

# SVD and QR method

Math 573 Fall 2021

Cesar Ramirez Ibañez

October 2021

## 1 QR algorithm for eigenvalues, Big Picture First

Ok, so now to the big picture of what the QR method for eigenvalues does: We want to find eigenvalues through successive QR factorizations or a “QR sequence” of factorizations: it builds a sequence  $\{A_i\}$  of matrices such that  $A_i$  as  $i \rightarrow \infty$  converges to a matrix with easy eigenvalues to find:

It converges to a matrix that is block upper triangular (when the matrix has complex conjugate pair eigenvalues, for example) and from which one can easily determine the eigenvalues.

In particular, for a special case, the notes prove the following result, though note the restrictive hypotheses required:

**Theorem 1.** *Suppose that  $A$  has eigenvalues  $\lambda_1, \dots, \lambda_n$  satisfying (strict inequalities here)*

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|.$$

*Let  $X$  denote the matrix whose  $i$ th column is an eigenvector of  $A$  corresponding to  $\lambda_i$  and suppose that  $X^{-1}$  has an LU decomposition. Then the subdiagonal elements of the matrices  $A_s$  of the basic QR algorithm tend to zero, and as for the diagonal elements:*

$$(A_s)_{kk} \rightarrow \lambda_k \quad \text{as } s \rightarrow \infty.$$

**Remember, the previous result is restrictive in its hypotheses, but it is relatively easy to understand how it works, that’s why it is given in the notes. But in general, instead of converging to an upper triangular matrix, one converges to a block upper triangular, or quasi upper triangular matrix.**

Let us mention first that in general QR type methods basically use the theoretically important Householder reflections to get the QR factor matrices. We first apply the reflections to get the upper Hessenberg form of a matrix  $A$ . Then to this upper Hessenberg form, we now use Householder reflections to successively obtain QR factorizations of a sequence of matrices which depend on previous QR factorizations. The general outline is:

## QR algorithm for eigenvalues (BIG PICTURE)

- (i) Reduce  $A$  to upper Hessenberg form. Rename this Hessenberg matrix as  $A$  and define  $A_0 := A$ . On this upper Hessenberg matrix  $A_0$ , apply the next step.
- (ii) Apply LOOP part of successive QR factorizations to obtain a sequence of matrices  $\{A_i\}$  where  $A_i = R_{i-1}Q_{i-1}$  and  $Q_{i-1}$  and  $R_{i-1}$  are QR decompositions of the previous matrix  $A_{i-1} = Q_{i-1}R_{i-1}$ . This LOOP part looks like

Listing 1: LOOP

---

```

A_0:=A

WHILE A_i not block upper triangular enough (according to tolerance) DO
  GET Q_i, R_i such that A_i=Q_i R_i
  DEFINE A_{i+1}:=R_i Q_i
END

```

---

- (iii) After some tolerance criterion, stop/halt when you got  $i$  large enough such that  $A_i$  is sufficiently block upper triangular-looking.
- (iv) From this last matrix  $A_i$  (for  $i$  sufficiently large), read off its eigenvalues by looking at the submatrices composing the blocks on the diagonal. These are the approximations to the eigenvalues of  $A$ .

We illustrate further details of steps (i) and (ii) in the section below. You can do Householder reflections however you want, Professor Wujun gave you a good and quick method to do these Householder reflections by quickly obtaining who is the vector  $u$  in a Householder reflection of type

$$P = I - 2uu^T$$

both for the upper Hessenberg form and the QR factorization.

## 2 QR method for eigenvalues and Householder reflections

### 2.1 Upper Hessenberg form of a matrix

As mentioned, first reduce  $A$  to its upper Hessenberg form, where we have:

**Theorem 2.** *Let  $A$  be an  $n \times n$  real matrix. Then there exists an orthogonal matrix  $Q$  such that  $Q^T A Q$  is quasi-triangular (i.e.,  $A$  is block triangular with each diagonal block of order  $\leq 2$ . Moreover,  $Q$  may be chosen so that any  $2 \times 2$  diagonal block of  $Q^T A Q$  has only complex eigenvalues (which must be complex conjugate since a polynomial with real coefficients can only have complex roots in conjugate pairs). Another terminology used is to call  $A$  upper Hessenberg if  $a_{ij} = 0$  when  $i > j + 1$ . So if  $A$  is real, we are saying there exists an orthogonal matrix  $Q$  making  $Q^T A Q$  upper Hessenberg.*

We do have though, the following result for the special case when  $A$  is real and symmetric, since  $Q^T A Q$  would also have to be symmetric, so we must have that it being upper triangular would imply the following:

**Corollary 1.** *If  $A$  is real and symmetric, there exists an orthogonal matrix  $Q$  such that  $Q^T A Q$  is tridiagonal.*

## 2.2 Step (i) of algorithm: Householder reflections for upper Hessenberg form

We start off with an  $n \times n$  matrix  $A$ . The algorithm used applies successive Householder reflections which can be viewed as successive matrix operators  $P_k$  applied on both sides of the matrix  $A$  which reduce/transform the matrix  $A$ , one column at a time (according to certain information from each row and column) until we reach the upper Hessenberg matrix  $H$  similar to  $A$ . To get a block upper triangular matrix we need  $n - 2$  column modifications. This is summarized neatly with matrix transformation notation as

$$H = P_{n-2}P_{n-1} \cdots P_1AP_1P_2 \cdots P_{n-2} \quad (1)$$

The notes go into more detail about how to define and motivate the Householder reflections  $P_r$  in (2), but in short we define matrices  $P_r$  through column vectors  $\mathbf{u}_r$  that are zero in the first  $r$  components (to maintain certain properties untouched) and rest of components carefully chosen in order to define the Householder reflections  $P_r$  by

$$P_r := I - \mathbf{u}_r \mathbf{u}_r^T / (2K_r^2) \quad (2)$$

where  $2K_r^2 = S_r^2 + a_{r+1,r}^{(r-1)}S_r$  and  $S_r$  is defined below in (3). Starting with  $A_0 := A$ , we define  $A_r = P_r^T A_{r-1} P_r$  for  $r = 1, 2, \dots, n - 2$ . We define the  $\mathbf{u}_r$  componentwise, by first letting  $a_{ij}^{(r-1)}$  denote the  $ij$ th element of  $A_{r-1}$  and setting

$$S_r = \text{sgn} \left( a_{r+1,r}^{(r-1)} \right) \left( \sum_{i=r+1}^n [a_{ir}^{(r-1)}]^2 \right)^{1/2}, \quad (3)$$

we define  $\mathbf{u}_r = ((\mathbf{u}_r)_1, (\mathbf{u}_r)_2, \dots, (\mathbf{u}_r)_n)^T$  componentwise by

$$\begin{aligned} (\mathbf{u}_r)_i &= 0, \quad i = 1, \dots, r, \\ (\mathbf{u}_r)_{r+1} &= a_{r+1,r}^{(r-1)} + S_r, \\ (\mathbf{u}_r)_i &= a_{ir}^{(r-1)}, \quad i = r + 2, \dots, n. \end{aligned}$$

With the previous, we define  $A_r = P_r^T A_{r-1} P_r$  for  $r = 1, 2, \dots, n - 2$  starting from  $A_0 = A$ , like said previously, which all in all is summarized as obtaining the Hessenberg matrix  $H$  similar to  $A$  as in (1).

Notice since similar matrices have the same eigenvalues, the  $H$  Hessenberg matrix similar to  $A$  has the same eigenvalues as  $A$ . If through Householder reflections we reach a Hessenberg matrix, for example, with a lone diagonal element, then that diagonal element would be an eigenvalue. this is because of the property that if  $H$  is an  $n \times n$  matrix made of block matrices, for instance as in

$$H = \begin{pmatrix} D_1 & C \\ 0 & D_2 \end{pmatrix}$$

with  $D_1$  an  $m \times m$  block matrix and  $D_2$  an  $l \times l$  block matrix, then

$$\det(H - \lambda I_n) = \det \begin{pmatrix} D_1 - \lambda I_m & C \\ 0 & D_2 - \lambda I_l \end{pmatrix} = \det(D_1 - \lambda I_m) \cdot \det(D_2 - \lambda I_l).$$

For instance, if we had obtained through Householder reflections the Hessenberg form

$$P_{n-2}P_{n-1} \cdots P_1AP_1P_2 \cdots P_{n-2} = H = \begin{pmatrix} 2 & -1/\sqrt{2} & 5/\sqrt{2} \\ -\sqrt{2} & 2 & 2 \\ 0 & 0 & -2 \end{pmatrix}$$

then we know one of the eigenvalues of  $A$  is  $-2$ .

### 3 Singular Value Decomposition Project

The **singular value decomposition** of an  $m \times n$  matrix  $M$  is a factorization of the form

$$M = U\Sigma V^T$$

where  $U$  is an  $m \times m$  orthogonal matrix,  $\Sigma$  is an  $m \times n$  rectangular diagonal matrix with non-negative real numbers on the diagonal, and  $V$  is an  $n \times n$  orthogonal matrix.

#### 3.1 How many types of SVDs?

There can be more than one type of SVD **for a rectangular**  $m \times n$  matrix  $M$  (i.e. when  $m \neq n$ ) as can be seen in the literature since we have at least:

1. **SVD on a padded matrix:** If  $M \in \mathbb{F}^{m \times n}$  is augmented through additional zero (columns or rows) to a matrix  $B \in \mathbb{F}^{N \times N}$  with  $N := \max\{m, n\}$ , and you do "square" SVD. You end up factoring  $B = U\Sigma V^T$  but you get according to the case present:

- If  $n < m$

$$B = [M \ 0] = U \underbrace{\begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}}_{V^T}^T$$

The matrix  $V_1$  is of size  $n \times n$ ,  $V_2$  of size  $(m - n) \times (m - n)$ .  $\Sigma_1$  a matrix of size  $n \times n$ .  $U$  is of size  $m \times m$ .

- If  $n > m$

$$B = \begin{bmatrix} M \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix}}_U^T \underbrace{\begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix}}_{\Sigma} V^T$$

$\Sigma_1$  a matrix of size  $m \times m$ ,  $U_1$  an  $m \times m$  matrix and  $U_2$  a size  $(n - m) \times (n - m)$ , while  $V$  of size  $n \times n$ .

In general for the padded case we have  $\Sigma_1$  a matrix of size  $\min\{m, n\} \times \min\{m, n\}$ .  $U, V, \Sigma$  are of size  $\max\{m, n\} \times \max\{m, n\}$ . In this case it is true that  $\Sigma\Sigma^T = \Sigma^T\Sigma$ .

2. **Complete SVD no-padding:** You can go straightforward to do SVD on

$$M = U \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V_1^T, \quad \text{when } n < m$$

or

$$M = U_1 \begin{bmatrix} \Sigma_1 & 0 \end{bmatrix} V^T, \quad \text{when } m < n$$

In this case it is NOT true that  $\Sigma\Sigma^T = \Sigma^T\Sigma$ .

3. **Economy SVD:** Alternatively, the matrix  $M$  can be represented as the sum of rank-one matrices,

$$M = U\Sigma V^T = \sum_{i=1}^{\min(n, m)} \sigma_i u_i v_i^T$$

where  $u_i$  and  $v_i$  are the  $i$ -th columns of the matrix  $U$  and  $V$  and  $\sigma_i$  the  $i$ -th diagonal element of  $\Sigma$ . Here if  $p := \min\{m, n\}$  then  $U \in \mathbb{F}^{m \times p}$  has orthonormal columns containing the left-singular vectors,  $\Sigma \in \mathbb{F}^{p \times p}$  is a diagonal matrix containing singular values in non-increasing order and  $V \in \mathbb{F}^{n \times p}$  has orthonormal columns made of the right-singular vectors. In this case it is true that  $\Sigma\Sigma^T = \Sigma^T\Sigma$ .

4. **Compact SVD:** If  $k := \text{rank}(M) < \min\{m, n\}$ , then the compact SVD form allows matrix  $M$  to be represented as the sum of even less rank-one matrices:

$$M = U\Sigma V^T = \sum_{i=1}^k \sigma_i u_i v_i^T \quad (4)$$

where  $u_i$  and  $v_i$  are the  $i$ -th columns of the matrix  $U$  and  $V$  and  $\sigma_i$  the  $i$ -th diagonal element of  $\Sigma$ . Here  $U \in \mathbb{F}^{m \times k}$  has orthonormal columns containing the left-singular vectors,  $\Sigma \in \mathbb{F}^{k \times k}$  is a diagonal matrix containing singular values in non-increasing order and  $V \in \mathbb{F}^{n \times k}$  has orthonormal columns made of the right-singular vectors. In this case it is true that  $\Sigma\Sigma^T = \Sigma^T\Sigma$ .

### 3.2 Truncated SVD

We can approximate  $M$  by even less rank-one operators than (4). The idea is to throw away the smaller singular values and chop off to get

$$M \approx \sum_{i=1}^{n_{\text{small}}} \sigma_i u_i v_i^T$$

### 3.3 The project itself

Alternatively, the matrix  $M$  can be represented as the sum of rank-one matrices,

$$M = \sum_{i=1}^{\min(n, m)} \sigma_i u_i v_i^T$$

where  $u_i$  and  $v_i$  are the  $i$ -th columns of the matrix  $U$  and  $V$  and  $\sigma_i$  the  $i$ -th diagonal element of  $\Sigma$ . In this project, we assume all the diagonal entries  $\sigma_i$  are non-negative.

The singular value decomposition (SVD) generalizes the eigen-decomposition of a square normal matrix with an orthonormal eigen-basis to any  $m \times n$  matrix.

The goal of this project is to develop a numerical algorithm to perform the singular value decomposition, based on QR decomposition taught in lecture, and the QR algorithm taught in class. We will also apply the singular value decomposition to low-rank matrix approximation.

(a)

Show that the columns of  $V$  (right-singular vectors) are eigenvectors of  $M^T M$  and Show that the columns of  $U$  (left-singular vectors) are eigenvectors of  $MM^T$ , i.e.

$$M^T M = V\Sigma^T \Sigma V^T \quad \text{and} \quad MM^T = U\Sigma \Sigma^T U^T.$$

(b)

Write a python function to perform the singular value decomposition, based on the QR algorithm applied to  $MM^T$ . You may use Hessenberg from (la.hessenberg) and QR decomposition in the scipy.linalg module. Type `help(la.hessenberg)` and `help(la.qr)` for more details. The function should returns two matrices  $U$  and  $\Sigma V^T$ . (No credit if you use the `la.svd` directly!)

(c)

Derive a formula to find the diagonal matrix  $\Sigma$  from the product  $\Sigma V^T$ . Show all steps!

(d)

We say  $\tilde{M}$  is the best rank  $r$  ( $r < \min(m, n)$ ) approximation of matrix  $M$  if

$$\tilde{M} = M = U\tilde{\Sigma}V^T$$

where  $\tilde{\Sigma}$  is the same matrix as  $\Sigma$  except that it contains only the  $r$  largest singular values (the other singular values are replaced by zero). Load the file `image.npy` as the matrix  $M$ , which is the pixel matrix of a hand written zero. Find its best rank 10 approximation  $\tilde{M}$  using the function in part b) and plot its corresponding image. Calculate the Frobenius norm of

$$M - \tilde{M}$$

by `LA.norm()`. The Frobenius norm of matrix  $A$  is defined as

$$\|A\|_F = \sqrt{\sum_{i,j=1}^{i=n, j=m} a_{ij}^2}.$$

### 3.4 Getting $\Sigma V^T$ and $U$

Note that you are getting  $\Sigma\Sigma^T$  as the diagonal matrix and you get  $U$  from the orthogonal transformations such that  $MM^T = U\Sigma\Sigma^T U^T$  but you also assume that  $M = U\Sigma V^T$  so that  $U^T M = \Sigma V^T$ . Since you know  $U$  and  $M$ , this last relation gives you  $\Sigma V^T$ .

### 3.5 What about $\Sigma$ ?

I would define  $\Sigma$  through the square roots of  $\Sigma\Sigma^T$ 's eigenvalues and be done with it. For some reason, everytime I get  $\Sigma\Sigma^T$ , I get the eigenvalues in the correct order.

### 3.6 What about $V$ ?

The problem is that applying the QR algorithm to  $M^T M$  seems to give  $V$  but I don't think its eigenvectors align with those for  $MM^T$  in order

ANOTHER OPTION: Now how about now that you know who  $\Sigma$  is, dividing the columns of  $\Sigma V^T$  by the adequate singular values  $\sigma_i$  to get a potential  $V^T$ ? The tricky part here becomes that to zero singular values you must associate an eigenvector. I would associate a dimension to some kernel somehow and find an orthonormal basis for it.

But actually you don't have to worry about that because in a compact SVD you don't need those zero right singular vectors (i.e., vectors corresponding to the zero singular value), since you would throw them out. So for a compact SVD you can eliminate those vectors from the  $V$  matrix as I show below.

## 4 What if we converged to something good enough?

For the project we have the matrix  $A = MM^T$ , or  $A = M^T M$ , whichever one you want to start with (most likely started with  $MM^T$ ). Note that  $MM^T$  is a symmetric matrix, with real coefficients. Also, it (and likewise  $M^T M$ ) is positive definite since

$$x^T(MM^T)x = (M^T x)^T(M^T x) = \|M^T x\|_2^2 \geq 0$$

Therefore, supposing the eigenvalues

$$\lambda_1 > \lambda_2 > \dots > \lambda_n$$

we converge along the diagonal to the eigenvalues, from the most basic convergence result. **But actually the symmetry of the matrix allows this to converge in particular to a diagonal matrix**  $A_i \rightarrow \text{diag}[\lambda_1, \dots, \lambda_n]$ . Note from the QR algorithm you have:

$$A_i = Q_i R_i \rightarrow A_{i+1} = R_i Q_i$$

so that supposing at step  $k$  you are within tolerance level to a diagonal matrix, so  $A_k \approx D$  a diagonal matrix. Then note from the way you successively operated starting from the original Hessenberg matrix  $A_0$ , that

$$\begin{aligned} D \approx A_k &= R_{k-1} Q_{k-1} = (Q_{k-1})^T \underbrace{Q_{k-1} R_{k-1}}_{=A_{k-1}} Q_{k-1} \\ &= (Q_{k-1})^T R_{k-2} Q_{k-2} Q_{k-1} \\ &= (Q_{k-1})^T (Q_{k-2})^T Q_{k-2} R_{k-2} Q_{k-2} Q_{k-1} \\ &= (Q_{k-1})^T (Q_{k-2})^T A_{k-2} Q_{k-2} Q_{k-1} \\ &\vdots \\ &= (Q_{k-1})^T (Q_{k-2})^T \cdots (Q_0)^T A_0 Q_0 \cdots Q_{k-2} Q_{k-1} \end{aligned}$$

so that if  $H_k := Q_0 \cdots Q_{k-2} Q_{k-1}$  (an orthogonal matrix, because it is a product of orthogonal matrices) we have that

$$D \approx A_k = H_k^T A_0 H_k \quad (5)$$

So that  $H_k$  approximately diagonalizes  $A_0$  which is the Hessenberg form of the original matrix  $A$ . Thus we could interpret  $H_k$  as being an approximation to the necessary change of basis matrix that diagonalizes  $A$ , with the columns of  $H_k$  containing approximations to the eigenvectors. Note that the matrix  $A = M^T M$  is positive semidefinite since  $x^T A x = x^T M^T M x = (Mx)^T Mx = \|Mx\|^2 \geq 0$  so it has non-negative eigenvalues, and taking the nonnegative roots of the eigenvalues

$$+\sqrt{\lambda_i} =: \sigma_i$$

then<sup>1</sup>

$$D = \Sigma \Sigma^T$$

Note that  $A_0$  in (5) was the Hessenberg form of the original matrix  $A = MM^T$  so it was obtained by applying successive orthogonal transformations such that  $A_0 = P^T A P$  is in upper Hessenberg form, therefore combining with equation (5) we have

$$D = \Sigma \Sigma^T \approx (PH_k)^T MM^T (PH_k) = U^T MM^T U \quad (6)$$

with  $U := PH_k$  an orthogonal matrix. The previous implies

$$U \Sigma \Sigma^T U^T = MM^T$$

## 5 Potential Difficulties project

In short the biggest potential difficulty is that you get something like

$$\Sigma \Sigma^T = \begin{bmatrix} \mathbf{0} & & \\ & \Sigma_1^2 & \\ & & \mathbf{0} \end{bmatrix}$$

---

<sup>1</sup>If we had a square matrix to begin with, then  $\Sigma$  would not be "rectangular" diagonal, but rather, perfectly diagonal and it would be true that  $D = \Sigma \Sigma^T = \Sigma^T \Sigma$ . But since we are in the case  $m \neq n$  in general, we cannot affirm  $\Sigma = \Sigma^T$  and  $\Sigma \Sigma^T = \Sigma^T \Sigma$ .

where

$$\Sigma_1^2 = \text{diag}[\sigma_1^2, \sigma_2^2, \dots, \sigma_r^2] \in \mathbb{R}^{r \times r} \quad (7)$$

with  $\sigma_i^2 > 0$  and  $r = \text{rank}(M)$ .

Imagine that the following happens: You get an SVD of form

$$U \Sigma V^T = \underbrace{\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{bmatrix}}_{\in \mathbb{F}^{3 \times 3}} \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \end{bmatrix}}_{\in \mathbb{F}^{3 \times 4}} \underbrace{\begin{bmatrix} v_{11} & v_{12} & v_{13} & v_{14} \\ v_{21} & v_{22} & v_{23} & v_{24} \\ v_{31} & v_{32} & v_{33} & v_{34} \\ v_{41} & v_{42} & v_{43} & v_{44} \end{bmatrix}}_{\in \mathbb{F}^{4 \times 4}}$$

which if we only have  $U$  and  $\Sigma V^T$  as in the project, becomes

$$M = U(\Sigma V^T) = \underbrace{\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{bmatrix}}_{U \in \mathbb{F}^{3 \times 3}} \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ \sigma_2 v_{12} & \sigma_2 v_{22} & \sigma_2 v_{32} & \sigma_2 v_{42} \\ \sigma_3 v_{13} & \sigma_3 v_{23} & \sigma_3 v_{33} & \sigma_3 v_{43} \end{bmatrix}}_{=\Sigma V^T \in \mathbb{F}^{3 \times 4}} \quad (8)$$

Let's see what is really happening here, to understand better the rank-1 economy SVD forms:

$$M = U(\Sigma V^T) = \quad (9)$$

$$\begin{aligned} & \begin{bmatrix} \sigma_2 u_{12} v_{12} + \sigma_3 u_{13} v_{13} & \sigma_2 u_{12} v_{22} + \sigma_3 u_{13} v_{23} & \sigma_2 u_{12} v_{32} + \sigma_3 u_{13} v_{33} & \sigma_2 u_{12} v_{42} + \sigma_3 u_{13} v_{43} \\ \sigma_2 u_{22} v_{12} + \sigma_3 u_{23} v_{13} & \sigma_2 u_{22} v_{22} + \sigma_3 u_{23} v_{23} & \sigma_2 u_{22} v_{32} + \sigma_3 u_{23} v_{33} & \sigma_2 u_{22} v_{42} + \sigma_3 u_{23} v_{43} \\ \sigma_2 u_{32} v_{12} + \sigma_3 u_{33} v_{13} & \sigma_2 u_{32} v_{22} + \sigma_3 u_{33} v_{23} & \sigma_2 u_{32} v_{32} + \sigma_3 u_{33} v_{33} & \sigma_2 u_{32} v_{42} + \sigma_3 u_{33} v_{43} \end{bmatrix} \\ &= \sigma_2 \begin{bmatrix} u_{12} v_{12} & u_{12} v_{22} & u_{12} v_{32} & u_{12} v_{42} \\ u_{22} v_{12} & u_{22} v_{22} & u_{22} v_{32} & u_{22} v_{42} \\ u_{32} v_{12} & u_{32} v_{22} & u_{32} v_{32} & u_{32} v_{42} \end{bmatrix} + \sigma_3 \begin{bmatrix} u_{13} v_{13} & u_{13} v_{23} & u_{13} v_{33} & u_{13} v_{43} \\ u_{23} v_{13} & u_{23} v_{23} & u_{23} v_{33} & u_{23} v_{43} \\ u_{33} v_{13} & u_{33} v_{23} & u_{33} v_{33} & u_{33} v_{43} \end{bmatrix} \\ &= \sigma_2 \begin{bmatrix} u_{12} \\ u_{22} \\ u_{32} \end{bmatrix} \begin{bmatrix} v_{12} & v_{22} & v_{32} & v_{42} \end{bmatrix} + \sigma_3 \begin{bmatrix} u_{13} \\ u_{23} \\ u_{33} \end{bmatrix} \begin{bmatrix} v_{13} & v_{23} & v_{33} & v_{43} \end{bmatrix} \\ &= \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \sigma_3 \mathbf{u}_3 \mathbf{v}_3^T \quad (\text{Compact SVD}) \end{aligned} \quad (10)$$

Note how the compact SVD reflects the rank of the original matrix, which was 2. If someone asked you to obtain the best rank 1 approximation to  $M$ , and  $\sigma_2 > \sigma_3$ , then the best rank one approximation is from retaining only the largest summand in (10), which since  $\mathbf{u}_i$  and  $\mathbf{v}_i$  are orthonormal, or in particular, have norm one, then the largest summand is

$$\begin{aligned} & \sigma_2 \begin{bmatrix} u_{12} v_{12} & u_{12} v_{22} & u_{12} v_{32} & u_{12} v_{42} \\ u_{22} v_{12} & u_{22} v_{22} & u_{22} v_{32} & u_{22} v_{42} \\ u_{32} v_{12} & u_{32} v_{22} & u_{32} v_{32} & u_{32} v_{42} \end{bmatrix} \\ &= \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T \approx M \quad (\text{Truncated SVD}) \end{aligned} \quad (11)$$

Thus we are approximating  $M$  in (9) by (11) But note we could have done the factorization in (8) a bit different: We could have realized we drop the zero row of  $\Sigma V^T$  (the first one) and the corresponding column vector of  $U$  (also the first one) to factor  $M$  as

$$M = \tilde{U} \tilde{\Sigma} V^T = \underbrace{\begin{bmatrix} u_{12} & u_{13} \\ u_{22} & u_{23} \\ u_{32} & u_{33} \end{bmatrix}}_{\in \mathbb{F}^{3 \times 2}} \underbrace{\begin{bmatrix} \sigma_2 v_{12} & \sigma_2 v_{22} & \sigma_2 v_{32} & \sigma_2 v_{42} \\ \sigma_3 v_{13} & \sigma_3 v_{23} & \sigma_3 v_{33} & \sigma_3 v_{43} \end{bmatrix}}_{\in \mathbb{F}^{2 \times 4}} \quad (12)$$

where  $\tilde{\Sigma} V^T$  replaces  $\Sigma V^T$  by eliminating the necessary row.