Class—SCLA SVD, Image Compression

Advanced Linear Algebra

Robert Beezer

Math 390, Spring 2021

We use PyLab (included in Sage, based on matplotlib) to read in an image as an array of grayscale intensities, and save it as an array. Then we compute the SVD and sum as many rank one terms as determined by the value set by the slider.

```
import pylab
# PNG from GIMP as "8bpc GRAY"
A_image = pylab.imread('mystery2.png', format='png')
@interact
def svd_image(i=(1,(1..50)), display_axes=True):
    u,s,v = pylab.linalg.svd(A_image)
    A     = sum(s[j]*pylab.outer(u[0:,j], v[j,0:]) for j in
        range(i))
    imgplot = pylab.imshow(A, cmap='gray')
    # can't scale larger
    show(imgplot, axes=display_axes)
    html('Compressed␣using␣%s␣singular␣values'%i)
```

If we use 8 bits (1 byte) for each pixel, we can have 256 different intensities. Each rank-one term requires storing/transmitting two vectors with 300 or 500 entries each. For 20 terms, then we require $20 \cdot (300 + 500 + 1) \approx 16K$ bytes for the image. This compares with perhaps 1 byte per pixel for a grayscale image, over all pixels, for $300 \cdot 500 \approx 150K$ bytes. (A compressed PNG color image file for this image is 188K).

It is worth having a peek at the singular values.

```
_, S, _ = pylab.linalg.svd(A_image)
S
```