CPL Project

Singular Value Decomposition (SVD)

Name: **Kavin A S B**; Roll No.: 1911085

Date of submission: November 21 , 2021

1 Terminology and Theorem

Given A $\in R^{m\times n}$, there exist orthogonal matrices U $\in R^{m\times m}$ and V $\in R^{n\times n}$, and a diagonal matrix $\Sigma \in R^{m\times n}$ with non-negative diagonal entries such that

$$A_{m\times n} = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n} \tag{1}$$

(if there are complex matrices, transpose can be replaced with complex conjugate) where U, V are unitary matrices. So,

$$U^T U = I_{m\times m} \tag{2}$$

$$V^T V = I_{n\times n} \tag{3}$$

The diagonal entries of Σ are called the singular values of A. The columns of U are called the left singular vectors, and those of V are called the right singular vectors. The singular values are unique, but U and V are not unique.
The number of nonzero singular values is equal to the rank of the matrix A.

2 Theory

The SVD represents an expansion of the original data in a coordinate system where the covariance matrix is diagonal. Calculating the SVD consists of finding the eigenvalues and eigenvectors of AA^T and $A^T A$. The eigenvectors of $A^T A$ make

up the columns of V , the eigenvectors of AA^T make up the columns of U. Also, the singular values in S are square roots of eigenvalues from AA^T or A^TA. The singular values are the diagonal entries of the S matrix and are arranged in descending order. The singular values are always real numbers. If the matrix A is a real matrix, then U and V are also real.
Here, we chose to find V by computing A^TA. We use an approximate method to compute eigenvalues of a diagonalizable matrix called the "Power method".

$$A^TA = (U\Sigma V^T)^T(U\Sigma V^T) = (V\Sigma U^T)(U\Sigma V^T) = V\Sigma^2V^T \tag{4}$$

So we can completely eliminate U from the discussion, and look at just V Σ^2V^T. And what's nice about this matrix is that we can compute its eigen-vectors, and these turn out to be exactly the singular vectors. The corresponding eigenvalues are the squared singular values. If you apply (V Σ^2V^T) to any v_i(singular vector), the only parts of the product that are non-zero are the ones involving v_i with itself, and the scalar σ_i^2 factors in smoothly.
Call $V\Sigma^2V^T$ = B. Let us choose a random vector x. Expanding it the singular vector basis v_i of B, we get $x = c_iv_i$. Let σ_i be the corresponding eigenvalue of v_i. Applying B 'k' times to the above equation and taking dot product with v_1, we can see that after some iterations, the x vector will shift to v_1 since the coefficient corresponding to the first singular vector dominates all of the others. And so if we normalize, the coefficient of B^kx corresponding to v_1 tends to 1, while the rest tend to zero. We won't compute a precise number of iterations. Instead we'll just compute until the angle between $B^{k+1}x$ and B^kx is very small(in the order of ϵ).i.e. we start with a random unit vector x, and then loop computing $x_{s+1} = Bx_s$, re-normalizing at each step. The condition to stop is that the magnitude of the dot product between x_s and x_{s+1} is very close to 1.
We started with the matrix A and computed v_1. We can use v_1 to compute u_1 and $\sigma_1(A)$. Now, let us ignore all vectors in the span of v_1 for our next step, and to do this we can simply subtract the rank 1 component of A corresponding to v_1. i.e., set A' = A - $\sigma_1(A)u_1v_1^T$. Then, we see that $\sigma_1(A') = \sigma_2(A)$ and basically all the singular vectors shift indices by 1 when going from A to A'. Then we can repeat this recursively to get U, V and diagonal elements of Σ. While doing this computation, we assume that $m > n$ by taking n columns as variables and m rows as values for a set of measurement.

References

[1] http://www.cs.yale.edu/homes/el327/datamining2013aFiles/07_singular_value_decomposition.pdf

[2] https://gregorygundersen.com/blog/2018/12/20/svd-proof/#:~:text=The%20existence%20claim%20for%20the,existence%20proof%20for%20the%20SVD

[3] https://www.itl.nist.gov/div898/handbook/pmc/section5/pmc541.htm

[4] https://ergodic.ugr.es/cphys/LECCIONES/FORTRAN/power_method.pdf

4