Principle Components Analysis

Aaron Avram

February 24 2025

Introduction

Principle Components Analysis (PCA) is a way to apply lossy compression to a dataset by uniformly reducing the dimensionality of each data point. An encoder function is used to embed each data point into the lower dimensional space. Then, a decoder function is used to reconstruct the original data point from its embedding. Clearly, there will be some deviation between the reconstructed and original data points. The purpose of PCA is to find the optimal encoder and its corresponding decoder for the any given dataset and embedding space.

Setup

Consider a set of m data points, each with n dimensions: $\{x^{(1)}, \dots, x^{(m)}\}$. Construct $X \in M_{m,n}(\mathbb{R})$ such that $X_{i,:} = x^{(i)T}$. Now suppose we want to embed X into a l-dimensional space, where each $x^{(i)}$ is mapped to a vector in this l-dimensional space, denoted $c^{(i)}$.

Finding The Optimal Encoder

First we will derive the optimal encoder function given an arbitrary decoder function. Label our given decoder function $g : \mathbb{R}^l \to \mathbb{R}^m$. We would want this function to be linear, so as to preserve some of the structure in the original dataset. Thus we can work with the matrix representation of g in the standard basis, which we will denote D, which for simplicity will have orthonormal columns. This is an obvious choice that does not lose any of the algorithm's generality. The optimal decoder function is then the matrix that maps a given input vector x to the point c^* such that:

$$c^* = \arg\min_c ||x - Dc||_2$$

Which minimizes the euclidean distance between the decoded value of c and the original point x. Note that squaring the function being minimized, does not change the optimal coding point, as $f(x) = x^2$ is increasing for non-negative inputs. Thus, we can substitute the euclidean distance squared and perform some manipulations to simplify the equation:

$$
\begin{aligned}
c^* &= \arg\min_c ||x - Dc||_2^2 \\
&= \arg\min_c [(x - Dc)^T(x - Dc)] \\
&= \arg\min_c [x^Tx - x^TDc - (Dc)^Tx + (Dc)^T(Dc)] \\
&= \arg\min_c [x^Tx - 2x^TDc + c^TD^TDc] \\
&= \arg\min_c [x^Tx - 2x^TDc + c^TI_lc] \\
&\text{(By the fact that the columns of D are orthonormal)} \\
&= \arg\min_c [x^Tx - 2x^TDc + c^Tc]
\end{aligned}
$$

To find c^* we compute the value of c such that the gradient of the function being minimized is 0:

$$
\begin{aligned}
0 &= \nabla_c(x^Tx - 2x^TDc + c^Tc) \\
0 &= -2D^Tx + 2c \\
c &= D^Tx
\end{aligned}
$$

Thus the optimal encoding function is $f(x) = D^Tx$, where D is the matrix representation of the decoder function.

Finding The Optimal Encoder

To find the optimal encoder function, we want to find a matrix D that minimizes the error produced by encoding a vector then decoding it. I.e. we want to find:

$$
\begin{aligned}
D^* &= \arg\min_D \sqrt{\sum_{ij}(x_j^{(i)} - (DD^Tx^{(i)})_j)^2} \\
&\text{(This is equivalent of taking the squared Frobenius norm of } X - XDD^T\text{)} \\
&= \arg\min_D ||X - XDD^T||_F^2
\end{aligned}
$$

Since the Frobenius norm of a matrix A is equivalent to the root of the trace of A^TA we have:

$$
\begin{aligned}
D^* &= \arg\min_D \operatorname{Tr}((X - XDD^T)^T(X - XDD^T)) \\
&= \arg\min_D \operatorname{Tr}(X^TX) - \operatorname{Tr}(X^TXDD^T) - \operatorname{Tr}(DD^TX^TX) + \operatorname{Tr}(X^TXDD^TDD^T) \\
&\text{(Because the first term depends only on X, we can omit it, as it will not affect the optimal D value)} \\
&= \arg\min_D -2\operatorname{Tr}(X^TXDD^T) + \operatorname{Tr}(X^TXDD^TD^T) \\
&= \arg\min_D -\operatorname{Tr}(X^TXDD^T) \\
&= \arg\max_D \operatorname{Tr}(D^TX^TXD)
\end{aligned}
$$

Thus we want to maximize:

$$D^* = \text{Tr}(D^T X^T X D)$$

To maximize this equation, we proceed by induction on l.

Base Case: $l = 1$ This equation thus reduces to solving for the optimal vector d^* such that:

$$d^* = \arg\max_d d^T X^T X d$$
$$\text{(Since the trace of a scalar is just itself)}$$

For simplicity let us label $X^T X$, as A, which is clearly a symmetric matrix. We can solve for d^* by finding the value of d for which the gradient of the equation is 0:

$$0 = \nabla_d (d^T A d)$$

However, we know that $d^T d = 1$, as d has unit norm. Thus, it lies on the unit disc and we can use the Lagrangian to solve for the optimal value of d

$$\begin{aligned} \mathcal{L}(d, \lambda) &= d^T A d - \lambda(d^T d - 1) \\ &\text{Setting the gradient of the Lagrangian to 0:} \\ 0 &= \nabla_d (d^T A d - \lambda(d^T d - 1)) \\ 0 &= 2Ad - 2\lambda d \\ \lambda d &= Ad \\ \lambda d &= X^T X d \end{aligned}$$

Thus the optimal solution to the equation is when d is an unit eigenvector of $X^T X$. Specifically, the one with the largest eigenvalue.

Inductive Step Suppose that for a given $l \in \mathbb{N}$ the optimal decoder matrix D has the unit eigenvectors with the largest eigenvalues, in decreasing order from left to right, as its columns. Now consider $l + 1$.

As before, we want to optimize the equation:

$$D^* = \arg\max_D \text{Tr}(D^T X^T X D)$$

Now consider:

$$\begin{aligned} \text{Tr}(D^T X^T X D) &= \sum_{i=1}^{l+1} (D^T X^T X D)_{ii} \\ &= \sum_{i=1}^{l+1} (d^{(i)^T} X^T X d^{(i)} \\ &= \sum_{i=1}^{l} (d^{(i)^T} X^T X d^{(i)} + d^{(l+1)^T} X^T X d^{(l+1)} \end{aligned}$$

Since the columns of D are orthonormal, the two summands can be optimized independently. The optimization of the first summand, follows from the inductive hypothesis. The second follows from the base case, except that the maximal eigenvalue is now the the unit eigenvector with the largest eigenvalue of $X^T X$ that is linearly independent with the columns of the optimal solution of the first summand. Thus, we see that the optimal decoder matrix D has the first $l + 1$ unit eigenvectors with the largest eigenvalues, in decreasing order from left to right, as its columns.

Conclusions

Thus, we have found the formula for the optimal encoder and decoder functions for a given dataset X. One note, PCA tends to perform better when applied on normalized data. This is because normalization prevents any difference in the scale of the data from skewing the value of each datapoint in the encoding space.