Singular Value Decomposition

For any $A \in \mathbb{C}^{m \times n}$, its (full) SVD is given by the factorization

$$A = U \Sigma V^*$$

where

$$U \in \mathbb{C}^{m \times m}, V \in \mathbb{C}^{n \times n}, \Sigma \in \mathbb{R}^{m \times n}$$
and $U, V$ are unitary, $\Sigma$ is diagonal with entries

\[ \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0 \]

and

\[ p = \min(m, n). \]
SVD example

Draw pictures for \( m < n, m = n, m < n \).
SVD example

\[
\begin{bmatrix}
1 & 0 \\
0 & 1 \\
1 & 1
\end{bmatrix}
= \begin{bmatrix}
-\frac{\sqrt{2}}{2\sqrt{3}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{3}} \\
\frac{\sqrt{2}}{2\sqrt{3}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{3}} \\
-\frac{\sqrt{2}}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}}
\end{bmatrix}
\begin{bmatrix}
\sqrt{3} & 0 \\
0 & 1 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
-\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}
\end{bmatrix}
\]
SVD, cont

Since \( A = U \Sigma V^* \), we have \( AV = U \Sigma \). So,

\[
A \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix} \Sigma.
\]

Equating columns,

\[
Av_i = \sigma_i u_i, \quad i = 1, \ldots p, \quad p = \min m, n.
\]

Why only to \( p \)?
SVD, cont

When $A$ is tall and skinny $m \times n$, it is possible to write the reduced SVD:

$$A = \hat{U} \hat{\Sigma} V^*$$

where now $\hat{U}$ has only columns 1 to $n$ of $U$, and $\hat{\Sigma}$ is $n \times n$. 
Since $A = UΣV^*$, the $u_i$’s are called the left singular vectors and the $v_i$’s are called the right singular vectors.
Geometry

Assume $A$ is real. FACT: $U, \Sigma, V$ are real.

Now $V$ is a unitary matrix, so $\|v_i\|_2 = 1$, $i = 1, ..n$ and also $v^*_i v_j = \delta_{i j}$.

So the $v_i$’s “live” on the unit sphere (on the hypersphere) in $\mathbb{R}^n$. 

What does the action $A v_i$ do, geometrically, to $v_i$?
Geometry, cont

Recall $A v_j = \sigma_j u_j$, and $\sigma_j \geq 0$, and $u_j$ also form an orthonormal set (so they live on the unit sphere in $\mathbb{R}^m$.)

Where is $\sigma_j u_j$? What is the significance of the $\sigma_j$?
Existence/Uniqueness; Thm 4.1

Every matrix has an SVD. Furthermore, the $\sigma_j$’s are uniquely determined, and, if $A$ is square and the $\sigma_j$ are distinct, then the left and right singular vectors $u_j$ and $v_j$ are uniquely determined up to complex signs (i.e. complex scalar factors of absolute value 1.)

skip proof
Examples

Determine the SVDs of

\[
A = \begin{bmatrix}
5 & 0 \\
0 & -1
\end{bmatrix}, \quad A = \begin{bmatrix}
0 & 2 \\
0 & 0 \\
0 & 0
\end{bmatrix}.
\]

For the 2nd example, what is the rank of \( A \)? Now what is the number of non-zero singular values of \( A \)? (The fact these match is not a coincidence!)
Let us see how the SVD can be used to solve

\[ Ax = b \]

where \( A \) is square and invertible.

Let \( A = U\Sigma V^* \). We get \( U\Sigma V^* x = b \). But, \( U^*U = I \), so....
\[ U^* U \sum_{I} V^* x = U^* b. \]

So, we now have to solve a diagonal system

\[ \sum_{x'} (V^* x) = b'. \]

THIS IS EASY TO SOLVE!!!! What is \( x'_i \)?
Note the change of coordinates.

\[ x' = V^* x \iff Vx' = x \]

This means the components of \( x' \) are the expansion coefficients of \( x \) in the columns of \( V \).

\[ b' = U^* b \iff Ub' = b \]

This means.............?
SVD as a system solver

- How many flops does it take to find $x$ using Gaussian elimination (or LU factorization or Gauss-Jordan)? etc. for a general $n \times n$ matrix $A$?
- How many flops to solve the diagonal system for $x'$?
- How many flops to then compute $x$ via backtransformation?
So how many flops do you think it takes to compute $x$ using the SVD??? **WARNING:** you get nothing for free.
SVD: Matrix Properties

Let $A$ be $m \times n$.

**Theorem 5.1** The rank of $A$ is $r$, the number of non-zero singular values.

**Theorem 5.2** $\mathcal{R}(A) = \text{span}\{u_1, \ldots, u_r\}$, $\mathcal{N}(A) = \text{span}\{v_{r+1}, \ldots, v_n\}$

For all but the math grad students, let’s ignore the proofs.
SVD: Matrix Properties

Theorem 5.3 \( \|A\|_2 = \sigma_1 \) and \( \|A\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2} \).

The proofs follow easily from the unitary invariance of these two matrix norms.

Let’s do the proof on the board........
**Theorem 5.4**: The nonzero singular values of $A$ are the square roots of the nonzero eigenvalues of $A^*A$ OR $AA^*$. 

Let’s do the proof. It will jog your memory about some stuff you learned in linear algebra about similar matrices.
Now that you know that the relation between the singular values of $A$ & eigenvalues of $A^*A$ ($AA^*$), let’s look at the eigenvectors.

\[
A^*A = V\Sigma^T\Sigma V^*, \quad AA^* = U\Sigma\Sigma^TU^*
\]
Thus, if you have an algorithm to compute the eigenvalues/vectors of a matrix, you can use it to compute the SVD, in theory. This is NOT usually the best approach. (see Lecture 31)
**Theorem 5.5**: If $A = A^*$, then the singular values of $A$ are the absolute values of the eigenvalues of $A$.

The proof relies on the fact that a Hermetian matrix has a complete set of orthonormal eigenvectors: i.e., that $A = QΛQ^*$.
Low-Rank Approximations

Let $A$ be $m \times n$, and have rank $r \leq \min(m, n)$. That means $\sigma_1, \ldots, \sigma_r > 0$, but all others are 0.
Consider $m > n$.

$$A = U\Sigma V^* = \begin{bmatrix} u_1 | u_2 | \ldots | u_r | \ldots | u_m \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \cdots & \cdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \cdots & 0 & \sigma_r & \cdots & 0 \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \cdots & \cdots & \cdots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix} \begin{bmatrix} v_1^* \\ v_2^* \\ \vdots \\ v_r^* \\ \vdots \\ v_n^* \end{bmatrix}$$

We can compress the SVD even further than the “reduced” form of the SVD!
Low-Rank Approximations

\[ A = \tilde{U}\tilde{\Sigma}\tilde{V}^* \]

**Theorem 5.7**: \( A \) is the sum of rank-one matrices:

\[ A = \sum_{j=1}^{r} \sigma_j u_j v_j^*. \]
Proof:

1. Multiplication $\tilde{\Sigma}\tilde{V}^*$ scales row $j$ ($v^*_j$) of $V^*$ by $\sigma_j$.

2. Remember the product of two matrices can be written as a sum of outer products (see homework).

3. Remember that an outer product of 2 vectors has rank-1.
Low-Rank Approximations

Property: the $k$th partial sum captures as much of the energy, in 2 norm, of $A$ as possible...

Theorem 5.8: For any $0 \leq k \leq r$, let $A_k = \sum_{j=1}^{k} \sigma_j u_j v_j^*$. Then

$$\|A - A_k\|_2 = \min_{\text{B has rank } \leq k} \|A - B\|_2 = \sigma_{k+1}.$$
Approximation and Geometry

• What’s the “best” approx. of a hyperellipsoid by line segment? The line is the longest axis.

• ” ” by a 2D ellipsoid? Take the longest & second longest axes.

• ” ” by a hyperellipsoid in k dimensions? $A_k$. 
F-norm Approximation

**Theorem 5.9:** The matrix $A_k$ also satisfies

$$
\|A - A_k\|_F = \min_{B \text{ has rank } \leq k} \|A - B\|_F = \sqrt{\sigma_{k+1}^2 + \cdots + \sigma_r^2}
$$
SVD and Image Compression

• A gray-scale image is an array (matrix) of numbers (pixels) we call $A$.

• The “compressed” image is called $A_k$.

• Storage of $A$ (trivially) requires $mn$ terms (let’s assume double precision)

• Storage of $A_k$ requires how many double precision terms?

DEMO
Aside

Interesting (sort-of open) problem:

If $A$ has only signed (or unsigned) integers, can we get a matrix factorization (in reasonable time) to do the compression that also only uses signed (or unsigned) integers?
SVD Computation

• This application shows that there are times when one only wants to compute the $k$ most dominant singular vectors. More examples to come (LSI and PCA).

• In particular, note that to compute $\|A\|_2$ requires computation of $\sigma_1$.

• So-called “iterative” algorithms seek to do this with a minimum of memory required. (Lecture 31)
• The algorithm to compute the SVD of an $n \times n$ matrix is $O(n^3)$, so it’s not any cheaper to use this to solve linear systems!!!

• More **stable**, however.
Rank vs. Numerical Rank

Beware the curse of finite precision arithmetic. A rank-deficient matrix may appear to be full rank when rank is numerically calculated!

More important is numerical rank. In the image example, I would say that the numerical rank is much lower than the dimension.