Journal of Data Science 0 (0), 1–22
0000
DOI: 0000
Computing in Data Science

Assessment of projection pursuit index for classifying high dimension low sample size data in R

Zhaoxing Wu[1,0] and Chunming Zhang[2,*]

[1] *University of Wisconsin-Madison, Department of Statistics, U.S.A.*
[2] *University of Wisconsin-Madison, Department of Statistics, U.S.A.*

Abstract

Analyzing "large p small n" data is becoming increasingly paramount in a wide range of application fields. As a projection pursuit index, the Penalized Discriminant Analysis (PDA) index, built upon the Linear Discriminant Analysis (LDA) index, is devised in Lee and Cook (2010) to classify high-dimensional data with promising results. Yet, there is little information available about its performance compared with the popular Support Vector Machine (SVM). This paper conducts extensive numerical studies to compare the performance of the PDA index with the LDA index and SVM, demonstrating that the PDA index is robust to outliers and able to handle high-dimensional datasets with extremely small sample sizes, few important variables, and multiple classes. Analyses of several motivating real-world datasets reveal the practical advantages and limitations of individual methods, suggesting that the PDA index provides a useful alternative tool for classifying complex high-dimensional data. These new insights, along with the hands-on implementation of the PDA index functions in the R package *classPP*, facilitate statisticians and data scientists to make effective use of both sets of classification tools.

Keywords *large p small n; linear discriminant analysis; penalized discriminant analysis; supervised classification;* SVM

1 Introduction

The projection pursuit is a statistical method, proposed by Kruskal (1969) and Friedman and Tukey (1974), for uncovering "interesting" structures from the original multivariate dataset. It is operated by pursuing an optimal projection of original data onto a low-dimensional space to reveal interesting structural features, such as clusters, inhomogeneity and subgroup information. The desired projection direction is found by numerically maximizing a "projection pursuit index". Yet, relatively few works on performing projection pursuit in high dimensions are available in the statistics literature. Refer to Zhang et al. (2022) and references therein for a recent review.

This paper mainly investigates a projection pursuit index extended from the linear discriminant analysis (LDA) for multi-class classification. LDA is a classical statistical method, aimed at developing a low-dimensional linear representation of multivariate data that separates the classes as much as possible. In practice, consider the dataset of p-dimensional vectors,

$$\boldsymbol{X}_{ij}, \quad i = 1, \ldots, g, \; j = 1, \ldots, n_i, \tag{1}$$

of the jth observation in the labeled class π_i, where g is the total number of labeled classes, and n_i is the number of observations in class π_i. Then $n = \sum_{i=1}^{g} n_i$ gives the total number of

*Corresponding author. Email: zwu363@wisc.edu or cmzhang@stat.wisc.edu.

observations. The between-class sums of squares and within-class sums of squares of the data vectors are captured by

$$\begin{aligned}\mathbf{B} &= \sum_{i=1}^{g} n_i(\overline{\boldsymbol{X}}_{i\cdot} - \overline{\boldsymbol{X}}_{\cdot\cdot})(\overline{\boldsymbol{X}}_{i\cdot} - \overline{\boldsymbol{X}}_{\cdot\cdot})^T,\\ \mathbf{W} &= \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\overline{\boldsymbol{X}}_{ij} - \overline{\boldsymbol{X}}_{i\cdot})(\overline{\boldsymbol{X}}_{ij} - \overline{\boldsymbol{X}}_{i\cdot})^T,\end{aligned}$$

respectively, where $\overline{\boldsymbol{X}}_{i\cdot} = \sum_{j=1}^{n_i}\boldsymbol{X}_{ij}/n_i$ is the ith group mean and $\overline{\boldsymbol{X}}_{\cdot\cdot} = \sum_{i=1}^{g}\sum_{j=1}^{n_i}\boldsymbol{X}_{ij}/n$ is the total mean. The LDA projection of the original dataset (1) onto a k-dimensional space seeks the optimal matrix $\mathbf{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k) \in \mathbb{R}^{p\times k}$, consisting of orthonormal column vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k$, to maximize the projection pursuit index (Lee et al. (2005)),

$$I_{\mathrm{LDA}}(\mathbf{A}) = \begin{cases} 1 - \frac{|\mathbf{A}^T\mathbf{W}\mathbf{A}|}{|\mathbf{A}^T(\mathbf{W}+\mathbf{B})\mathbf{A}|}, & \text{for } |\mathbf{A}^T(\mathbf{W}+\mathbf{B})\mathbf{A}| \neq 0,\\ 0, & \text{for } |\mathbf{A}^T(\mathbf{W}+\mathbf{B})\mathbf{A}| = 0,\end{cases} \tag{2}$$

where $|A|$ denotes the determinant of a matrix A. Particularly, the one-dimensional LDA projection direction is equivalent to solving the optimization problem:

$$\max_{\boldsymbol{a}\in\mathbb{R}^p} \frac{\boldsymbol{a}^T\Sigma_B\boldsymbol{a}}{\boldsymbol{a}^T\Sigma_W\boldsymbol{a}}, \tag{3}$$

where Σ_B is the between-class covariance matrix and Σ_W is the within-class covariance matrix.

This linear projection provides the basis for a classification rule, and can be modified to solve the supervised classification problem of data consisting of p features measured on n observations, each of which belongs to one of g classes. Nevertheless, the high-dimensional setting ($p > n$), also known as the "large p small n", causes the data piling issue in LDA. E.g., $\boldsymbol{a}^T\Sigma_W\boldsymbol{a}$ in (3) tends to be very small for some $\boldsymbol{a}$ when the sample size is much smaller than the dimension. One remedy is the Penalized Discriminant Analysis (PDA, Hastie et al. (1994)), which derives the optimal projection $\boldsymbol{a}$ by solving the maximization problem:

$$\max_{\boldsymbol{a}\in\mathbb{R}^p} \frac{\boldsymbol{a}^T\Sigma_B\boldsymbol{a}}{\boldsymbol{a}^T(\Sigma_W + \lambda\Omega)\boldsymbol{a}},$$

where Σ_W is regularized with a penalty matrix Ω and a regularization parameter λ.

Motivated from the PDA, Lee and Cook (2010) formulated a new projection pursuit index, called Penalized Discriminant Analysis index, to resolve the data piling issue. Their approach starts with the matrix $\widetilde{\Sigma} = (1-\lambda)\widehat{\Sigma} + \lambda\cdot\mathrm{diag}(\widehat{\Sigma})$, where $0 \leqslant \lambda < 1$ and $\widehat{\Sigma}$ denotes the sample covariance matrix of the data vectors. Using the standardized data vectors

$$\boldsymbol{X}^s_{ij} = \{\mathrm{diag}(\widehat{\Sigma})\}^{-1/2}(\boldsymbol{X}_{ij} - \overline{\boldsymbol{X}}_{\cdot\cdot}),$$

reduces the matrix $\widetilde{\Sigma}$ to the matrix $\widetilde{\mathbf{R}} = (1-\lambda)\widehat{\mathbf{R}} + \lambda\mathbf{I}_p$, where $\widehat{\mathbf{R}}$ is the sample correlation matrix of the data vectors, and $\mathbf{I}_p$ is a $p\times p$ identity matrix. Similarly, the between-class and within-class sums of squares of standardized data vectors are

$$\begin{aligned}\mathbf{B}^s &= \sum_{i=1}^{g} n_i(\overline{\boldsymbol{X}}^s_{i\cdot} - \overline{\boldsymbol{X}}^s_{\cdot\cdot})(\overline{\boldsymbol{X}}^s_{i\cdot} - \overline{\boldsymbol{X}}^s_{\cdot\cdot})^T,\\ \mathbf{W}^s &= \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\boldsymbol{X}^s_{ij} - \overline{\boldsymbol{X}}^s_{i\cdot})(\boldsymbol{X}^s_{ij} - \overline{\boldsymbol{X}}^s_{i\cdot})^T,\end{aligned}$$

respectively, with $\mathbf{B}^s + \mathbf{W}^s = n\widehat{\mathbf{R}}$, where $\overline{\boldsymbol{X}}^s_{i\cdot}$ is the i-th group mean of the standardized data and $\overline{\boldsymbol{X}}^s_{\cdot\cdot}$ is the total mean of the standardized data, namely 0. Consequently, the PDA index is then defined as

$$I_{\mathrm{PDA}}(\mathbf{A},\lambda) = 1 - \frac{|\mathbf{A}^T\{(1-\lambda)\mathbf{W}^s + n\lambda\mathbf{I}_p\}\mathbf{A}|}{|\mathbf{A}^T\{(1-\lambda)(\mathbf{B}^s+\mathbf{W}^s) + n\lambda\mathbf{I}_p\}\mathbf{A}|}, \tag{4}$$

which reduces to the LDA index (2) when $\lambda = 0$. Likewise, the PDA orthogonal projection onto a k-dimensional space is numerically obtained by maximizing $I_{\text{PDA}}(\mathbf{A}, \lambda)$ over $\mathbf{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k)$ with orthonormal columns.

The PDA index serves as an alternative way in classifying large p small n data, by incorporating regularization into the optimization algorithm to avoid the data piling issue. Besides projection pursuit indexes, classical multivariate analysis methods could solve classification problems and typically rely on taking the inverse of the sample covariance matrix $\widehat{\Sigma}$; but usually fail in the setting $p > n$, because $\widehat{\Sigma}$ is not full rank, and is not invertible. The Support Vector Machine (SVM), proposed by Cortes and Vapnik (1995) as a supervised machine learning technique for classification, aims to find a hyperplane that maximizes the separation of the data points to their potential classes in the feature space; the SVM algorithm can be implemented to linearly separable data points with a linear kernel or applied to linearly non-separable data with the Radial Basis Function (RBF) kernel. Refer to Marron (2015) for a new view of the SVM's performance in the "large p small n" context. Regardless of the kernel functions, Marron (2015) indicated that SVM will lose generalizability in the high-dimensional setting due to the data piling problem.

This paper contributes to assessing the performance of the PDA index under the "large p small n" setup and, for the first time, comparing its classification outcomes with the LDA index and SVM simultaneously. These three methods are trained on both simulated datasets and a wide range of real-world datasets to reveal the advantages and limitations of each method. Another main contribution of this paper is to illustrate the hands-on R implementation of the PDA index, together with some helper functions to facilitate more data analysts, machine learning engineers, and biologists to make effective use of the PDA index. Section 2 presents how to implement LDA and PDA indexes in R and some useful helper methods. Section 3 investigates PDA performances in simulation studies by comparing with the LDA index and SVM. Section 4 illustrates the performance of LDA, PDA, and SVM on real-world datasets. Section 5 concludes as well as makes recommendations regarding the suitable datasets and scenarios to apply supervised classification tools.

2 Description of PDA index

The LDA and PDA indexes can be implemented in the package *classPP* in Lee and Cook (2010). `PPindex.class()` calculates the projection pursuit index for the given data and the class information. The argument `PPmethod` supplies the choice of the projection pursuit index, by taking either "LDA" or "PDA" as the input. If "PDA" is given as the input, a hyperparameter input is also required, and then the argument `lambda` should not be set as `NULL`. The input data must be a numeric data matrix without class information, while the input `class` could be a character vector or a numeric vector.

```
library(classPP) #load the package
str(PPindex.class)
```

```
function (PPmethod, data, class, weight = TRUE, r = NULL, lambda = NULL,
    ...)
```

`PP.optimize.anneal()` applies the "simulated annealing optimization algorithm for projection pursuit" (SAPP) to find the optimal projection that maximizes the projection pursuit index for the given dataset. This R function provides the argument `projdim` to change the dimension

of projection users want to find. To implement SAPP, some default optimization parameters are given in the function, but with more flexibility, users are allowed to specify `cooling`, `temp`, and `energy`. They may use `temp` to determine the convergence speed or change `cooling` to control the number of iterations required to converge, which may decide whether the final result is the local maximum or global maximum. This function eventually returns two values, `index.best` and `proj.best`. Here, `index.best` is the projection pursuit index and could also be derived from `PPindex.class()`, while `proj.best` is the optimal projection from p dimensions to `projdim` dimensions. Other optimization functions, including `PP.optimize.random()` and `PP.optimize.Huber()`, are also provided in the package for users to choose from.

```
str(PP.optimize.anneal)
```

```
function (PPmethod, projdim, data, class, std = TRUE, cooling = 0.999,
    temp = 1, energy = 0.01, r = NULL, lambda = NULL, weight = TRUE, ...)
```

`PP.optimize.plot()` returns an optimal projection plot by taking in `PP.opt`, the optimal projection retrieved from `PP.optimize.anneal()` or any other optimization functions. The function is capable of plotting projections in one dimension or two dimensions.

```
str(PP.optimize.plot)
```

```
function (PP.opt, data, class, std = TRUE)
```

The code below provides an example of using `PP.optimize.plot()` and `PP.optimize.anneal()` in the package *classPP*. It first generates a toy dataset with 40 samples and 40 dimensions in 2 classes. Data points in one of the dimensions have differing means in each class, while data in other dimensions are normally distributed, unable to separate data into groups. The code computes the LDA and PDA ($\lambda = 0.7$) projection onto one dimension and two dimensions, with corresponding plots for visualization in Figure 1.

```
set.seed(123) # set a seed for reproducibility
n = 40 #number of observations
p = 40 #number of dimensions
#generate a simulated dataset
df = scale(as.data.frame(cbind(
      rbind(matrix(rnorm(n/2,2.2),,1),
            matrix(rnorm(n/2, -2.2),,1)),
      matrix(rnorm(n*(p-1)),,p-1)))) #data matrix
class = c(rep(1, n/2), rep(2, n/2)) #class vector
#optimal projection plots
par(mfrow = c(2, 2))
PP.opt = PP.optimize.anneal(''LDA'',1,df,class)
PP.optimize.plot(PP.opt, df, class)
title(''LDA'')
PP.opt = PP.optimize.anneal(''LDA'',2,df,class)
PP.optimize.plot(PP.opt, df, class)
title(''LDA'')
```

```
PP.opt = PP.optimize.anneal(''PDA'',1,df,class, lambda = 0.7)
PP.optimize.plot(PP.opt, df, class)
title(''PDA'')
PP.opt = PP.optimize.anneal(''PDA'',2,df,class, lambda = 0.7)
PP.optimize.plot(PP.opt, df, class)
title(''PDA'')
```

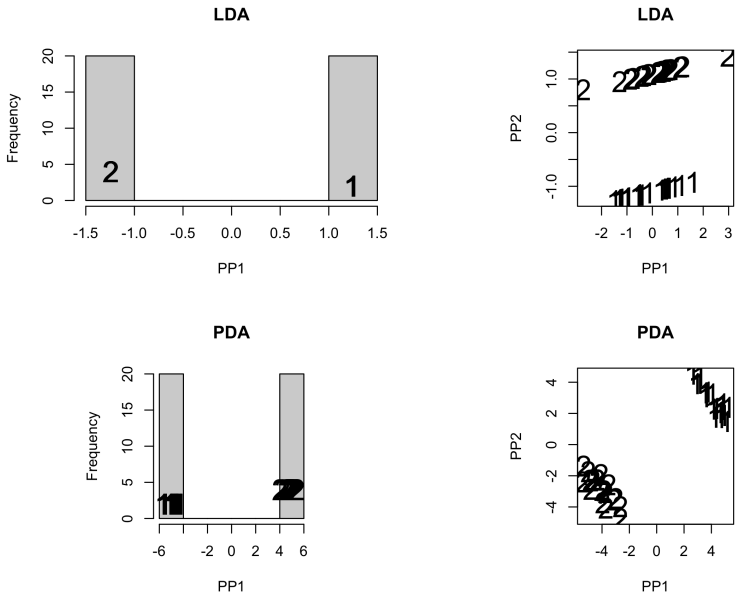

Figure 1: Projection plots generated by `PP.optimize.plot()`. The top left is LDA projection into one dimension and the top right is LDA projection into two dimensions. For PDA with a hyperparameter $\lambda = 0.7$, the bottom left shows data projected into one dimension and the bottom right shows data projected into two dimensions.

Even though the package *classPP* is a powerful tool that is easy to implement, it lacks several significant features when it comes to practice for people unfamiliar with projection pursuit methods. First of all, to validate classification accuracy, people prefer to split data into training and test sets, and they might need to visualize the two different projections on the same plot for comparison. However, the plotting method in the package *classPP* fails to plot both datasets simultaneously. Second, the absence of a method performance scale in *classPP* makes it harder for users to evaluate performance for the task of supervised classification. A more precise and quantitative evaluation like an accuracy score might be necessary other than the visual inspection. Lastly, users who are not familiar with the projection pursuit index might need a data-driven approach for selecting the hyperparameter in the PDA index. To learn the utility of this method, a helper function for parameter selection should be provided. The next part focuses on demonstrating the code and examples for the significant features that are missing from the package *classPP*.

Since `PP.optimize.plot()` fails to plot both training and test sets simultaneously, a helper function `plot_test_train()` is provided to draw projected training and test data on the same

plot for evaluation. In the output plot, the training data are plotted as numerical labels in black while the test data are plotted as geometric shapes in dark grey. The first two arguments of the function are the test and training data after projection, and the third and fourth arguments are classes of the test and training data. Users need to input a vector of all distinct classes as `cls_tot`. Figure 2 demonstrates an example of the projection of the toy dataset mentioned earlier after the train-test-split.

```
#proj.data.test: test dataset after projection
#proj.data.train: train dataset after projection
#cls_test: vector of classes of test dataset
#cls_train: vector of classes of train dataset
#cls_tot: vector of distinct classes in the entire dataset
plot_test_train = function(proj.data.test, proj.data.train, cls_test, cls_train, cls_tot){
  lim = c(min(min(proj.data.test), min(proj.data.train)),
          max(max(proj.data.test), max(proj.data.train)))

  plot(x=NA, y=NA, xlim = lim, ylim= lim, xlab='''', ylab=''', main=''',
       axes=FALSE, frame.plot=TRUE)
  for (i in 1:length(cls_tot)){
    #plot projected test data
    points(proj.data.test[,1][cls_test == cls_tot[i]],
           proj.data.test[,2][cls_test == cls_tot[i]],
           pch = i, col = ''darkgrey'')
  }
  #plot projected train data
  text(proj.data.train[,1], proj.data.train[,2], as.numeric(factor(cls_train)), cex=1)
}

#split the dataset into test and train dataset
ind = sample(1:n, n/4, replace=FALSE)
test = df[ind,]
train = df[-ind,]
cls_test = class[ind]
cls_train = class[-ind]

PP.opt = PP.optimize.anneal(''PDA'', 2, train, cls_train, lambda = 0.6)
proj.data.test = as.matrix(test)%*%PP.opt$proj.best
proj.data.train = as.matrix(train)%*%PP.opt$proj.best
plot_test_train(proj.data.test, proj.data.train, cls_test, cls_train,
                levels(as.factor(class)))
```

This paper adopts an accuracy score to quantitatively evaluate the percentage of data points correctly allocated to previously defined groups. The R function `acc()` takes in the optimal projection, training and test datasets, and class labels of training and test datasets, with a return of the accuracy score.

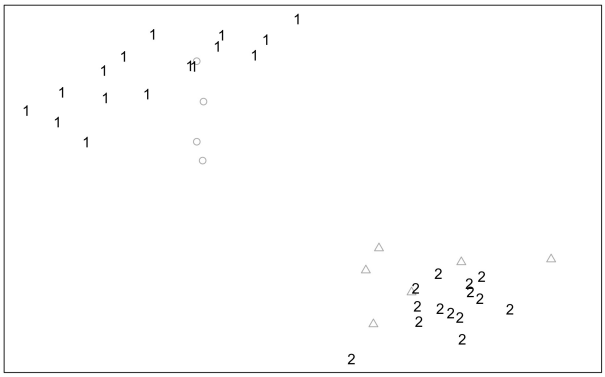

Figure 2: The projected training and test dataset from the toy data using `plot_test_train()`. The black digits represent the training data, and the grey symbols represent the test data.

```
#a: the optimal projection (PP.opt$proj.best)
#train: traversed train dataset
#test: traversed test dataset
#cls_test: vector of classes of test dataset
#cls_train: vector of classes of train dataset
acc = function(a, train, test, cls_train, cls_test){
  #get the average of classes in the training dataset
  cls_mean = data.frame(rep(NA, nrow(train)))
  for (i in unique(cls_train))
    cls_mean = cbind(cls_mean, rowMeans(train[, cls_train==i]))
  cls_mean = cls_mean[,-1] #remove the first column of NA

  rst = c() #storing predicted class of each test data point
  for (i in 1:ncol(test)){ #iterate through each test data point
    temp=c() #storing the distance of one data point to each class
    for (j in 1:ncol(cls_mean)){ #iterate through each class
      x = 0 #storing the distance of one data point to one class
      for (k in 1:ncol(a)) #iterate through the projected dimension
        x=x+(a[,k]%*%(test[,i]-cls_mean[,j]))^2
      temp=c(temp, x)
    }
    rst = c(rst, unique(cls_train)[which.min(temp)])
  }
  return (sum(rst==cls_test)/length(cls_test))
}

acc(PP.opt$proj.best, t(train), t(test), cls_train, cls_test)
```

```
[1] 0.9
```

As the PDA index requires a hyperparameter λ, Lee and Cook (2010) suggests an empirical way of selecting an appropriate hyperparameter for a certain dataset. Let W^{PP} be the within-

class sum of squares of the optimal projected data using the PDA index with λ, and define $S(\lambda)$ as follows:

$$S(\lambda) = \mathrm{tr}(W^{\mathrm{PP}})/(nk)$$

where n is the total number of observations and k is the projection dimension. The optimal hyperparameter is chosen to satisfy $S(\lambda) = 1$, when there is a balance in the within-class sum of squares between the training dataset and the test dataset. In the code below, the R function `S()` calculates a list of $S(\lambda)$ for the input λ vector and returns a plot $S(\lambda)$ vs. λ helping users select the optimal λ. Figure 3 shows a PDA hyperparameter selection plot for the Leukemia dataset, gathered by Golub et al. (1999), containing 3571 dimensions and 72 observations of 3 different classes. Each point symbolizes $S(\lambda)$ for the corresponding λ, and the red line is a fitted smoothing spline of all $S(\lambda)$, allowing users to better visualize the overall trend. The intersection between the red line and the blue line gives the optimal hyperparameter value, which in this case is $\lambda = 0.95$.

```
library(DiscriMiner) #import function withinSS()
#data: the original dataset
#class: vector of classes of the dataset
#method: ''PDA'' or ''LDA''
#dim: projection dimensions
#hyperparameter: vector of multiple hyperparameters
S = function(data, class, method, dim, hyper){
  S = c()
  for (i in hyper){
    PP.opt = PP.optimize.anneal(method,dim,data,class,lambda = i)
    W = withinSS(data%*%PP.opt[[''proj.best'']], class) #calculate within-class sum of squares
    S = c(S, sum(diag(W))/dim/nrow(data))
  }

  smoothingSpline = smooth.spline(hyper, S, spar=0.8)
  plot(hyper, S)
  lines(smoothingSpline, col = ''red'')
  abline(h=1, col=''blue'')
}
```

3 Simulation evaluation

To study the performance of the PDA index, we conducted simulation experiments under four different scenarios, where all datasets were randomly generated under certain rules and standardized. To prevent contingency and ensure the reliability and robustness of results, we repeated the data generation process for 20 times under the same condition and trained as well as tested all methods repetitively 20 times, where the mean of classification accuracy is computed for performance evaluation. The PDA index is compared with the LDA index and SVM (using both linear kernel and RBF kernel). The PDA indexes with $\lambda \in \{0.4, 0.6, 0.8\}$ are experimented with to examine the results for different choices of λ. For the SVM, hyperparameters are selected by 10-fold cross-validation.

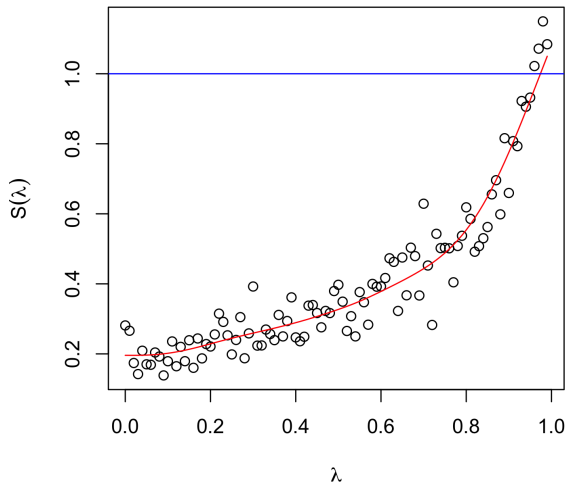

Figure 3: $S(\lambda)$ vs. λ for the Leukemia dataset. The optimal hyperparameter value is selected to be $\lambda = 0.95$.

3.1 Percentage of important variables

The important variable is defined as any variable able to separate different classes, and conversely, the unimportant variable is any variable that failed to distinguish between class information. All simulated datasets contain 40 samples with 2 classes in $p = 500$ dimensions, and thus each class has 20 data points. Each data point is generated from standard Gaussian distributions, except that the means of the two classes for the important variables are respectively shifted to 2.2 and -2.2. We simulated 21 sets of data with different percentages of important variables, where 21 different percentages are generated from 3% to 5% with a step of 0.1%.

A plot of how the accuracy changes with different percentages of important variables is shown in Figure 4. The LDA index exhibits the worst performance among all methods, and its accuracy score never exceeds 85%. For both the LDA index and SVM (RBF), their accuracy scores are significantly lower if the proportion of important variables is lower than 3.3%, proving that they are not suitable to classify datasets containing little discrimination information. Nevertheless, there is an increasing trend in accuracy scores when the percentage of important variables increases from 3.0% to 3.5% for both methods. This can be explained by more class information received by the model with an increasing percentage of important variables. For large percentages of important variables, the SVM (RBF) outperforms the LDA index but is less stable compared with the PDA index and SVM (linear). The PDA indexes and SVM (linear) hardly misclassify any sample, and they are not very sensitive to the percentage of important variables.

3.2 Ratio of sample sizes to dimensions

The ratio of sample sizes to dimensions is defined as n/p. All simulated datasets here contain 2 different classes in 500 dimensions. The number of important variables is always 10, and therefore,

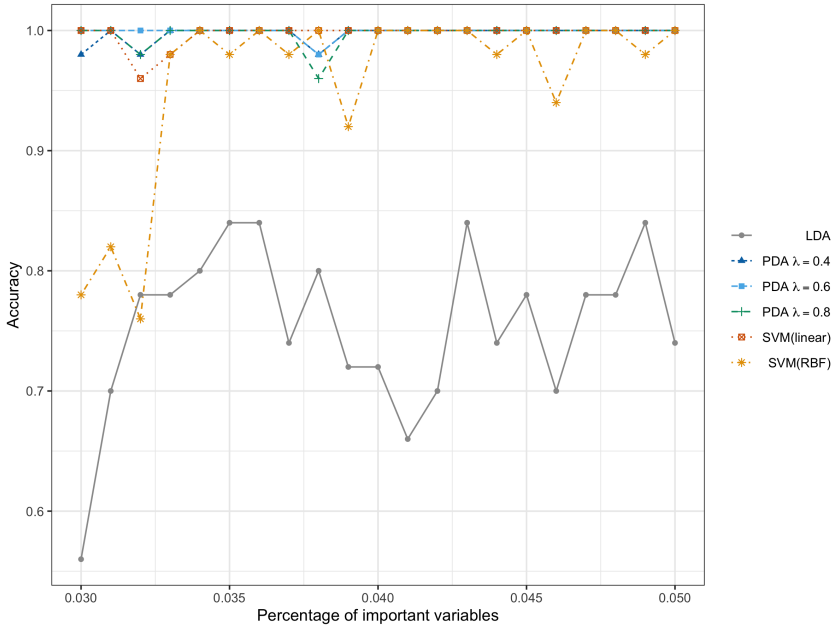

Figure 4: A line plot of the accuracy score of LDA indexes, PDA indexes with $\lambda \in \{0.4, 0.6, 0.8\}$, SVM with linear kernel and SVM with RBF kernel versus the percentage of important variables. Different colors and linetypes in the graph correspond to methods.

the amount of information used for classification is kept the same for different datasets. In this case, with dimensionality keeping the same, the number of observations in each class varies according to the ratio between sample sizes and dimensions. The number of observations in each class is chosen between 5 and 50 with a step of 3.

When the ratio is more than 15%, all models except LDA achieved an accuracy score of over 95% as shown in Figure 5, proving that more observations provide more information to the classification methods. As the ratio decreases, all models exhibit a decline in classification performance. This might be explained by "the curse of dimensionality" where more data features lead to a sparsity problem as the distance between data points becomes larger. Despite the low ratio complicating the task of allocating new samples, the PDA indexes manage to obtain accuracy no lower than 70%, demonstrating its ability to handle "large p small n" data. Even though SVM (linear) has stable performance with high accuracy when the ratio is over 5%, its performance plunges to only 20% when there is an extreme disparity between sample size and dimensionality. SVM (RBF) has unstable performance with small ratios, and in this scenario unable to surpass the performance of SVM (linear). For the LDA index, its accuracy score is vibrating between 55% and 80%, and there is no indication that its disappointing performance is affected by different ratios.

3.3 Number of classes

The number of classes for simulation ranges from 2 to 10. For the important variables, the mean of the first labeled class is 4, and the mean increase by 4 for the second labeled class, and the pattern continues for the rest of the classes. All simulated datasets contain 20 observations for each class in 500 dimensions, including 10 important variables.

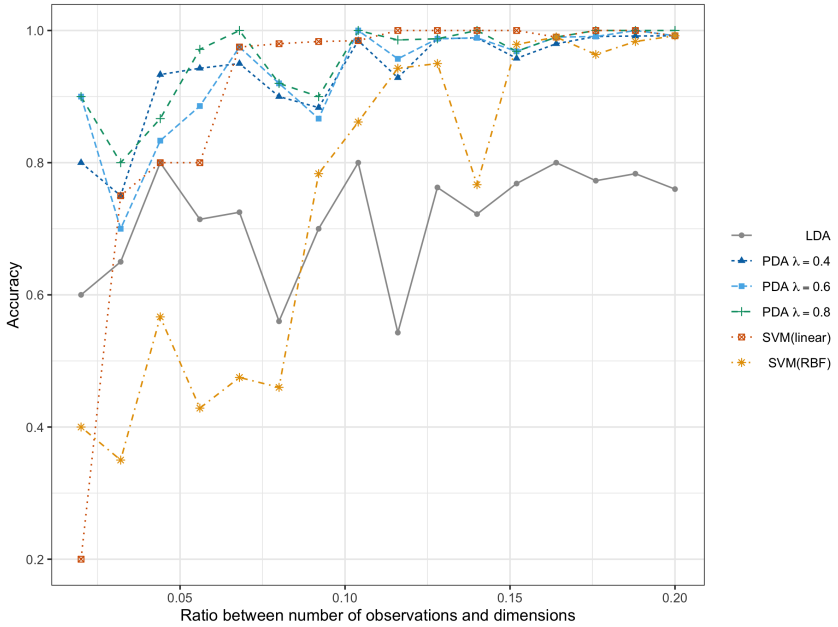

Figure 5: A line plot of the accuracy score of LDA indexes, PDA indexes with $\lambda \in \{0.4, 0.6, 0.8\}$, SVM with linear kernel and SVM with RBF kernel versus the ratio (n/p) of the total number of observations to dimensions. Different colors and line types in the graph correspond to methods.

Different performance metrics are evaluated other than the accuracy score, since there is an inherent ordering between classes. According to Gaudette and Japkowicz (2009), Mean Squared Error (MSE) and Mean Absolute Error (MAE) are the best metrics for ordinal classification so they are also measured in this case. As the number of classes increases for all methods, classification accuracy diminishes in Figure 6, and both MSE and MAE increase in Figure 7 and Figure 8, probably due to the great difficulty of maximizing differences among multiple classes. When there are more than 4 different classes, no method is able to achieve an accuracy score higher than 50%. Nevertheless, PDA indexes rank at the top among all methods when there are more than 8 classes with the highest accuracy and smallest MSE and MAE. For fewer class labels, PDA indexes and SVM (linear) obtain similar performance. The LDA index underperforms in comparison to PDA indexes. The SVM (RBF) has the lowest accuracy score and highest MSE and MAE among all methods when allocating more than 2 class labels. In conclusion, no method is able to address the challenge of multi-class classification with high accuracy or small error, but the PDA index generally has the best performance among all methods.

3.4 Effect of outliers

Including outliers in the simulated datasets could better resemble datasets collected in practical applications. Under this scenario, two types of datasets are simulated according to whether outliers are added to all unimportant variables or outliers are only added to important variables. For both cases, outliers are generated from a normal distribution with a mean of 0 and a standard deviation of 15, and the number of outliers in each variable ranges from 0 to 16. Again, all simulated datasets contain 20 observations for two classes in 500 dimensions, including 10 important variables. The means of the two classes for the important variables are respectively

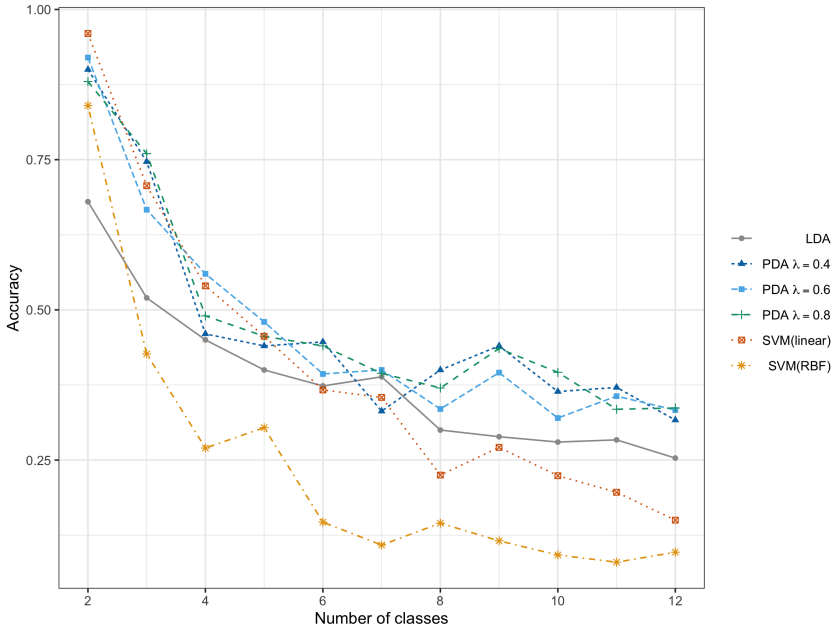

Figure 6: A line plot of the accuracy score of LDA indexes, PDA indexes with $\lambda \in \{0.4, 0.6, 0.8\}$, SVM with linear kernel and SVM with RBF kernel versus the number of classes. Different colors and linetypes in the graph correspond to methods.

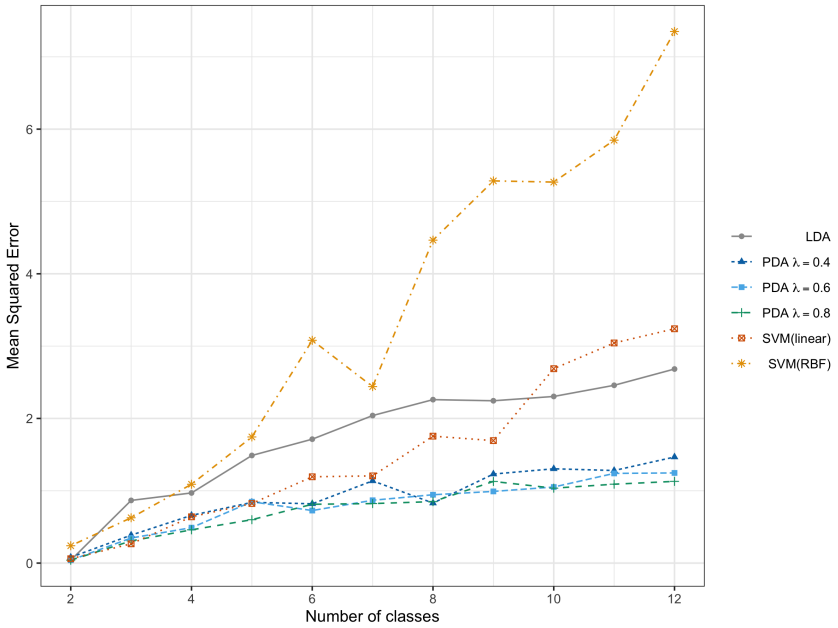

Figure 7: A line plot of MSE of LDA indexes, PDA indexes with $\lambda \in \{0.4, 0.6, 0.8\}$, SVM with linear kernel and SVM with RBF kernel versus the number of classes. Different colors and linetypes in the graph correspond to methods.

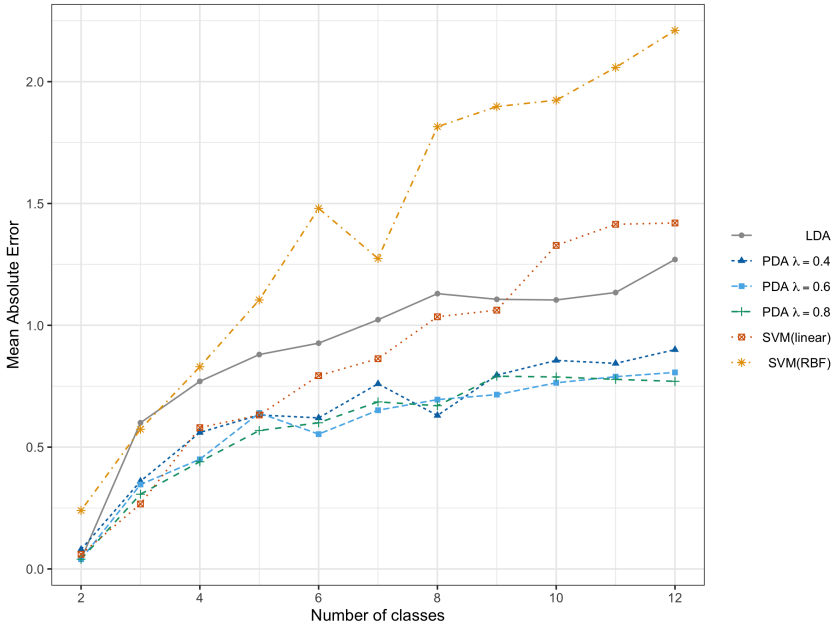

Figure 8: A line plot of MAE of LDA indexes, PDA indexes with $\lambda \in \{0.4, 0.6, 0.8\}$, SVM with linear kernel and SVM with RBF kernel versus the number of classes. Different colors and linetypes in the graph correspond to methods.

shifted to 2.2 and -2.2.

When the outliers are added to only important variables, there is no distinct advantage observed in any classification method for a large number of outliers. As shown in the top panel of Figure 9, for more than two outliers in each variable, the accuracy scores for all models are mostly lower than 60% and there is no significant difference among all models. Due to the small sample size, a small increase in the number of outliers might substantially contaminate the datasets, and the added noise makes it harder for the classification task. Even though all methods are sensitive to the effect of many outliers, the PDA index and SVM (linear) are able to identify one or two outliers from important variables.

When the outliers are added to unimportant variables as shown in the bottom panel of Figure 9, the performance of the PDA indexes is hardly influenced, maintaining an accuracy score around 95%. This indicates that the PDA index is able to identify the important variables in the datasets and robust to any outliers added to unimportant variables. Similarly, SVM (linear) has an accuracy slightly lower than the PDA index but generally greater than 80%. In contrast, the accuracy lines for the LDA index and SVM (RBF) are unstable and there are several sudden drops and rises in performance as the number of outliers added to unimportant variables increases, especially when there are more than 5 outliers. This observation signifies that the outliers added to unimportant variables sometimes confuse the LDA and SVM (RBF) algorithm, leading to an undesired performance. Nevertheless, the LDA and SVM (RBF) index sometimes are able to identify the outliers in unimportant variables and instead use the important variables for classification when the number of outliers is around 12, or 15, reaching an accuracy score as high as 80%.

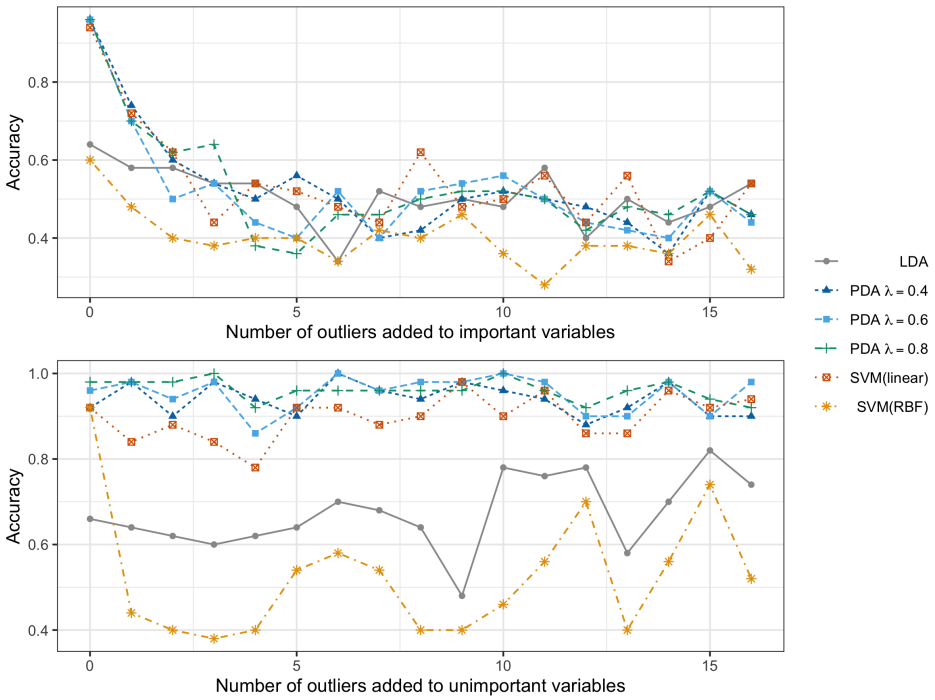

Figure 9: A line plot of the accuracy score of LDA indexes, PDA indexes with $\lambda \in \{0.4, 0.6, 0.8\}$, SVM with linear kernel and SVM with RBF kernel versus the number of outliers added to each variable. Different colors and linetypes in the graph correspond to methods. Top: All outliers are added to important variables. Bottom: All outliers are added to all variables except important variables.

4 Real-world data analysis

The simulated dataset might not be able to capture the complexity or characteristics of real-life data, and thus it is paramount to explore the performances of PDA indexes on real-world data as well to draw meaningful insights. The PDA index is compared with the LDA index and SVM in this section. All datasets are standardized before training. For the SVM, hyperparameters are selected by 10-fold cross-validation.

4.1 Microarray data

Not all real-world data are high-dimensional with small sample sizes, but microarray data generally satisfies this requirement and is one of the most important applications of the "large p small n" problem. Each patient sample is usually analyzed into a large list of gene expressions but patient samples for each class are hard to collect. We investigated 7 different microarray datasets with the data collection process for each dataset listed below. A summary of the number of samples, dimensions, and classes of microarray datasets is displayed in Table 1.

(i) Leukemia: Yeoh et al. (2002) acquired the diagnostic bone marrow samples from 248 pediatric acute lymphoblastic leukemia (ALL) patients who were determined to have one and only one of the six known pediatric ALL prognostic subtypes, which include T-cell lineage ALL (T-ALL), E2A-PBX1, TEL-AML1, MLL rearrangements, BCR-ABL, and hyperdiploid

karyotypes with more than 50 chromosomes (HK50). The 248 patients included 43 T-ALL, 27 E2A-PBX1, 79 TEL-AML1, 15 BCR-ABL, 20 MLL, and 64 HK50 patients.

(ii) Breast cancer: Sørlie et al. (2001) examined 85 experimental samples gathered from cDNA microarrays to identify breast carcinoma. The data consist of 456 cDNA clones from 427 unique genes for 78 carcinomas, 3 benign tumors, and 4 normal tissues.

(iii) Prostate cancer: Singh et al. (2002) have examined 235 radical prostatectomy specimens from surgery patients between 1995 and 1997. The authors used oligonucleotide microarrays containing probes for approximately 12, 600 genes and expressed sequence tags. They have reported that 102 of the radical prostatectomy specimens, including 52 prostate tumor samples and 50 non-tumor prostate samples.

(iv) Crohn's disease: According to Burczynski et al. (2006), Crohn's Disease and Ulcerative Colitis patients were classified according to transcriptional profiles in peripheral blood mononuclear cells from 42 healthy individuals, 59 Crohn's Disease patients, and 26 Ulcerative Colitis patients by hybridization to microarrays interrogating 22283 sequences.

(v) Sarcoma: Soft tissue sarcomas were studied by Nakayama et al. (2007) genetically, using 105 samples from 10 types of soft tissue tumors containing 22283 probe sets, consisting of 16 synovial sarcoma, 19 myxoid/round cell liposarcoma, 3 lipoma, 3 well-differentiated liposarcoma, 15 dedifferentiated liposarcoma, 15 myxofibrosarcoma, 6 leiomyosarcoma, 3 MPNST, 4 fibrosarcoma and 21 MFH.

(vi) Central nervous system disorder: Pomeroy et al. (2002) predicted central nervous system embryonal tumor outcome based on gene expression with 60 samples, containing 39 medulloblastoma survivors and 21 treatment failures.

(vii) Lung cancer: Malignant pleural mesothelioma (MPM)and adenocarcinoma (ADCA) of the lung were distinguished by Gordon et al. (2002) based on the expression levels of genes from 181 tissue samples (31 MPM and 150 ADCA).

Table 1: Real-life microarray datasets of different diseases.

Disease	**# Samples**	**# Dimensions**	**# Classes**
Leukemia	248	12625	6
Breast cancer	85	456	5
Prostate cancer	102	12600	2
Crohn's disease	127	22283	3
Sarcoma	105	22283	10
Central nervous system disorder	60	7128	2
Lung cancer	181	12533	2

Figure 10 illustrates the projection pursuits of LDA and PDA for seven microarray datasets. Since the SVM is not devised from the projection pursuit index, its performance cannot be examined on the projected data, but Table 2 includes the accuracy score for both projection pursuit indexes and SVM with different kernel functions.

LDA projections of breast cancer, prostate cancer, Crohn's disease, and central nervous system disorder contain data points from the same class piling up in the same projection space. For breast cancer in Figure 10(b), LDA seems to get confused with classes 2, 4, and 5, resulting in a low accuracy score of 45.5%. In contrast, the PDA index is able to project the 3 classes in

Table 2: The accuracy scores of 7 real-life microarray datasets of different diseases using LDA indexes, PDA indexes with $\lambda \in \{0.4, 0.6, 0.8\}$ and SVM.

Disease	LDA	PDA ($\lambda = 0.4$)	PDA ($\lambda = 0.6$)	PDA ($\lambda = 0.8$)	SVM (linear)	SVM (RBF)
Leukemia	60.3%	74.6%	76.2%	74.6%	98.4%	82.5%
Breast cancer	45.5%	77.3%	81.9%	72.7%	77.3%	81.8%
Prostate cancer	68.0%	98.0%	100.0%	96.0%	88.1%	88.1%
Crohn's disease	71.0%	90.3%	77.5%	87.1%	84.4%	65.6%
Sarcoma	14.3%	42.9%	32.1%	28.6%	59.2%	59.2%
Lung cancer	46.7%	60.0%	53.4%	66.7%	100%	97.8%
Central nervous system disorder	100.0%	100.0%	100.0%	100.0%	66.7%	66.7%

different data spaces, achieving accuracy as high as 81.9%, similar to SVM. For prostate cancer in Figure 10(c), there is a large between-class variance in the training set of the LDA index, but test samples from different class labels seem to aggregate together. PDA accuracy scores of prostate cancer under different hyperparameters are between 92.0% and 100.0%. However, the LDA accuracy score is approximately 30% lower than PDA, and SVM is around 10% lower than PDA, even though this is a binary classification task. Similarly for Crohn's disease, the difference in accuracy scores between PDA indexes and the LDA index could be as high as 20.0% as shown in Table 2. For SVM (RBF), its accuracy score is even more than 5% lower than the LDA index. More specifically, as the value of the hyperparameter increases in the PDA method, data points are projected in a more scattered pattern in Figure 10(d), corresponding to the larger penalty added to the method. For central nervous system disorder, apparently, no sample is misclassified for projection pursuit indexes. In Figure 10(g), the PDA indexes seem to project test data points with lower within-class variance than the LDA index, informing that PDA indexes are more robust regardless of the same accuracy metrics. In contrast, SVM achieved an accuracy score of only 66.7%, indicating that projection pursuit indexes are more suitable for the classification task in this case. In a nutshell, when handling the 4 microarray datasets mentioned above, the PDA index displays better performance than SVM and the LDA index, where the latter suffers from a severe data piling issue.

To project leukemia, sarcoma, and lung cancer, the PDA indexes fail to maintain in the first place among all methods but still achieve decent performance. In Figure 10(a), employing different hyperparameters of PDA indexes in the Leukemia dataset has little impact on both its projection pursuits and the performances. Even though the accuracy score of SVM (linear) is more than 20% higher than the PDA indexes, PDA accuracy scores are more than 10% higher than the LDA index. The task of classifying 6 different leukemia class labels is challenging, and achieving an accuracy of around 75% is a satisfactory performance for the PDA indexes. For sarcoma, both projection pursuit indexes struggle to classify all categories except categories 9 and 8 as shown in Figure 10(e). The LDA prediction accuracy is only 14.3%, namely almost all samples in the test set are misclassified, while PDA obtains a score around 30% higher than the LDA index in Table 2. The accuracy scores of SVM are slightly lower than 60% in this case. A large number of classes might justify the poor performance of three methods, conforming to the simulation result in Section 3.3. To predict lung cancer, the LDA index fails to reach an accuracy

Table 3: This table demonstrates the number of samples composed by different artists in different genres.

Genre	**Artist**	**# Samples**
Classical music	Beethoven	7
	Mozart	6
	Vivaldi	10
Rock music	Abba	10
	the Beatles	10
	the Eels	9

score above 50.0% while PDA accuracy is only around 60.0%, even though there are only two classes in total. In Figure 10(f), both methods manage to position training data in two different clusters for different classes, but test data points from different classes mix with each other. In contrast, SVM has almost perfect performance, implying that this method is more suitable for classifying lung cancer than projection pursuit indexes.

For the LDA index, there is a smaller within-class variance in the training set than that in the PDA index, suggesting a tendency for overfitting in LDA. This might corroborate why most of LDA accuracy scores are over 10% lower than PDA accuracy scores as found in Table 2. Moreover, there are certain datasets that certain methods are more suitable to analyze. For example, classifying central nervous system disorder is a simple task for projection pursuit indexes but a hard problem for SVM. On the other hand, projection pursuit indexes find it hard but SVM find it simple to classify leukemia and lung cancer datasets. There is no absolute advantage of one method over the other method, but there are scenarios where some method exhibit relative advantages. It is recommended that users mainly adopt the PDA index and may sometimes refer to SVM in practice for comparison, but avoid using the LDA index under the high-dimensional setting.

4.2 Music data

One music dataset, similar to the one generated by Lee and Cook (2010), is explored, and in this paper, we present an even more detailed analysis and different results from the original paper. In particular, we examine two classification problems, classifying music genres and classifying music artists. For music data collected, rock music tracks are created by Abba, the Beatles, and the Eels, while classical music tracks are composed by Vivaldi, Mozart, and Beethoven. With the python package `Librosa`, 81 different features are extracted, including *mean, standard deviation, kurtosis, skew, zero crossing rate, spectral centroids, spectral rolloff, mel-frequency cepstral coefficients, chromogram*, as well as *spectral bandwidth*. In total, there are 52 different samples in 81 dimensions, as demonstrated in Table 3. We treat the music genre as the major class, including rock and classical music, and meanwhile, we regard the music artist as the minor class, including six different classes. Even though the ratio between the total number of observations and dimensions in this scenario is not as extreme as the microarray data, it is meaningful to compare the model performances in classifying the same data on different class labels (major class labels and minor class labels).

All methods accomplish classifying genre labels with zero misclassification errors. Figure 11

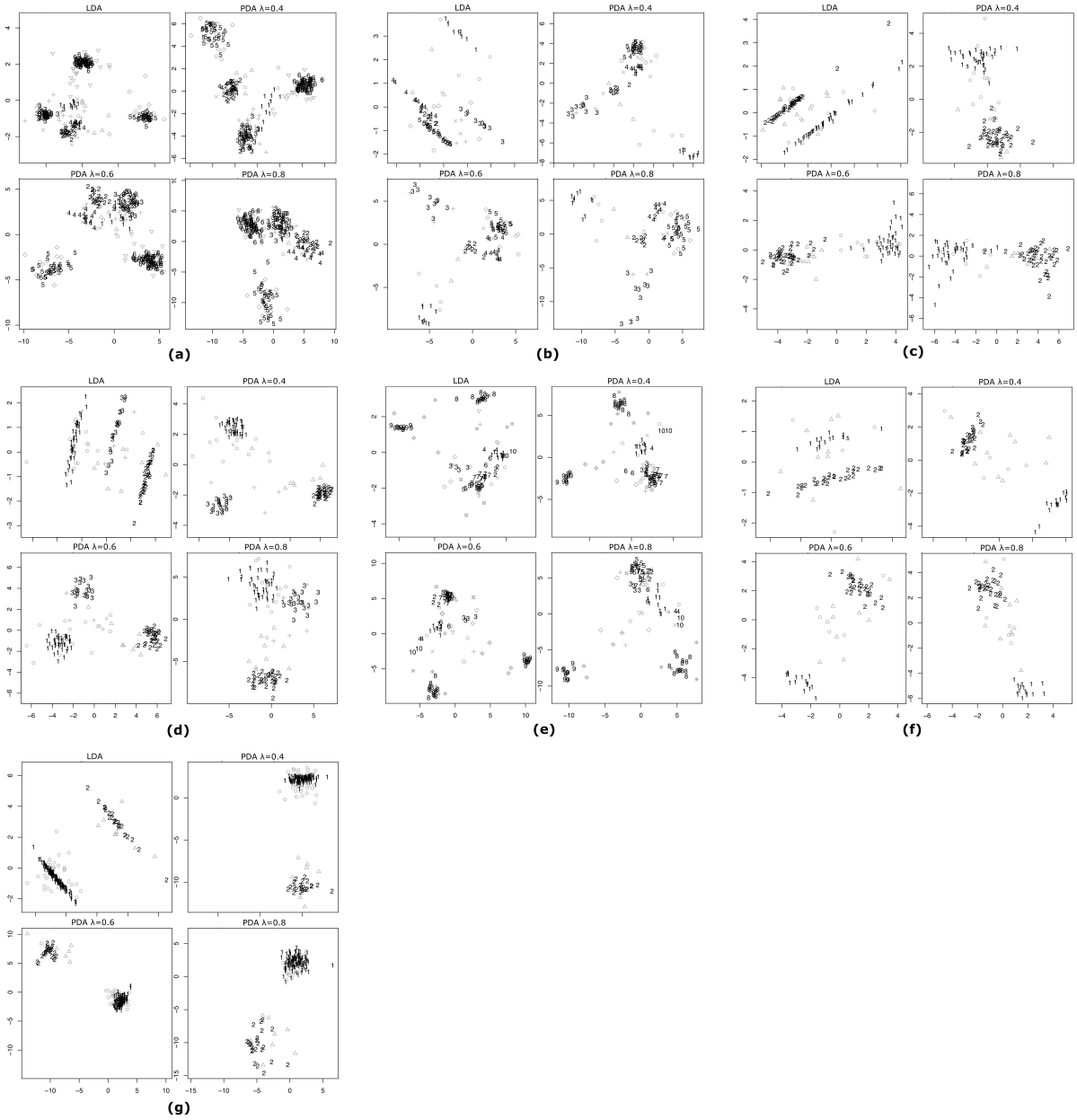

Figure 10: Projection pursuit on real-life microarray datasets of different diseases using different LDA indexes and PDA indexes with $\lambda \in \{0.4, 0.6, 0.8\}$. Numbers in the plot represent the training set classes, and symbols in the plot represent the test set classes. Plot (**a**): Leukemia. Plot (**b**): Breast cancer. Plot (**c**): Prostate cancer. Plot (**d**): Crohn's disease. Plot (**e**): Sarcoma. Plot (**f**): Lung cancer. Plot (**g**): Central nervous system disorder.

illustrates the data piling problem in LDA projection pursuits, where training labels fall onto two straight lines. This implies that the LDA index might not be as robust as PDA indexes due to the low training set variance and high test set variance.

Classifying artists should be a difficult task as there are more class labels but fewer samples in each class, where the methods not only need to decipher between rock music and classical music

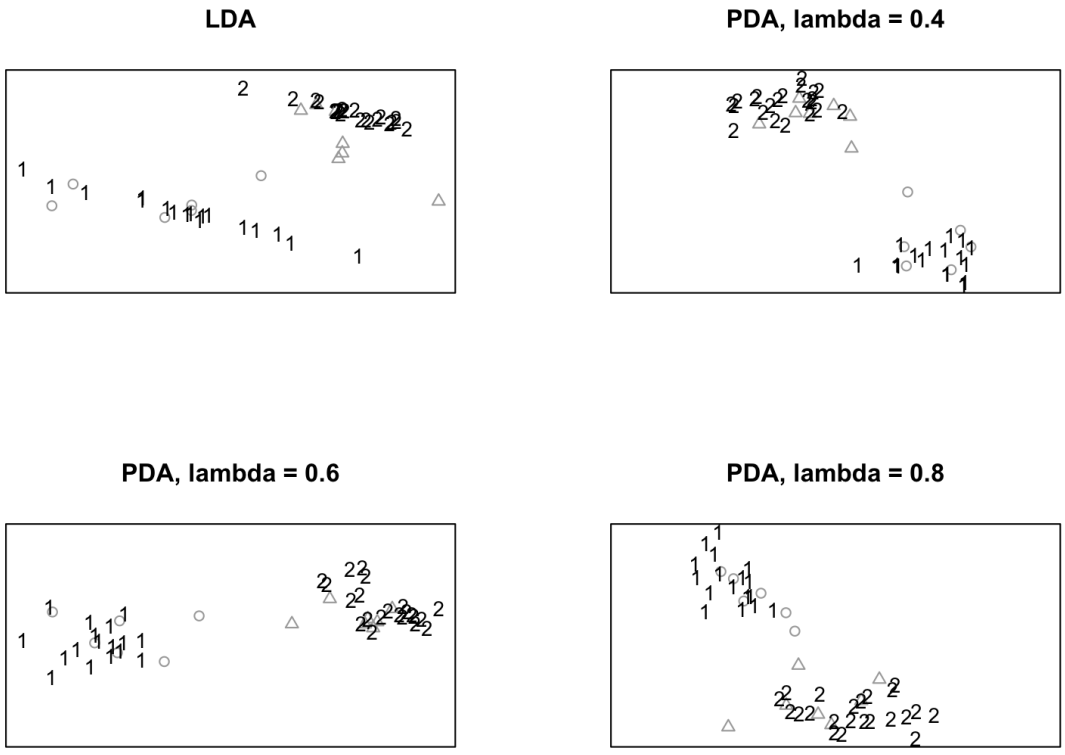

Figure 11: Projection pursuit on the music data for genre classification between rock music and classical music. The accuracy scores for LDA indexes and PDA indexes with $\lambda \in \{0.4, 0.6, 0.8\}$ are all 100%. Numbers in the plot represent the training set classes, and symbols in the plot represent the test set classes.

but also be required to learn the differences in artist styles. Both SVM and PDA($\lambda \in \{0.6, 0.8\}$) achieve a perfect accuracy score of 100% while the LDA index and PDA($\lambda = 0.4$) only give an accuracy score of 69%. According to Figure 12, distinguishing between class 6 and class 3 puzzles the LDA index and PDA with $\lambda = 0.4$, as the projection of the two classes seems to be mixed with each other. For PDA($\lambda \in \{0.6, 0.8\}$), the within-class variance of the training set is much larger than the other two methods, but it manages to differentiate all test classes. It seems like giving more weight to the PDA penalty matrix improves model performance when there are more class labels and fewer samples in each class.

5 Discussion and conclusion

In the statistics literature, relatively few works relevant to performing projection pursuit on high-dimensional data are available. For classification of large p small n data, Lee and Cook (2010) devised and implemented a new projection pursuit index, called the "PDA index", with

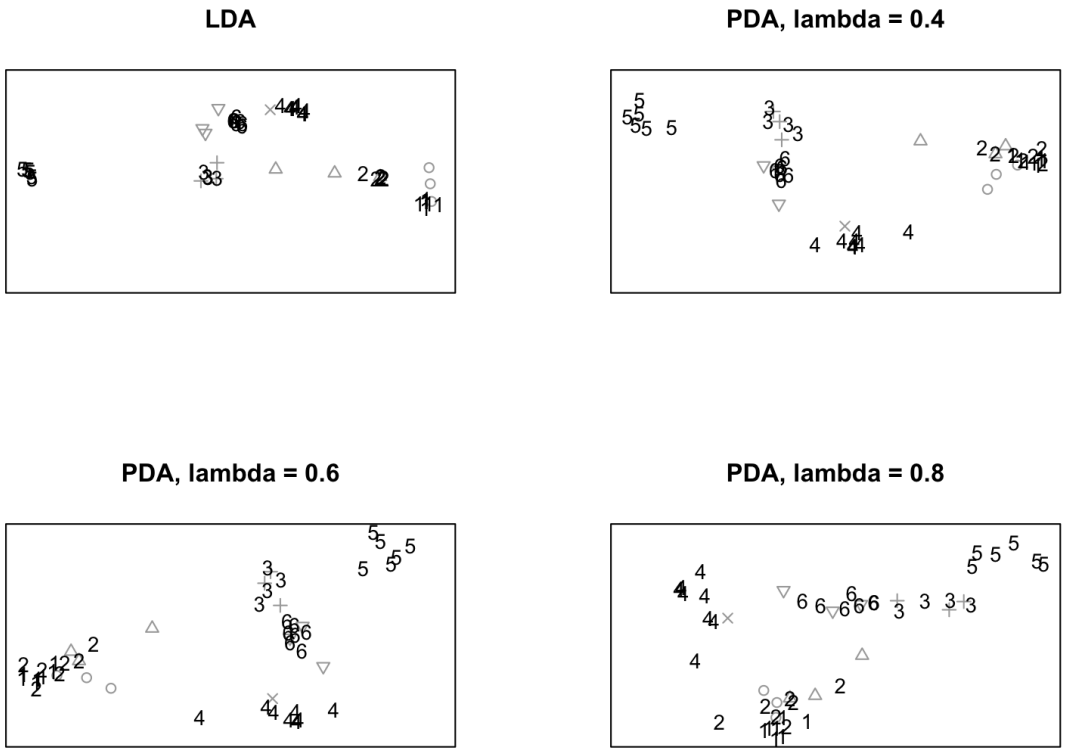

Figure 12: Projection pursuit of the music data for artist classification. The accuracy scores for LDA indexes and PDA indexes with $\lambda \in \{0.4, 0.6, 0.8\}$ are respectively 85%, 91%, 100%, 100%. Numbers in the plot represent the training set classes, and symbols in the plot represent the test set classes.

promising results. Yet, there is little information available about its performance compared with the popular Support Vector Machine (SVM) in supervised classification.

This paper focuses on, for the first time, comparing the performances of the LDA index, PDA index, and SVM (using two different kernels). As projection pursuit indexes, the LDA index and PDA index are mainly applied for classification tasks and dimension reduction and provide a visually appealing way to project high-dimensional data. As a supervised learning method, SVM has been popularly used for classification. From the simulated and empirical evaluations, the PDA index is able to outperform the other two methods in most cases. In practice, it is recommended to apply the PDA index under the "large p small n" setting because PDA could better deal with the additional penalty matrix to avoid the data piling issue. Users may apply the PDA index for visual inspection of projected data and meanwhile train the SVM for further comparison. On the other hand, it is not ideal to apply the LDA index which suffers from the data piling issue on high dimensional data.

According to the simulation results, the PDA index especially outperforms other methods on datasets with certain characteristics. As long as there are some distinguishing features for

different class labels, the PDA index gives perfect performance regardless of the percentage of important variables, meaning that adding or deleting more important variables from the data does not affect the PDA performance. One potential application is gene expression analysis where the number of genes for a given sample is substantial, but only few genes are contributing to class prediction. For the different ratios n/p of the number of observations to dimensions, the PDA index gives excellent and stable performance except when the ratio is below 3%. This is particularly helpful for biologists to analyze microarray data where information about gene expression profiles is massive but only limited samples are available. Even if the ratio is below 3%, the PDA index still has performance superior or comparable to the other two methods. Furthermore, the PDA index is robust to outliers in the simulation studies, suggesting that it avoids learning statistical noise by generalized training. Data scientists and statisticians are encouraged to apply this method to their own datasets in practice without worrying about noisy data. Even though the PDA index suffers from inaccuracy when dealing with a larger number of classes, it still has better performance than the LDA index and SVM. To classify a large number of class labels, users may apply the PDA index and some other classification algorithms to make a comparison and choose the best method for their needs.

To explore the performances of projection pursuit indexes for classifying non-Gaussian data, we randomly generate a bimodal dataset ($n = 40$, $p = 500$, $k = 2$), where unimportant variables follow the mixture Gaussian distribution $0.5\mathcal{N}(-2, 1/4) + 0.5\mathcal{N}(2, 1/4)$, while the distribution of class 1 important variables is $0.5\mathcal{N}(0, 1/4) + 0.5\mathcal{N}(4, 1/4)$ and the distribution of class 2 important variables is $0.5\mathcal{N}(-4, 1/4) + 0.5\mathcal{N}(0, 1/4)$. Both SVM and PDA achieved an accuracy score of more than 90%, but LDA only reached around 65% accuracy. From this exploratory study, both SVM and PDA are preferable to LDA in handling bimodal data. For various other types of non-Gaussian data points, a systematic comparison of different projection pursuit indices could be an interesting future work.

Overall, it is recommended for researchers to apply the PDA index in classifying high-dimensional datasets due to its robustness to outliers and high accuracy score under a wide range of circumstances. It might not be the optimal method in all cases but can be implemented as an alternative to the SVM. Both statisticians and other data-driven researchers may use the package *classPP* and the helper methods introduced in this paper to draw provoking insights from their own "large p small n" datasets.

Supplementary Material

All of our code is open source in the following GitHub repository `https://github.com/zwu363/projection-pursuit-index`.

Acknowledgment

We thank the Co-Editor Michael Grosskopf and a reviewer for insightful comments. C. Zhang's work was partially supported by U.S. National Science Foundation grants DMS-2013486 and DMS-1712418, and provided by the University of Wisconsin-Madison Office of the Vice Chancellor for Research and Graduate Education with funding from the Wisconsin Alumni Research Foundation.

References

Burczynski ME, Peterson RL, Twine NC, Zuberek KA, Brodeur BJ, Casciotti L, et al. (2006). Molecular classification of crohn's disease and ulcerative colitis patients using transcriptional profiles in peripheral blood mononuclear cells. *The Journal of Molecular Diagnostics*, 8(1): 51–61.

Cortes C, Vapnik V (1995). Support-vector networks. *Machine Learning*, 20: 273–297.

Friedman J, Tukey J (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C-23(9): 881–890.

Gaudette L, Japkowicz N (2009). Evaluation methods for ordinal classification. In: *Advances in Artificial Intelligence* (Y Gao, N Japkowicz, eds.), 207–210. Springer Berlin Heidelberg, Berlin, Heidelberg.

Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286 5439: 531–7.

Gordon GJG, Jensen RVR, Hsiao LLL, Gullans SRS, Blumenstock JEJ, Ramaswamy SS, et al. (2002). Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Research*, 62(17): 4963–4967.

Hastie TJ, Tibshirani R, Buja A (1994). Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association*, 89: 1255–1270.

Kruskal JB (1969). Toward a practical method which helps uncover the structure of a set of multivariate observations by finding the linear transformation which optimizes a new "index of condensation. In: *Statistical Computation* (RC Milton, JA Nelder, eds.), 427–440. Academic Press.

Lee EK, Cook D (2010). A projection pursuit index for large p small n data. *Statistics and Computing*, 20(3): 381–392.

Lee EK, Cook D, Klinke S, Lumley T (2005). Projection pursuit for exploratory supervised classification. *Journal of Computational and Graphical Statistics*, 14(4): 831–846.

Marron JS (2015). Distance weighted discrimination. *WIREs Computational Statistics*, 7: 109–114.

Nakayama R, Nemoto T, Takahashi H, Ohta T, Kawai A, Seki K, et al. (2007). Gene expression analysis of soft tissue sarcomas: characterization and reclassification of malignant fibrous histiocytoma. *Nature*, 20(7): 749–759.

Pomeroy SL, Tamayo P, Gaasenbeek M, Sturla LM, Angelo M, McLaughlin ME, et al. (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415(6870): 436–442.

Singh D, Febbo PG, Ross K, Jackson DG, Manola J, Ladd C, et al. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2): 203–209.

Sørlie T, Perou CM, Tibshirani R, Aas T, Geisler S, Johnsen H, et al. (2001). Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences*, 98: 10869–10874.

Yeoh EJ, Ross ME, Shurtleff SA, Williams WK, Patel D, Mahfouz R, et al. (2002). Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1(2): 133–143.

Zhang C, Ye J, Wang X (2022). A computational perspective on projection pursuit in high dimensions: feasible or infeasible feature extraction. *International Statistical Review*.