Image Compression through Singular Value Decomposition

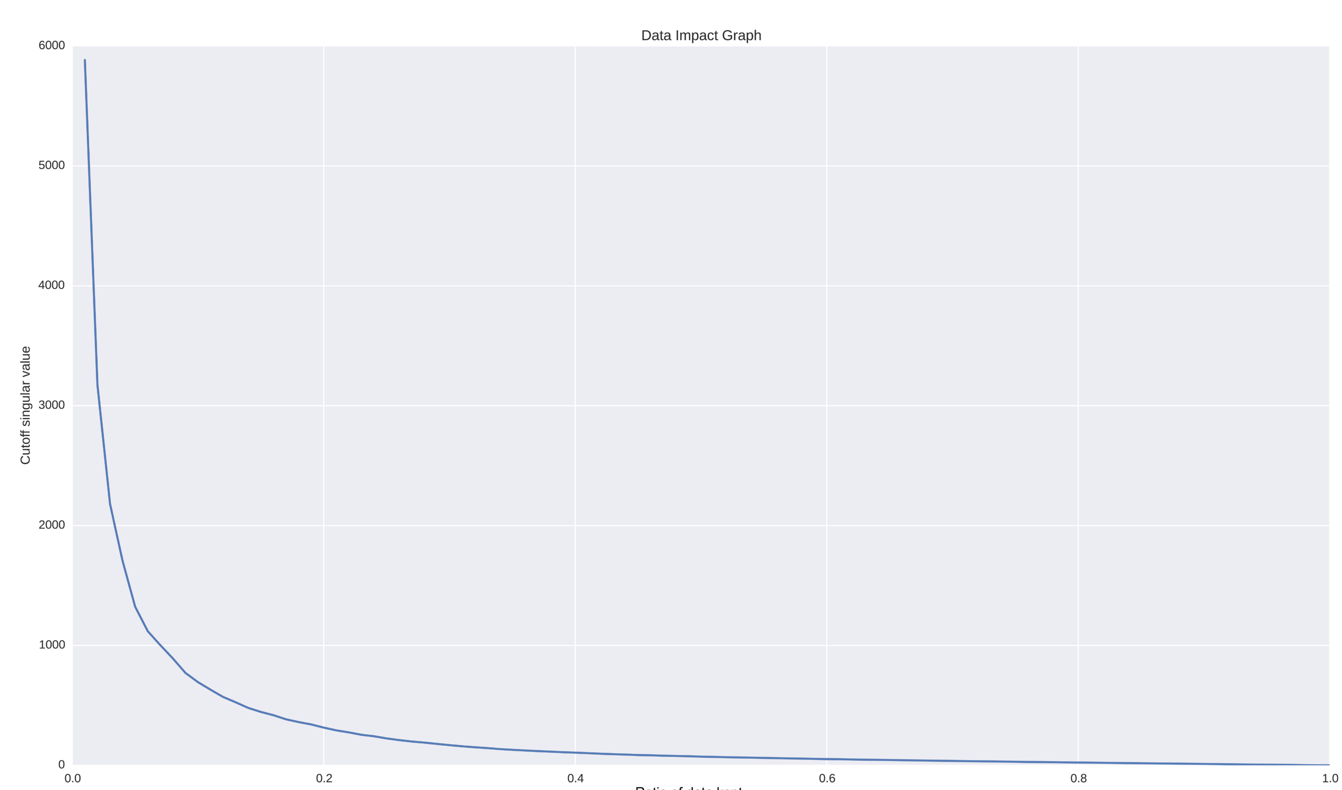
Joey Maalouf, Sean Carter, Ian Tang

WHAT OUR PROJECT DOES WITH SVD

Using SVD, we can reconstruct an image from some subset of its original data, which allows us to store images in a "compressed" form. By "compressed", we mean that a smaller amount of data is stored because we just mathematically reconstruct the image approximation when "decompressing" the file back into an image.

Original Image

Singular Value Impact Graph



1% 2% 20% 100%

Reconstruction of Kept Data

Reconstruction of Discarded Data

99% 98% 80% 0%

OVERVIEW

SVD, which stands for singular value decomposition, is a technique for representing some matrix A as the result of matrix multiplication between three matrices U, S, and V^T. The columns of U are the orthonormal eigenvectors of AA^T, the rows of V^T are the transposed orthonormal eigenvectors of A^TA, and S is a diagonal matrix of the singular values of A. If A is sized MxN, then U is MxM, S is MxN, and V^T is NxN.

ORTHONORMAL

Yes, the eigenvectors that make up the matrices U and V^T must be orthonormal. Since "orthonormal" is a portmanteau of "orthogonal" and "normal", it makes sense that orthonormal vectors are both orthogonal (at right angles, dot products of 0) and normal (unit length, magnitude of 1). MATLAB automatically outputs its eigenvectors as orthonormal, but it's good to know how to do it by hand; for this purpose, we learned the Gram-Schmidt process.

SINGULAR VALUES

We mentioned earlier that S was composed of the singular values of A; any singular value is just the square root of a nonzero eigenvalue. The eigenvalues that we use are those of A^TA and AA^T, which are the same. If A is not square, A^TA and AA^T will have a different number of eigenvalues, but all of the non-overlapping ones should be 0. Higher singular values correspond to larger impact on the matrix values. As the graph to the left shows, the ratio of value to importance is not linear; the few highest values have the biggest impact, and after the exponential drop-off, the majority of the singular values actually have very little impact on the final data.

REASONING

We can recall a few facts we know about matrices:

- A^TA and AA^T must be square matrices
- we can diagonalize any square matrix as PDP^{-1}
- for any orthonormal matrix M, $M^{-1} = M^T$

so we can say that $AA^T = US_1U^T$ and $A^TA = VS_2V^T$, where S_1 and S_2 (as stated earlier) contain the same singular values but may be padded with zeroes for proper sizing. (explain how that leads into $A = USV^T$)

PUTTING IT ALL TOGETHER

Singular value decomposition can be used to represent any matrix A as the sum of n rank-1 matrices, with n being equal to the smaller dimension of A. These matrices are calculated from the vectors that make up individual rows and columns in our U, S, and V^T matrices. If we have example U, S, V^T:

$$\begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} S_{11} & 0 & 0 \\ 0 & S_{22} & 0 \end{bmatrix} \begin{bmatrix} V_{11} & V_{12} & V_{13} \\ V_{21} & V_{22} & V_{23} \\ V_{31} & V_{32} & V_{33} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{bmatrix}$$

We can write A as

$$\begin{bmatrix} U_{11} \\ U_{21} \end{bmatrix} S_{11} \begin{bmatrix} V_{11} & V_{12} & V_{13} \end{bmatrix} + \begin{bmatrix} U_{12} \\ U_{22} \end{bmatrix} S_{22} \begin{bmatrix} V_{21} & V_{22} & V_{23} \end{bmatrix}$$

where each of these multiplications results in a rank-1 matrix that we'll call a component matrix, and we sum them together to get A. In our image compression and decompression program, we treat "compression" as taking the image and removing the component matrices that correspond to smaller singular values. Our "decompression" is just approximating a reconstruction of the original image using what's left.