STA315 Final Report

A DIRECT ESTIMATION APPROACH TO SPARSE LINEAR DISCRIMINANT ANALYSIS

Si Tong Liu, Chi Sheng

September 16, 2024

Contents

1 Introduction

With the development of technology, we face more and more high-dimensional classification problems in both academic research and practical areas. As a result, it became necessary for us to develop applicable new methods, since original methods like LDA are no longer efficient under high-dimensional settings.

Linear discriminant analysis (LDA) is an important conventional model for data classification, which is also one of the most common methods we use in high-dimensional settings. Consider two p-dimensional normal distributions with different means and same covariance matrix. Let Z be a random vector that is drawn from one of these two distributions with equal prior probabilities. The purpose of the classification is to find out which class Z is classified to. When the parameters $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ and covariance matrix are known, this question could be straightforward. In this case, the determination rule is: $\boldsymbol{\psi}_F(Z) = I\{(Z-\boldsymbol{\mu})'\boldsymbol{\Omega}\boldsymbol{\delta} \geq 0\}$, which is also been called as the Fisher's linear discriminant rule. The $\boldsymbol{\mu}$ here is $\boldsymbol{\mu} = (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2, \boldsymbol{\delta} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$, and $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$.

Compared to LDA, LPD depends on ,instead of either of them individually, $\boldsymbol{\delta}$ and $\boldsymbol{\Omega}$ through their product $\boldsymbol{\delta} \times \boldsymbol{\Omega}$. The LPD rule introduces a direct estimation method by estimating $\boldsymbol{\delta} \times \boldsymbol{\Omega}$ through a constrained $\boldsymbol{\ell}_1$ minimization method. Specifically, we propose to estimate

$\hat{\boldsymbol{\beta}} \in \arg\min_{\boldsymbol{\beta} \in R^P} \{|\beta|_1 \text{ subject to } |\hat{\boldsymbol{\Sigma}}_n \boldsymbol{\beta} - (\hat{\boldsymbol{X}} - \hat{\boldsymbol{Y}})|_\infty \leq \boldsymbol{\lambda}_n\}$

where $\boldsymbol{\lambda}_n$ is a tuning parameter and classifies Z to class 1 if and only if $(Z - \hat{\boldsymbol{\mu}})'\hat{\boldsymbol{\beta}} \geq 0$, where $\hat{\boldsymbol{\mu}} = (\bar{\boldsymbol{X}} + \bar{\boldsymbol{Y}})/2$. The estimator $\hat{\boldsymbol{\beta}}$ can be implemented easily using linear programming.

The LPD rule that we are about to discuss is data-driven and easy to implement. It has a significant computational advantage over the existing methods. In the following paper, we will first introduce the existing method LDA, its limitations, and the motivations for using the Linear Programming Discriminant. Moreover, there would be numerical simulations in our paper to show how prominent the LPD works compared to other methods.

2 Pros and Cons of the existing LDA method

As mentioned, the LDA method is usually what we look at when we encounter a classification problem. LDA, linear discriminant analysis, uses a linear combination of features as the classification criterion. It has been proven that LDA performs well and has certain optimality given a fixed dimension and as the sample size approaches infinity.

When we do classification, we have two p-dimensional normal distributions labeled as class 1 and class 2, $N = (\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and $N = (\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$. They share the same covariance matrix $\hat{\boldsymbol{\Sigma}}$. Assume we have a random vector Z, the main idea of a classification problem is finding out to which class we put Z into. From what we learned in class, LDA can provide us with a fast and easily-implemented discriminant function which helps us classify efficiently. Among all the algorithms we discussed, LDA is our top choice in most cases.

With that being said, LDA runs into fatal challenges when handling high-dimensional data sets. As we discussed in lecture, assume the dimension of the data matrix is p-by-n. In the high dimensional data set, it is highly possible that the data matrix we obtain is p>>n, meaning one dimension of the matrix is a lot greater than the other dimension. In situations like this, LDA becomes inefficient, since it relies heavily on the sample covariance matrix $\hat{\mathbf{\Sigma}}$ to be invertible. However, this assumption could not be fulfilled any more because in high-dimensional settings, $\hat{\mathbf{\Sigma}}$ might not be full rank.

Thus, even though LDA performs well in most cases, we might not be able to use it in high-dimensional data cases.

3 Motivation of the LPD method

The main reason for developing the LPD method is straightforward: we need better substitution for the LDA rule. As technology advances dramatically, we obtain high-dimensional data routinely for a wide range of applications. Classification for these data is crucial and essential in turning data sets into productive statistical results. Surely, difficulties come along the path.

The original Fisher's linear discriminant rule states:

$\boldsymbol{\psi}_F(Z) = I\{(Z-\boldsymbol{\mu})'\mathbf{\Omega}\boldsymbol{\delta} \geq 0\}$

μ here is $\boldsymbol{\mu} = (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2, \boldsymbol{\delta} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$, and $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$. Z would be classified into class 1 if and only if $\boldsymbol{\psi}_F(Z)$=1. The reason LDA performs poorly is that in high-dimensional settings, we have regularity conditions on not only $\mathbf{\Omega}$ (or $\mathbf{\Sigma}$) but also on δ as well. This was done in order to ensure that they could be estimated consistently. The most common assumption is that $\mathbf{\Omega}$ (or $\mathbf{\Sigma}$) and $\boldsymbol{\delta}$ are both sparse. With this being said, we still need to estimate them separately and then plug them back into Fisher's LDA rule.

However, $\mathbf{\Sigma}$ could be singular and has no well-defined inverse function. And unfortunately, any remedy towards this issue looks pale.

Bickel and Levina (2004) demonstrated that in cases where $p/(\boldsymbol{n}_1+\boldsymbol{n}_2) \to \infty$ the LDA performs no better than guessing blindly. They tried to use instead the generalized inverse of the sample covariance matrix, but the estimate is highly biased and unstable with poor performance. One solution is to simply ignore the dependence among variables and replace $\mathbf{\Sigma}$ with the diagonal of the sample covariance matrix, leading to the naive Bayes rule, also called the sample covariance matrix.

Fan and Fan (2008) proposed the features annealed independence rule applying the naive independence rule to a set of selected important features of $\boldsymbol{\delta}$ that are chosen by the threshold, which ignored the correlations between the variables and could be inefficient as well. All and all, we are in desperate need of a new method that can overcome these difficulties.

4 Explanation of the LPD method

In the present paper, we will show how the LPD rule works under high-dimension. Under such conditions, the inverse of covariance matrix is nearly impossible to be computed by humans without using statistic tools like R-language, and even with tools like R, it still takes a long time for machine to run the code and give us the simulation results.

To start with, we do simple random samplings on each class we define. Mostly, we assume these two samples have the same sizes. Let the size of data taken from class 1 be n_1, and the size of data taken from class 2 be n_2.

Let $(\boldsymbol{X}_k : 1 \leq k \leq \boldsymbol{n}_1)$ $(\boldsymbol{Y}_k : 1 \leq \boldsymbol{n}_2)$ be the samples we chose from both classes. They are identically and independently distributed from class1 $N = (\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and $N = (\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$. Set $\bar{\boldsymbol{X}} = \frac{1}{n_1}\sum_{i=1}^{n_1} x; \bar{\boldsymbol{Y}} = \frac{1}{n_2}\sum_{i=1}^{n_2} y; \hat{\boldsymbol{\delta}} = (\bar{\boldsymbol{X}} - \bar{\boldsymbol{Y}}), \hat{\boldsymbol{\mu}} = (\bar{\boldsymbol{X}} + \bar{\boldsymbol{Y}})/2$ Denote the sample matrix by $\hat{\boldsymbol{\Sigma}_x} = \frac{1}{n_1}\sum_{i=1}^{n_1}(X_i - \bar{\boldsymbol{X}})(X_i - \bar{\bar{\boldsymbol{X}}})'\hat{\boldsymbol{\Sigma}_y} = \frac{1}{n_2}\sum_{i=1}^{n_2}(Y_i - \bar{\boldsymbol{Y}})(Y_i - \bar{\bar{\boldsymbol{Y}}})'$ And set $\hat{\boldsymbol{\Sigma}_n} = \frac{1}{n}(n_1\hat{\boldsymbol{\Sigma}_x} + n_2\hat{\boldsymbol{\Sigma}_y})$, where $n = (n_1 + n_2)$.

It is clear that the Fisher's rule depends on Ω and Σ only through their product. Thus, we consider a constrained minimization method to directly estimate the product $\Omega\Sigma$ by exploiting the sparsity of $\Omega\Sigma$. We should note that here the sparsity of their product is weaker and more flexible than both of them separately. (Remark 1)

Specifically, we propose to estimate $\beta := \Omega\Sigma$ by the solution to the following optimization problem:

$\hat{\boldsymbol{\beta}} \in \arg\min_{\boldsymbol{\beta} \in R^P}\{|\beta|_1 \text{ subject to } |\hat{\boldsymbol{\Sigma}}_n\boldsymbol{\beta} - (\hat{\boldsymbol{X}} - \hat{\boldsymbol{Y}})|_\infty \leq \boldsymbol{\lambda}_n\}$

where λ_n is a tuning parameter, $\lambda_n = C\sqrt{\triangle_p logp/n}$, which delta $\triangle_p = \delta'\Omega\delta$ After we obtain $\hat{\boldsymbol{\beta}}$, we could follow the Fisher's rule to determine which class vector Z belongs to by using the β we got. The vector Z will be classify into class 1 if and only if $(Z - \hat{\boldsymbol{\mu}})'\hat{\boldsymbol{\beta}} \geq 0$, and into class2 otherwise.

Remark 1

Let δ be k_1-sparse and Ω be k_2-sparse, then $\Omega\delta$ is at most k_1k_2-sparse. Furthermore, the sparsity of $\Omega\delta$ does not require Ω being sparse. Suppose δ is k_1-sparse and without loss of generality assume the nonzeros are among the first k_1 coordinates,

$$\boldsymbol{\delta} = \begin{pmatrix} \delta_1 \\ 0 \end{pmatrix}$$

where $delta_1$ is a k_1-dimensional vector. Write Ω as

$$\boldsymbol{\Omega} = \begin{pmatrix} \Omega_{11} & \Omega_{21}' \\ \Omega_{21} & \Omega_{22} \end{pmatrix}$$

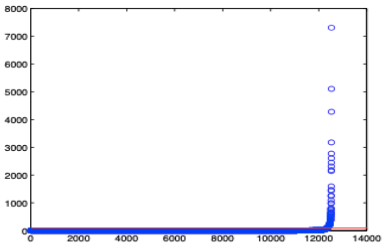
where Ω_{11} is $k_1 \times k_1$, Ω_{21} is $(p - k_1) \times k_1$ and Ω_{22} is $(p - k_1) \times (p - k_1)$. Then the sparsity of $\Omega\delta$ does not depend on Ω_{22} since $\Omega\delta = \begin{pmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{pmatrix}$ is sparse if Ω_{21} is sparse.

5 Real Case Simulation

Now, let's talk about two real cases simulations discussed in the paper. The first case is regarding the classification of different genes in a lung cancer data set(Gordon, et al. (2002)), and the second case is regarding the classification of different genes in a leukemia data set (Golub, et al. (1999)). Both of these sets have very high dimensions and are practical applications of our LPD method studies.

Lung cancer Data set: The lung cancer data set consists of 181 tissue samples and each sample is described by 12533 genes. Among the 181 tissue samples, there are two classes of tissue samples including 31 malignant pleural mesothelioma (MPM) and 150 adenocarcinoma (ADCA). Distinguishing MPM from ADCA is our main task in this case.

Since the sample variances of the genes range over a wide interval, we rescale it by 104 for numerical stability. The plot on the right is the sorted sample covariances.



(a) Lung cancer data

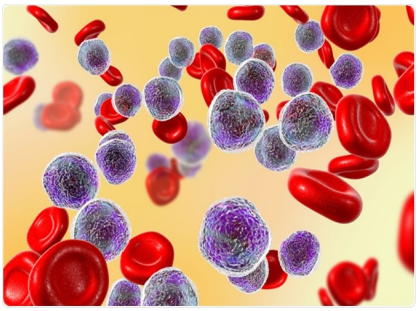
After using 32 training samples with 16 from MPM and 16 from ADCA, 149 testing sample with 15 from MPM and 134 from ADCA, we obtain the above result through multiple methods. LPD rule classifies all 149 of the testing samples correctly, which is more than satisfactory. On the other hand, naive-Bayes rule misclassified 12 of 149 testing samples, GLDA misclassified 7 of 149 testing samples, FAIR misclassified 7 of 149 testing samples as well and NSC misclassified 11 of 149 testing samples. Below is the table of classification error of Lung cancer data by various methods.

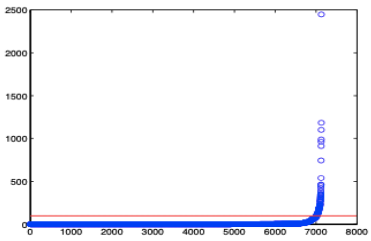
Leukemia Data set: The leukemia data set consists of 72 tissue samples, which were all from acute leukemia patients, either acute lymphoblastic leukemia (ALL) or acute myelogenous leukemia (AML). Distinguishing ALL from AML is our main task in this case. Since the sample variances of the genes range over a wide interval as in the lung cancer case, we rescale it by 105 for numerical stability. The plot on the right is the sorted sample covariances.

After using 38 training samples with 27 from ALL and 11 from AML, 34 testing sample with 20 from ALL and 14 from AML, we obtain the above result through multiple methods. LPD rule only misclassified 1 out of the entire

Table 5: Classification error of Lung cancer data by various methods.

	LPAD	FAIR	NSC	Naive-LDA	GLDA
Training error	0/32	0/32	0/32	0/32	0/32
Testing error	0/149	7/149	11/149	12/149	7/149





(b) Leukemia data

testing samples and made no training error. On the other hand, naive-Bayes rule misclassified 7 of 34 testing samples with 1 training error, GLDA misclassified 3 of 34 testing samples with 1 training error, FAIR misclassified 1 of 34 testing samples and 1 training sample as well and NSC misclassified 3 of 34 testing samples and also 1 training sample.

Table 6: Classification error of Leukemia data by various methods.

	LPAD	FAIR	NSC	Naive-LDA	GLDA
Training error	0/38	1/38	1/38	1/38	1/38
Testing error	1/34	1/34	3/34	7/34	3/34

Through these two real data analysis, it is fair to conclude that LPD has a better performance compared to other methods in a reality settings.

6 Numerical Simulations

We now present the simulation results and comparisons of the numerical performance between the LPD classifier and the LDA classifier.

- **Our own numerical simulation procedure**

  Firstly, the setup of the simulation study is as follows. We fix the sample sizes $n_1 = n_2 = 200$ and set $\boldsymbol{\mu_1}$ and $\boldsymbol{\mu_2}$=(1,..., 1, 0,..., 0), where the number of 1's is $s_0 = 10$, therefore, the number of 0 is equal to $n_2 - s_0$, and in this case, equals to 190.

  The two models are considered as following:

  Model 1: $\boldsymbol{\Omega} = \boldsymbol{\Sigma} = (\sigma_{ij})_{p\times p}{}^{1}$ with $\sigma_{ii} = 1$ for $1 \leq i \leq p$, which is the diagonal of the covariance matrix, and $(\sigma_{ij}) = \rho$ with $\rho = 0.5$ for i $\neq j$.
  Model 2 :$\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$, where $\boldsymbol{\Sigma} = (\sigma_{ij})_{p\times p}$ with $\sigma_{ij} = 0.8^{|i-j|}$ for $1 \leq i, j \leq p$.

  $\boldsymbol{\Omega}$ in Model 1 is approximately sparse matrix. It is diagonally dominant with the off-diagonal entries of order p-1. In Model 2, $\boldsymbol{\Sigma}$ can be well approximated by a sparse matrix and the inverse $\boldsymbol{\Sigma}$ is a 3-sparse matrix.

  In the simulation, we generate $n_1 = n_2 = 200$ training and test samples of the same size according to both models with the multivariate normal distribution, setting class 1 and class 2 to $N = (\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and $N = (\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$. To start with, we randomly draw a vector Z from class 1 or class 2, and record which class it is really chosen from in the "true" set. Secondly, we use both methods(LPD and LDA) to classify which class the Z belongs to. Recording both methods' results, and then, comparing the results with the true result, and getting the classifier by dividing to the size of the classification.

- **The simulation result**

  Our numerical result is as following

MODEL1		
p	LDA correctness	LPD correctness
200	0.62	0.63
400	0.63	0.63
600	0.63	0.63
800	0.63	0.63

MODEL2		
p	LDA correctness	LPD correctness
200	0.63	0.62
400	0.63	0.62
600	0.63	0.62
800	0.63	0.62

  The result suggests that in Model 1, when p = 200, LPD even gives a better result compared to LDA, and in other situations, the LPD has the same result as LDA. In Model 3, the difference between 2 methods is not significant.

- **Original paper's author's conclusion**

  The original paper's author used similar simulations with several other methods. After obtaining the results, the LPD method could offer a similar result to LDA, and much better result than other methods in the simulation. Thus, the author suggested that when using LPD methods in low dimension, the LPD could give a result with similar correctness compared to LDA, which is much better than other methods.

- **Our simulation conclusion compare to original paper's author's conclusion**

  Compared to the author's simulation result, our simulation result gives the same conclusion: the LPD method gets a result which has a similar correctness with the LDA method. And since the machine simulation time will increase significantly when p gets larger for LDA, LPD could be the method to replace the LDA in order to save the cost.

7 Conclusion

In conclusion, as our society and technology progress, we obtain huge amounts of high-dimensional data routinely through a wide range of applications. Classification for these data has become a main task in turning data sets into productive statistical results, and surely, difficulties come along the path. When data size become larger and larger, the LDA could be less efficient, and even impossible to perform directly. This is due to in high-dimension settings, matrix $\mathbf{\Sigma}$ is not consistently invertible. Using the LPD can avoid the problem of having to obtain an exact $\mathbf{\Omega}$.

In the future, when people are trying to do a classification for a group of data with large size, such as population surveys to a countries, double-blind vaccine tests and gene sequence determinations, we will suggest using the LPD method instead of the LDA method. This will not only save the costs of resources and but also a lot of time for doing the classification.

8 Reference

1. Bickel, P., and Levina, L. (2004), "Some Theory for Fisher's Linear Discriminant Function, 'Naive Bayes,' and Some Alternatives When There Are Many More Variables Than Observations," Bernoulli, 10, 989–1010. [1566]
2. Fan, J., and Fan, Y. (2008), "High Dimensional Classification Using Features Annealed Independence Rules," The Annals of Statistics, 36, 2605–2637. [1566,1571,1573,1574]
3. Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D. and Lander, E. S. (1999), Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science, 286, 531-537.
4. Gordon, G.J., Jensen, R.V., Hsiao, L.L., Gullans, S.R., Blumenstock, J.E., Ra- maswamy, S., Richards, W.G., Sugarbaker, D.J. and Bueno, R. (2002), Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma, Cancer Research, 62, 4963-4967.
5. Tony Cai Weidong Liu (2011) A Direct Estimation Approach to Sparse Linear Discriminant Analysis, Journal of the American Statistical Association, 106:496, 1566-1577