A Direct Approach to Sparse Discriminant Analysis in Ultra-high Dimensions

STA315 Final Project

Petar Solaja and Benny Huynh

April 2023
Professor Dehan Kong
University of Toronto Mississauga

1 Summary

Sparse discriminant methods are useful for selecting features and classifying data with high dimensions. However, they have limitations such as ignoring correlations among features which can lead to poor feature selection and classification. The authors of *A Direct Approach to Sparse Discriminant Analysis in Ultra-high Dimensions* propose a new approach for sparse discriminant analysis called lasso penalized least squares, which is motivated by the least squares formula of linear discriminant analysis. It is shown through theoretical analysis and experiments to consistently identify the subset of discriminative features and estimate the Bayes classification direction, even when the dimension of the data is high and when there are correlations among features. The proposed method is compared with other sparse discriminant methods using simulated examples in a consistent simulation setting.

2 Methods

2.1 Linear Discriminant Analysis

The paper starts off by introducing linear discriminant analysis (LDA). It is one of the oldest and most widely used techniques for classification, and is still very popular today due to its simplicity and effectiveness. LDA is a supervised learning method, meaning that it requires labeled data to train a model. The goal of LDA is to find a linear combination of variables that can best separate two or more classes of observations. Specifically, it seeks to find a set of coefficients that maximizes the separation between the means of the different classes, while minimizing the variance within each class.

LDA assumes the data is normally distributed and that the covariance matrices for each class are equal. Next the class priors are calculated, which are the probabilities of each class occurring in the training set. This is calculated by dividing the number of samples in each class by the total number of samples. Next the discriminant functions for each class are calculated using Bayes' Rule. The estimated means and covariance matrices for each class and the class priors are used to calculate the discriminant functions. Finally, the data point is classified to class 2 if and only if the discriminant function for class 2 is larger than the discriminant function for class 1. Shown below are the general steps of LDA.

(i) Assumes

$$x \mid G = g \sim N(\mu_g, \Sigma)$$

(ii) Calculate priors

$$\pi_1 = n_1/n$$

$$\pi_2 = n_2/n$$

(iii) Calculate discriminant functions
and by Bayes' Rule classify data point to class 2 if and only if:

$$\hat{\delta_2} > \hat{\delta_1} \Rightarrow \hat{\delta_2} - \hat{\delta_1} > 0$$

$$\rightarrow \{x - (\mu_1 - \mu_2)/2\}^T \Sigma^{-1}(\mu_2 - \mu_1) + log(\pi_2/\pi_1) > 0$$

where

$$\mu_1 = \hat{\mu_1} = \text{class 1 sample mean vector}$$

$$\mu_2 = \hat{\mu_2} = \text{class 2 sample mean vector}$$

$$\Sigma = \hat{\Sigma} = \frac{n_1 - 1}{n_1 + n_2 - 2} S_1 + \frac{n_2 - 1}{n_1 + n_2 - 2} S_2 = \text{pooled sample covariance estimate}$$

LDA can suffer from several issues when dealing with high-dimensional data. One issue is that when the number of predictor variables is much larger than the number of observations, the sample covariance matrix can become unstable. This can lead to poor LDA performance and inaccurate classification results. Another issue with LDA in high-dimensional data is the "small sample size" problem. When the number of observations is small relative to the number of predictor variables, the estimation of covariance matrices becomes unreliable. This problem can result in overfitting, where the model fits the noise in the data. However, even if we use a "better" estimator for the covariance matrix, there is no guarantee that the classifier will improve. This is because the performance of a classifier depends not only on the estimation of the covariance matrix, but also on the distribution of the data.

2.2 Nearest Shrunken Centroids

To address the issues with LDA, Tibshirani et al. (2002) proposed the nearest shrunken centroids (NSC) classifier. It is a modification of LDA but for high-dimensional data. The NSC classifier works by computing d_{jg}, the moderated statistic for feature j in class g which measures the difference in means between the two classes for feature j, while considering the variability of the feature within and between the classes. This is found by computing a centroid for each class using the mean of the training samples belonging to that class. This is denoted by $\mu_g j$, for feature j in class g, a.k.a the within-class sample mean. Then the sample variance of feature j is calculated and s_0 which is a small positive constant is added for robustness, but it equals to zero in the paper for simplicity. Next the shrunken centroids mean for feature j in class g is calculated using the marginal sample mean of feature j, defined as the weighted average of the within-class sample mean, denoted $\bar{x}_j$. λ is a pre-chosen positive constant that determines the amount of shrinkage applied to the sample mean. This shrinks each feature towards its class centroid. The shrunken centroids mean is used to compute the shrunken feature values and select the subset of features for classification. The amount of shrinkage applied to the sample mean depends on the magnitude of the moderated statistic d_{jg}, with larger values of d_{jg} resulting in more shrinkage. This helps to reduce the influence of noisy and irrelevant features, and improve the performance of the classifier. Finally, a data point x is classified to class 2 if and only if the last formula below is positive. Intuitively, the last formula computes a weighted sum of the differences between the values of the predictor variables for the observation and the shrunken centroid means for each class. The constant log term reflects the prior probability of each class. Shown below are the general steps for NSC.

(i) For each variable compute:

$$d_{jg} = (1/n_g + 1/n)^{-1/2} (\hat{\mu_{gj}} - \bar{x_j})(s_j + s_0)^{-1} \quad (g = 1, 2)$$

where

$$\hat{\mu_{gj}} = \text{within-class sample mean}$$

$$(s_j)^2 = \text{sample estimate of } \Sigma_{jj}$$

$$s_0 = \text{small positive constant (=0 in the paper)}$$

(ii) Then, the shrunken centroids is given by:

$$\hat{\mu}'_{gj} = \bar{x_j} + (n_g^{-1} + n^{-1})^{1/2} s_j d_{jg}^{\lambda} \quad (g = 1, 2)$$

where

$$\bar{x}_j = (n_1\hat{\mu_1} + n_2\hat{\mu_2})/n = \text{marginal sample mean of } x_j$$

$$\lambda = \text{pre-determined positive constant}$$

$$d_{jg}^{\lambda} = \text{sign}(d_{jg}) \cdot \max(0, |d_{jg}| - \lambda)$$

(iii) Finally, a data point x is classified to class 2 if and only if:

$$\Sigma_{j=1}^{p}\{x_j - (\hat{\mu}'_{2j} + \hat{\mu}'_{1j})/2\} s_j^{-2} (\hat{\mu}'_{2j} - \hat{\mu}'_{1j}) + log(n_2/n_1) > 0$$

The NSC addresses two issues that the LDA does not, which are the covariance matrix and selection of relevant features. Firstly, NSC uses only the diagonal of the sample covariance matrix to estimate the covariance matrix. This is done by using a shrinkage factor λ, which shrinks the sample variances towards a common value. If λ is set to zero, the resulting NSC classifier reduces to diagonal LDA. This modification works better in higher dimensions as it reduces the risk of overfitting. Secondly, NSC uses shrunken centroids means instead of standard means to estimate μ_1 and μ_2. The shrunken centroid means are obtained by shrinking the sample means towards a common centroid, resulting in a more stable estimate of the mean vectors. For large enough λ, the shrunken centroid means become equal to the sample means, resulting in no contribution to the classifier for that feature. This effectively performs feature selection and reduces the problem of high dimensions.

Experiments have shown that NSC is very competitive for high-dimensional classification tasks. However, there is little evidence that NSC can effectively discover the discriminant set, which is the set of features that are most relevant for classification. This is because NSC is a linear classifier and may not capture the complex nonlinear relationships that can exist between the features and the classes.

2.3 The Discriminative Set and the Signal Set

One idea that the paper talks about that is important to the method it proposes is the disriminative set and the signal set. In classification, the discriminative set refers to the set of variables that are most useful in a classification problem. That is, the variables that are the most relevant in determining the class of a given data point. In the paper, the discriminative set contains the variables that are in the same direction as the Bayes classification direction. Therefore, by definition, the discriminative set is given by A, where $A = \{j : \{\Sigma^{-1}(\mu_2 - \mu_1)\}_j \neq 0\}$. The variables in A are called discriminative variables. On the other hand, the signal set is the set containing the variables whose within-class means are different. So, the signal set is given by $\tilde{A} = \{j : \mu_{1j} \neq \mu_{2j}\}$ and the variables in $\tilde{A}$ are called signals. In an ideal setting where Σ is diagonal, $\tilde{A} = A$. However in the more common setting where there is a general covariance matrix, the discriminative set and the signal set can vary drastically.

2.4 Lasso Penalized Least Squares

The authors propose using a penalized sparse least squares approach to derive sparse discriminant methods when dealing with high-dimensional data, where the number of features is greater than the number of observations (p>n). In the standard p<n, there is a connection between least squares and LDA, were the least squares slope coefficient is a scalar multiple of the LDA direction. However, when p>n, the sample covariance matrix $\hat{\Sigma}$ is no longer invertible, and as a result, the LDA direction is undefined. Traditional methods such as LDA do not work well, thus the authors propose using a sparse discriminant method that employs a sparsity-inducing penalty, such as the lasso, to produce a classification rule in high-dimensional settings. The approach involves minimizing a penalized least squares problem. This approach allows for the identification of important features in the classification process and can lead to better performance compared to traditional methods such as LDA. Shown below are the general steps for the penalized least squares approach.

Least Squares:

$$(\hat{\beta}^{ols}, \hat{\beta}_0^{ols}) = \underset{\beta,\beta_0}{\text{argmin}} \Sigma_{i=1}^{n} (y_i - \beta_0 - x_i^T \beta)^2$$

Then,

$$\hat{\beta}^{ols} = c\hat{\Sigma}^{-1}(\hat{\mu_2} - \hat{\mu_1}), \quad (c > 0)$$

A penalized least squares formula is considered in order to produce a classification direction:

$$(\hat{\beta}^{ols}, \hat{\beta}_0^{ols}) = \underset{\beta,\beta_0}{\text{argmin}} \{n^{-1} \Sigma_{i=1}^{n} (y_i - \beta_0 - x_i^T \beta)^2 + \Sigma_{j=1}^{p} P_\lambda(|\beta_j|)$$

Here, $P_\lambda(\cdot)$ can be any sparsity-inducing penalty. In the paper, the lasso penalty is used as it is the most popular in literature. Would work with others such as elastic net, adaptive lasso, etc.

Then the classification rule assigns x to class 2 if:

$$x^T \hat{\beta}^\lambda + \hat{\beta}_0 > 0$$

Instead of using $\hat{\beta}_0^\lambda$ as the intercept in the classifier, an optimal intercept is estimated by:

$$\hat{\beta}_0 = \hat{\beta}_0^{opt} = -(\hat{\mu_1} + \hat{\mu_2})^T \hat{\beta}^\lambda / 2 + (\hat{\beta}^\lambda)^T \hat{\Sigma} \hat{\beta}^\lambda \{(\hat{\mu_2} - \hat{\mu_1})^T \hat{\beta}^\lambda\}^{-1} log(n_2/n_1)$$

After calculating $\hat{\beta}_0$, the classifier now assigns x to class 2 if:

$$\{x - (\hat{\mu_1} + \hat{\mu_2})/2\}^T \hat{\beta}^\lambda + (\hat{\beta}^\lambda)^T \hat{\Sigma} \hat{\beta}^\lambda \{(\hat{\mu_2} - \hat{\mu_1})^T \hat{\beta}^\lambda\}^{-1} log(n_2/n_1) > 0$$

The sparsity assumption on $\hat{\beta}^\lambda$ ensures that $(\hat{\beta}^\lambda)^T \hat{\Sigma} \hat{\beta}^\lambda$ is a good estimator for $(\hat{\beta}^\lambda)^T \Sigma \hat{\beta}^\lambda$ when p>>n.

3 Numerical Results

3.1 Simulation Procedure

Similar to the paper, we use simulated data to showcase the respectable performance of the paper's lassoed discriminant analysis. While the paper compares the classifier to five other competitors our simulations only included three other classifiers for comparison. These include the nearest shrunken centroids classifier (Tibshirani, 2002), the l_1-penalized linear discriminant (Witten & Tibshirani, 2011) and the t-test classifier. For the t-test classifier, we first performed Bonferroni adjusted t-tests with $\alpha = 0.05$ and then used only the variables that passed the t-test in the linear discriminant analysis. The nearest shrunken centroids classifier is implemented using the pamr R package while the l_1-penalized linear discriminant is implemented using the R package penalizedLDA.

To generate the simulation data, we first generated n class labels such that $\pi_1 = \pi_2 = 0.5$, so there is an equal probability for a data point to be labelled as class 1 or class 2. Conditioning on the class labels $g\,(g = 1, 2)$, we generated the p variables for the predictor x from a multivariate normal distribution with mean vector μ_g and covariance Σ. We set $\mu_1 = 0$ and $\mu_2 = \Sigma\beta^{Bayes}$. While the paper considers six different models with varying values for n, p, Σ and β^{Bayes} our review only simulations the paper's first simulation setting. In this model (Model 1 in the paper), the values of the four parameters are as follows: $n = 100$, $p = 400$, $\Sigma_{ij} = 0.5^{|i-j|}$ and $\beta^{Bayes} = 0.556(3, 1.5, 0, 0, 2, 0_{p-5})^T$. One thing that was not mentioned in the paper's simulation procedure was the way in which the simulated data was split into training and testing data. Therefore, we split the data in the same manner in which the data in the *Real Data* section was split, which was a random 2:1 training to testing data ratio. That means that in this simulation setting where $n = 100$, the data is trained on 67 data points and tested on 33 with 400 predictor variables being taken into account. One other difference between our simulation and the paper's is that ours only runs 100 replications as opposed to the paper's 2000 replications. So for each replication, the simulation data is generated, split, and then the four different classifiers are trained and then tested with their error percentages being calculated. Then the median error percentage of the 100 replications for each classifier is calculated and recorded.

3.2 Simulation Results

Table 1. *Simulation results. The methods from left to right: lassoed discriminant analysis (paper's method), Nearest Shrunken Centroids (Tibshirani), l_1-penalized discriminant (Witten) and the t-test classifier. Median error percentages with their standard error in parentheses are included.*

Model 1	Paper's method	Tibshirani	Witten	t-test classifier
Error %	12.12	12.12	12.12	9.09
SE	(6.13)	(5.85)	(6.12)	(5.83)

Looking at our results, we see that in the the setting where $n = 100$ and $p = 400$ the paper's method performs at the same level as NSC and the l_1-penalized discriminant classifier while the t-test classifier performs the best. In comparison with the paper's results in the same simulation setting, we notice that the first three methods perform slightly poorer while the t-test classifier performs slightly better. This can be attributed to the paper's much higher number of replicated simulations. As our results are based on only 100 replications versus the paper's 2000, it is expected that the paper's results show the methods to perform at a higher level. This is also the reason for the higher standard errors for the medians. Our standards errors are calculated with $n = 100$ (the number of replications), while the paper's standard errors were calculated with $n = 2000$. It is well know that standard error decreases as sample size increases. One other important aspect of the simulation results that should be be noted is the identical performances of the first three classifiers. This can be attributed to the way we split our data into training and testing sets. Since our data was split with a 2:1 ratio, our classifiers were trained using 67 data points and tested on only 33 data points, making it more likely to obtain identical error percentages. On the other hand, a technique such as leave-one-out cross validation trains the data on n-1 points and tests on 1 point, where the number of folds $k = n$. This allows for a better trained model and, as a result, a better performing classifier.

3.3 Discussion

From our results, we see that sparse discriminant analysis is computationally effective for high-dimensional classification when based on independence rules. However, independence rules may lead to unfavourable feature selection and, as a result, poor classification performance. This can be attributed to the difference between discriminative and signal variables. We see that when doing feature selection in classification, one should aim to obtain the discriminative set, which aids in accomplishing the goal of hypothesis testing on a large scale. Another thing that is important to note is that the paper and our review only focuses on lassoed discriminant analysis as the central goal is to demonstrate the effectiveness of the penalized least squares method in sparse discriminant analysis. The type of penalty function that is used is not crucial in the paper, and subsequently, in our review. For different data sets, certain penalty functions may be more appropriate than others. Further testing involving the use of different penalty terms could help uncover the most effective one to use in specific situations.

4 References

TIBSHIRANI, R. J., HASTIE, T. J., NARASIMHAN, B. & CHU, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proc. Nat. Acad. Sci. 99, 6567–72

WITTEN, D. M. & TIBSHIRANI, R. J. (2011). Penalized classification using Fisher's linear discriminant. J. R. Statist. Soc. B 73, 753–72.