

## 1 Singular Value Decomposition

Remember that that *rank* of a matrix  $A \in \mathbb{R}^{m \times n}$  is equal to to any of the following equivalent things:

1. The number of linearly independent rows of  $A$ .
2. The number of linearly independent columns of  $A$ .
3. The number of non-zero eigenvalues of  $A$  if  $m = n$ , or the number of non-zero singular values of  $A$  if  $m \neq n$ .

If a matrix is rank  $\ell$ , then it can be written as a sum of  $\ell$  rank 1 matrices. For any rank 1 matrix  $A \in \mathbb{R}^{m \times n}$ , we can write  $A = uv^T$  for  $u \in \mathbb{R}^m, v \in \mathbb{R}^n$ .

Notice that eigenvalues are only defined for matrices  $A \in \mathbb{R}^{n \times n}$ . If  $A \in \mathbb{R}^{m \times n}$  for  $m \neq n$ , then  $Ax$  cannot possibly equal  $\lambda x$  because  $Ax \in \mathbb{R}^m$  and  $x \in \mathbb{R}^n$ . For matrices with  $m \neq n$ , the analog of an eigenvalue is a *singular value*.

Before we talk more about singular values, recall the spectral theorem for symmetric matrices we introduced earlier in the course:

**Theorem 1.1** (Spectral Theorem). *Let  $A \in \mathbb{R}^{n \times n}$  be a symmetric matrix. Then, there are  $n$  eigenvalues  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  with corresponding orthonormal eigenvectors  $v_1, \dots, v_n$  so that*

$$A = \sum_{i=1}^n \lambda_i v_i v_i^T = V \Lambda V^T$$

where  $V$  has columns  $v_1, \dots, v_n$  (so  $V^T V = I$ ) and  $\Lambda$  is the diagonal matrix with  $\Lambda_{ii} = \lambda_i$ .

A natural question is: what about for general matrices  $A \in \mathbb{R}^{m \times n}$ , or for matrices in  $\mathbb{R}^{n \times n}$  which are not symmetric? It turns out all you need to do is turn  $A$  into a symmetric matrix using  $A^T A$ , and then apply the spectral theorem.

**Theorem 1.2** (Singular Value Decomposition (SVD)). *Let  $A \in \mathbb{R}^{m \times n}$ . Then there are orthonormal vectors  $u_1, \dots, u_\ell$ , orthonormal vectors  $v_1, \dots, v_\ell$ , and singular values  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_\ell$  such that:*

$$A = \sum_{i=1}^{\ell} \sigma_i u_i v_i^T = U \Sigma V^T$$

In addition, we have  $\ell \leq \min\{m, n\}$  and  $\sigma_i \in \mathbb{R}_{>0}$  for all  $i$ . (And here  $U$  is the matrix with columns  $u_1, \dots, u_\ell$  and  $V$  the matrix with columns  $v_1, \dots, v_\ell$ .)

The  $u_i$  are called the left singular vectors and the  $v_i$  the right singular vectors. There are a few differences in this statement compared to the spectral theorem. The most important difference is that instead of  $v_i v_i^T$ , we have  $u_i v_i^T$ , i.e., these vectors can differ. Second, we typically list the singular values in decreasing order, whereas eigenvalues are in increasing order. Finally, singular values are all positive: we can negate  $u_i$  to flip the sign of the singular value, and then by convention we just delete the singular values of value 0.

To get a handle on why things are a bit different, consider the following matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

This is a rank 1 matrix. But it's clear you can't write it in the form  $\lambda v v^T$ . So, we need to relax the

criteria: we need to write it in the form  $\lambda u v^T$ , which is easy: just pick  $u = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$  and  $v = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ .

The SVD gives us the following important fact: **every matrix can be written as the sum of rank 1 matrices**. Now let's prove it.

*Proof of SVD.*  $A^T A \in \mathbb{R}^{n \times n}$  is PSD since by definition it has a square root. Since it is symmetric, we can apply the spectral theorem, so there are non-negative eigenvalues  $\lambda_1, \dots, \lambda_n$  and orthonormal  $v_1, \dots, v_n$  such that  $A^T A = \sum_{i=1}^n \lambda_i v_i v_i^T$ . First, throw away all  $\lambda_i$  which are 0 so that  $A^T A = \sum_{i=1}^\ell \lambda_i v_i v_i^T$  after re-indexing for some  $\ell \leq \min\{n, m\}$  since the rank of the matrix is at most  $\min\{m, n\}$ . Let  $v_1, \dots, v_\ell$  be the orthonormal set above. First, notice that

$$\|A v_i\|_2^2 = v_i^T A^T A v_i = v_i^T \lambda_i v_i = \lambda_i$$

since  $v_i$  is an eigenvector of  $A^T A$ . So,  $\|A v_i\|_2 = \sqrt{\lambda_i}$ . Now, define for all  $i$ :

$$u_i = \frac{A v_i}{\|A v_i\|_2} = \frac{A v_i}{\sqrt{\lambda_i}} = \frac{A v_i}{\sigma_i}$$

In other words, set  $\sigma_i = \sqrt{\lambda_i}$ . By definition, these vectors have norm 1. Furthermore, they are orthonormal, because when  $i \neq j$  we have:

$$u_i^T u_j = \frac{(A v_i)^T A v_j}{\sigma_i \sigma_j} = \frac{v_i^T A^T A v_j}{\sigma_i \sigma_j} = \frac{v_i^T \lambda_j v_j}{\sigma_i \sigma_j} = 0$$

where we used that the  $v_i$  are orthonormal. So, the only thing remaining to prove is that  $A = \sum_{i=1}^\ell \sigma_i u_i v_i^T$ . It suffices to prove that  $A v_j = (\sum_{i=1}^\ell \sigma_i u_i v_i^T) v_j$  for all  $1 \leq j \leq n$  for the basis  $v_1, \dots, v_n$ . This is enough, because then:

$$A x = A \left( \sum_{j=1}^n v_j \langle x, v_j \rangle \right) = \sum_{j=1}^n A v_j \langle x, v_j \rangle = \sum_{j=1}^n \sum_{i=1}^\ell (\sigma_i u_i v_i^T) v_j \langle x, v_j \rangle = \sum_{i=1}^\ell \sigma_i u_i v_i^T \sum_{j=1}^n v_j \langle x, v_j \rangle$$

and this is  $(\sum_{i=1}^{\ell} \sigma_i u_i v_i^T)x$ . So, since they have the same product with every vector, they are the same matrix. So let's prove that  $Av_j = (\sum_{i=1}^{\ell} \sigma_i u_i v_i^T)v_j$  for all  $1 \leq j \leq n$  for the basis  $v_1, \dots, v_n$ .

$$(\sum_{i=1}^{\ell} \sigma_i u_i v_i^T)v_j = \sum_{i=1}^{\ell} \sigma_i u_i \langle v_i, v_j \rangle = \sigma_j u_j v_j^T v_j = \sigma_j u_j = \sigma_j \frac{Av_j}{\sigma_j} = Av_j \quad \square$$

## 2 Low Rank Approximation

A common task is to take a matrix  $A \in \mathbb{R}^{m \times n}$  and find a new *low rank* matrix that approximates  $A$ . This has many applications:

1. The most tangible application is image compression. Given a matrix of pixels, we can use a low-rank approximation to compress the image.
2. In recommendation systems, we have a matrix encoding user ratings. For example, suppose each row is a user, and each column is a movie. The entry is 0 if the user has not rated the movie, and otherwise is some integer rating, perhaps 1 if they liked it and  $-1$  otherwise. Now, we want to figure out what movies a particular user will like. It turns out a pretty good approach here is to find a low rank approximation of this matrix. For example, if everyone likes exactly the same movies (all movies are just good or bad), that's a rank 1 matrix since every row is the same. If that is the ground truth, then a rank 1 approximation is likely to figure this out.

Unsurprisingly, the best way to approximate a matrix  $A$  with a matrix of rank  $k$  is to use the top  $k$  singular values. There is even a theorem:

**Theorem 2.1.** *Let  $A \in \mathbb{R}^{m \times n}$ . Then, an optimal rank  $k$  approximation of  $A$  can be obtained by taking the top  $k$  singular values of  $A$ , i.e.,  $\tilde{A} = \sum_{i=1}^k \sigma_i u_i v_i^T$ . Formally:*

$$\inf_{\text{rank } \tilde{A} = k} \|A - \tilde{A}\|_2 = \sigma_{k+1}$$

where recall  $\|A\|_2$  for a matrix  $A$  is the spectral norm, or its largest singular value.

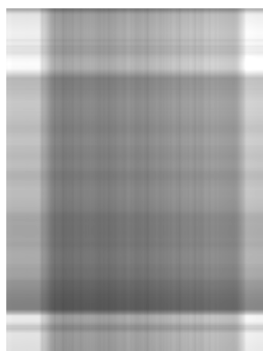
We will prove that  $\|A - \tilde{A}\| \leq \sigma_{k+1}$  by showing  $\tilde{A} = \sum_{i=1}^k \sigma_i u_i v_i^T$  achieves this. We will leave the other direction as an exercise. Notice that:

$$A - \tilde{A} = \sum_{i=k+1}^{\ell} \sigma_i u_i v_i^T$$

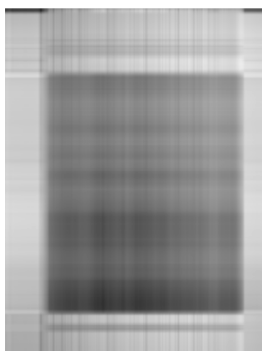
So,  $\|A - \tilde{A}\|_2 = \sigma_{\max} \sum_{i=k+1}^{\ell} \sigma_i u_i v_i^T = \sigma_{k+1}$ .

On your homework, you will visualize the effects of low rank approximation on images to produce something like the below. Naïvely, this is an image of about  $1000 \times 1000$  pixels, which would require about half a megabyte if every pixel has 32 different possible values. To store a rank 60 version would require about a tenth of that.

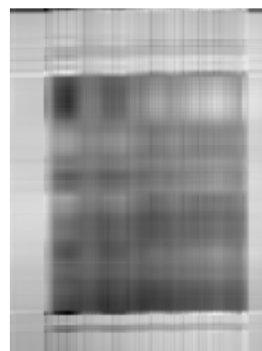
rank 1



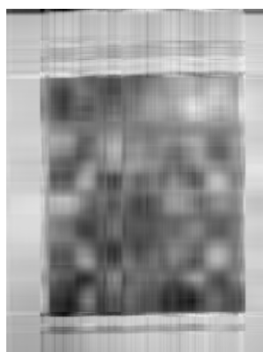
rank 2



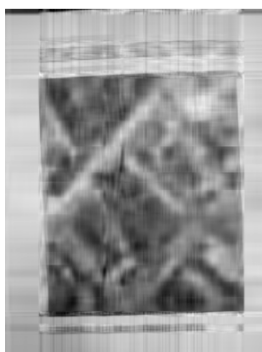
rank 3



rank 5



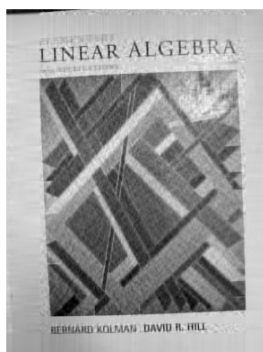
rank 10



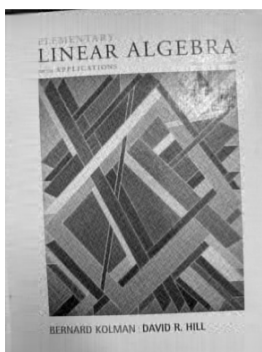
rank 20



rank 40



rank 60



rank 100

