Linear Dimensionality Reduction Methods for Prediction Purposes: Comparisons and Applications on Continuous and Binary Response Data

Giang Vu
University of Pittsburgh
STAT 2221 - Fall 2021
gtv4@pitt.edu

Abstract

Principle component analysis (PCA) and Linear discriminant analysis (LDA) are commonly used methods for linear dimensionality reduction of data sets with high dimension. They both find linear combinations of the variables in the original high-dimensional data and transform them into a low-dimensional space but still retain the characteristics of the original variables. As a result, these methods are utilized for prediction purposes to shrink the dimensionality of predictor variables before doing linear regression with a continuous response (PCA) or classification with a categorical response (LDA). This paper explores two alternative methods, partial least squares (PLS), and supervised PCA (Barshan), in the literature designed specifically with linear dimensionality reduction for predictive purposes in mind, applies them to simulated and real datasets, and compares their performance with classical PCA and LDA. For continuous response, PLS outperforms PCA and Barshan, while for binary response, no method seems to be consistently better than the others.

1 Objective

High dimensional data have become the norm for many fields of science. Carrying out data analysis for this type of data requires some pre-processing step to reduce the dimensionality, making them easier to work with. For regression and classification purposes, principal component analysis (PCA) and linear discriminant analysis (LDA), respectively, have been popular methods that a statistician would use to transform complex datasets with many variables into simpler structures. They are often employed as the initial step before regression/classification to reduce dimensionality of the predictors so that the results are calculated with lower computational costs and easier to interpret.

PCA is a widely used dimensionality reduction method for a linear regression problem [7] [8]. It seeks to compute principal components (PC) of the data, usually by eigendecomposition of the data covariance/correlation matrix, and project the data onto the subspace of the first few PCs and ignore the rest. The resulting transformed dataset will have lower dimensions while still containing as much variation from the original data as possible. In the context of prediction purposes, this method referred to as principal component regression (PCR), meaning PCA will be carried out on the predictors right before linear regression. However, since this algorithm does not involve information from the response variable, the PCA result does not necessarily provide the most relevant projection of the predictors for the purpose of prediction.

LDA is another popular dimensionality reduction method, but used mostly in classification problems [4][8]. It looks for linear combinations of predictors that best explain the data, and because it takes into account the difference between classes of the data, which is in fact the categorical response variable in a classification problem, LDA tends to have good prediction performance.

Xu et. al [10] reviewed and compared several supervised linear dimension-reduction methods with an attempt to find better alternatives of PCA for regression purpose, but left the application of such methods for classification purpose for further investigation.

This final project aims to replicate some of the analyses mentioned in [10], namely, the partial least squares (PLS) [5] and supervised PCA (Barshan) [1] methods, applies them on regression problems with continuous responses and compares their predictive performance with PCA. The paper also extends the discussion for classification problems with binary responses, applies the same two methods here and compares their performance with LDA.

This paper is organized as follows. Section 2 is dedicated to reviewing the general theoretical framework and algorithms of the four methods PCR, LDA, PLS, and Barshan.

In section 3, the methods are applied on two simulated data sets, with PCR, PLS, and Barshan for the data with continuous response while LDA, PLS, and Barshan are used for the data with binary response. Comparisons of the methods and limitations of the analyses are also discussed.

Similarly, in section 4, the same analyses are carried out, but on two real data sets, one with a continuous response, and the other with a binary response.

Finally, section 5 serves as the conclusion for the project with some takeaways and further discussions.

2 Literature Review

2.1 Principal Component Analysis/Regression (PCA/PCR)

Introduced in [7] and summarized in [8], PCA is a technique to find a set of linear orthogonal projections of a correlated set of variables, where the projections are ordered by decreasing variances. The goal is to reduce dimensionality of a centered matrix $\mathbf{X} \in \mathbb{R}^{p\times n}$, which contains n observations and p predictor variables, while still retaining variation among the rows of $\mathbf{X}$. This is achieved by finding the eigen-decomposition of the covariance matrix $\mathbf{\Sigma} = \mathbf{X}\mathbf{X}^{\mathbf{T}}$ and transforming the data with the k-leading eigenvectors of $\mathbf{\Sigma}$. We have

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathbf{T}}, \quad \mathbf{U}^{\mathbf{T}}\mathbf{U} = \mathbf{I_p} \tag{1}$$

where $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{\Sigma}$, and each column $\mathbf{u_j}$ of $\mathbf{U}$ contains the eigenvectors of $\mathbf{\Sigma}$, ordered by their corresponding eigenvalues, from largest to smallest. The transformation of $\mathbf{X}$ is done by calculating the first k principal components of $\mathbf{X}$.

$$\mathbf{z}_j = \mathbf{u}_j^T\mathbf{X}, \quad j = 1, 2, \ldots, k. \tag{2}$$

In the end, the matrix $\mathbf{X}$ has been reduced into a set of k ordered and uncorrelated linear projections of its columns. Linear regression is then carried out with the vectors $\mathbf{z_j}$ as the predictor variables instead of the matrix $\mathbf{X}$, hence the name PCR.

In order to choose the number k of components to keep, there are multiple common methods, such as Kaiser's rules, likelihood ratio testing, scree plot methods, etc. [8]

The algorithm of this method is summarized below [8].

Procedure 1 PCA/PCR

```
Input: $\mathbf{X} \in \mathbb{R}^{p\times n}$
Output: $\mathbf{z_1}, \ldots, \mathbf{z_k}$
  Compute covariance matrix $\mathbf{\Sigma} = \mathbf{X}\mathbf{X}^{\mathbf{T}}$.
  Compute eigen-decomposition of $\mathbf{\Sigma}$.
  Order the eigenvalues from largest to smallest, and get the corresponding $k$ leading eigenvectors $\mathbf{u_1}, \ldots, \mathbf{u_k}$.
  Transform the input data into principle components $\mathbf{z}_j = \mathbf{u}_j^T\mathbf{X}$, $j = 1, 2, \ldots, k$.
```

2.2 Linear Discriminant Analysis (LDA)

LDA is a popular method to reduce dimensionality in the context of a classification problem. The method was built upon Fisher's linear discriminant function, proposed in [4]. Due to time constraint, this paper only discusses and applies LDA on data with a binary response variable.

LDA is similar to PCA in the sense that it also seeks linear combinations of the predictors which best explain the data to shrink the predictors into a smaller matrix. However, unlike PCA, LDA is actually a supervised method, because it uses information from both the predictors and the response (labels of the observations) to construct a classification rule that separates the classes as much as possible [8]. The classification rule, or classifier, is then used to predict the class of new unlabeled observations.

Given a training data set of labeled observations, where $\mathbf{y} \in \mathbb{R}^n$ is the binary response variable that can take on values of 0 for the first class and 1 for the second class, and the matrix $\mathbf{X} \in \mathbb{R}^{p \times n}$ contains p predictor variables, LDA assumes the conditional density functions of $\mathbf{X}$ for the two classes, which are

$$\begin{aligned} P(\mathbf{X} = \mathbf{x}|\mathbf{y} = 0) = f_0(\mathbf{x}) \\ P(\mathbf{X} = \mathbf{x}|\mathbf{y} = 1) = f_1(\mathbf{x}), \end{aligned} \tag{3}$$

both follow the normal (Gaussian) distribution with parameters $(\mu_0, \mathbf{\Sigma}_0)$ and $(\mu_1, \mathbf{\Sigma}_1)$, and under the homogeneity assumption, $\mathbf{\Sigma}_0 = \mathbf{\Sigma}_1 = \mathbf{\Sigma}_{XX}$ [8].

Assuming prior probabilities

$$P(\mathbf{y} = i) = \pi_i, i = 0, 1, \tag{4}$$

the method then looks at the following logarithm of the ratio of the two posterior probabilities.

$$L(\mathbf{x}) = log\left\{\frac{f_0(\mathbf{x})\pi_0}{f_1(\mathbf{x})\pi_1}\right\} = b_0 + \mathbf{b}^T\mathbf{x} \tag{5}$$

where

$$\begin{aligned} b_0 &= -\frac{1}{2}\{\mu_0^T \mathbf{\Sigma}_{XX}^{-1} \mu_0 - \mu_1^T \mathbf{\Sigma}_{XX}^{-1} \mu_1\} + log(\pi_1/\pi_0) \\ \mathbf{b} &= \mathbf{\Sigma}_{XX}^{-1}(\mu_0 - \mu_1) \end{aligned} \tag{6}$$

The classification rule is: if $L(\mathbf{x}) > 0$, assign $\mathbf{x}$ to class 0, otherwise assign $\mathbf{x}$ to class 1.

The linear dimensionality reduction occurs at $\mathbf{b}^T\mathbf{x}$, which is a projection of multidimensional-space pont $\mathbf{x}$ onto a lower dimensional vector $\mathbf{b}$.

Because this method involves the prior probabilities, which contains information of the response variable $\mathbf{y}$, LDA has good predictive performance when applied in classification problem, unlike the unsupervised method PCA.

LDA's algorithm is summarized as follows [8].

```
Procedure 2 LDA
Input: $\mathbf{X} \in \mathbb{R}^{p \times n}, \mathbf{y} \in \mathbb{R}^n, y = 0, 1$
Output: $L(\mathbf{x})$
  Assume conditional densities $N(\mu_0, \mathbf{\Sigma}_0)$ and $N(\mu_1, \mathbf{\Sigma}_1)$.
  Assume prior probabilities $\pi_0$ and $\pi_1$.
  Compute $b_0$ and $\mathbf{b}^T$ in (6).
  $\mathbf{b}^T\mathbf{x}$ is when linear dimension reduction occurs.
  Compute $L(\mathbf{x})$ in (5) to construct a classifier.
```

2.3 Partial Least Squares (PLS)

Partial least squares (PLS) is an iterative method proposed as an alternative to PCA. Instead of just finding projections for the predictors $\mathbf{X} \in \mathbb{R}^{p \times n}$ like PCA, PLS actually projects both the predictors and the response $\mathbf{y} \in \mathbb{R}^n$ to a new lower-dimensional space [5][10].

Within each iteration i, a matrix $\mathbf{C} = \mathbf{X}^i \mathbf{y}^i (\mathbf{y}^i)^T (\mathbf{X}^i)^T$ is computed along with its leading eigenvector $\mathbf{u}^i$. The purpose of this is to maximize the covariance between the response variable and the transformed predictors [10].

After each iteration, the predictors $\mathbf{X}$ and the response $\mathbf{y}$ are both updated for the next iteration $i+1$ as follows.

$$\begin{aligned} (\mathbf{X}^{i+1})^T &= (\mathbf{X}^i)^T - (\mathbf{X}^i)^T \mathbf{u}^i (\mathbf{u}^i)^T \\ \mathbf{y}^{i+1} &= \mathbf{y}^i - \frac{\langle \mathbf{y}^i, (\mathbf{X}^i)^T \mathbf{u}^i \rangle}{||(\mathbf{X}^i)^T \mathbf{u}^i||_2^2} (\mathbf{X}^i)^T \mathbf{u}^i \end{aligned} \tag{7}$$

With a desired lower dimension k, after k iterations, a matrix $\mathbf{U} \in \mathbb{R}^{p \times k}$ can be formed with columns containing all the eigenvectors $\mathbf{u}^i$ in each iteration $i = 1, \dots, k$. This new matrix is then used as the transformation matrix to reduce dimensionality for any new predictor data, while retaining the covariance between the response and the predictors of the training data set.

[10] summarizes the algorithm for PLS as follows.

```
Procedure 3 PLS
Input: $\mathbf{X} \in \mathbb{R}^{p \times n}, \mathbf{y} \in \mathbb{R}^n$
Output: $\mathbf{U} \in \mathbb{R}^{p \times k}, \mathbf{U} = [\mathbf{u}^1, \dots, \mathbf{u}^k]$
  Initiate $\mathbf{X}^1 = \mathbf{X}$ and $\mathbf{y}^1 = \mathbf{y}$
  for $i = 1, \dots, k$ do
      Find the largest eigenvalue of $\mathbf{C} = \mathbf{X}^i \mathbf{y}^i (\mathbf{y}^i)^T (\mathbf{X}^i)^T$ and its corresponding eigenvector $\mathbf{u}^i$.
      Update the predictors and the response using (7).
  end for
  Form the matrix $\mathbf{U} \in \mathbb{R}^{p \times k}, \mathbf{U} = [\mathbf{u}^1, \dots, \mathbf{u}^k]$.
```

2.4 Supervised Principal Component Analysis (Barshan method)

[1] proposed a method of supervised principal component analysis, referred to as the Barshan method for short in this paper, which takes into account both a kernel matrix of the response $\mathbf{y} \in \mathbb{R}^n$ and the predictors $\mathbf{X} \in \mathbb{R}^{p \times n}$ of a centered training dataset.

This method is very similar to the classical PCA reviewed earlier, the only difference is instead of constructing principle components from the leading eigenvectors of the covariance matrix $\mathbf{\Sigma} = \mathbf{X}\mathbf{X}^\mathbf{T}$, the Barshan method computes principle components from the eigen-decomposition of the matrix

$$\mathbf{Q} = \mathbf{X}\mathbf{L}\mathbf{X}^T \tag{8}$$

where $\mathbf{L}$ is the kernel matrix of the response (target variable). For the discussion in this project, a target kernel matrix $\mathbf{L} = \mathbf{y}\mathbf{y}^T$ will be used.

After the k leading eigenvalues of $\mathbf{Q}$ and their corresponding eigenvectors are computed, the rest follows just like standard PCA, where the predictors are transformed with those eigenvectors. In the end, this method reduces dimensionality for the matrix $\mathbf{X}$ while maximizing the covariance between the response $\mathbf{y}$ and the transformed predictors.

Similar to PCA, methods like Kaiser's rules, scree plot, likelihood ratio testing, etc. to choose the desired value for k can also be employed along with Barshan's supervised PCA to find the right number of principal components to retain [8].

Below is an overview of the Barshan method's algorithm [1][10].

Procedure 4 Barshan's Supervised PCA

```
Input: $\mathbf{X} \in \mathbb{R}^{p \times n}, \mathbf{y} \in \mathbb{R}^n$
Output: $\mathbf{z_1}, \dots, \mathbf{z_k}$
  Compute matrix $\mathbf{Q} = \mathbf{X}\mathbf{y}\mathbf{y}^T\mathbf{X}^T$.
  Compute eigen-decomposition of $\mathbf{Q}$.
  Order the eigenvalues from largest to smallest, and get the corresponding $k$ leading eigenvectors $\mathbf{u_1}, \dots, \mathbf{u_k}$.
  Transform the input data into principle components $\mathbf{z}_j = \mathbf{u}_j^T\mathbf{X},\ j = 1, 2, \dots, k$.
```

3 Simulated Data Analysis

3.1 Continuous Response

Data Generation The methods reviewed above were applied to a simulated dataset with a continuous response. The predictors matrix $\mathbf{X}^T \in \mathbb{R}^{500\times 100}$, with 500 samples from 100 predictors, were generated from a multivariate Gaussian distribution $N_{100}(0, \mathbf{\Sigma})$, where Σ is a diagonal matrix containing the all ones in the diagonal, and the correlation between any two predictors is 0.5.

My simulation followed the same linear model from [10]

$$\mathbf{y} = \mathbf{X}\beta + \epsilon = \mathbf{X}\mathbf{\Phi}\alpha + \epsilon \tag{9}$$

where $\alpha \in \mathbb{R}^1 0$ is a vector of ones, and Φ is a 100×10 matrix containing the 10 eigenvectors corresponding to the 10 largest eigenvalues of $\mathbf{X}\mathbf{X}^T$. The coefficient β was constructed in this way so that it is in a subspace that is well aligned with the subspace containing maximal data variation of $\mathbf{X}$ [10]. Gaussian noise $\epsilon \sim N(0, 0.5)$ was also added to construct the continuous response variable $\mathbf{y} \in \mathbb{R}^{500\times 1}$.

Before the application of the methods, 75% of the data were randomly selected to be the training data set, while the rest became the test set. Then, PCR, PLS, and the Barshan methods were applied on the training data, and the mean squared errors (MSEs) of each algorithm for both the training and testing sets were recorded at each value for number of kept components $k = 1, \dots, 100$.

Results and Comparisons Figure 1 and Figure 2 contain the plots of Training MSE and Test MSE against the number of retained principle components, with the red color corresponding to PCR, green depicting PLS, and blue for Barshan method.

For all three methods, as the number k increases, both the Training MSE and Test MSE decrease, which makes sense since the more components were kept, the more overfitting my model became. PLS was the method that performed better (had lower MSEs) than the others, which agrees with the result in [10].

Limitations One interesting thing that happened is the result for Barshan and PCR are basically the same. This only happened with this simulated dataset, and did not appear again with the real dataset in the next section. Neither [1] nor [10] mention this phenomenon in their papers. There is some discussion about how PCR and Barshan's method are connected, in the sense that PCR is a special case of Barshan if we set the matrix $\mathbf{L}$ in (8) as the identity matrix $\mathbf{I}$ [1]. However, this was not the case for my simulated dataset, so this is one limitation of this project's analysis.

Another limitation is that due to time constraint, the project's analysis could only explore the methods with one simulated dataset, where the covariance matrix $\mathbf{\Sigma} = \mathbf{X}\mathbf{X}^\mathbf{T}$ has fast decaying eigenvalues, as shown in Figure 3 below. How fast the eigenvalues of $\mathbf{\Sigma}$ decay could potentially affect the performance of the methods discussed here, especially with classical PCA. As shown in [10], PLS still had the best predictive performance regardless of eigenvalues decaying speed.

Moreover, this paper only focused the analysis on the type of simulated data structure where the subspace containing the most variation in the predictors $\mathbf{X}$ and the subspace where the linear model coefficient β is in are well-aligned as described above. [10] also looked at different scenarios of alignment for these subspaces, and they did show impact on the performance of the discussed methods. Classic PCR, in particular, had worse predictive performance if the subspaces were not well aligned,

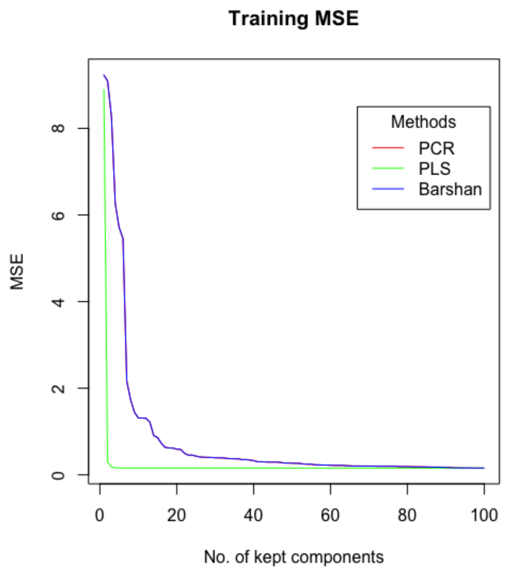

Figure 1: Training MSE vs. subspace dimension for Simulated data

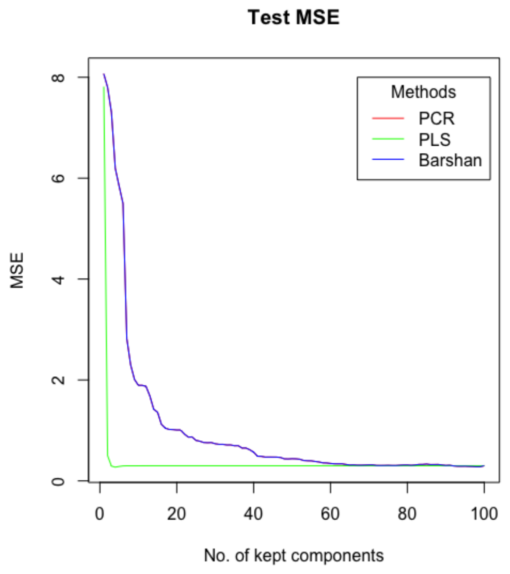

Figure 2: Testing MSE vs. subspace dimension for Simulated data

which made clearer the motivation of this entire discussion, i.e., PCR does not guarantee a resulting shrunken predictors matrix that contains the most relevant information for a regression problem [10].

3.2 Binary Response

Data Generation To obtain a simulated dataset with a binary response, I followed the same process to generate a predictors matrix $(\mathbf{X})^T \in \mathbb{R}^{500\times 50}$, with 500 samples from 50 predictors, which

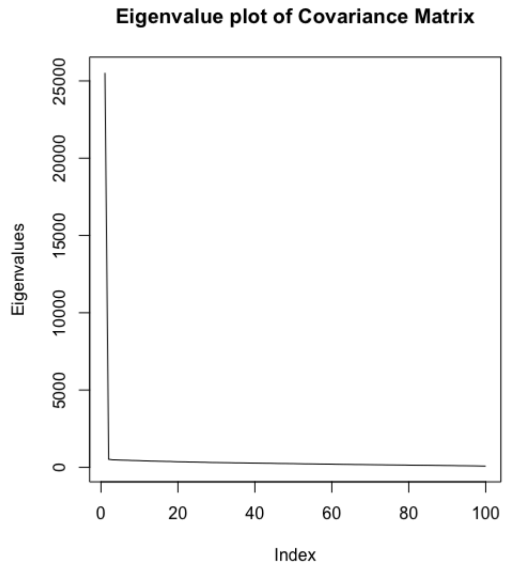

Figure 3: Eigenvalues of the covariance matrix $\mathbf{\Sigma}$ for simulated data

were generated from a multivariate Gaussian distribution $N_{50}(0, \mathbf{\Sigma})$, where Σ is a diagonal matrix containing the all ones in the diagonal, and the correlation between any two predictors is 0.5.

The simulated response was generated via this model below

$$\mathbf{Z} = \mathbf{e} + \mathbf{\Sigma_{i=1}^{50} i x_i^T} \tag{10}$$

where $\mathbf{e}$ is a 500×1 vector of all ones, and $\mathbf{x_i^T}$ contains the i-th predictor. The goal was to create a vector $\mathbf{Z} \in \mathbb{R}^{500}$ that is a linear combination of all the predictors with a bias.

Then this vector $\mathbf{Z}$ was fed into an expit function to convert into probabilities, and then the binary response vector/class labels $\mathbf{y} \in \mathbb{R}^{500 \times 1}$ were constructed, where $\mathbf{y}_i = 1$ if the probability in the i-th row of $\mathbf{Z}$ is greater than 0.5, and 0 otherwise.

Similarly to the previous simulation, 75% of the data were extracted to become the training set, while the rest were the testing set before LDA, PLS, and the Barshan methods were applied on the training data. For the PLS and Barshan methods with this classification problem, a value for the number of retained components k was selected via the screeplot method, before the misclassification rate of both training and test sets were computed for all three methods.

Results and Comparisons From Figure 4 below, which is a validation scree plot or RMSEP as function of the number of components after fitting a PLS model, I chose to keep only the first component based on the "elbow" rule.

With that number of components to keep decided, the misclassification rate of the three methods for both training data and testing data are presented in Table 1. Out of the three models, Barshan performed the best when it comes to prediction, with only 3.2% misclassification rate for the test set, followed by PLS at 4%, while LDA had the worst performance at 7.2%. Overall the performance of the methods were relatively close to each other as I suspected since LDA is itself a supervised method. There was not a clear discrepancy like in the first simulation with a continuous response, where PCA was clearly outperformed by the other methods.

Limitations Discussion of the PLS and Barshan methods in this paper is purely applied, as an general exploration rather than from a rigorous theoretical point of view. As a result, the PLS method was applied to the data with binary response but it treated the binary response as a continuous variable of numerical values 0 and 1. The Barshan method, on the other hand, was applied to the data before logistic regression was carried out with the new shrunken predictors matrix. For prediction, both

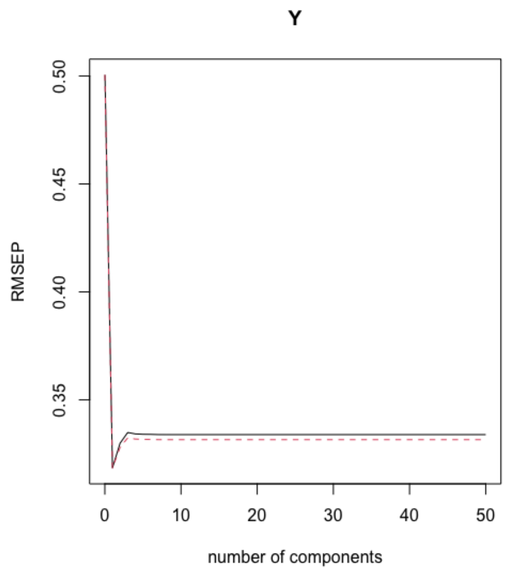

Figure 4: Validation plot post-PLS to determine k for simulated data

Table 1: Summary of simulated data results. Listed are sample size (n), number of predictors (r), number of classes (K), misclassifcation rates for linear discriminant analysis (LDA), partial least squares (PLS) and Barshan method (Barshan).

Data Set	n	r	K	LDA	PLS	Barshan
Training set	375	50	2	0.045	0.032	0.035
Test set	125	50	2	0.072	0.040	0.032

methods employed the basic rule that if predicted probability for an observation is greater than 0.5, then it belongs to class 1, otherwise it belongs to class 0. The results of the analysis might not be rigorous and accurate.

Another limitation is that the number of components to retain was chosen from the PLS model only, but it was used in both the PLS and Barshan methods, and due to the way the data was simulated, this dataset had a fast decaying rate for the eigenvalues of the covariance matrix as well. Most of the variation of the predictors can be explained by the very first component, leading to the choice of $k = 1$.

4 Real Data Analysis

4.1 Continuous Response

Data Description The data set "Boston" in the package *MASS* was used for this paper's analysis [2][6][9]. This data contains 506 samples of median property value in Boston, along with 14 columns of predictor variables including crime rate, average number of rooms, age of property, distances to employment centers, etc. The project's goal was to apply the methods on these 14 predictors and reduce their dimensionality before doing a linear regression to predict median property value in Boston.

Similar to previous analyses, 75% of the data were picked randomly to be the training data set, while the rest were the test set. Then, PCR, PLS, and Barshan methods were applied on the training set before a regression model was fitted and MSE was recorded for each method at each number of retained components $k = 1, \ldots, 13$

Results and Comparisons Figure 5 and Figure 6 contain the plots of Training MSE and Test MSE versus the number of kept components, with the red line depicting PCR, green for PLS, and blue for Barshan method.

The analysis showed the same result as the analysis on the simulated data. PLS performed the best in terms of having a low MSE at a low value of k, followed by Barshan method, and the worst performer was the classical PCR.

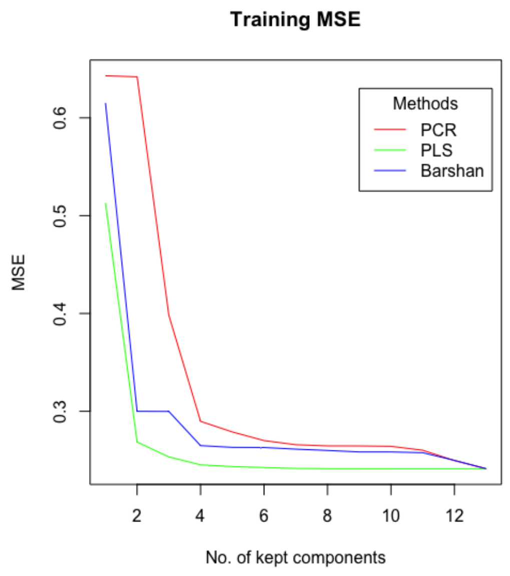

Figure 5: Training MSE vs. subspace dimension for Boston housing data

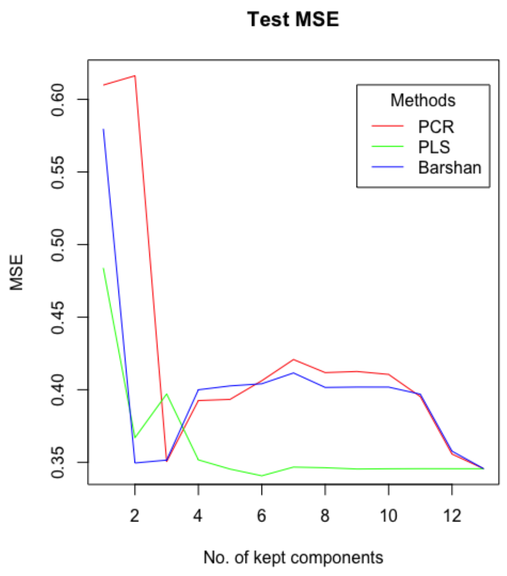

Figure 6: Testing MSE vs. subspace dimension for Boston housing data

Limitations The phenomenon of Barshan and PCR overlapping did not appear here like it did with simulated data. With this Boston dataset, there's a clear distinction between the three discussed methods, with PCR being outperformed by the other two. This could potentially point to an issue with the data simulation process previously, and further analysis is needed to explore this phenomenon better.

Another limitation here is, again, about the decaying speed of the covariance matrix's eigenvalues. The Boston data set is similar to the simulated data when it comes to how fast the eigenvalues of $\mathbf{\Sigma} = \mathbf{X}\mathbf{X}^{\mathbf{T}}$ decay. From Figure 7 below, it is clear that this data set does fall into the fast decaying scenario, with the first few eigenvalues having significantly larger values compared to the rest. As discussed earlier, this could impact predictive performance of the methods applied in this paper [10].

4.2 Binary Response

Data Description For a classification with binary response, the data set "Ionosphere" from UCI Machine Learning Repository was used [3][8]. The data contain 34 predictors that are 34 attributes for 17 pulse numbers (2 attribute per pulse) to describe electromagnetic signals collected in Goose Bay, Labrador. The response is a binary variable with classes "Good" and "Bad" to assess if the signal shows evidence of structure or no structure in the ionosphere, respectively.

One predictor was removed before the analysis because it is constant for all of 351 observations. Then, 75% of the data randomly became the training data while the remaining became the test data. The three methods, LDA, PLS, and Barshan were applied to the training set, and similar to what was done with the simulated data, a value of k was selected via screeplot for PLS and Barshan. Finally the misclassification rate for both the training set and test set were computed and compared between methods.

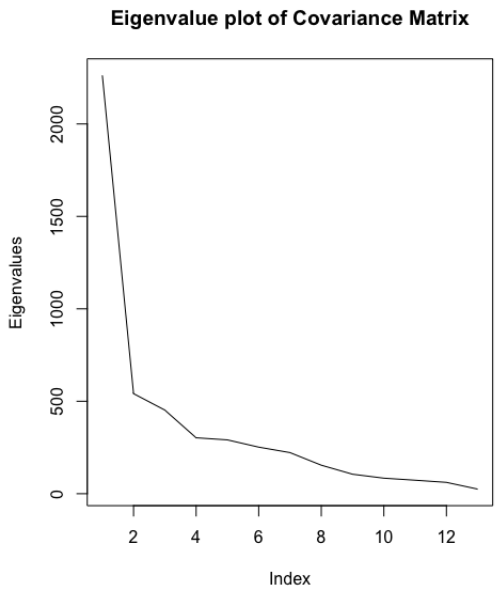

Figure 7: Eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$ for Boston housing data

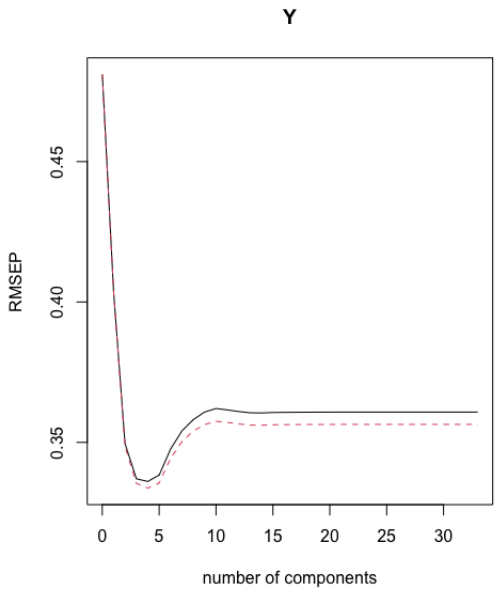

Figure 8: Validation plot post-PLS to determine k for Ionosphere data

Results and Comparisons From Figure 8, which is a screeplot of RMSEP versus number of components after fitting a PLS model, using the second elbow rule, a value $k = 4$ was chosen.

Table 2 contains the misclassification rate of the three methods for both the training data and the testing data. Unlike with the simulated data analysis above, this time LDA actually had a lower error rate than the other two. PLS did a bit better than Barshan for prediction with the test set, but both were outperformed by LDA. Overall, the predictive performance of the three methods were still quite close.

This result was not too surprising considering LDA is a supervised method, so I did not expect it to have poor predictive performance like PCA/PCR.

Table 2: Summary of Ionosphere data results. Listed are sample size (n), number of predictors (r), number of classes (K), misclassifcation rates for linear discriminant analysis (LDA), partial least squares (PLS) and Barshan method (Barshan).

Data Set	n	r	K	LDA	PLS	Barshan
Training set	263	33	2	0.091	0.106	0.106
Test set	88	33	2	0.159	0.170	0.193

Limitations Again, as discussed in previous simulated analysis, the discussion here is purely applied. PLS was applied to the data with binary response but treating it like a continuous response, and Barshan was applied to reduce dimensionality of the predictors before a logistic regression model was fitted. The same rule that if an observation's predicted probability is greater than 0.5 then it is class "Good", otherwise it belongs to class "Bad" was used here. And the value $k = 4$ was chosen based on the PLS model for both the PLS and Barshan methods. This analysis could be improved with a more thorough theoretical investigation of how to apply these methods to binary response data sets.

5 Conclusion

The paper has reviewed the general algorithms of four different methods for the same purpose of reducing dimensionality before prediction from a myriad of techniques in the literature [10]. The analyses carried out in [10] on data with a continuous response were replicated in this paper for both a simulated and a real data set. Furthermore, I also extended the discussion into data with a binary response as well, albeit naively and in a basic manner. The results showed that PCA/PCR does not have good predictive performance compared to other supervised methods like PLS and Barshan, while LDA, a supervised method itself, has performance on par with the other methods.

The analysis of this paper contains multiple limitations, since it only focused on linear dimension-reduction techniques, as well as the case of only one response variable. The applications of the methods were also not comprehensive and there were strange phenomena happening with the simulation process like discussed above.

Supplementary Materials

Please refer to the compressed .zip folder submitted along with the paper for more details of the analyses.

- *FinalProjectCode.Rmd* contains the R markdown code for the analyses and references of packages used
- *functions.R* contains the user-defined functions used in the markdown file
- *FinalProjectCode.html* contains the analyses result along with the code presented in a .html file that can be viewed in a Web browser.

References

[1] Barshan, E. et. al (2011) Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition* **44**(7): 1357-1371.

[2] Belsley D.A., Kuh, E. & Welsch, R.E. (1980) *Regression Diagnostics. Identifying Influential Data and Sources of Collinearity*. New York: Wiley.

[3] Dua, D. & Graff, C. (2019). Johns Hopkins University Ionosphere database. *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[4] Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. *Annals of Eugenics* **7**(2):179-188.

[5] Geladi, P. & Kowalski, B. R. (1986) Partial least-squares regression: a tutorial. *Analytica chimica acta* 185:1-17.

[6] Harrison, D. & Rubinfeld, D.L. (1978) Hedonic prices and the demand for clean air. *J. Environ. Economics and Management* **5**:81-102.

[7] Hotelling, H. (1933) Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* **24**(6):417-441.

[8] Izenman, A. J. (2013) *Modern Multivariate Statistical Techniques*. Springer Texts in Statistics, DOI 10.1007/978-0-387-78189-1_8, Springer Science+Business Media New York.

[9] Venables, W. N. & Ripley, B. D. (2002) *Modern Applied Statistics with S. Fourth Edition*. New York: Springer.

[10] Xu, S. et. al (2021) Supervised Linear Dimension-Reduction Methods: Review, Extensions, and Comparisons. *arXiv preprint arXiv: 2109.04244v1*.