Singular Value Decomposition

Mirza Mohammad Azwad

BSc in Software Engineering, Computer Science and Engineering(CSE) Department, Islamic University of Technology(IUT), Board Bazar, Gazipur, Dhaka

Objectives

This poster is to give a brief overview of the idea behind Singular Value Decomposition(SVD) along with a few key insights as to how it is applied in real life.

- Introduction to Eigen Values and Eigen Vectors
- Introduction to Diagonalization
- Introduction to Singular Value Decomposition(SVD)
- Practical Application of SVD: Facial recognition

Introduction to Eigen Values and Eigen Vectors

Eigenvalues and EigenVectors are one of the key takeaways from linear algebra. Eigen topics are essentially the power-up of linear transformations, determinants, linear systems, and change of basis. Eigenvectors are essentially vectors whose direction(span) remains unchanged upon a particular linear transformation. Eigenvalues are the values associated with these eigenvectors. Let us take an example to better understand the idea of eigenvectors. Take matrix A, $\begin{bmatrix} 1 & 4 \\ -4 & -7 \end{bmatrix}$ In order to find its eigenvalue, we find the determinant(det) such that $|A - \lambda I| = 0$ Which in this case is: $\begin{vmatrix} 1-\lambda & 4 \\ -4 & -7-\lambda \end{vmatrix}$ This can then be evaluated as: $(1-\lambda)(-7-\lambda)+16 = 0$
$\lambda = -3$. Hence, the corresponding eigenvectors can be calculated as the basis for the possible matrices of $A - \lambda I$ that being $\begin{bmatrix} 4 & 4 \\ -4 & -4 \end{bmatrix}$ which is given by $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ so any vector k$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ can be an eigenvector in this case.

[a] Explanation for Eigen values, Eigen vectors, Diagonalization and SVD taken from Introduction to Linear Algebra by Gilbert Strang

Diagonalization

Diagonalization is the process of factorizing a matrix such that $A = Q\Lambda Q^{T}$ The Q here represents orthogonal matrices. A property of Orthogonal matrices is that Q^{T} represents Q^{-1}. And Λ represents an Eigen matrix with the eigenvalues placed along its diagonal, preferably with the largest eigenvalue on the left-hand side. To diagonalize the matrix we first find the eigenvalues of the matrix followed by the eigenvectors. Let us say that we have eigenvalues $\lambda_1, \lambda_2, \lambda_3, ..., \lambda_n$ and the eigen vectors $x_1, x_2, ..., x_n$ then the matrices Q is the orthogonal matrix represented by $[x_1 \ x_2 \ ...]$ and Λ is given by $\begin{vmatrix} \lambda_1 & 0 & 0 & ... \\ 0 & \lambda_2 & 0 & ... \\ 0 & 0 & \lambda_3 & ... \\ ... & ... & ... & ... \end{vmatrix}$ To ensure that Q is orthogonal make sure only the unit vectors make up the columns, and then take the transpose as Q^{T} and thus we can find the required decomposition. Note that the diagonalization process only works for square matrices.

Singular Value Decomposition(SVD)

SVD is like the elder brother of diagonalization and is applicable even for rectangular matrices. The process goes as follows; let us say we have a matrix A, then we find $A^{T}A$ and AA^{T}. A fundamental property to note here is that multiplication of the matrix and its transpose yields a symmetric square matrix(Positive definite matrix, λ_i>0). Using this property, we can now diagonalize the matrix. Upon diagonalization, $A^{T}A$ yields $V\Lambda V^{T}$. Now taking into account the eigenvectors $v_1, v_2, ..., v_n$ and the eigenvalues $\lambda_1, \lambda_2, ..., \lambda_n$ We now introduce another term Σ, which is essentially a modification of Λ such that $\sigma^2 = \lambda$. In Σ the arrangement is similar to that of Λ, with one additional fact that being that the σs are arranged with the highest first, or in other words, $\sigma_1 > \sigma_2 > ... > \sigma_n > 0$. The eigenvectors v are called the right singular vectors.

SVD

[CONTINUED]This introduces us to another property, $Av = \sigma u$ and $A^T u = \sigma v$ where u is the left singular vectors obtained from the diagonalization of AA^{T}. Taking these into account, we can determine the left singular vectors, u, by $u_i = \frac{Av_i}{\sigma}$. The left singular vectors can be used to form the left singular matrix U, with the order of the left singular vectors matching that of σ s, and the right singular vectors can also be arranged according to the same order of σ s, which can be transposed to form the right singular matrix V. Hence, decomposing the matrix to form A=$U\Sigma V^T$. Following the idea of SVD, we come across something called **Principal Component Analysis(PCA)**. The idea is quite simple upon matrix multiplication of the individual parts we get the terms as $u_i\sigma_i v_i^T$ from i=1 to i=n and the previously explained idea of $\sigma_1 > \sigma_2 > ... > \sigma_n > 0$ comes into play here as the term $u_1\sigma_1 v_1^T$ holds more weight than the other terms so it is referred to as the **Principal Component**.

Results of Facial Recognition

True Title	Precision
Ariel Sharon	0.86
Colin Powell	0.80
Donald Rumsfeld	0.85
George W Bush	0.82
Gerhard Schroeder	0.95
Hugo Chavez	1.00
Tony Blair	1.00
accuracy	
macro avg	0.90
weighted avg	0.86

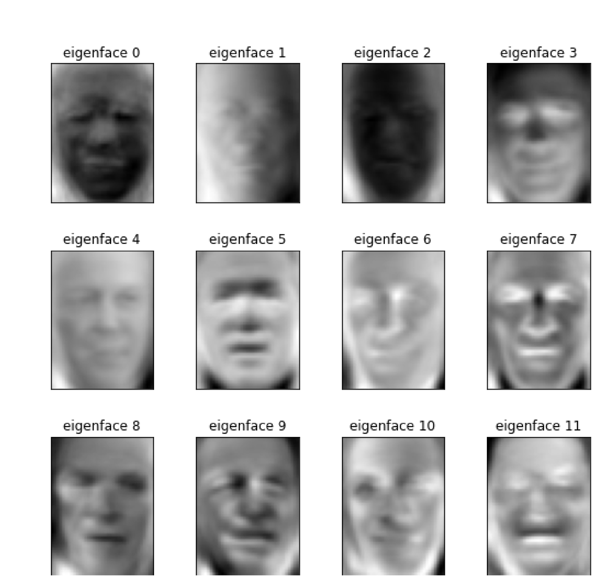
Table 1:Precision upon testing as per the training model

[b] Reference: Eigen Faces explanation obtained from https://heartbeat.comet.ml/

Eigen Faces

The Eigenfaces idea is used in computer vision to work with facial recognition. We have a collection of faces in the dataset. These faces all have some common abstract ingredients which are essentially eigenvectors that can be used as the basis for all these images with these images being a linear combination of the basis. We find the eigenvectors using PCA. We then perform dimensionality reduction so that the smaller set of basis images represents the original training images. After this we can record the difference between a particular image with the available eigenface and upon the computing and observing the minimum difference we can find how much an image is similar or different to the training data available. The idea of eigenfaces is what lays the foundation of facial recognition in modern systems. Further explanation can be shown in the code provided as footenote to this column.

Figure 1:Test Data



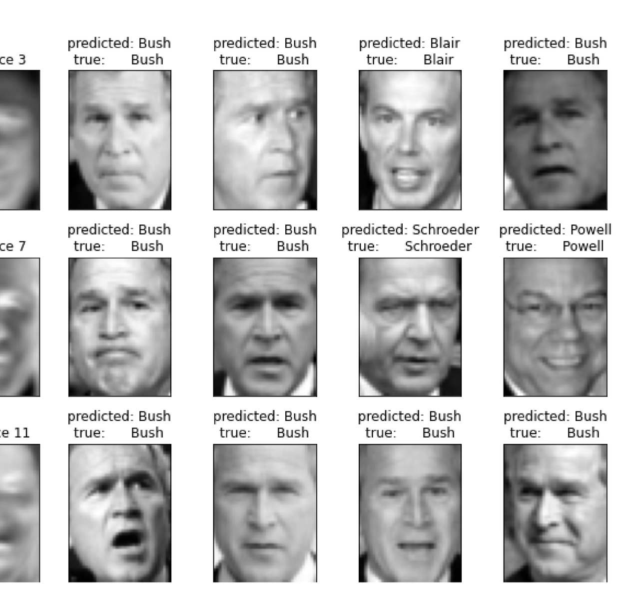


Figure 2 Eigen Face Figure 3 Results

[c] Facial Recognition using Eigen Faces Example Code View on Google Colab