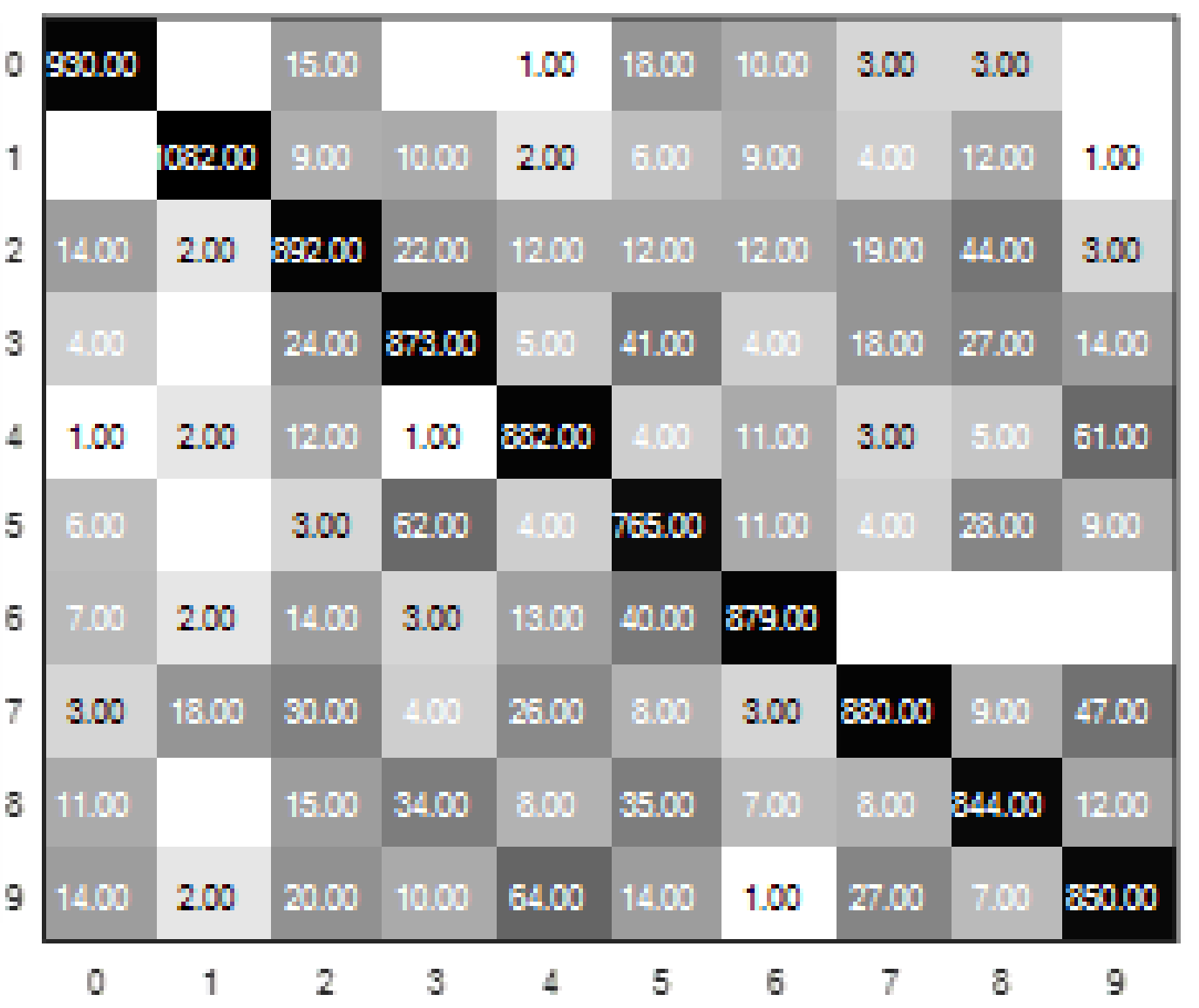
	0	1	2	3	4	5	6	7	8	9
0	969.00		1.00			3.00	1.00	1.00	5.00	
1		1098.00	12.00	1.00	2.00	1.00	1.00		20.00	
2	3.00		1000.00	3.00	2.00		3.00	1.00	19.00	1.00
3	1.00		8.00	976.00		5.00		2.00	14.00	4.00
4	1.00		3.00		965.00		3.00	2.00	2.00	6.00
5	2.00		1.00	16.00		862.00	2.00		8.00	1.00
6	6.00	1.00	1.00		3.00	15.00	925.00		7.00	
7	1.00	2.00	27.00	1.00	3.00	3.00		956.00	15.00	20.00
8	3.00		8.00	13.00	2.00	5.00	1.00	2.00	935.00	5.00
9	5.00	2.00	13.00	5.00	8.00	2.00		4.00	18.00	952.00

Table 3: The confusion matrix of Naïve Bayes classifier with normal distribution



	0	1	2	3	4	5	6	7	8	9
0	931.00		19.00	3.00		16.00	6.00	4.00	1.00	
1		1072.00	15.00	6.00	2.00	9.00	8.00	1.00	20.00	2.00
2	14.00	1.00	887.00	29.00	12.00	9.00	9.00	18.00	50.00	3.00
3	3.00		28.00	875.00	2.00	41.00	3.00	16.00	29.00	13.00
4	1.00	1.00	19.00	1.00	852.00	5.00	14.00	2.00	7.00	70.00
5	5.00	1.00	6.00	63.00	6.00	761.00	9.00	6.00	30.00	5.00
6	5.00	2.00	16.00	3.00	11.00	37.00	883.00		1.00	
7	2.00	18.00	34.00	5.00	24.00	12.00	3.00	870.00	13.00	47.00
8	11.00	1.00	19.00	35.00	11.00	36.00	5.00	9.00	834.00	13.00
9	10.00	1.00	25.00	8.00	54.00	17.00	1.00	23.00	14.00	856.00

Table 4: The confusion matrix of Naïve Bayes classifier with kernel distribution



	0	1	2	3	4	5	6	7	8	9
0	930.00		15.00		1.00	18.00	10.00	3.00	3.00	
1		1082.00	9.00	10.00	2.00	6.00	9.00	4.00	12.00	1.00
2	14.00	2.00	892.00	22.00	12.00	12.00	12.00	19.00	44.00	3.00
3	4.00		24.00	873.00	5.00	41.00	4.00	18.00	27.00	14.00
4	1.00	2.00	12.00	1.00	882.00	4.00	11.00	3.00	5.00	61.00
5	6.00		3.00	62.00	4.00	765.00	11.00	4.00	28.00	9.00
6	7.00	2.00	14.00	3.00	13.00	40.00	879.00			
7	3.00	18.00	30.00	4.00	26.00	8.00	3.00	880.00	9.00	47.00
8	11.00		15.00	34.00	8.00	35.00	7.00	8.00	844.00	12.00
9	14.00	2.00	20.00	10.00	64.00	14.00	1.00	27.00	7.00	850.00

Conclusion

- ❖ The high test errors LDA or NB produced means that the methods with their assumptions probably be inappropriate for the MNIST data.
- ❖ In general, QDA with low dimensions performs well and quickly than the other three classifiers, LDA, NB with normal, and NB with kernel.
- ❖ Instead of using 2DLDA, using PCA to deduct the dimensions in MNIST data is a better choice.
- ❖ Handling the singularity problem with MNIST data, PCA performs better results than the other three methods, 2DLDA, Direct QDA, and Psuedoinverse.
- ❖ The four classifiers revealed that the machines would be confused when classifying 3 and 5, 4 and 9, 2 and 8, and LDA had more difficulties about classifying these numbers than the other three classifiers.

Handling singularity

A low-definition images of size 28 by 28 implies a feature space of 28 28 = 784 dimensions, and those matrices are almost always singular. Due to the problem, there are some different methods to handle singularity.

❖ PCA + LDA/QDA: Dimensionality reduction is the important factor which is used to reduce the features of the original data without the loss of information. PCA algorithm is used in number of applications to reduce the features and transforms the higher dimensional space into lower space and performs well. (Van Der Maaten, Postma, & Van den Herik, 2009) To overcome this issue, we projected the data using principal component analysis before applying LDA or QDA, and from figure 7, the 50-dimensional feature vector represented 84% variances and the 154-dimensional feature vector represented 95% variances.

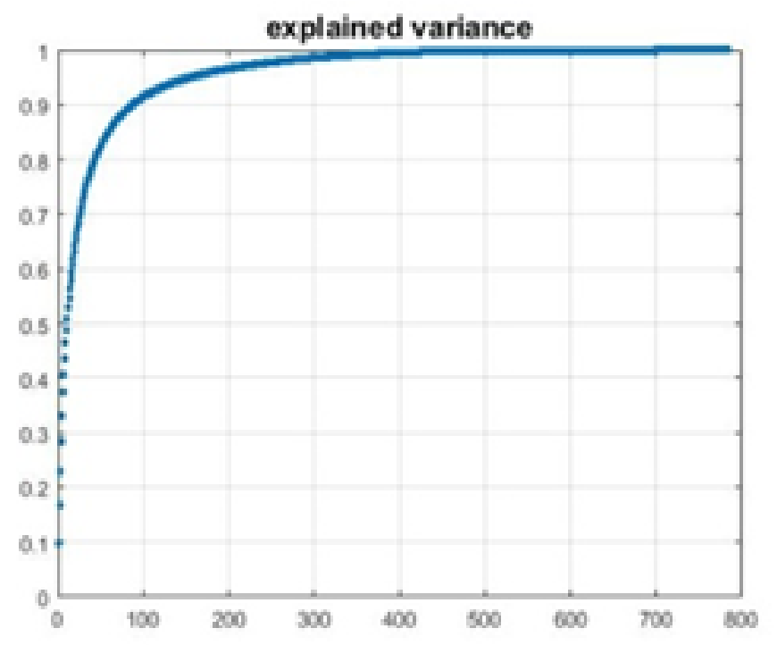


Figure 7: The explained variance of the 1-784 dimensional feature vectors

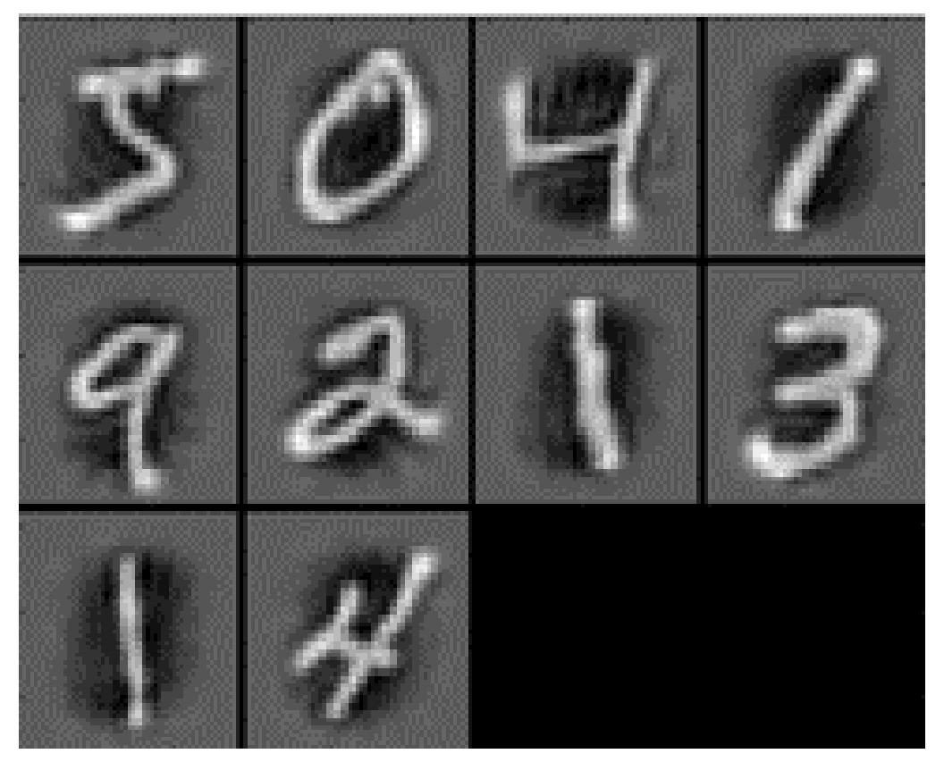


Figure 8: The Images with 154-dimensiona l feature vector

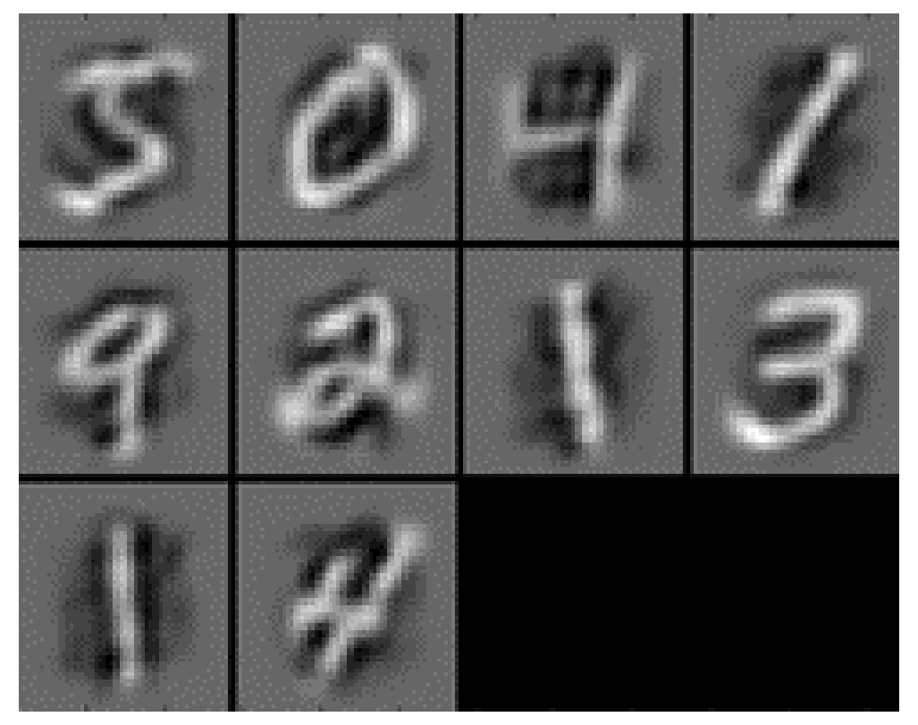


Figure 9: The images with 50-dimensional feature vector

❖ Psuedoinverse LDA/QDA: A common use of the pseudoinverse is to compute a best fit solution to a system of linear equations that lacks a unique solution. Another use is to find the minimum normal solution to a system of linear equations with multiple solutions. The pseudoinverse facilitates the statement and proof of results in linear algebra.

❖ Direct LDA/QDA: Considering the PCA criterion may not be compatible with the LDA criterion because the PCA step may discard dimensions that contain important discriminative information. A direct, exact LDA algorithm could accept high dimensional data as input, and optimizes Fisher's criterion directly, without any feature extraction or dimensionality reduction steps. (Yu & Yang, 2001)

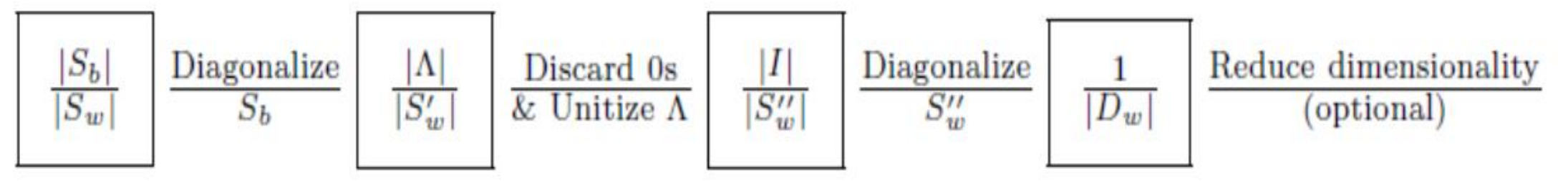


Figure 10: The algorithm of the direct LDA

❖ 2DLDA + LDA/QDA: PCA+LDA has high costs in time and space, due to the need for an eigen-decomposition involving the scatter matrices. 2DLDA overcomes the singularity problem implicitly, while achieving efficiency. The key difference between 2DLDA and classical LDA lies in the model for data representation. Classical LDA works with vectorized representations of data, while the 2DLDA algorithm works with data in matrix representation. (Ye, Janardan, & Li, 2004) More specifically, 2DLDA involves the eigen-decomposition of matrices which are much smaller than the matrices in classical LDA. This dramatically reduces the time and space complexities of 2DLDA over LDA.

Reference

- ❖ Van Der Maaten, L., Postma, E., & Van den Herik, J. (2009). Dimensionality reduction: A comparative. J Mach Learn Res, 10, 66-71.
- ❖ Ye, J., Janardan, R., & Li, Q. (2004). Two-dimensional linear discriminant analysis. Paper presented at the Advances in Neural Information Processing Systems, 1569-1576.
- ❖ Yu, H., & Yang, J. (2001). A direct LDA algorithm for high-dimensional data—with application to face recognition. Pattern Recognition, 34(10), 2067-2070.
- ❖ Murphy, K. P. (2012). Machine learning: A probabilistic perspective. Cambridge, MA: MIT Press.
- ❖ Classifier comparison. (n.d.). Retrieved March 30, 2016, from http://scikit-learn.org/0.16/auto_examples/classification/plot_classifier_comparison.html