AN EXPLORATION IN MATRIX COMPUTATIONS THROUGH THE SVD, BIPARTITE MATRICES, AND ITERATIVE METHODS.

David F. Gleich

August 21, 2023

The entire goal of our class was to help study matrix problems through their structure. Here we will consider matrices that have what we will call "bipartite" structure, following the conventions of a graph theory view on matrices. A more standard name for this structure is "consistently ordered" but that includes quite a few more details.

Where do bipartite matrices arise? The first place is in algorithms for the SVD. Another place is the 2d Laplacian, or any matrix derived from a bipartite graph. The point of this lecture is to look at the relationships between these persepctives.

0.1 ALGORITHMS FOR THE SVD

We have seen algorithms to compute eigenvalues of matrices. Algorithms for the SVD follow from two points of view. Assume without loss of generality that A has more rows than columns.

View 1. The singular values of the matrix A are the eigenvalues of $A^T A$

View 2. The singular values of the matrix A are the positive eigenvalues of $B = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$.

The matrix B in view 2 is a specific instance of a bipartite matrix!

More generally, the theorem underlying **View 2** is

THEOREM 1 *Let $B = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$ and let $A = U\Sigma V^T$ be the SVD of A. Then the eigenvalues of B are $\pm\sigma_i$ along with $m + n - 2n$ additional zeros. Given an eigenvalue $+\sigma_i$ eigenvector $Bz = \sigma z$ if we partition $z = \begin{bmatrix} x \\ y \end{bmatrix}$. Then, $Ay = \sigma x$ is one of the singular vectors and value sets.*

Proof We have $B = \begin{bmatrix} 0 & U\Sigma V^T \\ V\Sigma^T U^T & 0 \end{bmatrix} = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}\begin{bmatrix} 0 & \Sigma \\ \Sigma^T & 0 \end{bmatrix}\begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}$. — TODO –
Show more of this matrix, including the number of zeros. Then note that there exists a permutation matrix P such that

$$P\begin{bmatrix} 0 & \Sigma \\ \Sigma^T & 0 \end{bmatrix}P^T = \begin{bmatrix} 0 & \sigma_1 & & & \\ \sigma_1 & 0 & & & \\ & & 0 & \sigma_2 & \\ & & \sigma_2 & 0 & \\ & & & & \ddots \end{bmatrix}$$

— TODO – Work out more of this matrix, including the number of zeros.

Further, we have

$$\begin{bmatrix} 0 & \sigma_i \\ \sigma_i & 0 \end{bmatrix} = \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}\begin{bmatrix} +\sigma_i & \\ & -\sigma_i \end{bmatrix}\begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}.$$

At which point, we are done. ∎

This enables us to work with the matrix B instead of A. Of course, we do not actually form B, rather we work with it implicitly.

Just like eigenvalue algorithms, the first step is to reduce the matrix size via a set of orthogonal operations. For the SVD, we can make this two-sided! This enables us to reduce A to a bidiagonal matrix F.

— TODO – New nodes on doing the bidiagonal reduction via a full Householder step on the left, then a partial on the right, and then full on the left. …

— TODO – Note that we could do lower-bidiagonal too!

We can also see this via the Lanczos perspective. Consider running the Lanczos method on B.

```
EXAMPLE 2  A =
 -5.0  -5.0   5.0  -3.0
 -3.0  -2.0  -1.0   1.0
  4.0   3.0   0.0  -4.0
  0.0   2.0  -4.0   3.0
 -5.0   0.0  -3.0  -1.0
  5.0   2.0   2.0  -2.0


U,T,rho = lanczos(A, [ones(6); zeros(4)], 4)

U =
10 x 5 Array{Float64,2}:
 0.408248   0.0         0.558726    0.0         0.446931
 0.408248   0.0        -0.0423861   0.0         0.410261
 0.408248   0.0        -0.0192664   0.0        -0.687834
 0.408248   0.0        -0.5279      0.0         0.214123
 0.408248   0.0         0.466248    0.0        -0.332179
 0.408248   0.0        -0.435421    0.0        -0.0513024
 0.0       -0.549442    0.0        -0.540403    0.0
 0.0        0.0         0.0        -0.636057    0.0
 0.0       -0.137361    0.0         0.47354     0.0
 0.0       -0.824163    0.0         0.281345    0.0

T =
 0.0      2.97209
 2.97209  0.0        5.94127
          5.94127    0.0        7.37874
                     7.37874    0.0
```

There is a large amount of structure that emerges! Let's decode this structure!