Applying Principal Component Analysis to Regression and Optimization

Cassandra Overney

December 10, 2018

Abstract

Principal Component Analysis (PCA) is a fascinating method, which has strong connections to linear algebra concepts, such as singular value decomposition. This report serves to emphasize the power of PCA by delving into some of its many applications, including dimensionality reduction, clustering, and principal component regression (PCR). Furthermore, two new perspectives on solving PCA as optimization problems are explained.

1 Overview of PCA

Before attempting to apply Principal Component Analysis (PCA) to regression or optimization problems, it is important to have a clear understanding of how it works. PCA is a versatile data analysis tool that can be used to extract meaningful information from complex datasets. It is typically conducted to dimensionally reduce input data by finding linear combinations of variables that correspond to the principal components. A dimensionally reduced dataset can then be utilized for visualization (i.e. creating 2d and 3d scatter plots) and regression. This section serves to provide a short overview of how PCA works and its underlying math.

1.1 Summary of Steps

The steps required to perform PCA are the following (a python tutorial is available **here**):

1. Obtain the data (m variables and n samples) and structure into a matrix of size $m \times n$ or $n \times m$
2. Standardize the data (center rows by subtracting means and then divide by standard deviations)
3. Compute the covariance matrix, C
4. Find the eigenvalues and eigenvectors of C
5. Interpret the principal components (eigenvectors) and the amount of variance they each explain (eigenvalues)

1.2 Math

For consistency purposes, PCA will be done on a data matrix, X, of size $n \times m$ with n samples and m variables. The vectors $\vec{x}_i$ are row vectors corresponding to the data for the ith sample. The process begins by standardizing X. The most basic method for standardization is to re-center the data such that the mean of each variable is zero. This can be accomplished by subtracting the mean vector, $\vec{\mu}$, from each row vector in X. $\vec{\mu}$ is defined by the following equation

$$\vec{\mu} = \frac{1}{n}(\vec{x_1} + \cdots + \vec{x_n}) \tag{1}$$

Typically, further standardization is completed by dividing the elements in each column of the centered matrix by its standard deviation, resulting in z-scores for each data point. Computing z-scores is especially useful when X consists of variables measured on vastly different scales.

After standardizing X, the covariance matrix, C, can be found as follows

$$C = \frac{1}{n-1} X^T X \tag{2}$$

C is a $m \times m$ matrix where the ijth entry indicates the covariance between variables i and j. It is important to note that C is symmetric and its diagonal entries indicate the variance for each variable. The definitions for variance and covariance can be found in the appendix (5.1). Furthermore, if z-scores are utilized in the standardization, then C is also the correlation matrix.

The next step is to find the eigenvectors and corresponding eigenvalues of the covariance matrix. Since C is symmetric, its eigenvectors form an orthogonal eigenbasis that is useful for data reduction. These eigenvectors are known as the principal components, and their relative importance is given by their eigenvalues. In other words, the principal component $\vec{v}_i$ accounts for λ_i of the total variance. The eigenvector associated with the largest eigenvalue, $\vec{v}_1$, points in the direction of the dataset where the most variation occurs. The second principal component points in the direction of most variation left over after considering $\vec{v}_1$.

1.3 Connections to SVD

PCA can be seen as a specific application of Singular Value Decomposition (SVD). SVD involves breaking a matrix into simpler linear combinations that can reveal interesting patterns within a dataset. The SVD of X is

$$X = U \Sigma V^T \tag{3}$$

In this factorization equation, U and V contain the left and right singular vectors of X, while Σ contains the singular values. More specifically, the columns of U $(n \times n)$ are the orthonormalized eigenvectors of DD^T, while the columns of V $(m \times m)$ are the orthonormalized eigenvectors of $X^T X$. Σ is a diagonal matrix $(n \times m)$ with nonzero components obtained by taking the square roots of the eigenvalues of U and V.

The connection between PCA and SVD can be revealed by the spectral decomposition of the covariance matrix.

$$C = \frac{1}{n-1} X^T X = \frac{1}{n-1} (U \Sigma V^T)^T U \Sigma V^T = \frac{1}{n-1} V \Sigma^T U^T U \Sigma V^T = \frac{1}{n-1} V \Sigma^T \Sigma V^T \tag{4}$$

The above equation serves to re-write C into SVD form. In this case, V is the same as U, and Σ is equal to $\Sigma^T \Sigma$. Thus, the eigenvectors of C correspond to the columns of V, which are also the right singular vectors of X, and the eigenvalues of C are the m components of $\Sigma^T \Sigma$, which are identical to the squares of the singular values of $\frac{1}{n-1} X$.

2 Uses of PCA

PCA has a wide range of applications in data analysis. In this section, a few of them will be highlighted using a single dataset. The dataset, collected during road tests by Motor Trend magazine in 1974, consists of 11 variables quantifying various aspects of 32 automobiles (4). It is commonly referred to as the mtcars dataset and is widely used to understand PCA.

The columns of X correspond to particular car characteristics, while the rows correspond to each automobile. After standardizing X and conducting PCA, the principal components $(\vec{v}_1, \cdots, \vec{v}_m)$ can be arranged as columns in a $m \times m$ loadings matrix, V.

$$V = \begin{bmatrix} 0.36 & 0.02 & -0.23 & -0.02 & 0.1 & -0.11 & 0.37 & 0.75 & -0.24 & 0.14 & 0.12 \\ -0.37 & 0.04 & -0.18 & -0. & 0.06 & 0.17 & 0.06 & 0.23 & -0.05 & -0.85 & 0.14 \\ -0.37 & -0.05 & -0.06 & 0.26 & 0.39 & -0.34 & 0.21 & -0. & -0.2 & 0.05 & -0.66 \\ -0.33 & 0.25 & 0.14 & -0.07 & 0.54 & 0.07 & -0. & 0.22 & 0.58 & 0.25 & 0.26 \\ 0.29 & 0.27 & 0.16 & 0.85 & 0.08 & 0.24 & 0.02 & -0.03 & 0.05 & -0.1 & 0.04 \\ -0.35 & -0.14 & 0.34 & 0.25 & -0.08 & -0.46 & -0.02 & 0.01 & -0.36 & 0.09 & 0.57 \\ 0.2 & -0.46 & 0.4 & 0.07 & -0.16 & -0.33 & 0.05 & 0.23 & 0.53 & -0.27 & -0.18 \\ 0.31 & -0.23 & 0.43 & -0.21 & 0.6 & 0.19 & -0.27 & -0.03 & -0.36 & -0.16 & -0.01 \\ 0.23 & 0.43 & -0.21 & -0.03 & 0.09 & -0.57 & -0.59 & 0.06 & 0.05 & -0.18 & -0.03 \\ 0.21 & 0.46 & 0.29 & -0.26 & 0.05 & -0.24 & 0.61 & -0.34 & 0. & -0.21 & 0.05 \\ -0.21 & 0.41 & 0.53 & -0.13 & -0.36 & 0.18 & -0.17 & 0.4 & -0.17 & 0.07 & -0.32 \end{bmatrix}$$

The ith principal component can be defined as a linear function f(x) = $\vec{v}_i^T x$. Additionally, V can be interpreted as a rotation matrix with entries being the cosines of the angles formed by its original basis vectors and the loading vectors. Table 1 depicts this relationship for the first principal component.

Variable	v_1
Miles/Gallon	**0.36**
# Cylinders	**-0.37**
Displacement	**-0.37**
Horsepower	**-0.33**
Rear Axle Ratio	0.29
Weight	**-0.35**
1/4 Mile Time	0.20
V/S Engine	0.31
Transmission	0.23
# Gears	0.21
# Carburetors	-0.21

Table 1: Loadings of first principal component with most significant values in bold

The first entry in $\vec{v}_1$ is obtained by finding the cosine of the angle between $\vec{v}_1$ and the basis vector associated with miles/gallon. The basis vectors are given by the standard basis in Euclidean space.

$$0.36 = cos(\theta) = \frac{\vec{v}_1 \cdot \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T}{\left|\vec{v}_1\right| \times 1}$$

Another output of PCA includes the $n \times m$ scores matrix, Z, which contains the values of the principal component functions evaluated at each observation in X.

$$Z = XV \tag{5}$$

The scores on the first principal component ($\vec{z}_1$) are given in Table 2.

Car Model	z_1
Mazda RX4	0.66
Mazda RX4 Wag	0.63
Datsun 710	2.78
Hornet 4 Drive	0.31
Hornet Sportabout	-1.97
Valiant	0.06
Duster 360	-3.00
Merc 240D	2.06
Merc 230	2.29
Merc 280	0.53
Merc 280C	0.51
Merc 450SE	-2.25
Merc 450SL	-2.05
Merc 450SLC	-2.15
Cadillac Fleetwood	**-3.90**
Lincoln Continental	**-3.95**
Chrysler Imperial	**-3.59**
Fiat 128	**3.86**
Honda Civic	**4.25**
Toyota Corolla	**4.23**
Toyota Corona	1.90
Dodge Challenger	-2.18
AMC Javelin	-1.86
Camaro Z28	-2.89
Pontiac Firebird	-2.25
Fiat X1-9	3.57
Porsche 914-2	2.65
Lotus Europa	3.39
Ford Pantera L	-1.37
Ferrari Dino	0.00
Maserati Bora	-2.67
Volvo 142E	2.42

Table 2: Scores on first principal component with most significant values in bold

The values in Table 2 can be used to rank the automobiles. Cars with the lowest negative scores tend to have the most power and worst fuel economy, while cars with the highest positive scores tend to be smaller and have better gas mileage. The variables with the largest impact on the ranking system is determined by examining the values of the first principal component in Table 1. The variable with the most positive loading is miles per gallon, which is an attribute that less powerful cars typically have. The variables with the most negative loadings are cylinder number, displacement (volume of all cylinders in the engine), horsepower, and weight, which correspond to attributes that more powerful cars should have. The first principal component can be associated with **power**. A lower negative score indicates a powerful car, while a higher positive score indicates a smaller and more fuel-efficient car. Additional ranking systems can be established by examining the other principal components.

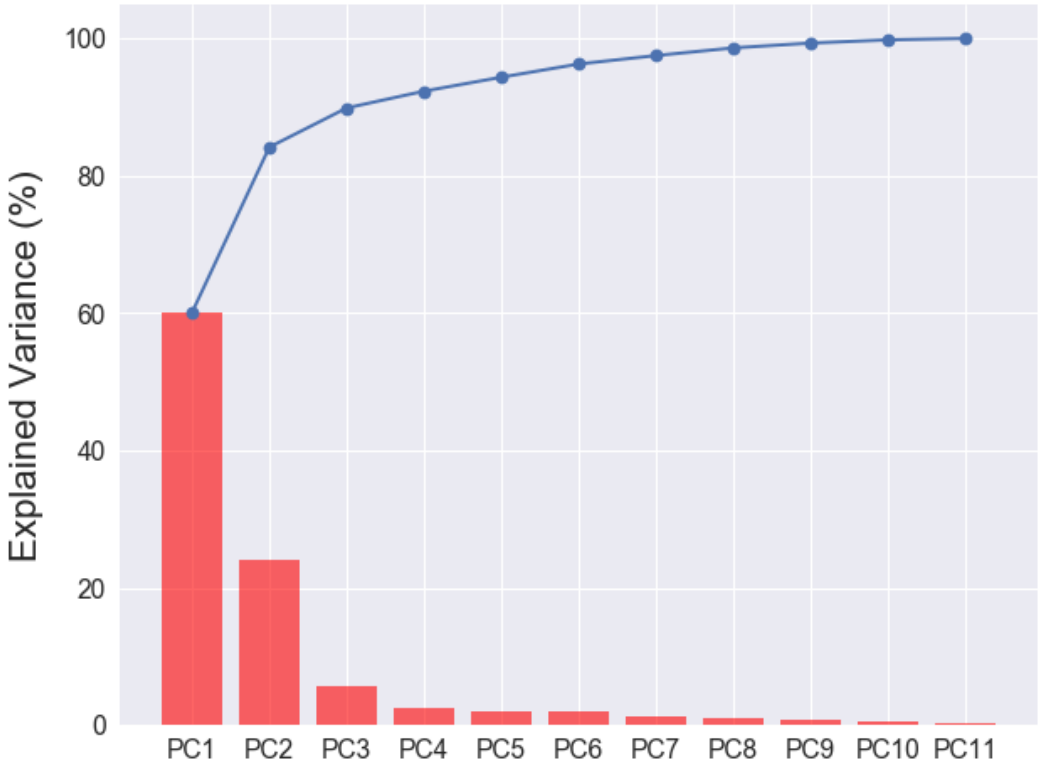

Figure 1: Bars represent variance of each principal component while scatter plot line is cumulative

2.1 Dimensionality Reduction

One of the most popular uses of PCA is dimensionality reduction, which involves using the most dominant principal components to project the data onto smaller dimensions. The mtcars dataset has a high ratio of variables to samples. The ratio can be decreased by creating a set of alternate variables that are linear combinations of the original variables (columns of V). Isolating the first few columns of Z results in a reduced dataset, which can be further analyzed. To determine the minimum number of columns that capture enough variance in the original data, it is important to examine the variance explained by each principal component.

Figure 1 shows the noncumulative and cumulative proportions of variance explained by each principal component. The first principal component captures around 60% of the variance in the data, the second principal component captures around 24%, and the third principal component captures around 6%. By using the first three columns of Z, around 90% of the information present in the original 11 variables is accounted for. Now that the dimensions of the dataset are reduced from 11 to 3, we can visualize relationships to see which cars are more similar to each other.

2.2 Clustering

With the aid of PCA, cluster analysis can be done to determine groups of automobiles with similar features. Figure 2 depicts clusters in the mtcars dataset. The clusters were found by conducting k-means clustering (n_clusters=4) on the first three columns of Z. In the green cluster, Ford Pantera, Maserati, and Ferrari are clustered together, forming a high end, high-performance sports car group. The red cluster contains a group of smaller cars, including the Toyota Corolla,

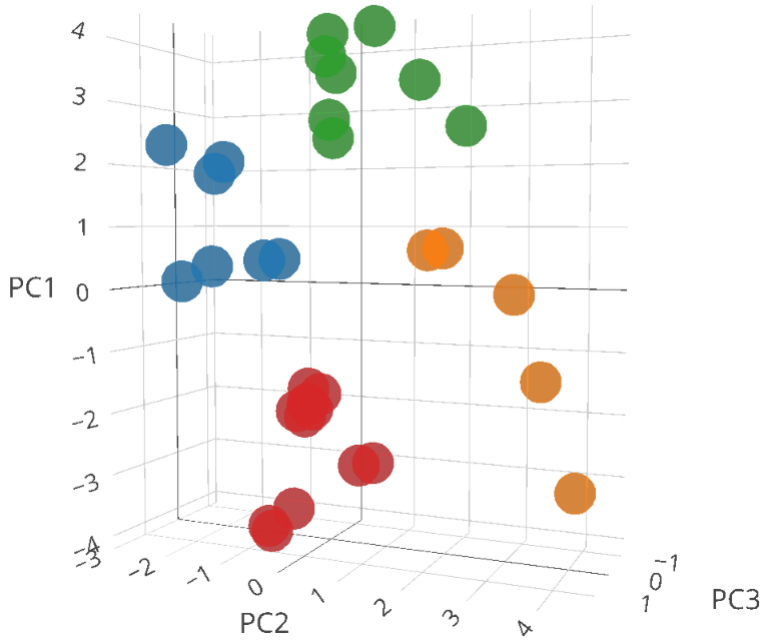

Figure 2: 3D scatter plot of scores for the first three principal components

Honda Civic, and Fiat 128. Using a dimensionally reduced dataset from PCA is a common way to validate and visualize cluster analysis.

2.3 Regression

PCA can also be used to improve multiple linear regression (aka least squares regression) by removing the issue of multicollinearity. This process is known as principal component regression (PCR). The goal of multiple linear regression is to find an error, e, and regression coefficients, B, for a model of the form

$$Y = XB + e \tag{6}$$

where Y is the dependent variable and X corresponds to the independent variables. Recall that X is $n \times m$ (m is actually m-1 if the dependent variable is taken from the original m variables).

Multicollinearity occurs when two or more independent variables are highly correlated with each other and can lead to unreliable estimates of regression coefficients. Using the dominant principal components as regressors to estimate unknown regression coefficients adds bias to the regression estimates, which reduces the standard error. The goal of PCR is to find an error, e, and coefficients, A, for a model of the form

$$Y = ZA + e \tag{7}$$

where Z is the appropriately reduced matrix of scores.

Eliminating the principal components associated with the smallest variances lessens the effect of multicollinearity because small variances indicate correlated independent variables. For the mtcars dataset, Figure 1 shows that the last six principal components only explain 5% of the data's variance, suggesting the presence of multicollinearity. As a result, multiple linear regression should be conducted using the first 3-5 principal component scores.

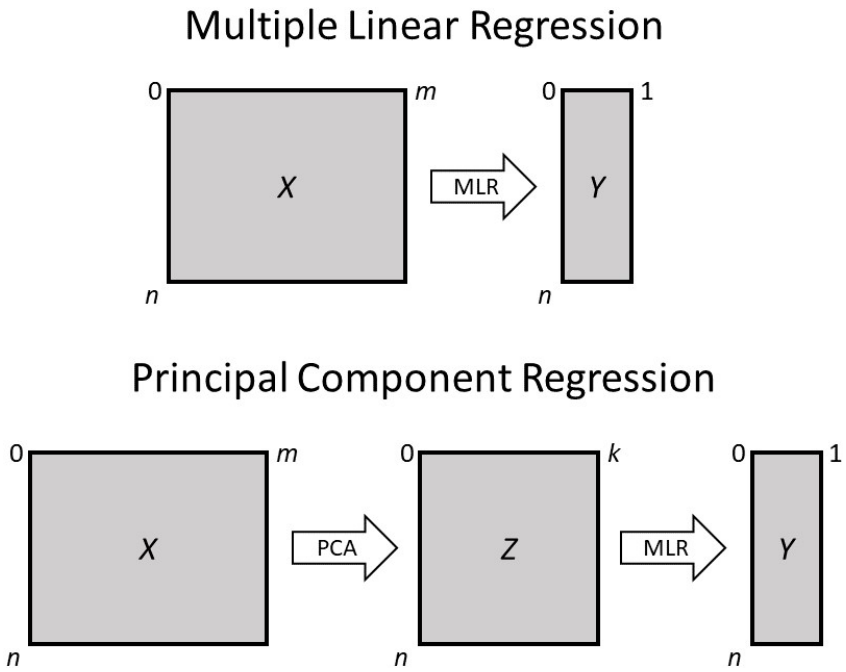

Figure 3: Diagram of multiple linear regression versus principal component regression

Figure 3 compares the processes of conducting multiple linear regression (or ordinary least squares) and PCR. The boxes represent matrices with the dimensions noted at the corners. Z is obtained from dimensionality reduction, and Y is XB (MLR) or ZA (PCR).

2.3.1 Math

This section delves into the math behind PCR. The input of PCR is the $n \times m$ standardized data matrix, X (containing z-scores). In ordinary least squares, the regression coefficients are found by

$$B = (X^T X)^{-1} X^T Y \tag{8}$$

where $X^T X$ is the correlation matrix of independent variables (similar to the covariance matrix C). The derivation of the above equation can be found in the appendix (5.2.1). B is of size $m \times 1$ where every entry corresponds to an independent variable.

PCR differs from least squares by transforming the independent variables into principal components.

$$X^T X = V D V^T = Z^T Z \tag{9}$$

where V contains the eigenvectors of $X^T X$, and D is a diagonal matrix of eigenvalues.

The principal components associated with the smallest eigenvalues are omitted to eliminate multicollinearity, and then ordinary least squares is used to regress Y on Z

$$A = (Z^T Z)^{-1} Z^T Y \tag{10}$$

where A contains the regression coefficients for the new set of independent variables, and Z is $n \times k$. A and B are related to each other by

$$B = V A \tag{11}$$

as shown in the appendix (5.2.2). To ensure that A and B are both $m \times 1$, A contains 0 wherever the principal components are omitted.

2.3.2 Code Workflow

The code workflow to conduct PCR in python is described below. The corresponding Jupyter notebook with the MSE plots can be found **here**.

1. Read in data (extract dependent variable and store separately if applicable)
2. Use cross-validation to determine how many principal components to use
   (a) Standardize data and conduct PCA
   (b) Transform data to obtain Z
   (c) Perform cross-validation (using scikit-learn's `cross_val_score` method) to measure the predictive performance of regression
      i. Separate data into smaller subsets using `KFold` (number of sets experimentally determined)
      ii. Set up ordinary least squares linear regression with `LinearRegression`
      iii. `cross_val_score` parameters
         A. `estimator`: ordinary least squares linear regression
         B. `X`: data to fit (array or matrix)
         C. `y`: target variable to try to predict (array)
         D. `scoring`: scorer object (`neg_mean_squared_error`)
         E. `cv`: splitting strategy generated by `KFold`
      iv. `cross_val_score` returns an array of scores from the estimator for each run
      v. Use `cross_val_score` to find average mean squared error (MSE) without considering principal components (y-intercept)
      vi. Use `cross_val_score` to find average MSEs for 1 to m principal components (adding an additional component with each iteration)
   (d) Plot average MSEs against the number of principal components and determine which number results in the lowest MSE (this number will be used when analyzing actual test data)
3. Run actual PCR
   (a) (Optional) Split data into training and test sets
   (b) Standardize training data and conduct PCA
   (c) (Optional) Verify the number of principal components to use with training data using method described above
   (d) Dimensionally reduce training data using the number of principal components determined above
   (e) Dimensionally reduce standardized test data using the same principal components as the previous step
   (f) Train regression model on training data
   (g) Use the model to predict values from the dimensionally reduced test dataset

(h) (Optional but recommended) Calculate MSE and SPE of expected and obtained values

Training the regression model (step 3f) is equivalent to solving the regression coefficients, A. To obtain Y from the test data, X, A needs to be converted to B using equation 11. Using the associative property, the regression equation can be re-written as

$$Y = X(VA) = (XV)A \tag{12}$$

XV is found in step 3e, which is then directly passed into the regression model, resulting in the correct Y.

PCR can be validated with several different statistic measures, such as MSE and SPE. Mean squared error (MSE) is calculated by finding the average of the squared differences between estimated and expected values. MSE is a measure of both the variance and bias of the estimator.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{13}$$

SPE stands for squared prediction error. It can be computed with the following equation

$$SPE = XV_0V_0^TX^T \tag{14}$$

where X is the test data matrix ($n \times m$) and the columns of V_0 consist of the eliminated principal components. The output should be a symmetric matrix ($n \times n$) with diagonal entries corresponding to the SPE for each test sample. Typically, 95% and 99% limits for SPE are found with training data and compared with new observations to determine whether the regression model is appropriate to use. Geometrically, SPE gives the orthogonal distance from a data point to the hyperplane spanned by the chosen number of principal components.

2.3.3 PCR Applications

Three different applications of PCR are available as Jupyter notebooks **here**. Two of the three notebooks use the code workflow described above to analyze the mtcars dataset and the hitters dataset. The hitters dataset consists of 322 observations of major league baseball players on 20 variables. PCR is conducted to predict the players' salaries based on the other 19 variables.

The third notebook uses PCA for logistic regression to predict whether a breast mass is benign or malignant. The data consists of 10 cell nuclei characteristics (i.e. radius, symmetry, and compactness) within 357 benign and 212 malignant images. For each characteristic, the mean, standard error, and worst values are recorded, resulting in 30 independent variables. PCA is typically implemented with logistic regression to dampen the effects of multicollinearity.

3 Representing PCA as Optimization Problems

This section explains two connections between PCA and optimization. Viewing PCA as an optimization problem can aid in the generation of improved methods for PCA, such as L1-PCA. L1-PCA is a locally-converging heuristic procedure that attempts to find the L1-norm best-fit subspace.

3.1 Subspace Optimization

A slightly different perspective on PCA involves finding a best-fit subspace of k dimensions with $k < m$. The goal of subspace optimization is to remove redundancies in the data such that it can be fully represented in fewer dimensions. The best-fitting k-dimensional subspace can be found with

$$\min_{V} \sum_{i=1}^{n} \left|\vec{x}_i - \vec{z}_i V^T\right|^2 \tag{15}$$

where $\vec{z}_i$ is the projection of $\vec{x}_i$ in the subspace and $\vec{z}_i V^T$ is $\vec{z}_i$ in terms of the original coordinates (5.3.1). In other words, the columns of V define a basis for the subspace. One solution to this optimization problem is to set the columns of V as the first k principal components and $\vec{z}_i$ as the rows of $Z = XV$. PCA is an optimal solution because of its connection to SVD. The subspace optimization problem can be restated as finding a V such that the difference between X and the projected data on V, $|X - Proj_V X|$, is minimized. Since V is a basis for the subspace, the vectors it is spanned by should be mutually orthogonal to each other.

From SVD, X can be written as a sum of singular values

$$X = \sum_{i=1}^{m} \sigma_i \vec{u}_i \vec{v}_i^T \tag{16}$$

where the singular vectors satisfy the conditions for V. As a result, the solution of subspace optimization would be the vectors corresponding to the k largest singular values.

3.2 Forward and Backward View of PCA

Another way to solve for PCA using optimization involves searching for a vector that maximizes the variance of projections of points in X onto a one-dimensional subspace and then finding a second vector that is orthogonal to the first and maximizes the leftover variance. This approach is very similar to the interpretation of PCA described at the end of section 1.2.

The variance of points projected on a vector, $\vec{v}_i$ is

$$\text{Var}(X\vec{v}_i) = \vec{v}_i^T C \vec{v}_i \tag{17}$$

where C is the covariance matrix. The derivation of equation 17 can be found in the appendix (5.3.2). From equation 17, the first part of the optimization problem is

$$\max_{\vec{v}_1} \vec{v}_1^T X^T X \vec{v}_1, \tag{18}$$

$$\vec{v}_1^T \vec{v}_1 = 1$$

The optimal solution to $\vec{v}_1$ is the eigenvector of C associated with the largest eigenvalue, which is equivalent to the first principal component. The maximized variance is given by the eigenvalue. (The $\frac{1}{n-1}$ is left out because it is a constant.) After finding $\vec{v}_1$, the direction of maximum remaining variance can be found by solving

$$\max_{\vec{v}_2} \vec{v}_2^T X^T X \vec{v}_2, \tag{19}$$

$$\vec{v}_2^T \vec{v}_2 = 1,$$

$$\vec{v}_1^T \vec{v}_2 = 0$$

Maximizing leftover variance is addressed by the constraint that $\vec{v}_1$ and $\vec{v}_2$ must be orthogonal to each other. The optimal solution to $\vec{v}_2$ is the second principal component. The process can continue and the kth principal component can be solved with

$$\max_{\vec{v}_k} \vec{v}_k^T X^T X \vec{v}_k, \tag{20}$$

$$\vec{v}_k^T \vec{v}_k = 1,$$
$$\vec{v}_i^T \vec{v}_k = 0, i = 1, \cdots, k-1$$

Finding successive directions of maximum variation is known as a forward view of PCA. Backward view involves seeking successive directions of minimum variation in the data. Using the minimization approach would produce the principal components with variances in ascending order, which could be useful when developing alternative PCA methods.

4 Conclusion

Overall, principal component analysis can be used to determine what linear combinations of variables most accurately explain the variance of a complex dataset. This feature enables data matrices to be dimensionally reduced into their most statistically significant components. Dimensionality reduction is useful for further analysis and eliminates noise and redundancy in the data. Additional applications of PCA include cluster analysis and principal component regression (PCR). PCR is an effective alternative to multiple linear regression because it eliminates the effect of multicollinearity by using dimensionality reduction. PCR is conducted on three different datasets (mtcars, hitters, and breast mass) located in the **Linearity2-PCA GitHub repo**. Furthermore, the connection between PCA and optimization is explored with two examples, subspace optimization and forward/backward view of PCA. Solving PCA as optimization problems can aid in the creation of more tailored PCA methods. Some extensions worth investigating include

1. Comparing PCR with Projection to Latent Structures
2. Studying how L1-PCA works
3. Understanding the connection between k-means clustering and PCA
4. Utilizing PCA to improve stochastic optimization (currently in progress)

5 Appendix

5.1 Statistics Background

$$\text{Var}(A) = \frac{1}{n-1} \sum_{i=1}^{n} (a_i - \mu_A)^2$$

$$\text{Cov}(A, B) = \frac{1}{n-1} \sum_{i=1}^{n} (a_i - \mu_A)(b_i - \mu_B)$$

Cov(A,B) = Cov(B,A) and Cov(A,A) = Var(A). A negative covariance implies that when one variable is larger, the other variable tends to be smaller. A positive covariance means that both variables tend to show similar trends.

$$\text{Corr}(A, B) = \frac{\text{Cov}(A, B)}{\sigma_A \sigma_B}$$

If data is fully standardized, then both σ_A and σ_B equals 1, making C the covariance and correlation matrix.

5.2 PCR Background

5.2.1 Derivation of $B = (X^T X)^{-1} X^T Y$

Given X and Y, find B while minimizing $\left|XB - Y\right|$. To minimize the error, $XB - Y$ should be orthogonal to the columns of X, which is the same as being orthogonal to the rows of X^T. As a result,

$$X^T(XB - Y) = 0$$
$$X^T XB - X^T Y = 0$$
$$X^T XB = X^T Y$$
$$B = (X^T X)^{-1} X^T Y$$

5.2.2 Verification of $B = VA$

Substitute $B = (X^T X)^{-1} X^T Y$ and $A = (Z^T Z)^{-1} Z^T Y$

$$(X^T X)^{-1} X^T Y = V(Z^T Z)^{-1} Z^T Y$$

Substitute $Z = XV$, multiply both sides by V^T, and use $V^T V = I$ to simplify

$$V^T (X^T X)^{-1} X^T Y = I(V^T X^T XV)^{-1} V^T X^T Y$$

Multiply both sides by $V^T X^T XV$

$$V^T X^T XVV^T (X^T X)^{-1} X^T Y = V^T X^T Y$$
$$V^T X^T XI(X^T X)^{-1} X^T Y = V^T X^T Y$$
$$V^T X^T Y = V^T X^T Y$$

5.3 PCA and Optimization Background

5.3.1 Breakdown of $\vec{z}_i V^T$

It is clear that $\vec{z}_i V^T$ gives $\vec{z}_i$ in terms of the original coordinates by examining the dimensions. $\vec{z}_i$ are row vectors of size $1 \times k$, and V^T is of size $k \times m$. $\vec{z}_i V^T$ would result in a row vector of size $1 \times m$, which matches in dimensions with $\vec{x}_i$.

When using principal components as the solution to subspace optimization, substitute $Z = XV$ as $\vec{z}_i$ in $\vec{z}_i V^T$ to get

$$ZV^T = XVV^T = X$$

5.3.2 Derivation of $\text{Var}(X\vec{v}_i) = \vec{v}_i^T C \vec{v}_i$

For a given matrix M, $\text{Var}(M) = \frac{1}{n-1} M^T M = C$. Since XV gives the matrix Z,

$$\text{Var}(Z) = \frac{1}{n-1}(Z)^T Z$$

Substitute $Z = XV$

$$\text{Var}(XV) = \frac{1}{n-1}(XV)^T XV$$

$$\text{Var}(XV) = \frac{1}{n-1} V^T X^T XV$$

$\frac{1}{n-1}$ is a constant so we can arrange the above equation into

$$\text{Var}(XV) = V^T \left[\frac{1}{n-1} X^T X \right] V = V^T C V$$

$\vec{v}_i$ corresponds to a column of V resulting in

$$\text{Var}(X\vec{v}_i) = \vec{v}_i^T C \vec{v}_i$$

6 References

[1] Dunn, K. 6.6. **Principal Component Regression (PCR)**. Process Improvement Using Data.

A very thorough explanation of PCR, which also mentions SPE

[2] Gorinevsky, D. **Lecture 2 Intelligent Energy Systems: Monitoring Basics**. Stanford University 33 (2012).

Lecture slides that explain how to find SPE and what it represents geometrically

[3] Hyndman, R. **Why every statistician should know about cross-validation**. Hyndsight.

A concise blog post explaining how cross-validation works

[4] Kuznetsova, A., Pons-Moll, G. & Rosenhahn, B. "PCA-Enhanced Stochastic Optimization Methods". in Pattern Recognition (eds. Pinz, A., Pock, T., Bischof, H. & Leberl, F.) 377–386 (Springer Berlin Heidelberg, 2012).

A very interesting paper that exposed me to the idea of using PCA to improve stochastic optimization algorithms, such as particle swarming and simulated annealing

[5] **Principal Components Regression**. NCSS Statistical Software.

Another tutorial that describes how PCR works

[6] Reris, R. & Brooks, J. P. "Principal Component Analysis and Optimization: A Tutorial". in Operations Research and Computing: Algorithms and Software for Analytics (eds. Borchers, B., Brooks, J. P. & McLay, L.) 212–225 (INFORMS, 2015). doi:10.1287/ics.2015.0016.

A great paper that introduced me to the idea of solving PCA as an optimization problem. The paper also motivated me to use the mtcars dataset to explain the uses of PCA

[7] Tayarani-N, M., Bennett, A. P., Xu, H. & Yao, X. "Improving the performance of evolutionary engine calibration algorithms with principal component analysis". in 2016 IEEE Congress on Evolutionary Computation (CEC) 5128–5137 (2016). doi:10.1109/CEC.2016.7748340.

This paper was my initial inspiration for this project. I wanted to use PCA to improve optimization algorithms, but I eventually decided to focus on PCR

6.1 Coding Resources

[1] **Linearity2-PCA**

Link to my GitHub repo with a few PCR examples

[2] Broll, S. **Logistic Regression Model using PCA components**. Kaggle.

Source of data for breast mass application

[3] **Hitters Dataset**. Kaggle.

Source of data for baseball application

[4] Romanini, R. **mtcars Dataset**. Kaggle.

Source of mtcars dataset

[5] **PCR Tutorial**

A great tutorial on using python to conduct PCR. I completed this tutorial to get a better idea of how to apply PCR to large datasets