# Singular Value Decomposition (SVD) Transform

## SVD Transform

Let the 2D image array be represented by a matrix $A = [a_{ij}]_{M \times N}$ with rank equal to $R$. Here we assume $R \leq M \leq N$. Now we consider the follow eigenvalue problems for $A^T A$ and $AA^T$:

$$AA^T u_i = \lambda_i u_i$$

$$A^T A u_i = \lambda_i v_i$$

where $\lambda_i$, $(i = 1, 2, \cdots, R)$ are the eigenvalues of both $AA^T$ and $A^T A$, and $u_i$ $(i = 1, 2, \cdots, R)$ and $v_i$, $(1, 2, \cdots, R)$ are the eigenvectors of $AA^T$ and $A^T A$, respectively.

Since these matrices are symmetric, their eigenvectors are orthogonal:

$$u_i^T u_j = v_i^T v_j = \delta_{ij}$$

and they form two orthogonal matrices $U = [u_1, \cdots, u_N]_{M \times M}$ and $V = [v_1, \cdots, v_N]_{N \times N}$ and

$$UU^T = U^T U = I$$

$$VV^T = V^T V = I$$

As there exist only $R$ non-zero eigenvalues, $\lambda_i > 0$, $(i = 1, \cdots, R)$, both matrices $U$ and $V$ have some zero columns and the matrix $I$ has only $R$ non-zero diagonal elements.

$U$ and $V$ will diagonalize $AA^T$ and $A^T A$ respectively:

$$U^T (AA^T) U = \Lambda_{M \times M} = diag[\lambda_1, \cdots, \lambda_R]$$

and

$$V^T (A^T A) V = \Lambda_{N \times N} = diag[\lambda_1, \cdots, \lambda_R]$$

From linear algebra, we know that the *singular value decomposition (SVD)* of $A$ is defined as:

$$U^T A V = \Lambda^{1/2} = diag[\ \sqrt{\lambda_1}, \cdots, \sqrt{\lambda_R}\ ]$$

This can be considered as the forward SVD transform and the inverse transform is

$$A = U\Lambda^{1/2}V^T = \sum_{i=1}^{R} \sqrt{\lambda_i}[u_i v_i^T]$$

This inverse SVD transform can be so interpreted that the original image matrix is decomposed into a set of $R$ *eigenimages* $\sqrt{\lambda_i}[u_i v_i^T]$, where the outer product $u_i v_i^T$ is an M by N matrix.

SVD transform pair:

$$\begin{cases} U^T A V = \Lambda^{1/2} \\ A = U\Lambda^{1/2}V^T \end{cases}$$

# Conservation of Degrees of Freedom

We now show that the degrees of freedom (d.o.f., the number of independent variables in the signal) are conserved in the SVD transform (same as any other transforms). In spatial domain the d.o.f. of the image matrix $A_{N\times N}$ (assuming $M = N = R$ for simplicity) is $N^2$. Now in the transform domain, both $U$ and $V$ have the same d.o.f. $(N^2 - N)/2$ for the following reason. The first column vector has $N$ elements subject to normalization, i.e., $N-1$ d.o.f.; the second vector is the same except it also has to be orthogonal to the first one, and therefore has $N-2$ d.o.g.; the third vector has to be orthogonal to the first two vectors and therefore has $N-3$ d.o.f.; etc. Now the total d.o.f. of all $N$ vectors are

$$(N-1) + (N-2) + \cdots + 1 = (N^2 - N)/2$$

Together with the $N$ d.o.f. in $\Lambda$, the over all d.o.f. is

$$2\ (N^2 - N)/2 + N = N^2$$

This indicates that the signal, in either the original spatial domain ($A$) or the transform domain ($\Lambda$, $U$ and $V$), always has the same degrees of freedom.

# Application in Image Compression

SVD image transform has different applications in image processing and analysis. We now consider how it can be used for data compression. First we write a matrix $A$ as

$$A = [a_1, \cdots, a_N]_{M \times N}$$

where $a_i = [a(1,i), \cdots, a(M,i)]^T$ is the ith column vector of $A$. Now consider the *norm* of $A$ defined as

$$\begin{aligned} \parallel A \parallel \triangleq tr\ [A^T A] &= tr \begin{bmatrix} a_1^T \\ \cdots \\ \cdots \\ a_N^T \end{bmatrix} [a_1, \cdots, a_N] = tr \begin{bmatrix} a_1^T a_1 & \cdots & \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & a_N^T a_N \end{bmatrix} \\ &= \sum_{i=1}^{N} a_i^T a_i = \sum_{i=1}^{N} \sum_{j=1}^{M} a^2(j,i) \end{aligned}$$

This shows that $\parallel A \parallel$, the norm of $A = [a_{ij}]$, represents the total energy contained in a matrix $A$.

Now we consider image compression achieved by using only the first $k$ eigenimages of the given image $A$:

$$A_k \triangleq \sum_{i=1}^{k} \sqrt{\lambda_i} u_i v_i^T$$

with error

$$E_k = A - A_k = \sum_{i=k+1}^{R} \sqrt{\lambda_i} u_i v_i^T$$

After compression, the energy (information) contained in $A_k$ is:

$$\begin{aligned} \parallel A_k \parallel &= tr\ [A_k^T A_k] = tr[\sum_{i=1}^{k} \sqrt{\lambda_i} v_i u_i^T][\sum_{j=1}^{k} \sqrt{\lambda_j} u_j v_j^T] \\ &= tr[\sum_{i=1}^{k} (\sum_{j=1}^{k} \sqrt{\lambda_i} \sqrt{\lambda_j} v_i u_i^T u_j v_j^T)] = tr[\sum_{i=1}^{k} \lambda_i v_i v_i^T] \\ &= \sum_{i=1}^{k} \lambda_i tr\ [v_i v_i^T] = \sum_{i=1}^{k} \lambda_i v_i^T v_i = \sum_{i=1}^{k} \lambda_i \end{aligned}$$

We can also see that the total amount of energy (information) contained in the original image $A$ is

$$\text{Information in } A = \sum_{i=1}^{R} \lambda_i$$

and the energy (information) lost (contained in $E_k$) is

$$\text{Information in } E_k = \sum_{i=k+1}^{R} \lambda_i$$

It is therefore obvious that minimum energy is lost if we range $\lambda_i$'s so that

$$\lambda_1 \geq \cdots \geq \lambda_k \geq \cdots \geq \lambda_R$$

To find out the data compression ratio, consider total degrees of freedom in $A_k$:

$$A_k = \sum_{i=1}^{k} \sqrt{\lambda_i} u_i v_i^T$$

The degrees of freedom in the k vectors $\{u_i \;\; i = 1, \cdots, k\}$ are

$$(N-1) + (N-2) + \cdots + (N-k) = Nk - k(k+1)/2$$

The same is true for $\{v_i \;\; i = 1, \cdots, k\}$. Including the k degrees of freedom in $\{\lambda_i, \;\; i = 1, \cdots, k\}$, we have the total

$$k + 2Nk - k(k+1) = 2Nk - k^2$$

and the compression ratio is

$$\frac{2Nk - k^2}{N^2} = 2\frac{k}{N} - (\frac{k}{N})^2$$