

Review.

- If the $m \times n$ matrix A has SVD $A = U\Sigma V^T$, then its pseudoinverse is $A^+ = V\Sigma^+U^T$.
Here, Σ^+ , the pseudoinverse of Σ , is the $n \times m$ diagonal matrix, whose nonzero entries are the inverses of the entries of Σ .
- The system $Ax = b$ has “optimal” solution $x = A^+b$.
Here, “optimal” means that x is the smallest least squares solution.

Example 170.

- Find the pseudoinverse of $A = [1 \ 2 \ 3]$.
- Find the smallest solution to $x_1 + 2x_2 + 3x_3 = 6$.

As before, smallest solutions means the solution x such that $\|x\|$ is as small as possible. One obvious solution is $[1, 1, 1]^T$, but is it the smallest?

Solution.

(a) $A^T A = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} [1 \ 2 \ 3] = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix}$ has 14-eigenvector $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ and 0-eigenvectors $\begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -3 \\ 0 \\ 1 \end{bmatrix}$.

$$u_1 = \frac{1}{\sigma_1} A v_1 = \frac{1}{\sqrt{14}} [1 \ 2 \ 3] \frac{1}{\sqrt{14}} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = 1$$

Hence, $A = U\Sigma V^T$ with $U = [1]$, $\Sigma = [\sqrt{14} \ 0 \ 0]$, $V = \begin{bmatrix} 1/\sqrt{14} & * & * \\ 2/\sqrt{14} & * & * \\ 3/\sqrt{14} & * & * \end{bmatrix}$.

$$A^+ = V\Sigma^+U^T = \begin{bmatrix} 1/\sqrt{14} & * & * \\ 2/\sqrt{14} & * & * \\ 3/\sqrt{14} & * & * \end{bmatrix} \begin{bmatrix} 1/\sqrt{14} \\ 0 \\ 0 \end{bmatrix} [1] = \frac{1}{14} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Comment. No surprise on U . The only options for U are $U = [1]$ and $U = [-1]$.

Comment. Realizing what we did here allows us to write down A^+ immediately for all $1 \times n$ matrices A . See Example 171.

Homework. Complete the SVD of A . That is, find an option for the two missing columns of V , so that V is an orthogonal matrix. In other words, find an orthonormal basis for the 0-eigenspace.

Comment. An even better approach would be to compute AA^T first (instead of $A^T A$) which would allow us to compute U first (rather than V first). Can you fill in the blanks?

- We are solving $Ax = [6]$ with $A = [1 \ 2 \ 3]$ as in the previous example.

We conclude that the smallest solution is $x = A^+[6] = \frac{3}{7} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$.

Compare. $\left\| \frac{3}{7} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \right\| = \frac{3}{7} \sqrt{14} \approx 1.604$ is indeed smaller than, say, $\left\| \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\| = \sqrt{3} \approx 1.732$.

Geometric picture. The equation $x_1 + 2x_2 + 3x_3 = 6$ describes a plane (not through the origin), and we are asking for the point on that plane which is closest to the origin. That's a typical question in Calculus III. Note that $[1 \ 2 \ 3]^T$ is the normal vector of the plane. Explain why the answer had to be a multiple of that normal vector!

Example 171. More generally, find the pseudoinverse of $A = [a_1 \ a_2 \ a_3]$.

Solution. As in the previous example, we see that the answer will be $A^+ = \frac{a}{\|a\|^2}$ with $a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$.

Comment. Likewise for $A = [a_1 \ a_2 \ \dots \ a_n]$.

Example 172. How is the rank of A reflected in its singular value decomposition $A = U\Sigma V^T$?

Solution. The rank of A is equal to the number of nonzero singular values.

Theorem 173. (matrix approximation lemma) Suppose A is a $m \times n$ matrix, and we want to approximate A using a matrix B of rank s (smaller than the rank of A). Let $A = U\Sigma V^T$ be the SVD of A (with singular values in decreasing order). Then, the best such approximation is $B = U_s \Sigma_s V_s^T$, where Σ_s is the $s \times s$ diagonal matrix with entries $\sigma_1, \sigma_2, \dots, \sigma_s$ and U_s, V_s are obtained from U, V by only taking the first s columns.

Comment. Note that, by choosing s small compared to r , we can store an approximation of A using much less data. This approximation will be good if the omitted singular values $\sigma_{s+1}, \sigma_{s+2}, \dots, \sigma_r$ are all “small”.

Comment. Equivalently, $B = U\Sigma_s V^T$, where Σ_s is now obtained from Σ by setting all but the largest s singular values to 0. In other words, Σ_s has the values $\sigma_1, \sigma_2, \dots, \sigma_s$ on its diagonal, followed by zeros.

In other words. Here is another common way to say the same thing:

- Observe that $A = U\Sigma V^T$ is equivalent to $A = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^T$.
- Each matrix $\mathbf{u}_i \mathbf{v}_i^T$ has rank 1.
- The best rank s approximation to A is $B = \sum_{i=1}^s \sigma_i \mathbf{u}_i \mathbf{v}_i^T$.

Advanced comment. Here, “best” approximation is measured using the Frobenius norm of a matrix A (which is the same as the norm of a vector with all the entries of A).

Example 174. Determine the best rank 1 approximation of $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & -1 \end{bmatrix}$.

Solution. We determine (do it!) that A has the SVD

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 & 0 \\ 0 & \sqrt{2} & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{3} & -1/\sqrt{2} & 1/\sqrt{6} \\ 1/\sqrt{3} & 0 & -2/\sqrt{6} \\ 1/\sqrt{3} & 1/\sqrt{2} & 1/\sqrt{6} \end{bmatrix}^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & -1 \end{bmatrix}.$$

Hence, the best rank 1 approximation of A is (that is, we keep 1 singular value only) is

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} \sqrt{3} \end{bmatrix} \begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{bmatrix}^T = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Comment. Equivalently, $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{3} & -1/\sqrt{2} & 1/\sqrt{6} \\ 1/\sqrt{3} & 0 & -2/\sqrt{6} \\ 1/\sqrt{3} & 1/\sqrt{2} & 1/\sqrt{6} \end{bmatrix}^T = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$.

Example 175. Determine the best rank 1 approximation of $A = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$.

Solution. Recall that $A = U\Sigma V^T$ with $U = \begin{bmatrix} -2/\sqrt{6} & 0 & -1/\sqrt{3} \\ 1/\sqrt{6} & 1/\sqrt{2} & -1/\sqrt{3} \\ -1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} \end{bmatrix}$, $\Sigma = \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$, $V = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}$.

Hence, the best rank 1 approximation of A is $\frac{1}{\sqrt{6}} \begin{bmatrix} -2 \\ 1 \\ -1 \end{bmatrix} \begin{bmatrix} \sqrt{3} \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix}^T = \frac{1}{2} \begin{bmatrix} 2 & -2 \\ -1 & 1 \\ 1 & -1 \end{bmatrix}$.

Example 176. (image compression) Let us load a 341x512 grayscale photo and store it as a matrix A . Each entry of the matrix is a value between 0 (black) and 1 (white).

The beautiful picture is taken from: <http://www.southalabama.edu/departments/publicrelations/brand/photography.html>

[The same approach works with color pictures. These are often represented by three matrices: one for the red component of the pixel, one for the green and for the blue component (RGB color scheme).]

```
Sage] import pylab
```

```
Sage] A = matrix(pylab.imread('/home/armin/photo.png'))
```

```
Sage] A.dimensions()
```

```
(341, 512)
```

```
Sage] A[0,0]
```

```
0.137254908681
```

```
Sage] matrix_plot(A, cmap='gray')
```



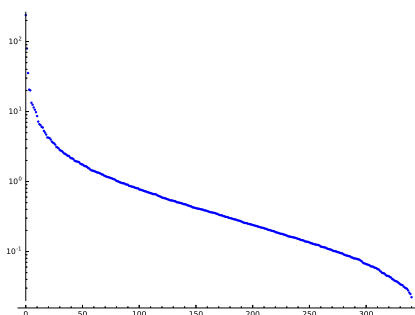
Next, we compute the SVD of A . Despite the size of A that takes the computer only a fraction of a second:

```
Sage] U,S,V = A.SVD()
```

```
Sage] S.diagonal()[:6]
```

```
[238.443435709, 79.4429775448, 35.4540786319, 20.5662302846, 20.0697710337, 13.3421216529]
```

```
Sage] list_plot(S.diagonal(), scale='semilogy')
```



As we can see, the magnitude of the singular values drops off quickly. We get a good approximation to A (our original photo) by computing a best rank s approximation to A by computing $U_s \Sigma_s V_s^T$ where Σ_s is the $s \times s$ diagonal matrix with entries $\sigma_1, \sigma_2, \dots, \sigma_s$ and U_s, V_s are obtained from the corresponding matrices in the SVD $A = U \Sigma V^T$ by only taking the first s columns.

```
Sage] def A_approx(s):
    U0 = U.matrix_from_columns([0..s-1])
    S0 = diagonal_matrix(S.diagonal()[0:s])
    V0 = V.matrix_from_columns([0..s-1])
    return U0*S0*V0.transpose()
```

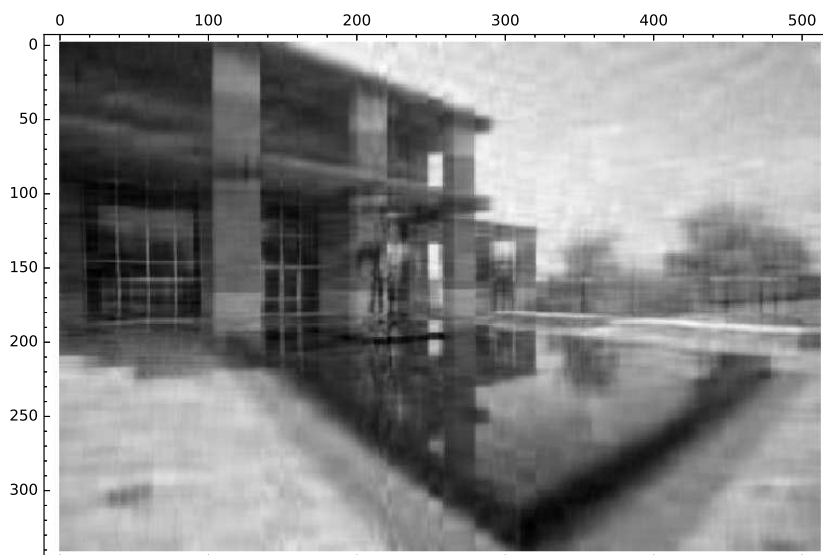
Taking only 100 of the 341 singular values, we get an approximation, which is almost as good as the original:

```
Sage] matrix_plot(A_approx(100), cmap='gray')
```



But notice the development of artifacts. Taking only 20 of the 341 singular values, a lot is lost:

```
Sage] matrix_plot(A_approx(20), cmap='gray')
```



Comment. Image compression is just one (nice visual) example of the power of SVD. A variation of this approach can, for instance, also be used for image denoising. Much more generally, the SVD is able to extract the most important features of any sort of data!