DSA5205 Project 1: High Dimensional Problems

Hangxiao Yang[1]

[1]Faculty of Science, National University of Singapore, 597159, Singapore.

Contributing authors: E1351047@u.nus.edu;

Abstract

The paper addresses challenges in high-dimensional data analysis, particularly when the number of features significantly exceeds the number of samples ($\boldsymbol{n_{features} \gg n_{samples}}$). The study explores various machine learning methods, including **L**inear **D**iscriminant **A**nalysis (LDA), **P**rincipal **C**omponent **A**nalysis (PCA), $\boldsymbol{L_1}$ regularization (Lasso), and $\boldsymbol{L_2}$ regularization (Ridge), to identify their effectiveness in such scenarios. The experiments on the public dataset GEO demonstrate that LDA and Lasso methods have significant advantages in high-dimensional data tasks. The related code is open-sourced at: https://github.com/yhx30/Project-5205.

Keywords: LDA, PCA, Lasso, $\boldsymbol{L_2}$ Regression, SVM

1 Introduction

In general, we assume that the number of data samples in a prediction task is sufficient, i.e., the number of features p is much smaller than the number of observations N. However, in specific cases, we encounter situations where only a small dataset is available, and the data is high-dimensional, i.e., $p \gg N$, such as in genomics and other areas of computational biology. In this case, the dataset exhibits high variance, and models designed for the $N > p$ scenario tend to overfit. A simple approach is to use highly regularized methods. Additionally, we need to develop entirely new methods to analyze high-dimensional data.

2 Methodologies

2.1 LDA and NSC

Diagonal **L**inear **D**iscriminant **A**nalysis (LDA) and **N**earest **S**hrunken **C**entroids (NSC) are regularization-based methods particularly effective in high-dimensional scenarios ($p \gg N$). Diagonal LDA simplifies the classification problem by assuming that features are independent within each class. Although this independence assumption may not hold in most real-world datasets, it becomes practical when data is insufficient to estimate feature dependencies accurately. By reducing the covariance matrix to a diagonal form, the number of model parameters decreases significantly, enabling efficient and interpretable classification.

- **Diagonal Linear Discriminant Analysis (LDA)**
  In diagonal LDA, the discriminant function for class k is expressed as:

$$\delta_k(x^*) = -\sum_{j=1}^{p} \frac{(x_j^* - \bar{x}_{kj})^2}{s_j^2} + 2\log \pi_k, \tag{1}$$

  where x_j^* is the $j-th$ feature of the test observation, $\bar{x}_{kj}$ is the mean of the $j-th$ feature in class k, s_j is the pooled within-class standard deviation, and π_k is the prior probability of class k. This formulation makes diagonal LDA equivalent to a nearest centroid classifier with standardization. However, a limitation of this approach is its reliance on all features, which can hinder interpretability.
- **Nearest Shrunken Centroids (NSC)**
  NSC extends diagonal LDA by incorporating regularization that selects relevant features. It shrinks the class-specific means $\bar{x}_{kj}$ toward the overall mean $\bar{x}_j$ using soft-thresholding:

$$d'_{kj} = \text{sign}(d_{kj})(|d_{kj}| - \Delta)_+ \tag{2}$$

  where $d_{kj} = \frac{\bar{x}_{kj} - \bar{x}_j}{m_k(s_j + s_0)}$, m_k is a scaling factor, s_0 is a small positive constant, and Δ is the threshold parameter determined through cross-validation. Features with $d'_{kj} = 0$ are effectively excluded from the model, enhancing both interpretability and robustness. The shrunken centroids are then used in place of the original centroids for classification, yielding a refined and interpretable model suitable for high-dimensional data.

2.2 Linear Classifiers with L_2 Regularization

Regularized **D**iscriminant **A**nalysis (RDA), regularized multinomial logistic regression, and **S**upport **V**ector **M**achines (SVM) are advanced methods designed to leverage multivariate information in high-dimensional data. These methods can incorporate L_1 or L_2 regularization to control model complexity and improve generalization.

- **Regularized Discriminant Analysis (RDA) L**inear **D**iscriminant **A**nalysis (LDA) often requires the inversion of a $p \times p$ within-class covariance matrix. In scenarios where $p \gg N$, this matrix becomes massive and typically has a rank of at most $N < p$, rendering it singular. RDA mitigates this issue by regularizing the within-class covariance matrix. The adjusted covariance estimate is expressed as:

  $$\hat{\Sigma}(\gamma) = \gamma\hat{\Sigma} + (1-\gamma)\text{diag}(\hat{\Sigma}), \quad \gamma \in [0, 1], \tag{3}$$

  where γ determines the degree of regularization, balancing between the original covariance matrix and its diagonal approximation.
- **Logistic Regression with Quadratic Regularization**
  Logistic regression with quadratic regularization is another essential method for classification, especially for multiclass problems. It uses a symmetric multinomial logistic model:

  $$\Pr(G = k|X = x) = \frac{\exp(\beta_{k0} + x^T\beta_k)}{\sum_{k'=1}^{K}\exp(\beta_{k'0} + x^T\beta_{k'})}, \tag{4}$$

  and applies L_2 regularization by maximizing a penalized log-likelihood:

  $$\max_{\{\beta_{0k},\beta_k\}_{k=1}^{K}} \left[\sum_{i=1}^{N}\log\Pr(g_i|x_i) - \frac{\lambda}{2}\sum_{k=1}^{K}||\beta_k||_2^2\right]. \tag{5}$$

  This approach ensures that parameter estimation remains stable even in high-dimensional scenarios.
- **Support Vector Classifiers (SVC)** Support vector classifiers (SVCs) are particularly effective when $p > N$, as classes are generally separable by a hyperplane unless identical feature vectors exist in different classes. Without regularization, SVCs identify the hyperplane with the largest margin, maximizing the gap between classes in the training data:

  $$\min_{\beta,\beta_0} \frac{1}{2}||\beta||_2^2 \quad \text{subject to } y_i(\beta^T x_i + \beta_0) \geq 1, \forall i. \tag{6}$$

  Interestingly, even in high-dimensional cases ($p \gg N$), unregularized SVCs often perform comparably to their regularized counterparts. For multiclass problems ($K > 2$), several generalizations of the two-class SVC have been developed.
- **Feature Selection** Feature selection is not performed automatically by LDA, logistic regression, or SVC, as these methods rely on quadratic regularization. To address

this, ad-hoc methods like recursive feature elimination have been proposed. This backward stepwise approach removes features with the smallest coefficients iteratively and refits the classifier. In nonlinear settings, these methods can be modified to incorporate kernels. Despite the high dimensionality, radial kernels often yield robust results, as they reduce the influence of distant points in the feature space, enhancing resistance to outliers.

- **Computational Efficiency via Singular Value Decomposition (SVD)** When $p \gg N$, computational efficiency can be improved using Singular Value Decomposition (SVD). Instead of working in the p-dimensional space, computations are performed in an N-dimensional space, as N points in a p-dimensional space lie in an $(N-1)$-dimensional affine subspace. SVD allows decomposition of the data matrix X as:

$$X = UDV^T = RV^T, \tag{7}$$

where $R = UD$ is an $N \times N$ matrix. For example, ridge regression can be reformulated using R instead of X:

$$\beta = (X^T X + \lambda I)^{-1} X^T y = V(R^T R + \lambda I)^{-1} R^T y. \tag{8}$$

This transformation reduces computational complexity and makes the analysis of high-dimensional models more tractable. Geometrically, this corresponds to rotating features into a coordinate system where only the first N coordinates are non-zero, aligning with the invariance properties of quadratic penalties.

2.3 Linear Classifiers with L$_1$*Regularization*

Linear classifiers using L_1 regularization differ from L_2-regularized models by enabling automatic feature selection through sparsity. This is achieved by penalizing the absolute values of the coefficients, which forces some of them to become exactly zero, effectively removing the associated features from the model. The L_1 regularization, as used in the lasso, minimizes the following objective function:

$$\min_{\beta} \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|, \tag{9}$$

where λ controls the amount of penalization and sparsity.

- **The Least Angle Regression (LARS)**
LARS algorithm efficiently computes solutions for the lasso across all values of λ. When $p > N$, the convex duality of the problem ensures that the number of non-zero coefficients is at most N, regardless of the value of λ. This makes L_1 regularization particularly valuable in high-dimensional settings, enabling severe feature selection.
- **Elastic Net**
Elastic net, a hybrid approach, combines L_1 and L_2 penalties to balance sparsity and the handling of correlated features. Its penalty term is given by:

$$\sum_{j=1}^{p} \left(\alpha|\beta_j| + (1-\alpha)\beta_j^2 \right), \tag{10}$$

where α determines the balance between the penalties. While the L_1 term enforces sparsity, the L_2 term encourages averaging of coefficients for correlated features. Elastic net can include more than N non-zero coefficients even when $p > N$, making it an attractive choice for some high-dimensional problems.

- **Fused Lasso**
For functional data, where features are indexed by a variable t, the fused lasso extends the lasso by incorporating a smoothness constraint on the coefficients. This is achieved by penalizing both the absolute values of coefficients and the differences between consecutive coefficients. The fused lasso minimizes:

$$\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p-1} |\beta_{j+1} - \beta_j| \right\}. \tag{11}$$

This approach is particularly useful for detecting patterns in ordered data, such as genomic or temporal sequences.
These methods showcase the adaptability of L_1-based regularization to various high-dimensional and structured data scenarios, enabling both effective modeling and interpretable results.

2.4 Classification When Features are Unavailable

Defining a feature vector is not always straightforward. However, if we can construct an $N \times N$ proximity matrix that measures the similarities between every pair of objects in the database, many classifiers can still be utilized by interpreting these similarities as inner-products. Beyond support vector machines, several other classifiers can operate solely with inner-product matrices. This capability also allows them to be kernelized in a manner similar to SVMs.

For instance, in nearest centroid classification, the distance from the test point to each centroid can be computed, enabling classification based on the closest centroid. Techniques such as K-means clustering, logistic or multinomial regularized regression, **L**inear **D**iscriminant **A**nalysis (LDA), and **P**rincipal **C**omponents **A**nalysis (PCA) can also be implemented using only inner-products.

That said, there are limitations when relying exclusively on inner-products. Variables cannot be standardized, individual variable contributions cannot be directly assessed, and it is impossible to distinguish between meaningful variables and noise. All variables are treated equally, without prioritizing their importance.

2.5 High-Dimensional Regression: Supervised Principal Components

Principal Components Analysis (PCA, Algorithm 1) is a powerful method for identifying linear combinations of features with large variance in a dataset. However, in many high-dimensional regression scenarios, the objective extends beyond merely identifying combinations with high variance. Instead, we seek linear combinations that exhibit both substantial variance and significant correlation with the outcome variable. To achieve this, **S**upervised **P**rincipal **C**omponents (SPC) focuses on features that individually have strong correlations with the outcome. This approach builds connections with other methods, such as Latent-Variable Modeling and **P**artial **L**east **S**quares (PLS) regression. Thresholded PLS can even be seen as a noisy variation of supervised principal components.

Algorithm 1 **S**upervised **P**rincipal **C**omponents.

```
Input: Data matrix, Response vector, Threshold $\theta$
Output: Principal component scores, loading matrix
Compute the standardized univariate regression coefficients for the outcome as a function of each feature separately.
for each value of the threshold $\theta$ from the list $0 \leq \theta_1 < \theta_2 < \ < \theta_K$ do
    Form a reduced data matrix consisting of only those features whose univariate coefficient exceeds $\theta$ in absolute value, and compute the first $m$ principal components of this matrix.
    Use these principal components in a regression model to predict the outcome.
Pick $\theta$ (and $m$) by cross-validation.
```

Despite its effectiveness in reducing test errors compared to competing methods, SPC does not inherently produce sparse models, often leading to redundancy due to highly correlated features being selected together. Preconditioning combines the strengths of SPC and the lasso method to address this. First, the supervised principal component predictor $\hat{y}_i$ is computed for each observation in the training set, with the threshold determined via cross-validation. Then, the lasso is applied using $\hat{y}_i$ as the response variable instead of the original outcome y_i , while retaining all features in the lasso fit. This process effectively denoises the response variable, reducing the impact of noisy features on the lasso and enhancing its performance.

2.6 Feature Assessment and the Multiple-Testing Problem

In some cases, the objective is not solely to achieve accurate outcome prediction but rather to evaluate the significance of individual features. This shifts the focus from prediction to a more traditional statistical problem: multiple hypothesis testing. For example, a two-sample t-statistic can be employed to determine which features are informative. In the context of the book's example, this involves comparing the mean gene expression levels between two groups of patients.

However, when dealing with a large number of features, such as genes, it is inevitable that some may appear significant purely by chance. This phenomenon is referred to as the multiple testing problem, which arises when evaluating a vast number of hypotheses simultaneously. This section primarily explores specific examples related to genetic testing and revisits traditional hypothesis testing methods commonly used in statistics.

3 Expriments

In this section, we compare the effectiveness of different methods (LDA, PCA, Lasso, and L_2 regularization) in practical high-dimensional problem tasks.

3.1 Datasets

We compare the methods introduced in Chapter 18 on two publicly available GEO datasets to assess their performance impact.

Gene **E**xpression **O**mnibus (GEO) is a public repository established by the **N**ational **C**enter for **B**iotechnology **I**nformation (NCBI) in 2000. It archives high-throughput gene expression data from various research institutions worldwide, encompassing diverse species and experimental conditions. The database includes data from microarray chips, next-generation sequencing, and other platforms, serving as a valuable resource for researchers in genomics and related fields.

We utilized the GEO Series ID GSE10072, which is a dataset designed for classifying lung tumor and normal lung tissues. This dataset comprises 107 samples, each with $22,283$ features, satisfying the condition $n_{features} \gg n_{samples}$, thus qualifying it as a high-dimensional dataset.

3.2 PCA VS LDA

We designed a simple experiment using the $sklearn.datasets.make\_classification()$ function to randomly generate a dataset. In this experiment, we consistently set $n\_informative = 3$, $n\_classes = 3$, and $random\_state = 42$. Initially, we configured $n\_samples = 100$ and $n\_features = 500$, meeting the condition where the number of features significantly exceeds the number of samples. The results showed that the model accuracy using LDA was 10% higher than that using PCA, with PCA accuracy at 27% and LDA accuracy at 37%. Figure 1 visually presents the dimensionality reduction outcomes of both methods.

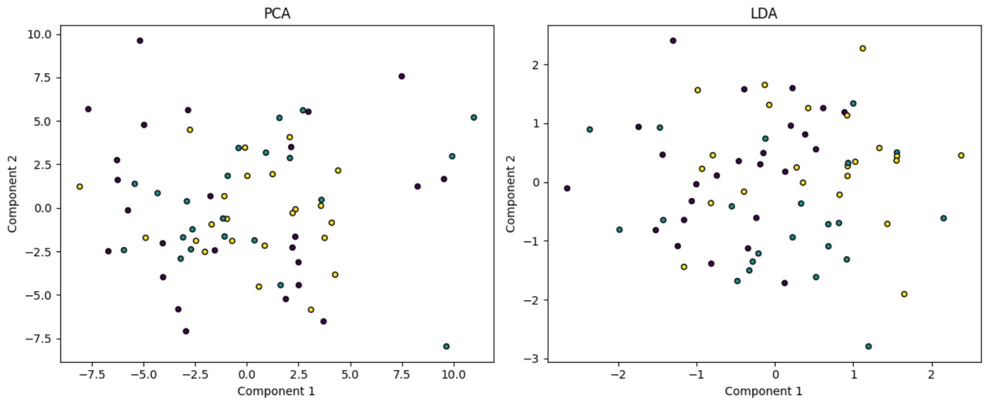

Fig. 1

Subsequently, we set $n\_samples = 10,000$ and $n\_features = 500$. This configuration provides a sufficient sample size to support the high-dimensional data. In this scenario, the model accuracy using PCA was 74%, while LDA achieved 64%, indicating that the model with PCA outperformed LDA by 10%. Figure 2 visually presents the dimensionality reduction results of both methods.

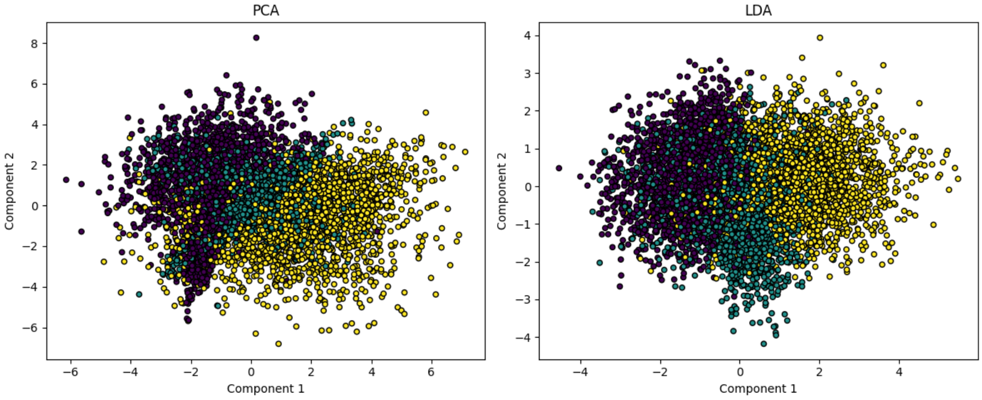

Fig. 2

3.3 Results in GEO

We conduct a comparative experiment using Logistic Regression on the GEO dataset (Series ID: GSE10072) to analyze the effects of four methods (LDA, PCA, Lasso, and L_2 regularization). Table 1 presents our experimental results.

Table 1: Cross-Validation Mean Accuracy of Different Models

Method	**Cross-Validation Mean Accuracy**
LDA	96.00
Supervised PCA	95.90
L_1 Regularization (Lasso)	100.00
L_2 Regularization (Ridge)	98.67

From Table 1, we observe that when $n\_features \gg n\_samples$, LDA outperforms PCA, and L1 regularization (Lasso) surpasses L2 regularization (Ridge). This is attributed to Lasso's inherent feature selection capability (as shown in Figure 3), which effectively reduces dimensionality.

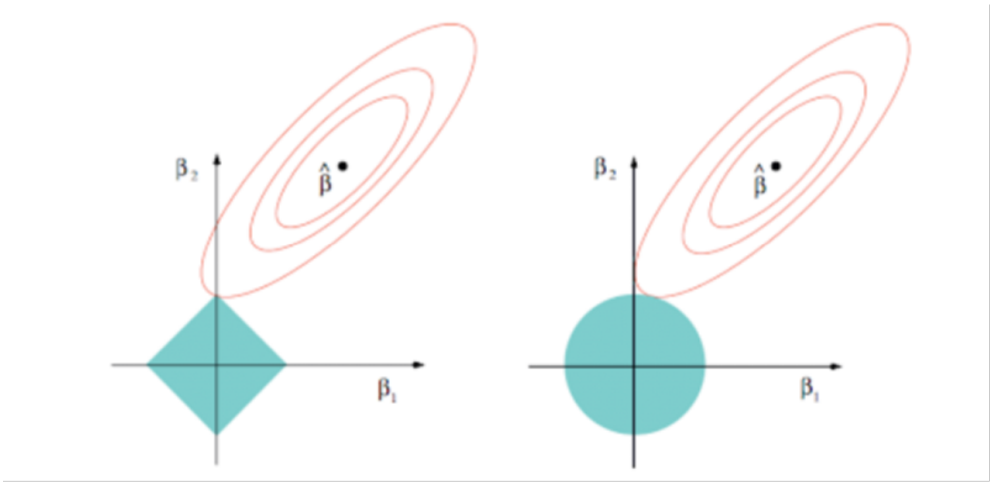

Fig. 3

4 Conclusion

The findings emphasize the importance of feature selection and regularization in high-dimensional data tasks. Lasso's ability to reduce dimensionality through sparsity proves beneficial, while LDA's simplified assumptions allow for efficient classification when data samples are scarce. The study suggests potential for further optimization and exploration of hybrid methods.