ME 7247: Advanced Control Systems **Fall 2022–23**

Lecture 05: Singular Value Decomposition

Friday September 23, 2022

Lecturer: Laurent Lessard *Scribe: Mayur Bhise*

Orthogonal Matrices, Singular Value Decomposition

1 Orthonormal Vectors

A vector is said to be normal if it has a length of one. Two vectors are said to be orthogonal if they're at right angles to each other (their dot product is zero). A set of vectors is said to be orthonormal if they are all normal, and each pair of vectors in the set is orthogonal.

Orthonormal vectors are usually used as a basis on a vector space. Establishing an orthonormal basis can make calculations significantly easier. For example, the length of a vector is simply the square root of the sum of the squares of its coordinates when expressed in an orthonormal basis.

Definition 1.1. A set of vectors $\{u_1, u_2, \dots, u_r\}$ in $\mathbb{R}^n$ is *orthonormal* if

$$u_i^\mathsf{T} u_j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

If we define the matrix $U \in \mathbb{R}^{n\times r}$ as $U = \begin{bmatrix} u_1 & u_2 & \cdots & u_r \end{bmatrix}$, then the definition above is equivalent to $U^\mathsf{T} U = I$. In this case, we say that U is a *semi-orthogonal matrix.* If $r = n$ (U is square), we say that U is an *orthogonal matrix.*

Some comments:

- We must have $r \leq n$ (U must be a tall matrix), because it is impossible for more than n vectors to be mutually orthogonal in an n-dimensional space.
- Since the columns of U are orthonormal, we should really call U a "semi-orthonormal matrix", but people have settled on the name "orthogonal".

Orthonormal vectors can be used as a basis for a subspace. In fact, *any* subspace of $\mathbb{R}^n$ has an orthonormal basis. For example, consider $S = \text{span}\{a_1, a_2, a_3\}$. The vectors a_i are not necessarily orthogonal, but we can find a orthogonal basis for S using the Gram–Schmidt process.

Gram–Schmidt. Given a set of vectors $\{a_1, a_2, \dots, a_r\}$, do the following:

1. Let $u_1 = a_1$, then normalize: $u_1 \mapsto \frac{u_1}{\|u_1\|}$.
2. Let $u_2 = a_2 - \text{proj}_{u_1}(a_2)$, then normalize: $u_2 \mapsto \frac{u_2}{\|u_2\|}$.
3. Let $u_3 = a_3 - \text{proj}_{u_1}(a_3) - \text{proj}_{u_2}(a_3)$, then normalize: $u_3 \mapsto \frac{u_3}{\|u_3\|}$.

We then continue in this fashion. The k^{th} step of the process is $u_k = a_k - \sum_{i=1}^{k-1} \text{proj}_{u_i}(a_k)$, then normalize: $u_k \mapsto \frac{u_k}{\|u_k\|}$. If at any point, we have $u_i = 0$, then this means the current a_i is a linear combination of the previous a_i's, so we can simply skip it and move onto the next a_i. When we are done, we will have $\text{span}\{a_i\} = \text{span}\{u_i\}$ with the $\{u_i\}$ forming are orthonormal set.

Here, $\text{proj}_b(a)$ is the projection of a onto the vector b, given by

$$\text{proj}_b(a) = \frac{\langle a, b\rangle}{\langle b, b\rangle} b = \frac{a^\mathsf{T} b}{\|b\|^2} b$$

When $b = u$ is a unit vector, the formula simplifies to $\text{proj}_u(a) = \langle a, u\rangle u = (a^\mathsf{T} u)u$.

Here is a visualization of Gram–Schmidt: https://www.youtube.com/watch?v=KOkuTXrv5Gg

Orthogonal matrices as transformations. Another way to interpret orthogonal and semi-orthogonal matrices is to view them as a transformation from one vector space to another (via matrix multiplication). So if $U \in \mathbb{R}^{n\times r}$ is semi-orthogonal, we think of the map $U : \mathbb{R}^r \to \mathbb{R}^n$ obtained via matrix multiplication. This transformation is an *isometry*; it preserves angles and distances between points.

For example, if $x \mapsto y$ (which means that $y = Ux$) then we have:

$$\langle Ux, Uy\rangle = (Ux)^\mathsf{T}(Uy) = x^\mathsf{T} U^\mathsf{T} U y = x^\mathsf{T} y = \langle x, y\rangle$$

In particular, if $x = y$, we have $\|Ux\| = \langle Ux, Ux\rangle = \langle x, x\rangle = \|x\|$. So angles and distances are preserved if we transform all the points using U.

Orthogonal matrices as coordinate frames. If we have a vector $x \in \mathbb{R}^n$ and an orthogonal matrix $U \in \mathbb{R}^{n\times n}$, we can view the columns of $U = \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix}$ as basis vectors for $\mathbb{R}^n$ and we can ask how to express x in these coordinates. Here is how:

$$\begin{aligned} x &= (UU^\mathsf{T})x && (1)\\ &= U(U^\mathsf{T} x) && (2)\\ &= (u_1^\mathsf{T} x)u_1 + \cdots + (u_n^\mathsf{T} x)u_n && (3) \end{aligned}$$

So the vector $U^\mathsf{T} x$ is the vector of *coordinates* that express x is the basis $\{u_1, \dots, u_n\}$. Because these basis vectors are mutually orthogonal, the coordinates satisfy the Pythagorean theorem!

$$(u_1^\mathsf{T} x)^2 + \cdots + (u_n^\mathsf{T} x)^2 = \|U^\mathsf{T} x\|^2 = x^\mathsf{T} U U^\mathsf{T} x = x^\mathsf{T} x = \|x\|^2$$

Matlab commands. Two useful commands in Matlab:

- `U = orth(A)` returns a semi-orthogonal matrix U whose columns are an orthonormal basis for $\text{range}(A)$. The number of columns of U is equal to $\text{rank}(A)$.
- `V = null(A)` returns a semi-orthogonal matrix U whose columns are an orthonormal basis for $\text{null}(A)$.

2 Singular Value Decomposition

Thin SVD. (also called the *economy SVD*) Let $A \in \mathbb{R}^{m \times n}$. There exists a factorization

$$A = U_1 \Sigma_1 V_1^\mathsf{T}$$

where $U_1 \in \mathbb{R}^{m \times r}$ and $V_1 \in \mathbb{R}^{n \times r}$ are semi-orthogonal, and $\Sigma_1 \in \mathbb{R}^{r \times r}$ is square and diagonal with positive entries that are decreasing along the main diagonal. In other words,

$$\Sigma_1 = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_r \end{bmatrix} \qquad \text{with } \sigma_1 \geq \cdots \geq \sigma_r > 0 \tag{4}$$

If we decompose $U_1 = \begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix}$ and $V_1 = \begin{bmatrix} v_1 & \cdots & v_r \end{bmatrix}$ into their columns, we can write A as a sum of r rank-1 matrices:

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^\mathsf{T} \tag{5}$$

The σ_i are called the *singular values* of A. The u_i are called the *left singular vectors* and the v_i are called the *right singular vectors.*

Uniqueness. The thin SVD is unique in the sense that any decomposition of A of the form $A = U_1 \Sigma_1 V_1^\mathsf{T}$ described above has the same singular values. The singular vectors are not unique, however. There can be ambiguity, e.g. if we flip the sign of u_i and v_i for some particular i, the sum (5) will be unchanged.

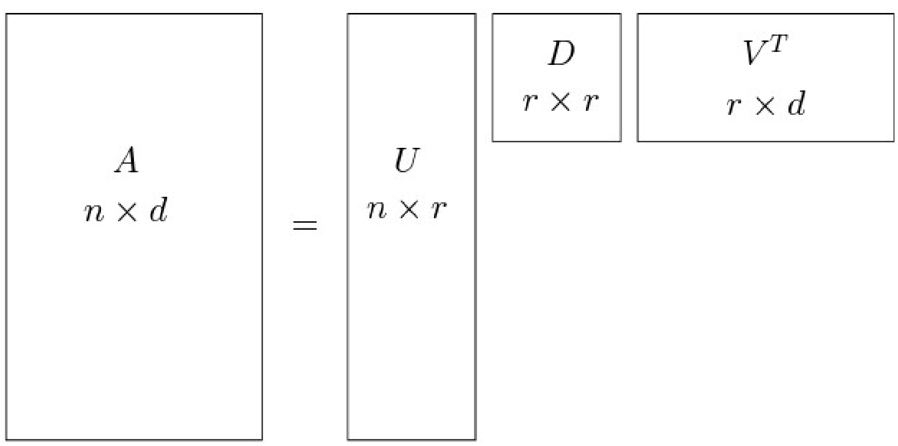

Figure 1: The thin SVD decomposition of an $n \times d$ matrix.

Orthogonal completions. We can group the singular vectors into matrices and find orthogonal completions. In other words, if $\{u_1, \ldots, u_r\}$ are the left singular vectors, Define $\{u_{r+1}, \ldots, u_m\}$ so that $\{u_1, \ldots, u_m\}$ is orthonormal. Then define the matrices:

$$U_1 = \begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix}, \quad U_2 = \begin{bmatrix} u_{r+1} & \cdots & u_m \end{bmatrix}, \quad U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}.$$

So $U_1 \in \mathbb{R}^{m \times r}$ and $U_2 \in \mathbb{R}^{m \times (m-r)}$ are semi-orthogonal, and $U \in \mathbb{R}^{m \times m}$ is orthogonal. Similarly, we can define $V_1 \in \mathbb{R}^{n \times r}$ and $V_2 \in \mathbb{R}^{n \times (n-r)}$ and $V \in \mathbb{R}^{n \times n}$.

We can also derive the following useful formulas, which show a correspondence between the left and right singular vectors:

$$\begin{aligned} Av_i = \sigma_i u_i \quad &\text{and} \quad A^\mathsf{T} u_i = \sigma_i v_i \qquad \text{for } i = 1, \ldots, r \\ Av_i = 0 \quad &\text{and} \quad A^\mathsf{T} u_i = 0 \qquad \text{for } i \geq r+1 \end{aligned}$$

The left and right singular vectors form orthonormal bases for various important subspaces:

$$\begin{aligned} \operatorname{span}\{u_1, \ldots, u_r\} &= \operatorname{range}(U_1) = \operatorname{range}(A) \\ \operatorname{span}\{u_{r+1}, \ldots, u_r\} &= \operatorname{range}(U_2) = \operatorname{range}(A)^\perp \\ \operatorname{span}\{v_1, \ldots, v_r\} &= \operatorname{range}(V_1) = \operatorname{null}(A)^\perp \\ \operatorname{span}\{v_{r+1}, \ldots, v_r\} &= \operatorname{range}(V_2) = \operatorname{null}(A). \end{aligned}$$

These facts can be proved using the definitions of range and nullspace, and SVD properties. For example, here is how we would prove that $\operatorname{range}(V_2) = \operatorname{null}(A)$.

1) *Proof that* $\operatorname{range}(V_2) \subseteq \operatorname{null}(A)$*:* Suppose $x \in \operatorname{range}(V_2)$. Then $x = V_2 w$ for some w. Take the thin SVD of A and compute:

$$Ax = U_1 \Sigma_1 V_1^\mathsf{T} V_2 w = U_1 \Sigma_1 (V_1^\mathsf{T} V_2) w = 0.$$

The last step follows because $V_1^\mathsf{T} V_2 = 0$. Therefore, $x \in \operatorname{null}(A)$ and $\operatorname{range}(V_2) \subseteq \operatorname{null}(A)$.

2) *Proof that* $\operatorname{null}(A) \subseteq \operatorname{range}(V_2)$*:* Suppose $x \in \operatorname{null}(A)$. Then $Ax = 0$. Again, take the thin SVD of A and compute:

$$Ax = 0 \quad \Longrightarrow \quad U_1 \Sigma_1 V_1^\mathsf{T} x = 0 \quad \Longrightarrow \quad \Sigma_1 V_1^\mathsf{T} x = 0 \quad \Longrightarrow \quad V_1^\mathsf{T} x = 0.$$

The second step followed by multiplying both sides by U_1^T and using $U_1^\mathsf{T} U_1 = I$. Using the fact that V is orthogonal, write $x = VV^\mathsf{T} x = V_1(V_1^\mathsf{T} x) + V_2(V_2^\mathsf{T} x) = V_2(V_2^\mathsf{T} x)$. Therefore, $x \in \operatorname{range}(V_2)$, and so $\operatorname{null}(A) \subseteq \operatorname{range}(V_2)$, and this completes the proof. ■

The Full SVD. Instead of using a "thin" U_1 and V_1 and square Σ_1, we can use the orthogonally completed U and V, and pad the Σ_1 with zeros. This leads to the full SVD, which is typically just called "the SVD".

$$A = U \Sigma V^\mathsf{T} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^\mathsf{T} \\ V_2^\mathsf{T} \end{bmatrix} \tag{6}$$

Here, $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal, and $\Sigma \in \mathbb{R}^{m \times n}$ is the same size as A.

3 Computing the SVD

Computing the SVD of a matrix $A \in \mathbb{R}^{m \times n}$ can be carried out as efficiently as computing the eigenvalues of a symmetric matrix of size $\min(m, n)$. Consider the matrix $A^\mathsf{T} A$. Using the SVD, we have:

$$\begin{aligned} A^\mathsf{T} A &= \left(U \Sigma V^\mathsf{T}\right)^\mathsf{T} \left(U \Sigma V^\mathsf{T}\right) \\ &= V \Sigma^\mathsf{T} \left(U^\mathsf{T} U\right) \Sigma V^\mathsf{T} \\ &= V(\Sigma^\mathsf{T} \Sigma) V^\mathsf{T} \\ &= V(\Sigma^\mathsf{T} \Sigma) V^{-1} \end{aligned}$$

Now notice that $\Sigma^\mathsf{T} \Sigma = \operatorname{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_r^2, 0, \dots, 0) \in \mathbb{R}^{n \times n}$. So what we have above is actually an eigenvalue decomposition of $A^\mathsf{T} A$. We can compute this directly using our favorite eigenvalue solver, and this will tell us what the σ_i and corresponding v_i are.

This confirms some facts about linear algebra concerning symmetric matrices:

- Symmetric matrices always have real eigenvalues and the eigenvectors can be chosen to be real as well.
- Symmetric matrices are always orthogonally diagonalizable, i.e. we can always find an eigenvalue decomposition such that the eigenvectors are orthonormal.

Finding the SVD

1. Find an eigenvalue decomposition of $A^\mathsf{T} A = V \Lambda V^{-1}$. Since $A^\mathsf{T} A$ is symmetric and positive semidefinite (we'll see what this means next lecture), the eigenvalues are real and nonnegative and the eigenvectors can be chosen to be orthonormal, so we can pick V such that $V^{-1} = V^\mathsf{T}$. From the derivation above, we also have $\Lambda = \Sigma^\mathsf{T} \Sigma$, so this reveals the right singular vectors v_i and corresponding singular values σ_i.
2. Use the fact that $A v_i = \sigma_i u_i$ to find each of the u_i for $i = 1, \dots, r$.
3. Find an orthogonal completion of the u_i to get $u_{r+1}, \dots, u_m$ such that $U = \begin{bmatrix} u_1 & \cdots & u_m \end{bmatrix}$ is an orthogonal matrix.

Matlab commands. Two useful commands in Matlab:

- `[U,S,V] = svd(A)` returns the full SVD of `A`. So `U` and `V` are orthogonal and `S` is the same size as `A`. These matrices satisfy `A = U*S*V'`.
- `[U1,S1,V1] = svd(A)` returns the thin (economy) SVD of `A`. So `U1` and `V1` are semi-orthogonal and `S1` is square (with dimension equal to the rank of `A`). These matrices satisfy `A = U1*S1*V1'`.

Under the hood, many matlab commands such as `orth`, `null`, and `rank` work by computing the SVD first and then extracting the desired result from U, V, and Σ.