PRINCIPAL COMPONENT ANALYSIS

ERIC XIA

Abstract. Principal components are orthogonal vectors in a basis which maximize the variance of a dataset in respect to themselves. Here, we show that the principal component matrix is equivalent to the eigenvectors of the covariance matrix. We demonstrate applications of principal components, where simplified structures can be found in high dimensional data. Finally, we discuss the limitations of PCA and discuss possible alternatives.

1. Introduction

Principal components are orthogonal vectors which maximize the variance of a dataset in respect to themselves. Here, we show that the matrix made up of the principal components is equivalent to the eigenvectors of the covariance matrix.

1.1. **Example: Fish swimming.**

We are interested in determining which directions fish are swimming, and take three pictures from random positions above the water. Intuitively, one important direction seems to be A_x. Principal Component Analysis allows us to find the exact base(s) along which a dataset varies, and reexpress data along them.

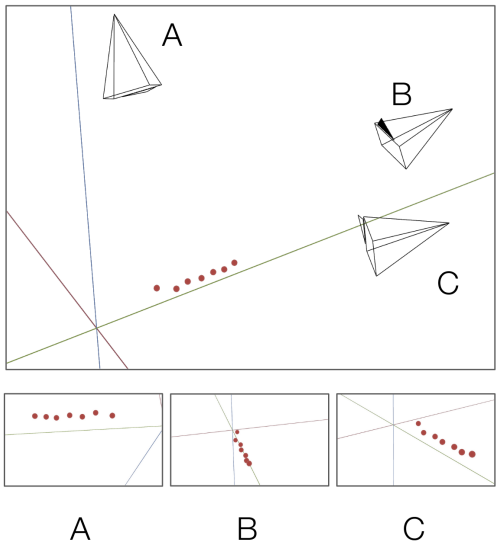

Figure 1. The fish swimming and positions of cameras.

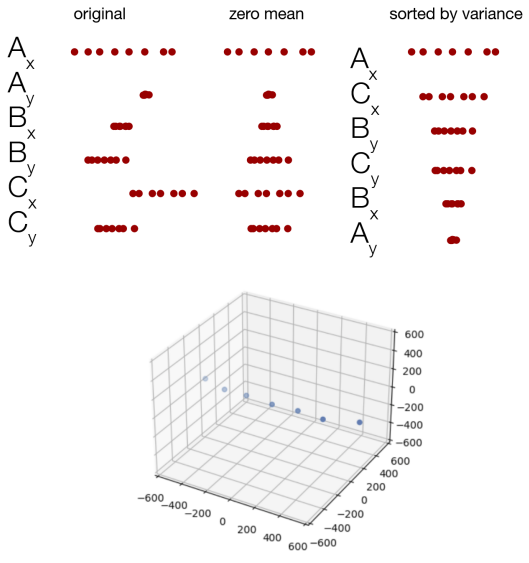

Figure 2. Approximations at finding bases, compared to the best reconstruction by PCA.

2. Background

2.1. **Set up: Naive Basis.** We will assume that we are working with an $m \times n$ matrix X. Each row of X is a m dimensional vector, where m[1] is the number of measurement types, and n is the number of measurements. Equivalently, our sample consists vectors that lies in an m-dimensional vector space. We have an orthonormal (orthogonal and normal) basis that reflects the method in which we gathered the data. In other words, for m measurement types we have

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = I_m$$

Example 2.1. For our fish, we have a basis of $\hat{A}_x, \hat{A}_y, \hat{B}_x, \hat{B}_y, \hat{C}_x, \hat{C}_y$ or

$$(1,0,0,0,0,0), (0,1,0,0,0,0), (0,0,1,0,0,0), (0,0,0,1,0,0), (0,0,0,0,1,0), (0,0,0,0,0,1)$$

We can write the observed position of any fish as a linear combination of these.

We assume that we have high-dimension measurements of a lower-dimension phenomenon; we will look for a better basis through which we can re-express our data set. In other words, we're looking for a change of basis P to apply to our naive basis.

Proposition 2.2. *Let P be the matrix form of a linear transformation from $\mathbb{R}^m \to \mathbb{R}^m$. We have $PX = Y$, where the rows of P are a set of basis vectors for expressing columns of X.*

Proof. Let p_i be a row of P, and x_i be a column of X. We have

$$PX = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix} \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}$$

So that

$$Y = \begin{bmatrix} p_1 x_1 & \cdots & p_1 x_n \\ \vdots & \ddots & \vdots \\ p_m x_1 & \cdots & p_m x_n \end{bmatrix}$$

Where a column y_i of Y looks like

$$y_i = \begin{bmatrix} p_1 x_i \\ \vdots \\ p_m x_i \end{bmatrix}$$

Each y_i going down Y is a dot product of x_i with $p_1 ... p_m$, so that taken together the rows of P, $\{p_1 ... p_m\}$, are a new set of basis vectors for expressing the columns of X.

□

[1] We will assume m and all following variables to be in $\mathbb{R}$

2.2. **Goal: no covariance, maximum variance.** The basis we actually want is not the naive basis.

Example 2.3. For the fish example, the axes on which the cameras lie don't correspond with the directions which the fish go. We would like to find an axis, or axes, that the fish are heading.

Here, we will define a statistically motivated goal for P, our change of basis. We will minimize redundancy, measured by covariance, and maximize the variance.

We will define variance and covariance as follows. Variance is a measure of how much a collection of measurements deviates from its best fit. If it is entirely predicted by its best fit line, it will have a variance of 0. For a single set of measurements adjusted to have means of zero, A:

Definition 2.4.

$$\text{variance of A} = \sigma_A^2 = \frac{1}{n}\sum_i a_i^2$$

Example 2.5.

If we have two sets of measurements (with zero means), we can generalize our definition. We define covariance between A and B as

Definition 2.6.

$$\text{covariance of A and B} = \sigma_{AB}^2 = \frac{1}{n}\sum_i a_i b_i$$

Remember that our data has a mean of 0, so that small values in a dataset are negative.

Example 2.7.

$$B_y = 423, 516, 568, 628, 691, 749, 781$$

$$\text{(zero mean) } B_y \approx -199.3, -106.3, -54.3, 5.7, 68.7, 126.7, 158.7$$

$$C_x = 958, 1021, 1158, 1246, 1389, 1479, 1603$$

$$\text{(zero mean) } C_x \approx -306.9, -243.9, -106.9, -18.9, 124.1, 214.1, 338.1$$

$$C_y = 316, 422, 467, 528, 585, 645$$

$$\text{(zero mean) } C_y \approx -202.5, -96.5, -51.5, 9.4, 66.4, 126.4, 148.4$$

$$\frac{1}{7}\sum_{i=1}^{7} B_{yi}C_{yi} \approx 13947.1$$

$$\frac{1}{7}\sum_{i=1}^{7} C_{xi}C_{yi} \approx 25221.9$$

We can see covariance of B_y and C_y is small, but covariance of C_x and C_y is big.
When comparing two measurements for all the fish, if we consistently get (small, big) measurements as we go over our fish, we will end up with a really negative covariance (a_i is $-$ while b_i is $+$). In contrast, if we get small and small or big and big values all the time, we'll end up with a really positive covariance. If our comparison doesn't tend towards either, we'll have a weak, small covariance; which is what we want.

Thus, the absolute magnitude of the covariance tells us how redundant two measurements are. Observe that the covariance of A and A is $\sigma^2_{AA} = \sum_i a_i a_i = \sum_i a_i^2 = \sigma^2_A$, the variance of A. We can convert measurements A and B into row vectors.

$$a = (a_1, a_2, ...a_n)$$
$$b = (b_1, b_2...b_n)$$

Now, the covariance is the scaled dot product of a $1 \times n$ matrix, a, by a $n \times 1$ matrix, b^T, that is $\sigma^2_{ab} = \frac{1}{n} a b^T$.

2.3. **2 to many dimensions.** Just as we have covariance for two sets of measurements, we can define a covariance matrix for several sets of measurements. Renaming a as x_1 and b as x_2, we have m measurements total so that we have

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$$

This is consistent with our definition of X above.

Definition 2.8. The covariance matrix of X is the scaled dot product of an $m \times n$ matrix X, by a $n \times m$ matrix X^T.

$$C_x = \frac{1}{n} X X^T$$

Now, the ij^{th} element of C_X is the dot product between the vector of the i^{th} measurement type with the vector of the j^{th} measurement type, equal to the $ji^{th} element$ (as the dot product is commutative). Thus, C_x is symmetric.
Observe that C_X is a square symmetric mxm matrix, that diagonal terms C_{jj} are the variances of the j^{th} measurement, and that all other terms are covariances between two measurements.

2.4. **diagonalizing our covariance matrix.** C_X is the covariance matrix for X, and C_Y is the covariance for Y. What P should we choose to multiply X by so that C_y has these properties? We define our goal to be

- minimizing redundancy, measured by covariance
- maximizing the variance

This means that our covariance matrix C_y should have

- all off-diagonal terms as 0, cause we don't want covariance.
- each successive dimension is ordered according to its variance

In conclusion, we want to find P where $Y = PX$ such that $C_y = \frac{1}{n} Y Y^T$ is a diagonal matrix.

2.5. **background proof.**

Proposition 2.9. *The inverse of an orthogonal matrix is its transpose, or* $A^{-1} = A^T$

Proof. Let A is an orthogonal matrix whose columns are $c_1, c_2, ..., c_n$; we will think about $A^T A$ looks like. The columns of an orthogonal matrix are mutually orthogonal unit vectors. Thus,

$$c_i^T c_j = \begin{cases} 1 \text{ when } i = j \\ 0 \text{ when } i \neq 0 \end{cases}$$

This makes sense intuitively, because the dot product of two orthogonal vectors will be zero except against itself. However, we also have the identity matrix as

$$a_{ij} = \begin{cases} 1 \text{ when } i = j \\ 0 \text{ when } i \neq 0 \end{cases}$$

Ultimately, we can conclude that $A^T A = I$. By definition of inverse we can write

$$A^T A = I = A^{-1} A$$

$$A^T = A^{-1}$$

□

3. Main Theorem

We are now ready to prove the principle behind PCA, following the proof given in [4].

Theorem A. *We want to find some orthonormal matrix P in $Y = PX$ such that $C_y = \frac{1}{n} YY^T$, our new covariance matrix, is some diagonal matrix D. We find that if P is the eigenvectors for our original covariance matrix $C_X = \frac{1}{n} XX^T$ arranged as rows, $C_y = D$.*

Proof. We rewrite C_y in terms of PX and then C_X:

$$C_y = \frac{1}{n} YY^T \qquad \text{definition}$$

$$C_y = \frac{1}{n} (PX)(PX)^T \qquad \text{definition}$$

$$C_y = \frac{1}{n} PXX^T P^T \qquad \text{shoes and socks}$$

$$C_y = P(\frac{1}{n} XX^T) P^T \qquad \text{moving scalar}$$

$$C_y = PC_X P^T \qquad \text{definition}$$

At this point, we want to write C_X in terms of D, so we will prove a surprising property: a symmetric matrix is diagonalized by a matrix of its orthogonal eigenvectors. Because C_X is symmetric, we can then write it in terms of D.

Proposition 3.1. *For a symmetric matrix A we have $A = EDE^T = E^T DE$, where D is diagonal and E is a matrix of orthogonal eigenvectors of A, arranged as columns.*

Proof. The proof is in two parts. In the first part, we see that any matrix can can be diagonalized if and only if its eigenvectors are all linearly independent. In the second part, we see that the eigenvectors of a symmetric matrix are not just linearly independent, but also orthogonal.

Proposition 3.2. *Let A be a matrix, with linearly independent eigenvectors. Let $E = [e_1 e_2 ... e_n]$ be the matrix of eigenvectors placed in the columns. Let D be a diagonal matrix where the i^{th} eigenvalue is placed in the ii^{th} position. We have $A = EDE^{-1}$.*

Proof. Let's look at two products, AE and ED. Let $A = \begin{bmatrix} A_1 \\ \vdots \\ A_n \end{bmatrix}$, so that A_i is the i^{th} row of A. The first column of AE is $\begin{bmatrix} A_1 * e_1 \\ \vdots \\ A_n * e_1 \end{bmatrix} = Ae_1$, the rows of A dotted with e_1. The second column is $\begin{bmatrix} A_1 * e_2 \\ \vdots \\ A_n * e_2 \end{bmatrix} = Ae_2$. Altogether, we can think of the columns of AE as $[Ae_1 Ae_2 ... Ae_n]$.

The first column of ED is $\lambda_1 e_1$. This is because D_1 has just λ_1 in the first position, and the column will retrieve elements $\begin{bmatrix} (\lambda_1 e_{11} + 0e_{12} + \cdots 0e_{1n}) \\ \vdots \\ (\lambda e_{n1} + 0e_{n2} + \cdots + 0e_{nn})) \end{bmatrix} = \begin{bmatrix} \lambda_1 e_{11} \\ \vdots \\ \lambda e_{n1} \end{bmatrix} = \lambda_1 e_1$.

Altogether, we can think of the columns of ED as $[\lambda_1 e_1 \lambda_2 e_2 ... \lambda_n e_n]$.

Because we have $Ae_i = \lambda_i e_i$ by the definition of the eigenvalue, we have $[Ae_1 Ae_2 ... Ae_n] = [\lambda_1 e_1 \lambda_2 e_2 ... \lambda_n e_n]$, and AE = ED. Rearranging, we have $A = EDE^{-}1$. This means that any matrix with linearly independent eigenvectors can be diagonalized by its eigenvectors. □

Proposition 3.3. *All symmetric matrices have orthogonal eigenvectors.*

Proof. For some symmetric matrix, let λ_1 and λ_2 be distinct eigenvalues for eigenvectors e_1 and e_2. We have

$$\begin{aligned} \lambda_1 e_1 \cdot e_2 &= (\lambda_1 e_1)^T e_2 && \text{definition} \\ &= (Ae_1)^T e_2 && \text{definition} \\ &= e_1^T A^T e_2 \\ &= e_1^T A e_2 && \text{by symmetry} \\ &= e_1^T (\lambda_2 e_2) && \text{definition} \\ \lambda_1 e_1 \cdot e_2 &= \lambda_2 e_1 \cdot e_2 \\ (\lambda_1 - \lambda_2)(e_1 \cdot e_2) &= 0 \end{aligned}$$

Because we have unique eigenvalues λ_1 and λ_2, we must have $e_1 \cdot e_2 = 0$. Therefore, we know that the eigenvectors of a symmetric matrix are orthogonal. □

Thus, a symmetric matrix is diagonalized by a matrix of its orthogonal eigenvectors. □

Now, we will return to the main thread of the proof, where we have $C_y = PC_X P^T$. We will choose P to be the matrix where each row p_i is an eigenvector of $C_x = \frac{1}{n} XX^T$. By 3.2 we have $C_x = E^T DE$, where we assume E to be the matrix of eigenvectors of C_x arranged as columns.

Thus, we have $P \equiv E^T$. As we assume P to be an orthogonal matrix, we also have $P^{-1} = P^T$. Then

$$\begin{aligned}
C_Y &= PC_XP^T \\
C_Y &= P(E^TDE)P^T && \text{by 3.1} \\
C_Y &= P(P^TDP)P^T \\
C_Y &= (PP^T)D(PP^T) \\
C_Y &= (PP^{-1})D(PP^{-1}) && \text{by 2.9} \\
C_Y &= IDI \\
C_Y &= D
\end{aligned}$$

Therefore when $P \equiv E^T$, we have C_Y as a diagonal matrix, which satisfies the properties we wanted for our transformation: maximum variance, minimum covariance. □

4. Applications

We will demonstrate two applications of PCA on data. First, we will show how PCA can reconstruct the direction of the fish. Next, we will show how PCA can be used on a four dimensional iris measurement dataset to identify species.

4.1. PCA on fish dataset: from 6 dimensions to 3.

Example 4.1. Here is python code for finding the principal components of the fish dataset with the eigenvectors of the covariance matrix, converted from Matlab in [4]. Coordinates are taken directly from the 3d environment through scripting.

```
#imports: matplotlib, numpy
#[Ax, Ay, Bx, By, Cx, Cy]
fish_data = [
[1275,223,783,423,958,316],
[1074,190,762,516,1021,422],
[912,236,823,568,1158,467],
[759,220,829,628,1246,528],
[598,249,881,691,1389,585],
[473,238,891,749,1479,645],
[336,229,922,781,1603,667]]

cols = ['ax','ay','bx','by','cx','cy']
x = pd.DataFrame(fish_data, columns=cols)
#removing the mean from each measurement
x = x - x.mean()
cov_mat = np.cov(np.transpose(x))
[eigvals, eigvecs] = np.linalg.eig(cov_mat)
for i in eigvals:
    print(i / sum(eigvals))
```

[2]

```
indices = np.argsort(eigvals)[::-1]
eigvecs = eigvecs[:, indices]
print(np.dot(eigvecs[0], eigvecs[1]))
```

[3]

```
y = np.matmul(x, eigvecs[:,:3])
#visualization code
```

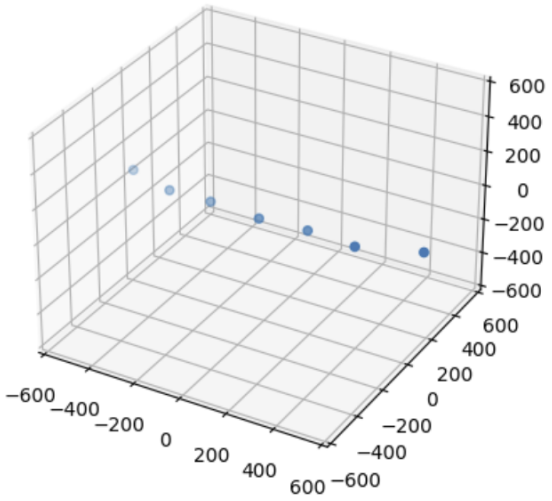

FIGURE 3. From a six-dimensional dataset of X and Y values, PCA reconstructs the relative positions of fish swimming. Unfortunately, the principal components themselves do not correspond to dimensions in the real world (there is not enough information given to do this). However, the structure is preserved very well.

4.2. **PCA on iris dataset: from 4 dimensions to 2.**

Example 4.2. Using scikit-learn[2], the process is considerably easier.

[2]This returns an explained variance of 0.993, 0.005, and 0.001 for our first three eigenvectors, confirming our intuition that the data is one-dimensional.

[3]This returns 0, confirming our eigenvectors are orthogonal.

	sepal length	sepal width	petal length	petal width	target
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

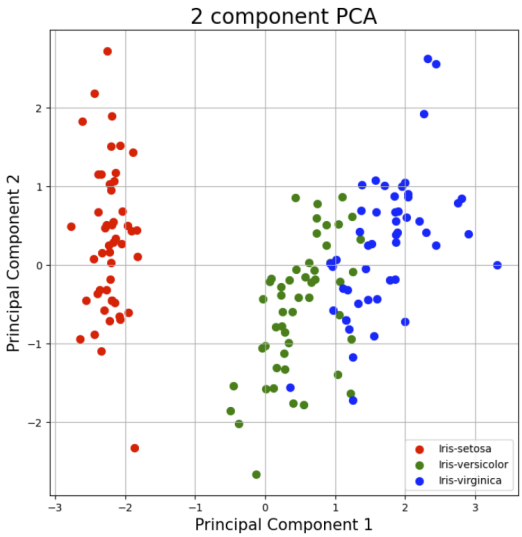

FIGURE 4. From a four-dimensional dataset of sepal length, sepal width, petal length, and petal width, PCA identifies two principal components, from which the species are easily separable.

5. Future directions

PCA has a long history of applications. The core ideas were developed in 1901 by Karl Pearson in "On Lines and Planes of Closest Fit to Systems of Points in Space"[1], in which he described a method for fitting a linear subspace to multivariate data. It was later independently found by Harold Hotelling in the 1930s. It is also closely related to singular value decomposition[4]. Under different names, the technique is used in signal processing, mechanical engineering, linear algebra, and meterological science.

PCA continues to be used widely today, as a very powerful way to understand or visualize the essence of a dataset. By looking at the eigenvalues of the covariance matrix, we are able to quantify the importance of each principal component. With the fish example, it can be made immediately obvious that the underlying data is one-dimensional, even though we have a six dimension set of measurements.

However, it has numerous flaws. We make several assumptions about our original dataset with PCA: that the best way to express our dataset is by linear combination, that maximum variance is always good, and the principal components must be orthogonal. These assumptions are what make PCA work – the eigenvectors of a symmetric matrix are orthogonal – but often do not work on real world data.

[4]SVD is a method of writing any matrix as a decomposition USV^T; it can be shown that the columns of V are the principal components of X. The python library scikit-learn uses SVD in their implementation of PCA!

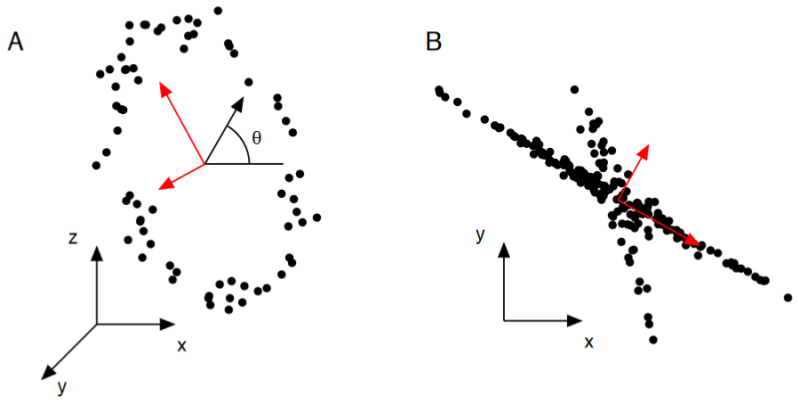

FIGURE 5. (Reproduced from [4]) A nonlinear dataset with non-orthogonal axes. PCA (on the right) fails to capture its intrinsic structure, as the directions of highest variance do not accurately represent the data.

Nonlinear phenomena, such as the ring above, cannot be modeled well with PCA, as it defines redundancy as necessarily being linear. Towards this end, one approach that improves on PCA is kernel PCA[3]: by using a kernel function instead of the dot product in our covariance matrix, it is possible to implicitly perform PCA on a nonlinearly related space to the input.

REFERENCES

[1] K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, Nov 1901.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[3] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. *Lecture Notes in Computer Science*, page 583–588, 1997.

[4] J. Shlens. A tutorial on principal component analysis, 2014.