

1 Planning

Recall from our understanding of control that any n -dimensional discrete-time system with state equation

$$\vec{x}(i+1) = A\vec{x}(i) + B\vec{u}(i)$$

can be controlled to reach any desired state from any other in at most n time steps if and only if the controllability matrix

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix}$$

is of full row rank. We will only consider the controllable case from now on.

For simplicity, let's consider the case of scalar control only, so B is in fact a column vector. Then, we know that, starting at an initial state $\vec{x}(0) = \vec{0}$, we can reach any target state \vec{x}^* in n time steps by applying the control

$$\vec{u} = \begin{bmatrix} u(n-1) \\ u(n-2) \\ \vdots \\ u(0) \end{bmatrix}$$

chosen such that

$$\vec{x}^* = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \begin{bmatrix} u(n-1) \\ u(n-2) \\ \vdots \\ u(0) \end{bmatrix} = \mathcal{C}\vec{u}.$$

In the case of scalar control, \mathcal{C} is an $n \times n$ matrix (of full rank, by our assumption of controllability) and so is invertible. Thus, we can uniquely choose our control inputs to be

$$\vec{u} = \mathcal{C}^{-1}\vec{x}^*.$$

However, what if we didn't want to arrive at the state \vec{x}^* after n time steps, but only need to be there after some longer duration $t > n$? Clearly, there are many ways to do so. In particular, notice that for the first $t - n$ steps, we can apply *any* controls we want, since the final n steps will always be sufficient to bring us to \vec{x}^* from wherever we might have ended up.

So are we done? Not quite. Imagine the linear system of our robot car from lab, and consider the problem of bringing it to a particular state (i.e. assigning particular values to the wheel velocities) at a certain time. One way (that is perhaps the most natural) would be to apply a steady input to gradually ramp up the wheel velocities, so we reach the target state at the desired time. Another way, however, could be to apply large random inputs, accelerating and decelerating each wheel, until just before the target time t , at which point

we would apply large controls to set the wheel velocities to their desired values. Clearly, though both approaches accomplish the same goal, the former is more “natural” than the latter.

More generally, in the case of arbitrary controllable linear systems, we will aim to choose a series of scalar inputs \vec{u} that reach the target state while minimizing a cost function, which expressed how “unnatural” our inputs are.

An obvious choice for this cost function would be the norm $\|\vec{u}\|$ of the inputs. In our earlier example of the robot car, it should be clear that the first case of applying steady inputs has a smaller norm than the large, randomly varying inputs of the second case, suggesting that this definition of the cost function is in accordance with our intuition. We will aim to compute the control input that reaches a target value while minimizing this quantity, known as the *minimum energy control*.

2 Minimum Energy Control

In the previous section, we focused on controllable systems with scalar inputs, where \mathcal{C} was of full rank. Now, we will consider the more general problem of arbitrary n -dimensional systems with scalar inputs, with an initial state $\vec{x}[0] = \vec{0}$. We will denote

$$\mathcal{C}_t = \begin{bmatrix} B & AB & A^2B & \cdots & A^{t-1}B \end{bmatrix}$$

where A and B are the matrices in the state transition equation.

By expanding out the state transition matrix, it is clear that we wish to choose the t -dimensional input vector \vec{u} such that

$$\vec{x}[t] = \mathcal{C}_t \vec{u} = \vec{x}^*.$$

We will assume that \vec{x}^* lies in the column space of \mathcal{C}_t , as otherwise the problem is clearly unsolvable since the target state can never be reached. But since \mathcal{C} may have a null space, it is possible for infinitely many solutions \vec{u} to exist, and we wish to pick the solution with minimum norm.

Let’s try to make sense of this problem by drawing a picture. Let \vec{u}_0 be a particular solution to the above equation (that is not necessary of minimum norm), and recall that the null space of \mathcal{C}_t is a subspace of t -dimensional space. Furthermore, observe that for any vector $\vec{a} \in \text{Null}(\mathcal{C}_t)$, $\vec{u}_0 + \vec{a}$ is a solution to our equation, since

$$\mathcal{C}_t(\vec{u}_0 + \vec{a}) = \mathcal{C}_t \vec{u}_0 + \mathcal{C}_t \vec{a} = \mathcal{C}_t \vec{u}_0 = \vec{x}^*.$$

Although we will not prove it here, it is extremely straightforward to see that any possible solution \vec{u} where $\mathcal{C}_t \vec{u} = \vec{x}^*$ can be written in the form $\vec{u}_0 + \vec{a}$ for some $\vec{a} \in \text{Null}(\mathcal{C}_t)$.

Thus, we can plot the space of possible \vec{u} by first plotting all vectors in $\text{Null}(\mathcal{C}_t)$, and then translating them by \vec{u}_0 .

Intuitively, it seems clear that the control input \vec{u} with minimum norm should be one that is entirely orthogonal to $\text{Null}(\mathcal{C}_t)$. This would seem to be analogous to what we saw when studying least squares, where again we were interested in choosing a point on a subspace such that the error vector was orthogonal to said subspace.

Let’s try to prove this rigorously. Imagine that we had some orthonormal basis

$$\vec{v}_1, \vec{v}_2, \dots, \vec{v}_t$$

of our t -dimensional space of inputs, such that the first n $\vec{v}_1, \dots, \vec{v}_n$ form a basis for those vectors orthogonal

to $\text{Null}(\mathcal{C}_t)$, and the remaining $t - n$ vectors $\vec{v}_{n+1}, \dots, \vec{v}_t$ form a basis for $\text{Null}(\mathcal{C}_t)$. We will worry about constructing this basis explicitly later - for now, let's just assume that it exists. (It has to — we can get a basis for the nullspace using 16A techniques and then orthonormalize it using Gram-Schmitt. Then, we can extend it to a full basis for the space again using Gram-Schmitt.)

Consider our \vec{u}_0 expressed in this basis, as follows:

$$\vec{u}_0 = \alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2 + \dots + \alpha_t \vec{v}_t.$$

Observe that the norm of \vec{u}_0 is

$$\begin{aligned} \|\vec{u}_0\| &= \sqrt{\vec{u}_0^\top \vec{u}_0} \\ &= \sqrt{\alpha_1^2 + \alpha_2^2 + \dots + \alpha_t^2}, \end{aligned}$$

since the \vec{v}_i are orthonormal, so $\vec{v}_i^\top \vec{v}_j$ is 1 if $i = j$ and 0 otherwise.

Now, observe that in this basis, any $\vec{a} \in \text{Null}(\mathcal{C}_t)$ can be represented as

$$\vec{a} = \beta_{n+1} \vec{v}_{n+1} + \beta_{n+2} \vec{v}_{n+2} + \dots + \beta_t \vec{v}_t,$$

since (by definition) it has components only in the the null space of \mathcal{C}_t . Thus, we can write any solution $\vec{u} = \vec{u}_0 + \vec{a}$ as

$$\vec{u} = \vec{u}_0 + \vec{a} = \alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2 + \dots + \alpha_n \vec{v}_n + (\alpha_{n+1} - \beta_{n+1}) \vec{v}_{n+1} + \dots + (\alpha_t - \beta_t) \vec{v}_t.$$

In a similar manner to what we did before, we see immediately that the norm of this expression is

$$\|\vec{u}\| = \sqrt{\alpha_1^2 + \alpha_2^2 + \dots + \alpha_n^2 + (\alpha_{n+1} - \beta_{n+1})^2 + \dots + (\alpha_t - \beta_t)^2}.$$

To minimize the norm of \vec{u} , therefore, it is clear that we should set the $\beta_i = \alpha_i$ for all valid $i > n$. Therefore, our minimum energy control must be

$$\begin{aligned} \vec{u} &= \vec{u}_0 + \vec{a} \\ &= \alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2 + \dots + \alpha_n \vec{v}_n + (\alpha_{n+1} - \beta_{n+1}) \vec{v}_{n+1} + \dots + (\alpha_t - \beta_t) \vec{v}_t \\ &= \alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2 + \dots + \alpha_n \vec{v}_n + (\alpha_{n+1} - \alpha_{n+1}) \vec{v}_{n+1} + \dots + (\alpha_t - \alpha_t) \vec{v}_t \\ &= \alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2 + \dots + \alpha_n \vec{v}_n. \end{aligned}$$

Observe that this solution is entirely orthogonal to $\text{Null}(\mathcal{C}_t)$, as we had expected. All that remains now is to demonstrate the existence of the \vec{v}_i .

3 Constructing an Orthonormal Basis

In truth, this task is rather simple. From Gaussian elimination, we know how to compute a basis of $\text{Null}(\mathcal{C}_t)$, which we can orthonormalize using Gram-Schmidt to produce the $\vec{v}_{n+1}, \dots, \vec{v}_n$. We can then extend this basis again using Gram-Schmidt to span all of t -dimensional space, to produce the $\vec{v}_1, \dots, \vec{v}_t$, demonstrating the existence of our desired basis. Although this description is not fully precise, it is very straightforward to make it rigorous and solve the problem of minimum energy control computationally. We don't get nice expressions however and so there is less insight to be had with this purely procedural approach.

Instead, we will choose to attack this problem from a different direction, which is not entirely unreasonable and will prove useful in our later derivation of the SVD. By definition, the \vec{v}_i are an orthonormal basis of t -dimensional space. But recall from earlier that, by the real spectral theorem, the eigenvectors of a real symmetric matrix can give such an orthonormal basis. Wouldn't it be interesting if some symmetric matrix Q had exactly these eigenvectors $\vec{v}_1, \dots, \vec{v}_t$? To be extra useful, it would be nice if $\vec{v}_{n+1}, \dots, \vec{v}_t$ are in fact all members of $\text{Null}(Q)$ — in other words, it would be great if these were all the eigenvectors corresponding to $\lambda = 0$. That is, we want Q to have the same nullspace as \mathcal{C}_t .

How do we generate such a symmetric matrix Q ? Since it has the same null space as \mathcal{C}_t , perhaps we could try writing it in the form

$$Q = D\mathcal{C}_t,$$

where D is some unknown matrix of full column rank, so Q and \mathcal{C}_t clearly have the same null space. Now, as Q is symmetric, we can take transposes to obtain

$$Q = Q^T = \mathcal{C}_t^T D^T \implies D\mathcal{C}_t = \mathcal{C}_t^T D^T.$$

Looking at the latter equality, a natural conjecture would be to try $D = \mathcal{C}_t^T$, so $Q = \mathcal{C}_t^T \mathcal{C}_t$. Let's see if this works out.

It is obvious that $\text{Null}(\mathcal{C}_t) \subseteq \text{Null}(Q)$. Let's try to prove the opposite relation, in order to show equality. Specifically, for any $\vec{v} \in \text{Null}(Q)$, we'd like to show that $\vec{v} \in \text{Null}(\mathcal{C}_t)$. Well, for any given such \vec{v} , by definition it is true that

$$Q\vec{v} = \vec{0} \implies \mathcal{C}_t^T \mathcal{C}_t \vec{v} = \vec{0}.$$

We'd like to show that $\mathcal{C}_t \vec{v} = \vec{0}$, which is equivalent to writing $\|\mathcal{C}_t \vec{v}\| = 0$. By the definition of the norm of a real vector, we'd like to show that

$$(\mathcal{C}_t \vec{v})^T (\mathcal{C}_t \vec{v}) = \vec{v}^T \mathcal{C}_t^T \mathcal{C}_t \vec{v} = 0.$$

But wait! This is basically what we had in the definition of \vec{v} , just pre-multiplied by \vec{v}^T . Putting everything together into a proof, we have

$$\begin{aligned} & \vec{v} \in \text{Null}(Q) \\ \implies & Q\vec{v} = \vec{0} \\ \implies & \mathcal{C}_t^T \mathcal{C}_t \vec{v} = \vec{0} \\ \implies & \vec{v}^T \mathcal{C}_t^T \mathcal{C}_t \vec{v} = 0 \\ \implies & \|\mathcal{C}_t \vec{v}\| = 0 \\ \implies & \mathcal{C}_t \vec{v} = \vec{0} \\ \implies & \vec{v} \in \text{Null}(\mathcal{C}_t), \end{aligned}$$

so $\text{Null}(Q) \subseteq \text{Null}(\mathcal{C}_t)$, as we desired. Since we've proven this inequality in both directions, we have $\text{Null}(Q) = \text{Null}(\mathcal{C}_t)$, as we had hoped.

Thus, we can produce an orthonormal basis of the eigenspace corresponding to $\lambda = 0$ of Q that provides us with the $\vec{v}_{n+1}, \dots, \vec{v}_t$ that we wanted.

Considering the remaining eigenvectors of Q for $\lambda \neq 0$, it is clear (by the real spectral theorem) that they are all mutually orthogonal and will all be orthogonal to $\text{Null}(Q)$. So we can choose n of them to form our $\vec{v}_1, \dots, \vec{v}_n$, completing our construction of the $\{\vec{v}_i\}$.

4 Singular Values

Now that we know how to construct this Q , and have demonstrated that its eigenvectors are exactly the $\vec{v}_1, \dots, \vec{v}_t$ that we had wanted, it is natural to speculate on the eigenvalues of each of these eigenvectors. We know that the eigenvalues of $\vec{v}_{n+1}, \dots, \vec{v}_t$ are all 0, since they lie in the null space of Q . But what about the first n eigenvectors? Let's try to work this out algebraically.

By the definition of eigenvectors, we can perform manipulations very similar to what we did earlier involving nullspaces, to see that

$$\begin{aligned}
 Q\vec{v}_i &= \lambda_i\vec{v}_i \\
 \implies \mathcal{C}_t^\top \mathcal{C}_t \vec{v}_i &= \lambda_i\vec{v}_i \\
 \implies \vec{v}_i^\top \mathcal{C}_t^\top \mathcal{C}_t \vec{v}_i &= \lambda_i\vec{v}_i^\top \vec{v}_i \\
 \implies \vec{v}_i^\top \mathcal{C}_t^\top \mathcal{C}_t \vec{v}_i &= \lambda_i\vec{v}_i^\top \vec{v}_i \\
 \implies \|\mathcal{C}_t\vec{v}_i\|^2 &= \lambda_i\|\vec{v}_i\|^2 \\
 \implies \lambda_i &= \frac{\|\mathcal{C}_t\vec{v}_i\|^2}{\|\vec{v}_i\|^2} \\
 &= \|\mathcal{C}_t\vec{v}_i\|^2.
 \end{aligned}$$

In particular, notice that $\lambda_i \geq 0$. For $i > n$, we already knew that $\lambda_i = 0$. But for $i \leq n$, since $\vec{v}_i \notin \text{Null}(\mathcal{C}_t)$, $\lambda_i \neq 0$, so $\lambda_i > 0$.

This sounds interesting. Without loss of generality, we can order our \vec{v}_i such that

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > \lambda_{n+1} = \lambda_{n+2} = \dots = \lambda_t = 0,$$

and place them as columns of the eigenvector matrix

$$V = \begin{bmatrix} | & & | \\ \vec{v}_1 & \dots & \vec{v}_t \\ | & & | \end{bmatrix}.$$

We can then write the eigendecomposition of Q as

$$Q = V\Lambda V^{-1} = V \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_t \end{bmatrix} V^\top.$$

Since all the $\lambda_i \geq 0$, we can choose

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > \sigma_{n+1} = \sigma_{n+2} = \dots = \sigma_t = 0$$

where $\lambda_i = \sigma_i^2$ for all i . Recall that $\lambda_i = \|\mathcal{C}_t\vec{v}_i\|^2$, so $\sigma_i = \|\mathcal{C}_t\vec{v}_i\|$. These σ_i are known as the *singular values* of \mathcal{C}_t , and will prove important in the next section.

5 Constructing Another Orthonormal Basis For the Output

Now, we can change coordinates of our input \vec{u} to be in V -basis. If we do that, we get a new matrix for \mathcal{C} that takes inputs in these new coordinates. Since $V^\top = V^{-1}$ by orthonormality, we know that $\tilde{\mathcal{C}} = \mathcal{C}VV^{-1} = (\mathcal{C}V)V^{-1}$. This means that if we consider $\vec{u} = V^{-1}\tilde{u}$ to be the input in V -basis, then $\tilde{\mathcal{C}} = \mathcal{C}V$ would be the counterpart of \mathcal{C} . Since $\vec{v}_{n+1}, \dots, \vec{v}_t$ are all in the nullspace of \mathcal{C} , we know that the last $t - n$ columns of $\tilde{\mathcal{C}}$ are all $\vec{0}$. So what about the first n columns of $\tilde{\mathcal{C}}$? These are $\mathcal{C}\vec{v}_i$ for $i = 1, \dots, n$. What are these like? After all, this is the matrix that we would have to invert in order to find the minimum energy controls in this new basis.

For any valid choice of i, j , we see that

$$\begin{aligned} (\mathcal{C}_i\vec{v}_i)^\top (\mathcal{C}_i\vec{v}_j) &= \vec{v}_j^\top \mathcal{C}_i^\top \mathcal{C}_i \vec{v}_i \\ &= \sigma_i^2 \vec{v}_j^\top \vec{v}_i = \begin{cases} 0 & \text{if } i \neq j \\ \sigma_i^2 & \text{if } i = j \end{cases} \end{aligned}$$

since \vec{v}_i and \vec{v}_j are orthonormal. This is great news. This suggests that this is going to be easy to invert.

If the $\mathcal{C}_i\vec{v}_i$ are orthogonal, it'd be reasonable to try and use them to create an orthonormal basis of the column space of \mathcal{C}_i .

We have n nonzero mutually orthogonal vectors corresponding to $i \leq n$, and the column space of \mathcal{C}_i is n dimensional, so we're still OK! To make each of these vectors of unit length, we should scale them down by their length $\|\mathcal{C}_i\vec{v}_i\| = \sigma_i$, to obtain the orthonormal basis $\vec{w}_1, \vec{w}_2, \dots, \vec{w}_n$, where we define

$$\vec{w}_i = \frac{\mathcal{C}_i\vec{v}_i}{\sigma_i}$$

for all valid $i \leq n$. Rearranging, we get

$$\mathcal{C}_i\vec{v}_i = \sigma_i\vec{w}_i.$$

Horizontally stacking this result for all $i \leq n$, we find that

$$\mathcal{C}_i \begin{bmatrix} | & & | \\ \vec{v}_1 & \dots & \vec{v}_n \\ | & & | \end{bmatrix} = \begin{bmatrix} | & & | \\ \vec{w}_1 & \dots & \vec{w}_n \\ | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix}.$$

Is anything missing? Well, recall that each of the \vec{v}_i was t dimensional, but we only use n of them here, so the matrix stacking them horizontally is not square! To fix this, we can simply "pad" that matrix with the remaining \vec{v}_i for $i > n$, and introduce additional zero columns to the diagonal matrix of the σ_i , to obtain

$$\mathcal{C}_i \begin{bmatrix} | & & | \\ \vec{v}_1 & \dots & \vec{v}_t \\ | & & | \end{bmatrix} = \begin{bmatrix} | & & | \\ \vec{w}_1 & \dots & \vec{w}_n \\ | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n & 0 & \dots & 0 \end{bmatrix},$$

Now, since V is a square orthogonal matrix, $V^{-1} = V^\top$, so we can rearrange to obtain

$$\mathcal{C}_t = \begin{bmatrix} | & & | \\ \vec{w}_1 & \dots & \vec{w}_n \\ | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} | & & | \\ \vec{v}_1 & \dots & \vec{v}_t \\ | & & | \end{bmatrix}^\top = W\Sigma V^\top,$$

defining U and Σ to be the horizontally stacked \vec{w}_i and the diagonal matrix of the σ_i respectively.

The above decomposition is called the Singular Value Decomposition (SVD) of \mathcal{C} . We will return to it in a moment.

6 Applications to Planning

The singular value decomposition of \mathcal{C}_t can be interpreted as follows - the columns of V formed a t -dimensional orthonormal basis of the domain of \mathcal{C}_t , and the columns of W formed an n -dimensional orthonormal basis of its column space.

Moreover, they mapped between each other in a very clean manner, with $\mathcal{C}_t \vec{v}_i = \sigma_i \vec{w}_i$ for $i \leq n$, and $\mathcal{C}_t \vec{v}_i = \vec{0}$ for $i > n$. Let's see how we can use this property of the SVD can help us find the minimum cost control input.

Recall that we showed that our control vector \vec{u} must have components only along $\vec{v}_1, \dots, \vec{v}_n$, in order to minimize its norm. Thus, for $\mathcal{C}_t \vec{u} = \vec{x}^*$, we can use projections onto these two bases to see that

$$\begin{aligned} \mathcal{C}_t \vec{u} &= \vec{x}^* \\ \implies \mathcal{C}_t \left(\sum_{i=1}^n \langle \vec{u}, \vec{v}_i \rangle \vec{v}_i \right) &= \sum_{i=1}^n \langle \vec{x}^*, \vec{w}_i \rangle \vec{w}_i \\ \implies \sum_{i=1}^n \langle \vec{u}, \vec{v}_i \rangle (\mathcal{C}_t \vec{v}_i) &= \sum_{i=1}^n \langle \vec{x}^*, \vec{w}_i \rangle \vec{w}_i \\ \implies \sum_{i=1}^n \sigma_i \langle \vec{u}, \vec{v}_i \rangle \vec{w}_i &= \sum_{i=1}^n \langle \vec{x}^*, \vec{w}_i \rangle \vec{w}_i, \end{aligned}$$

so

$$\sigma_i \langle \vec{u}, \vec{v}_i \rangle = \langle \vec{x}^*, \vec{w}_i \rangle \implies \langle \vec{u}, \vec{v}_i \rangle = \frac{\langle \vec{x}^*, \vec{w}_i \rangle}{\sigma_i}.$$

for all valid $i \leq n$. Substituting back into our expression for \vec{u} , recalling that it does not have any components in $\text{Null}(\mathcal{C}_t)$, we obtain

$$\vec{u} = \sum_{i=1}^n \frac{\langle \vec{x}^*, \vec{w}_i \rangle}{\sigma_i} \vec{v}_i,$$

which is a clean expression for the minimum energy control to reach our desired state in terms of the SVD.

7 Outer Products

Now, we will look at a new interpretation of matrix multiplication, in order to construct an alternative way of writing the SVD in terms of what are known as *outer products*.

Recall that, for real vectors \vec{x} and \vec{y} expressed as columns with n components, their inner product is defined as $\vec{y}^\top \vec{x}$, which yields a 1×1 matrix typically treated as a scalar.

Similarly, we will define their *outer product* to be $\vec{x}\vec{y}^\top$. Let's see what this means. Let

$$\vec{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix}^\top$$

$$\vec{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}^\top,$$

where it is possible that $m \neq n$. Then, by the definition of matrix multiplication,

$$\vec{x}\vec{y}^\top = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix} = \begin{bmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_n \\ x_2y_1 & x_2y_2 & \cdots & x_2y_n \\ \vdots & \vdots & \ddots & \vdots \\ x_my_1 & x_my_2 & \cdots & x_my_n \end{bmatrix}.$$

So while the inner product took two vectors of the same dimension and produced a scalar, the outer product takes two vectors of possibly different dimensions and yields a matrix!

Furthermore, notice that this matrix cannot be any arbitrary matrix - since each of its columns are a scalar multiple of \vec{x} , it cannot be of a rank greater than 1. It is straightforward to show that any matrix of rank 0 or 1 can be produced by an outer product of two vectors, but we will not discuss the details here.

Now, why are we interested in the outer product? Well, recall that we can express real matrix multiplication in terms of inner products. Specifically, we know that

$$\begin{bmatrix} - & \vec{x}_1^\top & - \\ \cdots & \cdots & \cdots \\ - & \vec{x}_m^\top & - \end{bmatrix} \begin{bmatrix} | & & | \\ \vec{y}_1 & \cdots & \vec{y}_n \\ | & & | \end{bmatrix} = \begin{bmatrix} \vec{x}_1^\top \vec{y}_1 & \vec{x}_1^\top \vec{y}_2 & \cdots & \vec{x}_1^\top \vec{y}_n \\ \vec{x}_2^\top \vec{y}_1 & \vec{x}_2^\top \vec{y}_2 & \cdots & \vec{x}_2^\top \vec{y}_n \\ \vdots & \vdots & \ddots & \vdots \\ \vec{x}_m^\top \vec{y}_1 & \vec{x}_m^\top \vec{y}_2 & \cdots & \vec{x}_m^\top \vec{y}_n \end{bmatrix},$$

where the element at the i th row and j th column of the product of two matrices X and Y is the dot product of the i th row of X and the j th column of Y .

However, what if we were interested in the columns of X and the rows of Y instead? As it turns out, it is the case that

$$\begin{bmatrix} | & & | \\ \vec{x}_1 & \cdots & \vec{x}_n \\ | & & | \end{bmatrix} \begin{bmatrix} - & \vec{y}_1^\top & - \\ \cdots & \cdots & \cdots \\ - & \vec{y}_n^\top & - \end{bmatrix} = \vec{x}_1\vec{y}_1^\top + \vec{x}_2\vec{y}_2^\top + \cdots + \vec{x}_n\vec{y}_n^\top.$$

This result can be proved by applying the definition of matrix multiplication, but it is tedious and so will be omitted.

Instead, we will look at an example that demonstrates the main ideas behind the proof. Consider the product

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}.$$

From our knowledge of the matrix product as the composition of dot products representing each element in

the result, we obtain

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 \cdot 5 + 2 \cdot 7 & 1 \cdot 6 + 2 \cdot 8 \\ 3 \cdot 5 + 4 \cdot 7 & 3 \cdot 6 + 4 \cdot 8 \end{bmatrix}.$$

We will not simplify this result, for reasons that will become clear in a moment. Now, calculating the product of these matrices using outer products, we obtain

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \begin{bmatrix} 5 & 6 \end{bmatrix} + \begin{bmatrix} 2 \\ 4 \end{bmatrix} \begin{bmatrix} 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 \cdot 5 & 1 \cdot 6 \\ 3 \cdot 5 & 3 \cdot 6 \end{bmatrix} + \begin{bmatrix} 2 \cdot 7 & 2 \cdot 8 \\ 4 \cdot 7 & 4 \cdot 8 \end{bmatrix}.$$

Notice that the terms in our matrix multiplication evaluated using dot products correspond exactly to the terms in our sums of outer products, so our outer product definition of matrix multiplication is consistent with our previous definitions, at least in this example. A slightly more rigorous form of this argument can be used to prove this result in general, but it is best to understand this result at an intuitive level.

8 Outer Product Form of the SVD

Now, we will use this new interpretation of matrix multiplication as a sum of outer products in order to obtain an alternative way of expressing the SVD. Recall that the SVD of a matrix A represented it as the product

$$A = U \Sigma V^T,$$

where Σ was a matrix with nonzero entries only along its main diagonal. Let A be an $m \times n$ matrix, so U is a square $m \times m$ matrix and V^T is a square $n \times n$ matrix. Additionally, without loss of generality, assume that $m \geq n$, so A is a “tall” matrix (the same results can be obtained if A is “wide” by considering A^T , which would become “tall”).

Let the columns of U be \vec{u}_1 through \vec{u}_m , the nonzero diagonal entries of Σ be σ_1 through σ_n (moving from the top-left to the bottom-right entry), and the rows of V^T be \vec{v}_1^T through \vec{v}_n^T . From our understanding of outer products, we know how to express UV^T in terms of the \vec{u}_i and \vec{v}_i . But how does Σ affect this interpretation?

Let’s consider just the first two terms of the SVD - the product $U\Sigma$. By the outer product interpretation of matrix multiplication, we have that

$$\begin{aligned} U\Sigma &= \begin{bmatrix} | & & | \\ \vec{u}_1 & \cdots & \vec{u}_m \\ | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \sigma_n \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \\ &= \vec{u}_1 \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \end{bmatrix} + \vec{u}_2 \begin{bmatrix} 0 & \sigma_2 & \cdots & 0 \end{bmatrix} + \cdots + \vec{u}_n \begin{bmatrix} 0 & 0 & \cdots & \sigma_n \end{bmatrix} \\ &\quad + \vec{u}_{n+1} \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix} + \cdots + \vec{u}_m \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix} \\ &= \begin{bmatrix} | & | & \cdots & | \\ \sigma_1 \vec{u}_1 & \sigma_2 \vec{u}_2 & \cdots & \sigma_n \vec{u}_n \\ | & | & & | \end{bmatrix}, \end{aligned}$$

since the inner products involving the zero row vector yield the zero matrix, and so can be disregarded.

Notice what has happened here. The σ_i have acted as coefficients for the first n columns of \vec{u} , with the subsequent columns vanishing entirely. Now, we can multiply by V and apply the outer-product interpretation of matrix multiplication to obtain

$$\begin{aligned} A &= U\Sigma V^T \\ &= \begin{bmatrix} | & | & \cdots & | \\ \sigma_1 \vec{u}_1 & \sigma_2 \vec{u}_2 & \cdots & \sigma_n \vec{u}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} - & \vec{v}_1^T & - \\ - & \vec{v}_2^T & - \\ & \cdots & \\ - & \vec{v}_n^T & - \end{bmatrix} \\ &= \sigma_1 \vec{u}_1 \vec{v}_1^T + \sigma_2 \vec{u}_2 \vec{v}_2^T + \cdots + \sigma_n \vec{u}_n \vec{v}_n^T, \end{aligned}$$

so any $m \times n$ matrix A (where $m \geq n$) can be expressed as the weighted sum of n rank-1 matrices of the form $\vec{u}_i \vec{v}_i^T$, where the set of \vec{u}_i and \vec{v}_i are each mutually orthonormal. This interpretation of the SVD is known as the *outer product form*.

Contributors:

- Rahul Arya.
- Anant Sahai.