Research Paper:
Principal Component Analysis

Roderick L. Renwick

The University of Michigan -- Dearborn

April 17, 2019

Table of Contents

Executive Summary

This report goes in depth about a mathematical algorithm called Principal Component Analysis (PCA), and its relevance to face classification applications. It is broken up into three sections. The first provides insight into what PCA is, and why it is used for classifications within data sets. The following section details an abstraction of the mathematics required for implementation of the PCA approach. The final section then examines a scenario in which PCA is used, to show how it is implemented, and what the results look like regarding face classification.

The primary purpose of this report is to summarize how to implement a fundamental PCA algorithm structure. This algorithm structure serves as the building blocks for scoring the variance of multidimensional data sets and classifying these sets from these variance scores. One of my biggest takeaways from researching this is how feasible and tangible this seemingly crazy-complicated idea becomes after mapping out the math behind it. It is my hope that you will feel the same way too, after reading this (or at least be able to make sense of it). As the history is still young, exciting, and experimental for machine learning through computer vision applications such as PCA, it is an advantageous time to jump into it – and this is a perfect place to start.

Given that this subject is extensively complication (to me, at least) and I have not yet had the free time I need to learn about more it, this research paper provided the perfect opportunity for me to do a deep-dive into the math behind the Principal Component Analysis.

Research Paper:

Principal Component Analysis

The fundamental subject matter at the core of Principal Component Analysis (PCA) is linear algebra, and more specifically, the use of orthogonal matrix transformations. Orthogonal transformations will be discussed in a latter section but first, there is a question to be addressed – what exactly is PCA? On the highest level of abstraction, PCA is a way to measure the variance within and between data sets. Data sets may then be classified with respect to their variance scores. Think of this as a compare-and-contrast tool, where PCA can recognize patterns associated with data as well as emphasize the anomalies. These anomalies are uncorrelated variables (called “principal components”) that best describe the variance, while the correlated variables are the overlapping values that create the patterns (Smith, 2012). The key here is that once we have established a pattern, we do not care about the redundant variables, only the uncorrelated ones. This provides great potential for dimensionality reduction of the data, i.e., large amounts of data compression without much sacrifice of useful information. So, why do we care? Well, this means that PCA is a powerful technique for real-time image classification.

Methodology

Before getting into the mathematics and implementation of PCA, I would like to note a few of the sources that helped facilitate my understanding of PCA and guided my research. Perhaps the most all-encompassing short paper was that of the *International Journal of Recent Trends in Engineering: Facial Recognition using Eigenfaces by PCA (IJRTE, 2009, p.587)*. This paper serves as a great academic publication on the overview of PCA, what eigenfaces are and why they are important, and the applications of PCA in facial recognition.

I also reference *A tutorial on Principal Components Analysis (Smith, 2002)*, for a more detailed analysis of the mathematics behind PCA, alongside a concise walkthrough which "covers standard deviations, covariance, eigenvectors and eigenvalues" (p.1). This provides fantastic step-by-step instructions on how to implement PCA on a set of data in order to analyze it. explanation for. For further understanding of the linear algebra required for PCA, specifically the orthogonal transformations at the most theoretical level, see *Orthogonal Neighborhood Preserving Projections: A projection-based dimensionality reduction technique (Kokiopoulou & Saad, 2006)*.

All these scholarly articles were found published online and their links are noted in the Reference List section at the end of this paper. I strongly encourage anyone with serious interest or unanswered questions leftover after reading this paper to go to these links. Also, for anyone seeking to implement PCA within a program, OpenCV (Open Source Computer Vision, 2015) is a unique and eloquent library, originally developed by Intel, with the tools to do so.

Required Mathematics

Orthogonal Transformations

In linear algebra, orthogonal (perpendicular) transformations are used to maintain the geometric lengths (as vectors) and angles (between vectors) of an object while transforming (via rotating or reflecting) the data set. This transformation is used to describe/map out uncorrelated or independent values (orthogonal projection) of specific objects within a given data set. Imagine a 3D cube held up against a bright light (whose face is perpendicular to the ground), the 2D square shadow underneath it (if the side lengths remained the same) is effectively an orthogonal projection from the higher-dimensional 3D space to the lower-dimensional 2D space, maintaining the angles and lengths of the original object. The key to remember here is that the orthogonal transformation (a projection in this case) produces a reduction in dimensionality while still being able to describe the shape of the object.

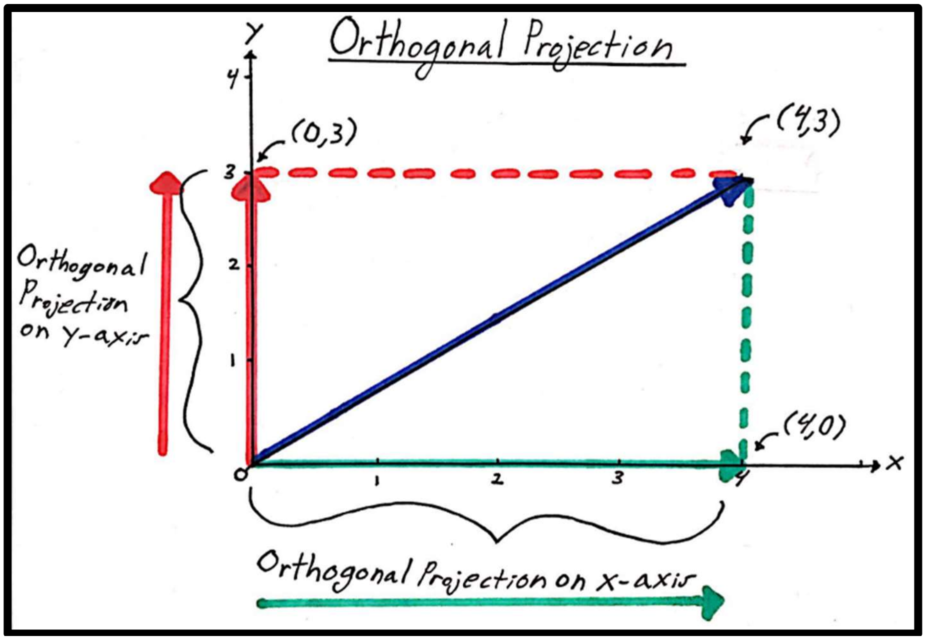

Figure-1: Orthogonal Projection Example

For example, the diagram above (figure-1) shows a 2-demensional vector space which can be described from two 1-demensional projections. The blue diagonal arrow on the x-y plane is the 2D vector, and the red vertical arrow represents it's 1D projection onto the y-axis while the green horizontal arrow represents it's 1D projection onto the x-axis.

Covariance Matrices

Matrices present concise representations of elements within a data set (in rows, and columns) which allow for simple manipulation (linear transformation) of the data. Much like how standard deviation and variance are measurements of 1-deminsional data sets, co-variance measurements are used to describe the distribution (relationship) of elements between data sets (i.e., the co-variance scores how correlated or uncorrelated variables are across data sets). This theory may be scaled to any n-dimensional data set, where "the covariance matrix defines both the spread (variance), and the orientation (covariance) of data" (Spruyt, 2014).

Eigenvalues & Eigenvectors

In matrices, eigenvectors and their corresponding eigenvalues are used to express a covariance matrix in a unique way. As Spruyt's (2014) article concisely explains:

> the largest eigenvector of the covariance matrix always points into the direction of the largest variance of the data, and the magnitude of this vector equals the corresponding eigenvalue. The second largest eigenvector is always orthogonal to the largest eigenvector, and points into the direction of the second largest spread of the data. (p.5)

The next diagram (figure-2) shows a visual representation of eigenvectors and their eigenvalues from a scatterplot of data after a linear transformation. Colored arrows show the "principal components" (rotated orientation axes) of the transformed data, which are the eigenvectors. The first principal component (blue arrow), is the largest eigenvector since in has the highest degree of variance associated with the data points projected onto it, and its variance magnitude gives us the eigenvalue of this eigenvector.

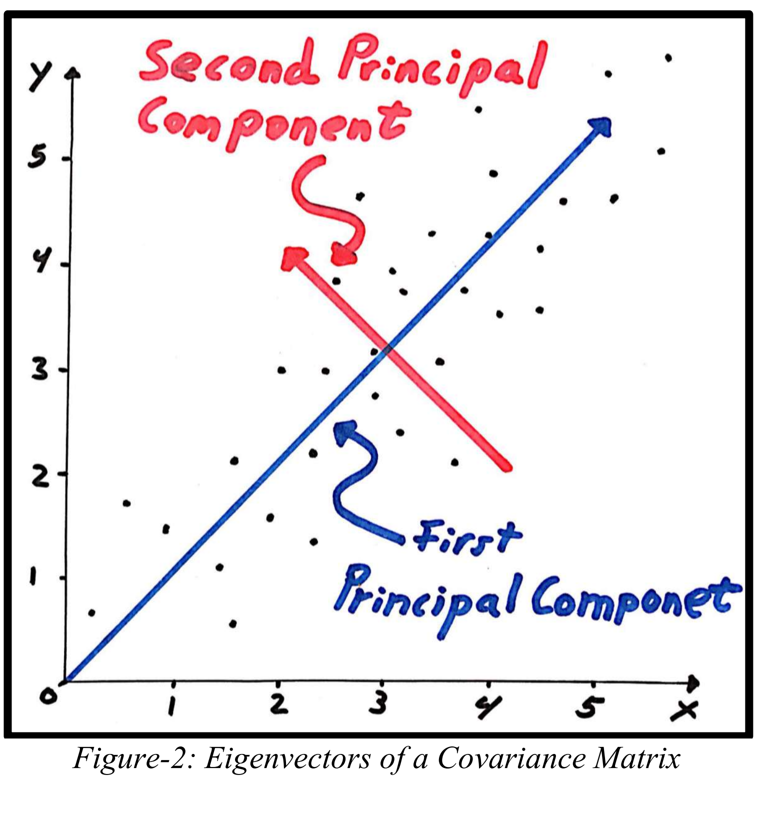

Figure-2: Eigenvectors of a Covariance Matrix

PCA Application: Eigenfaces

Now for another layer of abstraction – approaching this from a computer vision perspective. Assume the data sets we are taking in are 2D grayscale images of human faces, (a more complicated approach would be 3D colored images whose depths are the red, green, & blue pixel values). Imagine one image of a face (called an instance), whose grayscale pixel values ranging from 0-255. For face recognition, these images are represented as a matrix of their pixel values, and the matrix of a single instance would be transformed into one long vector of pixel values. Once each face in the data set has been vectorized, you can computer a mean of the vectors (or effectively the “average” face). If you subtract the mean face vector from each instance you, you end up with vector values which describe its variances from the mean face. You can then use Principal Component Analysis on these data sets (of face vector instances).

This will produce a covariance matrix of the face images, having a set of eigenvectors which best describes the distribution of values (keeping only the highest eigenvalues) between data sets (IJRTE, 2009). The eigenvectors for this application are called eigenfaces, which are the principal components projected onto the “face space” of all the images. Each eigenvector corresponds to one instance (or face image), where its largest eigenvalues best represent the variance of that face compared with respect to the mean. Visually these black and white eigenfaces would show a “ghostly face” for a unique individual, and the white represents where they vary most from the average face of the data set (IJARCET).

Conclusion

The bottom-line impact of this thought experiment is to demonstrate how PCA is used to take a high-dimensional data set of correlated values and turn them into a covariance matrix of uncorrelated values which are the principal components whose distribution of values best represent its variance compared to others in that dataset. This approach is key to minimizing similarities between data sets (which are redundancies of a global pattern found), while maximizing the variance within data sets. As this presents a way to classify patterns and empathize discrepancies, PCA is uniquely powerful and fundamental when it comes to face recognition applications.

Reflective Memo

From my perspective, I have always been so interested in the math that translates to object classification and face recognition in computer vision applications. While the math is still overwhelming for me to take in, as I learn more about it I get closer to recognizing how it translates to image classification. Hopefully someday I will able to implement my own math structures to perform PCA, and this is just the opportunity I needed to gain the knowledge for it.

I think this is fundamental to machine learning applications as it teaches computer programs how to extract data patterns and empathize variations within the data. The applications are endless, and we have yet to really witness the depths of machine learning capabilities. Not only for computer engineering and computer science, but all fields of math, science, engineering will benefit from computer vision related technologies.

I have learned more than I would have trying to learn about this any other way than writing a research paper and reading up on the core math involved. Hopefully I will be able to apply this in my senior design project and continue to specialize in this field.

Reference List

Kokiopoulou, E. K., & Saad, Y. S., (2006, March 21). Orthogonal Neighborhood Preserving Projections: A projection-based dimensionality reduction technique. Retrieved from https://www-users.cs.umn.edu/~saad/PDF/umsi-2006-16.pdf

Lata, Y.V. L., Tungathurthi, C. K. B. T., Rao, H. R. M. R., Govardhan, A. G., & Reddy, L. P. R., (2009, May). Facial Recognition using Eigenfaces by PCA. *International Journal of Recent Trends in Engineering, Vol. 1, No. 1*. Retrieved from https://www.researchgate.net/publication/228748710_Facial_recognition_using_eigenfaces_by_PCA

Open Source Computer Vision. (2015). *Introduction to Principal Component Analysis (PCA).* Retrieved from https://docs.opencv.org/3.1.0/d1/dee/tutorial_introduction_to_pca.html

Paul, L. C. P., & Sumam, A. A. S., (2012, November). Face Recognition Using Principal Component Analysis Method. Retrieved from http://ijarcet.org/wp-content/uploads/IJARCET-VOL-1-ISSUE-9-135-139.pdf

Smith, L. I. S., (2002, February 26). A tutorial on Principal Components Analysis. Retrieved from http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

Spruyt, V. S., (2014, April 24) A geometric Interpretation of the Covariance Matrix. Retrieved from http://www.visiondummy.com/2014/04/geometric-interpretation-covariance-matrix/