Singular Value Decomposition as Simply as Possible

Singular Value Decomposition (SVD) is powerful and ubiquitous tool for matrix factorization but explanations often provide little intuition. My goal is to explain SVD as simply as possible before working towards the formal definition.

PHILOSOPHY
10 November 2018

Beyond the definition

In my experience, singular value decomposition (SVD) is typically presented in the following way: any matrix $M \in \mathbb{C}^{m \times n}$ can be decomposed into three matrices,

$$M = U\Sigma V^* \tag{1}$$

where U is an $m \times m$ unitary matrix, Σ is an $m \times n$ diagonal matrix, and V is an $n \times n$ unitary matrix. V^* is the conjugate transpose of V. Depending on the source, the author may then take a few sentences to comment on how this equation can be viewed as decomposing the linear transformation M into two rotation and one dilation transformations or that a motivating geometric fact is that the image of a unit sphere under any $m \times n$ matrix is a hyperellipse. But how did we arrive at this equation and how do these geometrical intuitions relate to the equation above?

The goal of this post is simple: I want explain SVD beyond this definition. Rather than present Equation 1 in its final form, I want to build up to it from first principles. I will begin with an explanation of SVD without jargon and then formalize the explanation to provide the definition in Equation 1. Finally, I will discuss an application of SVD that will demonstrate its utility. In future posts, I hope to prove the existence and uniqueness of the SVD and explore connections between SVD in detail.

This post will rely heavily on a geometrical understanding of matrices. If you're unfamiliar with that concept, please read my [previous post on the subject](#).

SVD without jargon

Jargon is useful when talking within a community of experts, but I find that, at least for myself, it is easy to use jargon to mask when I do not understand something. If I were to present SVD to a mathematician, the power of jargon is that I could convey the key ideas rapidly and precisely. The downside is that I could also simply use the words without fully understanding the underlying idea, while my expert listener filled in the gaps. As an exercise, I want to first present the SVD without jargon, as if explaining it to an interested 14-year-old—think eighth-grade-level mathematical maturity. I will then formalize this intuitive explanation to work towards the standard definition we see above.

Imagine we have a square. The square, like your hand forming an "L" has a particular orientation which we graphically represent with arrows (Figure 1).

Figure 1: A square with oriented arrows.

We can manipulate this square in various ways. For example, we can pull or push on an edge to stretch or compress the square (Figure 2A and 2B). We can rotate the square (Figure 2C) or flip it to change its orientation—imagine turning your hand over so that the "L" becomes an "J" (Figure 2D). We can even shear the square, which means to deform it by applying a force either up, down, left, or right at one of the square's corners (Figure 2E).

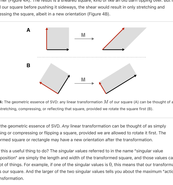

Figure 2: Our original square under different types of transformations: (A) stretched, (B) compressed, (C) rotated, (D) reflected or flipped, and (E) sheared.

The only constraint is that our transformation must be linear. Intuitively, a linear transformation is one in which a straight line before the transformation results in a straight line after the transformation. To visualize what this means, imagine a grid of evenly spaced vertical and horizontal lines on our square. Now let's draw a diagonal line on the square and perform some transformation. A linear transformation will be one such that after the transformation, this diagonal line is still straight (Figure 3B). To imagine the nonlinear transformation in Figure 3C, imagine bending a piece of engineering paper by pushing two sides together so that it bows in the middle.

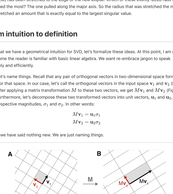

Figure 3: (A) Our original square under a linear transformation M (B) and a nonlinear transformation M' (C).

Now that we've defined our square and what we can do to it, we can state a cool mathematical fact. Consider any linear transformation M, which we will apply to our square. If we are allowed to rotate our square before applying M, then we can find a rotation such that, by first applying the rotation and then applying M, we transform our square into a rectangle. In other words, if we rotate the square before applying M, then M just stretches, compresses, or flips our square. We can avoid our square being sheared.

Let's visualize this with an example. Imagine we sheared our square by pushing horizontally on its top left corner (Figure 4A). The result is a sheared square, kind of like an old barn-tipping over. But if we rotated our square before pushing it sideways, the shear would result in only stretching and compressing the square, albeit in a new orientation (Figure 4B).

Figure 4: The geometric essence of SVD: any linear transformation M of our square (A) can be thought of as simply stretching, compressing, or reflecting that square, provided we rotate the square first (B).

This is the geometric essence of SVD. Any linear transformation can be described as simply stretching or compressing or flipping a square, provided we are allowed to rotate it first. The transformed square or rectangle may have a new orientation after the transformation.

Why is this a useful thing to do? The singular values referred to in the name "singular value decomposition" are simply the length and width of the transformed square, and these values can tell you a lot of things. For example, if one of the singular values is 0, this means that our transformation flattens our square. And the larger of the two singular values tells you about the maximum "action" of the transformation.

If that second statement doesn't make sense, consider visualizing our transformation without the second rotation, which does not effect the size of the rectangle in any way. Also, instead of stretching (or flattening) a square into a rectangle, imagine stretching a circle into an ellipse (We'll see why in a second.) (Figure 5).

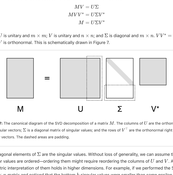

Figure 5: (A) An oriented circle. If it helps, imagine that circle inscribed in our original square. (B) Our circle transformed into an ellipse. The length of the major and minor axes of the ellipse have values σ_1 and σ_2 respectively, called the singular values.

What does Figure 5 represent geometrically? The larger of the two singular values is the length of major axis of the ellipse. And since we transformed a perfect circle, every possible radii (the edge of the circle) has been stretched to the edge of the new ellipse. Which of these evenly-sized radii was stretched the most? The one pulled along the major axis. So the radius that was stretched the most was stretched an amount that is exactly equal to the largest singular value.

From intuition to definition

Now that we have a geometrical intuition for SVD, let's formalize these ideas. At this point, I am going to assume the reader is familiar with basic linear algebra. We want re-embrace jargon to speak precisely and efficiently.

First, let's name things. Recall that any pair of orthogonal vectors in two-dimensional space forms a basis for the plane. In our case, let's call the orthogonal vectors in the input space $\mathbf{v}_1$ and $\mathbf{v}_2$ (Figure 6A). After applying a matrix transformation M to these two vectors, we get $M\mathbf{v}_1$ and $M\mathbf{v}_2$ (Figure 6B). Furthermore, let's decompose these two transformed vectors into unit vectors, $\mathbf{u}_1$ and $\mathbf{u}_2$, times their respective magnitudes, σ_1 and σ_2, in other words:

$$\begin{aligned} M\mathbf{v}_1 &= \mathbf{u}_1\sigma_1 \\ M\mathbf{v}_2 &= \mathbf{u}_2\sigma_2 \end{aligned} \tag{2}$$

So far, we have said nothing new. We are just naming things.

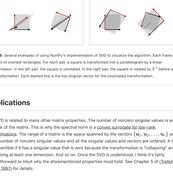

Figure 6: Formalizing the geometric essence of SVD: if we properly rotate our domain defined by the basis vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, then any linear transformation M is just a transformation by a diagonal matrix (dilating, reflecting) in a potentially-rotated range defined by $\mathbf{u}_1$ and $\mathbf{u}_2$.

But now that we have things named, we can do a little algebraic manipulation. First, note that any vector $\mathbf{x}$ can be described using the basis vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ in the following way,

$$\mathbf{x} = (\mathbf{x} \cdot \mathbf{v}_1)\mathbf{v}_1 + (\mathbf{x} \cdot \mathbf{v}_2)\mathbf{v}_2 \tag{3}$$

where $\mathbf{a} \cdot \mathbf{b}$ denotes the dot product between vectors $\mathbf{a}$ and $\mathbf{b}$. If you're unsure about the above formulation, consider a similar formulation with the standard basis vectors:

$$\mathbf{x} = x_1\begin{bmatrix}1\\0\end{bmatrix} + x_2\begin{bmatrix}0\\1\end{bmatrix}$$

where x_i denotes the i-th component of $\mathbf{x}$. In Equation 3, we are [projecting](#) via the dot product $\mathbf{x}$ onto $\mathbf{v}_1$ and $\mathbf{v}_2$, before decomposing the terms using the basis vectors $\mathbf{v}_1$ and $\mathbf{v}_2$.

Next, let's left-multiply both sides of Equation 3 by our matrix transformation M. Since the dot product produces a scalar, we can distribute and commute M as follows:

$$\begin{aligned} M\mathbf{x} &= (\mathbf{x} \cdot \mathbf{v}_1)M\mathbf{v}_1 + (\mathbf{x} \cdot \mathbf{v}_2)M\mathbf{v}_2 \\ \\ \text{Next, we can rewrite } M\mathbf{v}_1 \text{ as } \mathbf{u}_1\sigma_1\text{:} \\ \\ M\mathbf{x} &= (\mathbf{x} \cdot \mathbf{v}_1)\mathbf{u}_1\sigma_1 + (\mathbf{x} \cdot \mathbf{v}_2)\mathbf{u}_2\sigma_2 \end{aligned}$$

We're almost there. Now since the dot product is commutative, we can switch the ordering, e.g. $\mathbf{x} \cdot \mathbf{v}_1 = \mathbf{v}_1 \cdot \mathbf{x}$. And since each dot product term is a scalar, we move each one to the end of their respective expressions:

$$M\mathbf{x} = \mathbf{u}_1\sigma_1(\mathbf{v}_1 \cdot \mathbf{x}) + \mathbf{u}_2\sigma_2(\mathbf{v}_2 \cdot \mathbf{x})$$

Now we can remove $\mathbf{x}$ from both sides of the equation because, in general, $A\mathbf{x} = B\mathbf{x} \implies A = B$. For details, see the discussion [here](#). It might be easier to intuit if we rewrite the claim as $(A - B)\mathbf{x} = \mathbf{0} \implies A - B = 0$. Removing $\mathbf{x}$, we get:

$$M = \mathbf{u}_1\sigma_1\mathbf{v}_1 + \mathbf{u}_2\sigma_2\mathbf{v}_2 \tag{4}$$

Given appropriately defined matrices, Equation 4 becomes SVD in its canonical form (Equation 1) for 2×2 matrices:

$$M = \underbrace{\begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2\end{bmatrix}}_{U}\underbrace{\begin{bmatrix}\sigma_1 & 0\\ 0 & \sigma_2\end{bmatrix}}_{\Sigma}\underbrace{\begin{bmatrix}\mathbf{v}_1^*\\ \mathbf{v}_2^*\end{bmatrix}}_{V^*} \tag{5}$$

Furthermore, you should have an intuition for what this means. Any matrix transformation can be represented as a diagonal transformation (dilating, reflecting) defined by Σ provided the domain and range are properly rotated first.

The vectors $\mathbf{u}_i$ are called the left singular vectors, while the vectors $\mathbf{v}_i$ are called the right singular vectors. This orientating terminology is a bit confusing because "left" and "right" come from the equation above, while in our diagrams of rectangles and ellipses, the $\mathbf{v}_i$ vectors are on the left.

The standard formulation

Now that we have a simple geometrical intuition for SVD and have formalized it for all 2×2 matrices, let's re-formulate the problem in general and in the standard way. If jumping from 2-dimensions to n-dimensions is challenging, recall the advice of Geoff Hinton:

> To deal with hyper-planes in a 14-dimensional space, visualize a 3-D space and say "fourteen" to yourself very loudly. Everyone does it.

Consider that given our definitions of $\mathbf{v}_i$, $\mathbf{u}_i$, and σ_i, we can rewrite Equation 2 for an arbitrary $m \times n$ matrix M as:

$$M\begin{bmatrix}\\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n\\ \\ \end{bmatrix} = \begin{bmatrix}\\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_m\\ \\ \end{bmatrix}\begin{bmatrix}\sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n\end{bmatrix}$$

which immediately gives us Equation 5 again, but for $m \times n$ matrices:

$$\begin{aligned} M\mathbf{V} &= U\Sigma \\ M\mathbf{V}\mathbf{V}^* &= U\Sigma\mathbf{V}^* \\ M &= U\Sigma\mathbf{V}^* \end{aligned}$$

Here, U is unitary and its $= m \times U$ is unitary and $n \times n$; and Σ is diagonal and $m \times n$. $V^* = I$ since V is orthonormal. This is schematically drawn in Figure 7.

Figure 7: The canonical diagram of the SVD decomposition of a matrix M. The columns of U are the orthonormal left singular vectors, Σ is a diagonal matrix of singular values, and the rows of V^* are the orthonormal right singular vectors. The dashed areas are padding.

The diagonal elements of Σ are the singular values. Without loss of generality, we can assume the singular values are ordered—ordering them might require reordering the columns of U and V^*. And the geometric interpretation of them holds in higher dimensions. For example, if we performed the SVD on an $m \times n$ matrix and noticed that the bottom 5 singular values were smaller than some epsilon, you could visualize this matrix transformation as flattening a hyperellipse along those 5 dimensions. Or if you have an exponential decay in the value of the singular values, it suggests that all the "action" of the matrix happens along only a few dimensions.

And that's it! That's the SVD. You can read about the distinction between the full and truncated SVD, about handling low-rank matrices, and so forth in better resources than this. My goal was to arrive at Equation 1 through as simple reasoning as possible. At least in my opinion, going slowly through the geometrical intuition makes explanations of the SVD make more sense.

Re-visualizing the SVD

Now that we understand the SVD, we can convince ourselves that how we have been visualizing it is correct. Consider Figure 8, which was generated using MatplotLib and NumPy's implementation of SVD, `np.linalg.svd`. We can see that these figures exactly match our intuition for what is happening for SVD in 2-dimensions.

Figure 8: Several examples of using NumPy's implementation of SVD to visualize the algorithm. Each frame has two pairs of overlaid rectangles. For each pair, a square is transformed into a parallelogram by a linear transformation. In the left pair, the square is unrotated. In the right pair, the square is rotated by U^* before applying the transformation. Each dashed line is the top singular vector for the associated transformation.

Applications

The SVD is related to many other matrix properties. The number of nonzero singular values is equal to the rank of the matrix. This is why the matrix rank is a [convex surrogate for low-rank approximations](#). The range of a matrix is the space spanned by the vectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ where r is the number of nonzero singular values and all the singular values and vectors are ordered. A matrix is uninvertible if it has a singular value that is zero because the transformation is "collapsing" an m-cube along at least one dimension. And so on. Once the SVD is understood, I think it's fairly straightforward to intuit why the aforementioned properties must hold. See Chapter 5 of (Trefethen & Bau III, 1997) for details.

One algorithm I find much easier to understand knowing the standard SVD is principal component analysis (PCA). Rather than assume you know how PCA works, let's look at what PCA computes and then let's reason about what PCA is doing with our knowledge of SVD. Consider a real-valued data matrix X that is $n \times p$ where n is the number of samples and p is the number of features. Its SVD is

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

Now consider the covariance matrix of X, namely X^*X, in the context of the SVD decomposition:

$$\begin{aligned} X^*X &= (U\Sigma V^*)^*(U\Sigma V^*) \\ &= V\Sigma^*U^*U\Sigma V^* \\ &= V\Sigma^*\Sigma V^* \\ &= V\Sigma^2V^* \end{aligned}$$

We already know that V^* and V are just unitary matrices, where U^* disappears because $U^*U = I$. And even without knowing about eigenvectors or eigenvalues, we can see what PCA is doing by understanding SVD: it is diagonalizing the covariance matrix of X. So PCA is finding the major axes along which our data varies.

If it's not clear what SVD or eigendecomposition on data means, Jeremy Kun has a good blog post about that.

Conclusion

The singular value decomposition or SVD is a powerful tool in linear algebra. Understanding what the decomposition represents geometrically is useful for having an intuition for other matrix properties and also helps us better understand algorithms that build on the SVD.

Acknowledgments

Writing this post required that I synthesize from a number of excellent resources. Most notably, for the derivation in the section formulating the SVD, I used this tutorial from the American Mathematical Society. I thank Mahmoud Abdelkhalek for bringing a typo in the derivation to my attention.

1. Trefethen, L. N., & Bau III, D. (1997). *Numerical linear algebra* (Vol. 50). Siam.