Last time

- Started with arbitrary real m-by-n matrix $\mathbf{A}$ of rank r
- Generated $\mathbf{S} = \mathbf{A}^{\mathrm{T}}\mathbf{A}$, a real, symmetric, positive semidefinite matrix
  - $\mathbf{S}$ is n-by-n
  - $\mathbf{S}$ is likewise of rank r, since (shown earlier) $\mathrm{null}\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right) = \mathrm{null}\left(\mathbf{A}\right)$
  - by Spectral Theorem, $\mathbf{S}$ has decomposition $\mathbf{S} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}$, where
    - entries of λ (eigenvalues) are all nonnegative, so may be labeled as σ_i^2, with $\sigma_1^2 \geqslant \sigma_2^2 \geqslant \cdots \geqslant \sigma_r^2 > \sigma_{r+1}^2 = \cdots = \sigma_n^2 = 0$
    - the eigenvectors $\mathbf{v}_i$ of $\mathbf{S}$ corresponding to eigenvalues σ_i^2 and comprising the columns of $\mathbf{V}$ form an orthonormal basis of $\mathbb{R}^n$
    - the eigenvectors $\mathbf{v}_{r+1}, \ldots, \mathbf{v}_n$ correspond to zero eigenvalues, so form a basis of $\mathrm{null}\left(\mathbf{S}\right) = \mathrm{null}\left(\mathbf{A}\right)$
    - the eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$, being orthogonal to $\mathrm{null}\left(\mathbf{A}\right)$, form a basis for $\mathrm{null}\left(\mathbf{A}\right)^{\perp} = \mathrm{col}\left(\mathbf{A}^{\mathrm{T}}\right)$, the row space of $\mathbf{A}$
- For $i = 1, \ldots, r$ we defined vectors $\mathbf{u}_i = \dfrac{1}{\sigma_i}\mathbf{A}\mathbf{v}_i$. These $\mathbf{u}_i$
  - are in $\mathrm{col}\left(\mathbf{A}\right)$, by definition.
  - are mutually orthogonal unit vectors—that is,

$$\begin{aligned}\left\langle \mathbf{u}_i, \mathbf{u}_j \right\rangle &= \left\langle \frac{1}{\sigma_i}\mathbf{A}\mathbf{v}_i, \frac{1}{\sigma_j}\mathbf{A}\mathbf{v}_j \right\rangle = \frac{1}{\sigma_i\sigma_j}\left\langle \mathbf{A}\mathbf{v}_i, \mathbf{A}\mathbf{v}_j \right\rangle = \frac{1}{\sigma_i\sigma_j}\left\langle \mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{v}_i, \mathbf{v}_j \right\rangle \\ &= \frac{1}{\sigma_i\sigma_j}\left\langle \sigma_i^2\mathbf{v}_i, \mathbf{v}_j \right\rangle = \frac{1}{\sigma_i\sigma_j}\sigma_i^2\left\langle \mathbf{v}_i, \mathbf{v}_j \right\rangle = \begin{cases} 0, & i \neq j, \\ 1, & i = j. \end{cases}\end{aligned}$$

  - are, thus, a basis of $\mathrm{col}\left(\mathbf{A}\right)$.
  - satisfy $\mathbf{A}\mathbf{A}^{\mathrm{T}}\mathbf{u}_i = \mathbf{A}\mathbf{A}^{\mathrm{T}}\left(\dfrac{1}{\sigma_i}\mathbf{A}\mathbf{v}_i\right) = \dfrac{1}{\sigma_i}\mathbf{A}(\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{v}_i) = \dfrac{1}{\sigma_i}\mathbf{A}(\sigma_i^2\mathbf{v}_i) = \sigma_i^2 \cdot \dfrac{1}{\sigma_i}\mathbf{A}\mathbf{v}_i = \sigma_i^2\mathbf{u}_i$, which says the $\mathbf{u}_i$ are eigenvectors of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ corresponding to eigenvalues σ_i^2.
- As $\mathrm{col}\left(\mathbf{A}\right)$ has $\mathrm{null}\left(\mathbf{A}^{\mathrm{T}}\right)$ as its orthogonal complement, with the latter being of dimension $m - r$, we may find an orthonormal basis $\mathbf{u}_{r+1}, \ldots, \mathbf{u}_m$ of the left nullspace, so that the matrix $\mathbf{U}$ whose columns are $\mathbf{u}_1, \ldots, \mathbf{u}_m$ form an orthonormal basis of $\mathbb{R}^m$. Moreover, we have

$$\mathbf{A}\mathbf{V} = \mathbf{U}\boldsymbol{\Sigma}, \qquad \text{or} \qquad \mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}} = \sigma_1\mathbf{u}_1\mathbf{v}_1^{\mathrm{T}} + \sigma_2\mathbf{u}_2\mathbf{v}_2^{\mathrm{T}} + \cdots + \sigma_r\mathbf{u}_r\mathbf{v}_r^{\mathrm{T}},$$

where

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & & & & & \\ & \sigma_2 & & & & \\ & & \ddots & & & \\ & & & \sigma_r & & \\ & & & & 0 & \\ & & & & & \ddots \end{bmatrix}_{m\times n}.$$

In Octave, consider the 4-by-3 matrix **A** having rank 2:

```
octave:11> A = [6 1 11; 4 7 1; 2 1 3; 3 8 -2]
A =

   6   1   11
   4   7    1
   2   1    3
   3   8   -2

octave:12> rank(A)
ans =  2
```

We generate its svd

```
octave:14> [U,Sigma,V] = svd(A)
U =

  -0.810510   0.492371   0.282812   0.143745
  -0.440022  -0.485858  -0.031679  -0.754532
  -0.259610   0.078428  -0.952159   0.140872
  -0.286457  -0.717887   0.111388   0.624639

Sigma =

Diagonal Matrix

   1.4034e+01            0            0
            0   1.0865e+01            0
            0            0   2.9622e-17
            0            0            0

V =

  -0.570173  -0.090750  -0.816497
  -0.459028  -0.789067   0.408248
  -0.681319   0.607567   0.408248
```

and note that various products we expect to hold, do so:

```
octave:23> A'*A      % original matrix S
ans =

   65    60    70
   60   115     5
   70     5   135

octave:24> V*diag([14.034^2 10.865^2  0])*V'     % spectral decomposition of S
ans =

   65.0012    60.0008    70.0016
   60.0008   114.9993     5.0024
   70.0016     5.0024   135.0009

octave:25> A*A'
ans =

   158    42    46     4
    42    66    18    66
    46    18    14     8
     4    66     8    77

octave:26> U*diag([14.034 10.865 0  0])^2*U'     % spectral decomp of AA'
ans =

   158.0019    42.0019    46.0007     4.0017
    42.0019    66.0002    18.0005    65.9996
    46.0007    18.0005    14.0002     8.0004
     4.0017    65.9996     8.0004    76.9991

octave:27> U*Sigma*V'
ans =

   6.00000    1.00000   11.00000
   4.00000    7.00000    1.00000
   2.00000    1.00000    3.00000
   3.00000    8.00000   -2.00000

octave:28> U'*U       % should be 4-by-4 identity matrix
ans =

   1.00000  -0.00000  -0.00000   0.00000
  -0.00000   1.00000   0.00000   0.00000
  -0.00000   0.00000   1.00000   0.00000
   0.00000   0.00000   0.00000   1.00000

octave:29> V'*V       % should be 3-by-3 identity matrix
ans =
```

```
 1.00000   0.00000  -0.00000
 0.00000   1.00000   0.00000
-0.00000   0.00000   1.00000
```

Image compression

For what follows, I will use the digital image found at this url:

http://www.calvin.edu/~scofield/courses/m355/materials/octave/images/owl.png

which you can download and place in your working directory, as I have.

Now, consider the digital image

```
octave:35> im = imread("owl.png");
octave:36> im = im (:,:,1) ;
octave:37> size(im)
ans =

   480   640
```

The image consists of $(480)(640) = 307,200$ pixels, stacked 480 high and 640 wide. We have stored it as a matrix, where the entries are numbers ranging from 0 (black) through 255 (white), with numbers in between these two extremes representing various shades of gray.

As with any real matrix, we can find the SVD. The shapes/sizes of the various matrices are predictable, knowing the size of the image.

```
octave:38> [U,Sigma,V] = svd(im);
octave:39> size(U)
ans =

   480   480

octave:40> size(Sigma)
ans =

   480   640

octave:41> size(V)
ans =

   640   640
```

Now, technically, all of the singular values are nonzero.

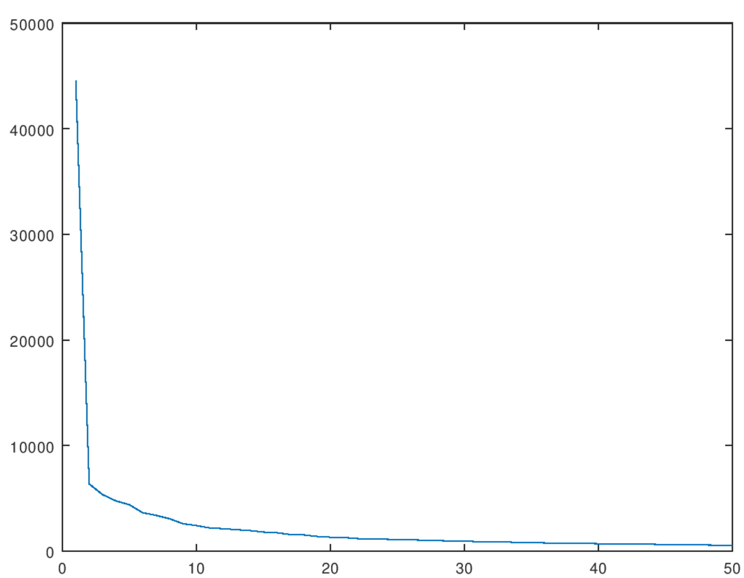
```
octave:42> length(diag(Sigma))
ans =   480
octave:43> singVals = diag(Sigma)
octave:44> singVals(1:4)        % the largest  four
ans =

    44559.28117
     6372.09432
     5357.72008
     4759.38851

octave:45> singVals(477:480)      % the smallest  four
ans =

    10.1009
     9.6046
     9.0802
     8.9670
```

To fully recover the original digital image from its SVD, we would include all 480 terms

$$\sigma_1\mathbf{u}_1\mathbf{v}_1^{\mathrm{T}} + \sigma_2\mathbf{u}_2\mathbf{v}_2^{\mathrm{T}} + \cdots + \sigma_{480}\mathbf{u}_{480}\mathbf{v}_{480}^{\mathrm{T}}$$

But here I plot the first 50 to show how the singular values fall off in size quickly:

```
octave:46> plot(singVals(1:50))
```



If we choose to keep only the first 20 singular values with the corresponding columns of **U** and **V**, this amounts to saving just $(20)(1+480+640) = 37,320$ numbers, roughly 7.3% of the numbers stored in the original matrix.

```
octave:47> k=20;
octave:48> recIm=zeros(size(im));
octave:49> for jj=1:k, recIm += S(jj,jj)*U(:,jj)*V(:,jj)'; end;
octave:50> imshow( uint8(recIm) )
```

The resulting image is displayed below right, with the original appearing on the left.