Math Tools Review

September 18th, 2018

Review outline

- Matrix * vec multiplication
- Diagonal matrices
- Orthogonal matrices
- SVD basics
- SVD example

Matrix * vec

- The more you practice, the easier your life in this class will be
- Check out online resources (like the link I posted on piazza)
- Seriously though, practice. Passively understanding it isn’t going to cut it in this class.

Matrix * vec

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \end{bmatrix} = v_x \begin{bmatrix} 1 \\ -2 \end{bmatrix} + v_y \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \end{bmatrix} = v_x \begin{bmatrix} 1 \\ -2 \end{bmatrix} + v_y \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \end{bmatrix} = v_x \begin{bmatrix} 1 \\ -2 \end{bmatrix} + v_y \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

Matrix * vec

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} (V_x)(1) + (V_y)(3) \\ \\ \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} (V_x)(1) + (V_y)(3) \\ (V_x)(-2) + (V_y)(0) \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} (V_x)(1) + (V_y)(3) \\ (V_x)(-2) + (V_y)(0) \end{bmatrix}$$

Matrix * vec

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} (v_x)(1) + (v_y)(3) \\ (v_x)(-2) + (v_y)(0) \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} (v_x)(1) + (v_y)(3) \\ (v_x)(-2) + (v_y)(0) \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} v_x \begin{bmatrix} 1 \\ -2 \end{bmatrix} & \begin{matrix} + (v_y)(3) \\ + (v_y)(0) \end{matrix} \end{bmatrix}$$

Matrix * vec

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} (V_x)(1) + (V_y)(3) \\ (V_x)(-2) + (V_y)(0) \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} (V_x)(1) + (V_y)(3) \\ (V_x)(-2) + (V_y)(0) \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} V_x \begin{bmatrix} 1 \\ -2 \end{bmatrix} \begin{matrix} + (V_y)(3) \\ + (V_y)(0) \end{matrix} \end{bmatrix}$$

Matrix * vec

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} (v_x)(1) + (v_y)(3) \\ (v_x)(-2) + (v_y)(0) \end{bmatrix} \qquad \begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} (v_x)(1) + (v_y)(3) \\ (v_x)(-2) + (v_y)(0) \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} (v_x)(1) + (v_y)(3) \\ (v_x)(-2) + (v_y)(0) \end{bmatrix} \qquad \begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} (v_x)(1) + (v_y)(3) \\ (v_x)(-2) + (v_y)(0) \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} v_x \begin{bmatrix} 1 \\ -2 \end{bmatrix} & + (v_y)(3) \\ & + (v_y)(0) \end{bmatrix} \qquad \begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} v_x \begin{bmatrix} 1 \\ -2 \end{bmatrix} + v_y \begin{bmatrix} 3 \\ 0 \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 \\ -2 & 0 \end{bmatrix} * \begin{bmatrix} v_x \\ v_y \end{bmatrix} = v_x \begin{bmatrix} 1 \\ -2 \end{bmatrix} + v_y \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

Diagonal matrices

- All off-diagonal entries are zero

$$\begin{bmatrix} 2 & 0 \\ 0 & 1/2 \end{bmatrix} * \begin{bmatrix} V_x \\ V_y \end{bmatrix} = V_x \begin{bmatrix} 2 \\ 0 \end{bmatrix} + V_y \begin{bmatrix} 0 \\ 1/2 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 0 \\ 0 & 1/2 \end{bmatrix} * \begin{bmatrix} V_x \\ V_y \end{bmatrix} = V_x \begin{bmatrix} 2 \\ 0 \end{bmatrix} + V_y \begin{bmatrix} 0 \\ 1/2 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 0 \\ 0 & 1/2 \end{bmatrix} * \begin{bmatrix} V_x \\ V_y \end{bmatrix} = V_x \begin{bmatrix} 2 \\ 0 \end{bmatrix} + V_y \begin{bmatrix} 0 \\ 1/2 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 0 \\ 0 & 1/2 \end{bmatrix} * \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} 2 \; V_x \\ 1/2 \; V_y \end{bmatrix}$$

Orthogonal matrices

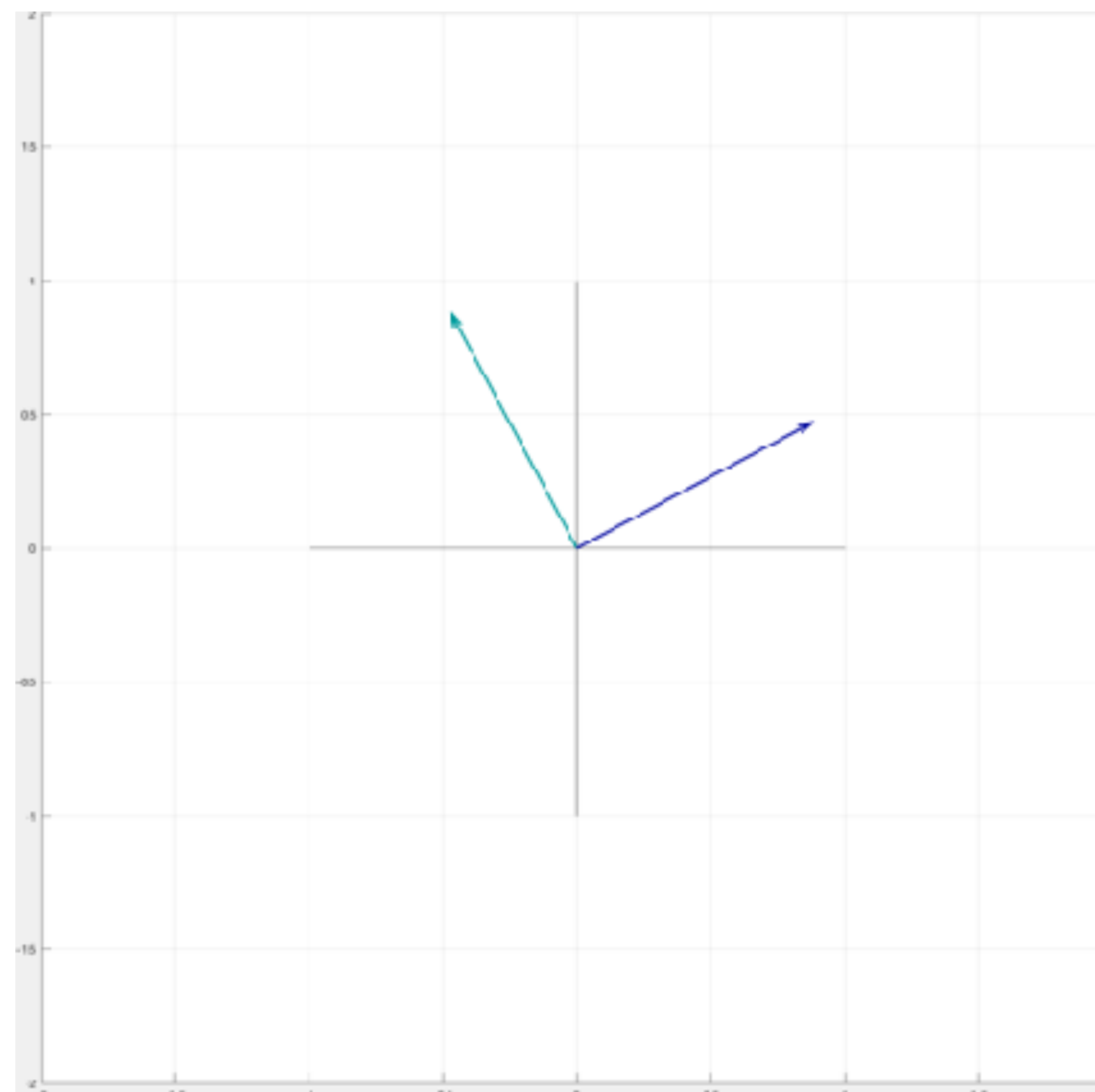
- Columns are orthogonal unit vectors
- Perform rotations

$$\begin{bmatrix} -0.47 & 0.88 \\ 0.88 & 0.47 \end{bmatrix}$$

Orthogonal matrices

- Visualizing the orthogonal vectors:

$$\begin{bmatrix} -0.47 & 0.88 \\ 0.88 & 0.47 \end{bmatrix}$$

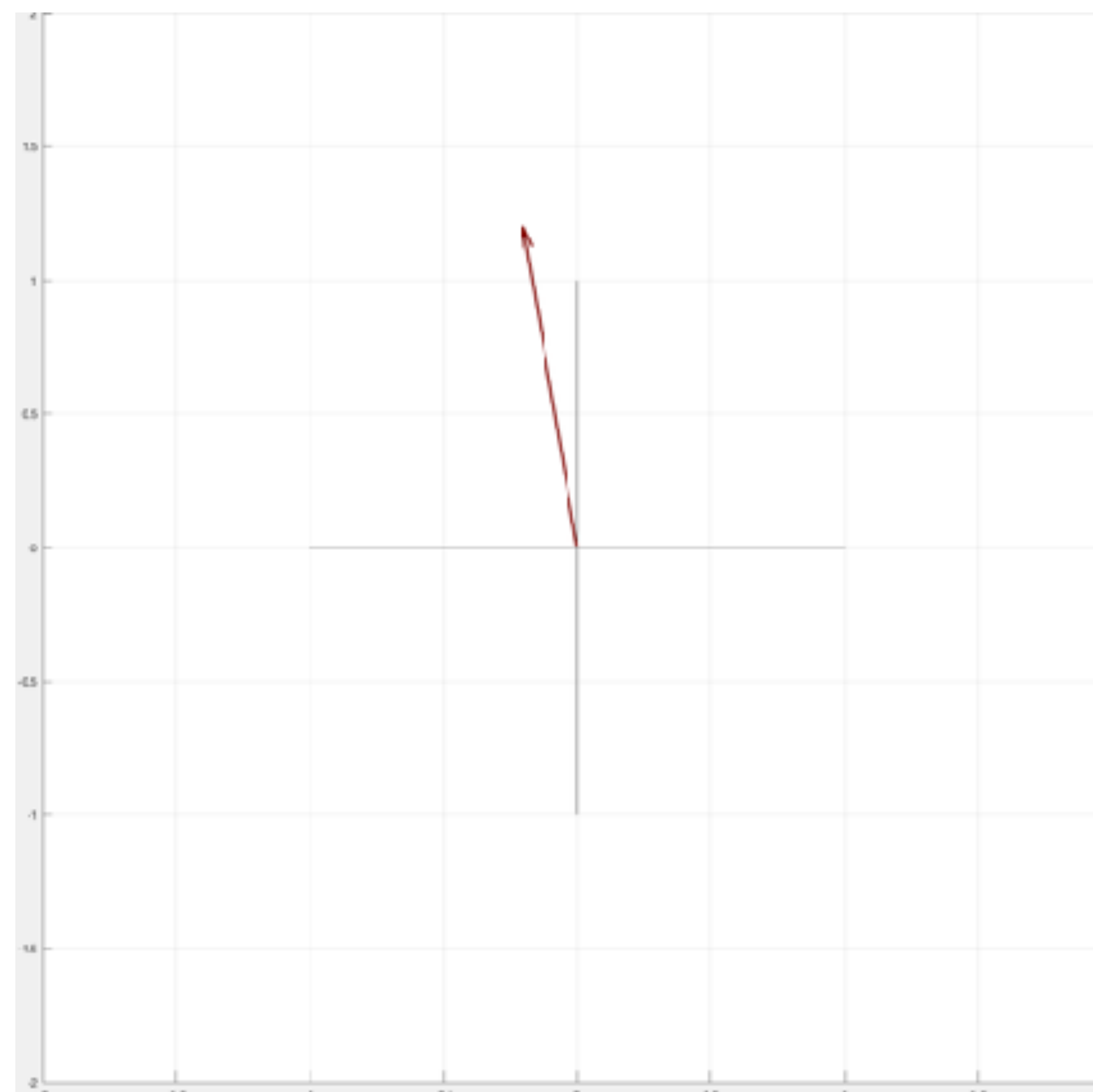


Orthogonal matrices

- Putting a vector "through" the matrix:

$$\begin{bmatrix} -0.47 & 0.88 \\ 0.88 & 0.47 \end{bmatrix}$$

$$\begin{bmatrix} -0.47 & 0.88 \\ 0.88 & 0.47 \end{bmatrix} * \begin{bmatrix} -0.2 \\ 1.2 \end{bmatrix} = -0.2 \begin{bmatrix} -0.47 \\ 0.88 \end{bmatrix} + 1.2 \begin{bmatrix} 0.88 \\ 0.47 \end{bmatrix}$$



Orthogonal matrices

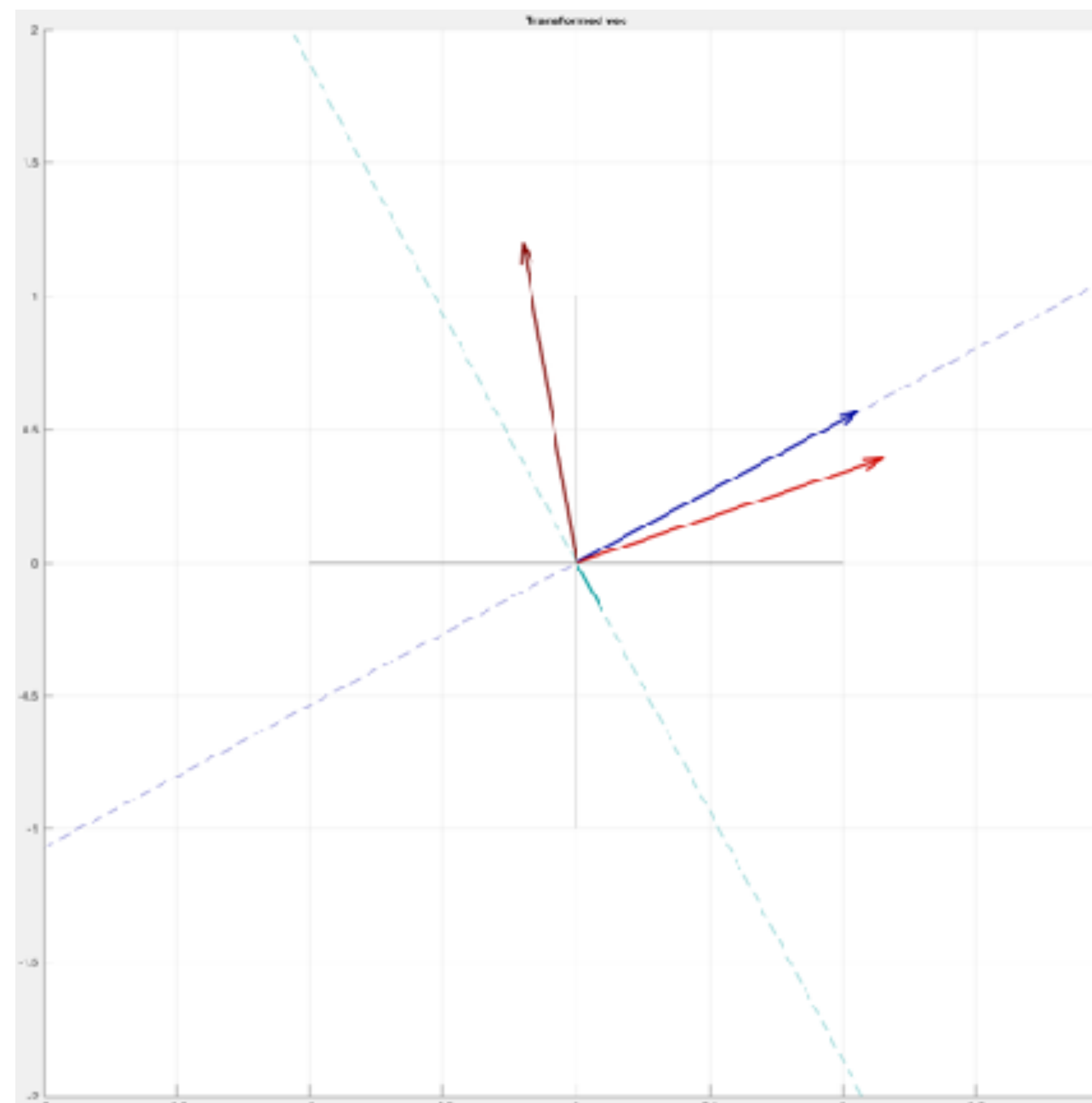
- Putting a vector “through” the matrix:

$$\begin{bmatrix} -0.47 & 0.88 \\ 0.88 & 0.47 \end{bmatrix}$$

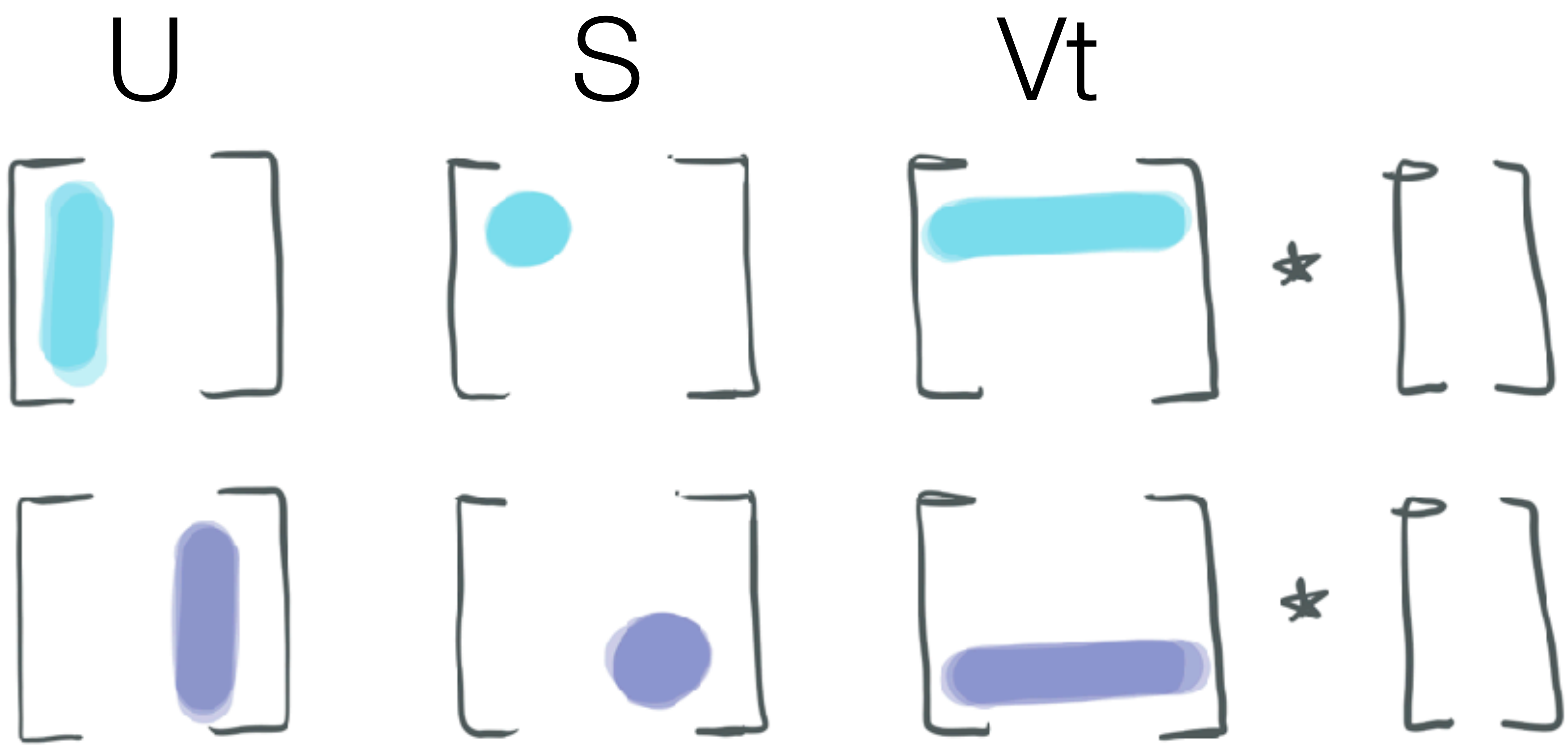
$$\begin{bmatrix} -0.47 & 0.88 \\ 0.88 & 0.47 \end{bmatrix} * \begin{bmatrix} -0.2 \\ 1.2 \end{bmatrix} = -0.2 \begin{bmatrix} -0.47 \\ 0.88 \end{bmatrix} + 1.2 \begin{bmatrix} 0.88 \\ 0.47 \end{bmatrix}$$

$$\begin{bmatrix} -0.47 & 0.88 \\ 0.88 & 0.47 \end{bmatrix} * \begin{bmatrix} -0.2 \\ 1.2 \end{bmatrix} = -0.2 \begin{bmatrix} -0.47 \\ 0.88 \end{bmatrix} + 1.2 \begin{bmatrix} 0.88 \\ 0.47 \end{bmatrix}$$

$$\begin{bmatrix} -0.47 & 0.88 \\ 0.88 & 0.47 \end{bmatrix} * \begin{bmatrix} -0.2 \\ 1.2 \end{bmatrix} = -0.2 \begin{bmatrix} -0.47 \\ 0.88 \end{bmatrix} + 1.2 \begin{bmatrix} 0.88 \\ 0.47 \end{bmatrix}$$

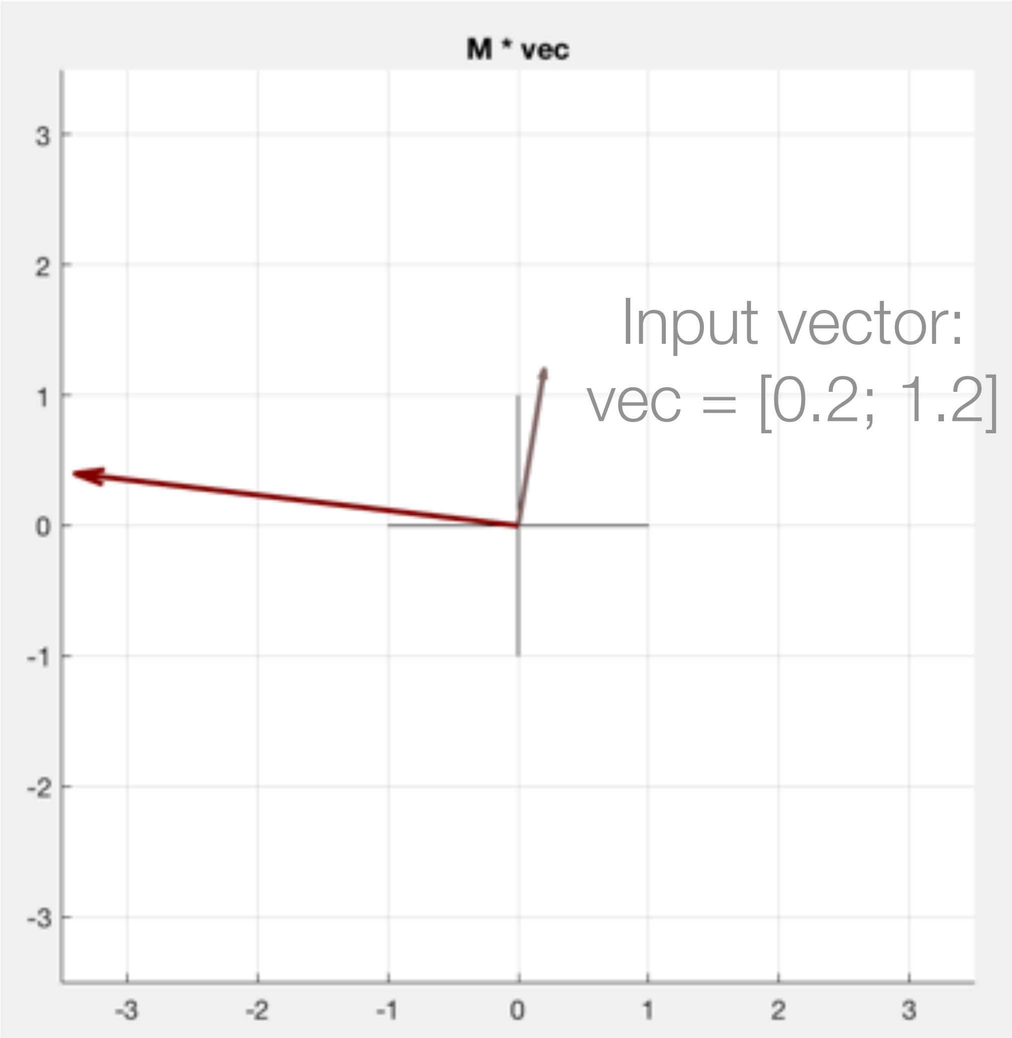


SVD Basics



- We can think about Vt decomposing the input vector into a v1 component and a v2 component.
- s1 stretches the input's v1 component and s2 stretches the v2 component.
- u1 rotates the v1 component to its "final destination" and u2 does the same to the v2 component

Example: M * vec



Input vector:
vec = [0.2; 1.2]

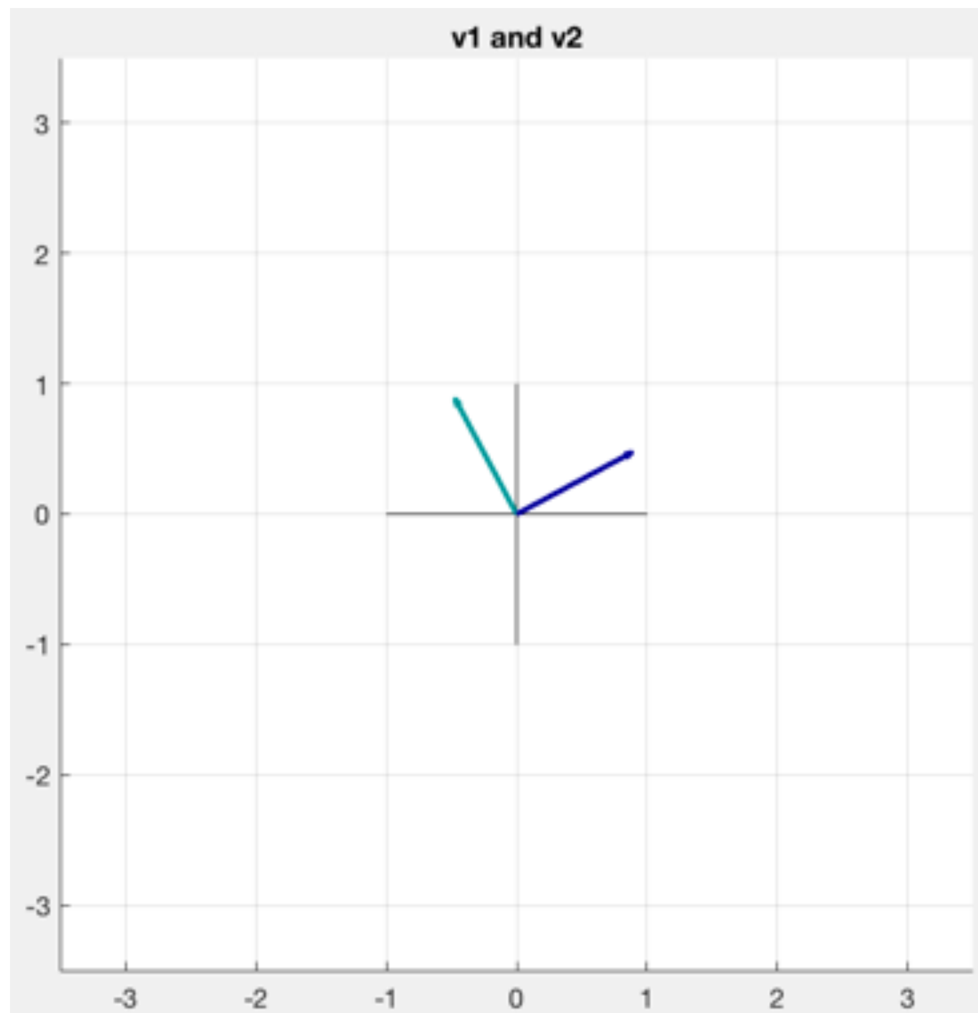
Final vector:
M*vec = [-3.4 0.4]

As we know, matrices can stretch and rotate.
M is a matrix that does both. Check out how vec got stretched and rotated

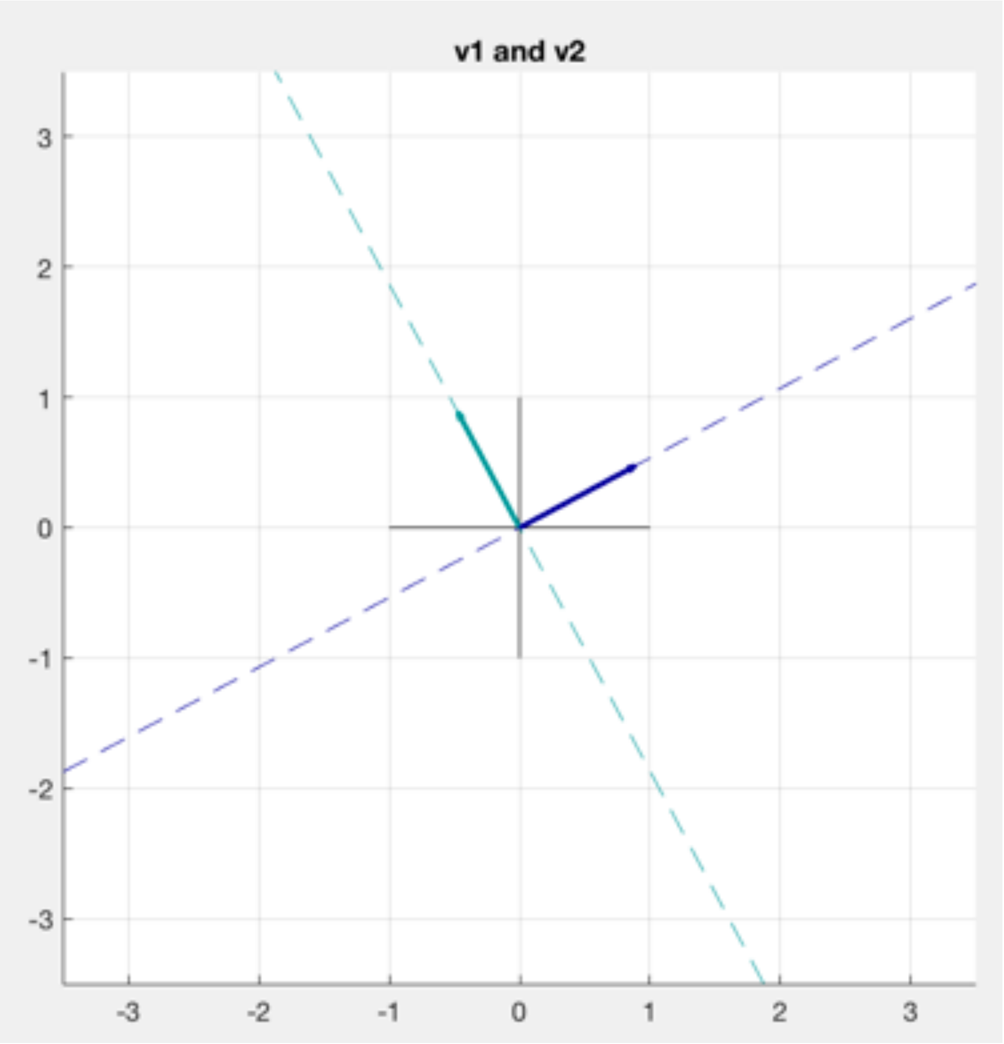
Matrix M:
M = [1 -3; 2 0]

svd(M) — the V

Matrix V has 2 orthogonal unit vectors.
Let's plot them:



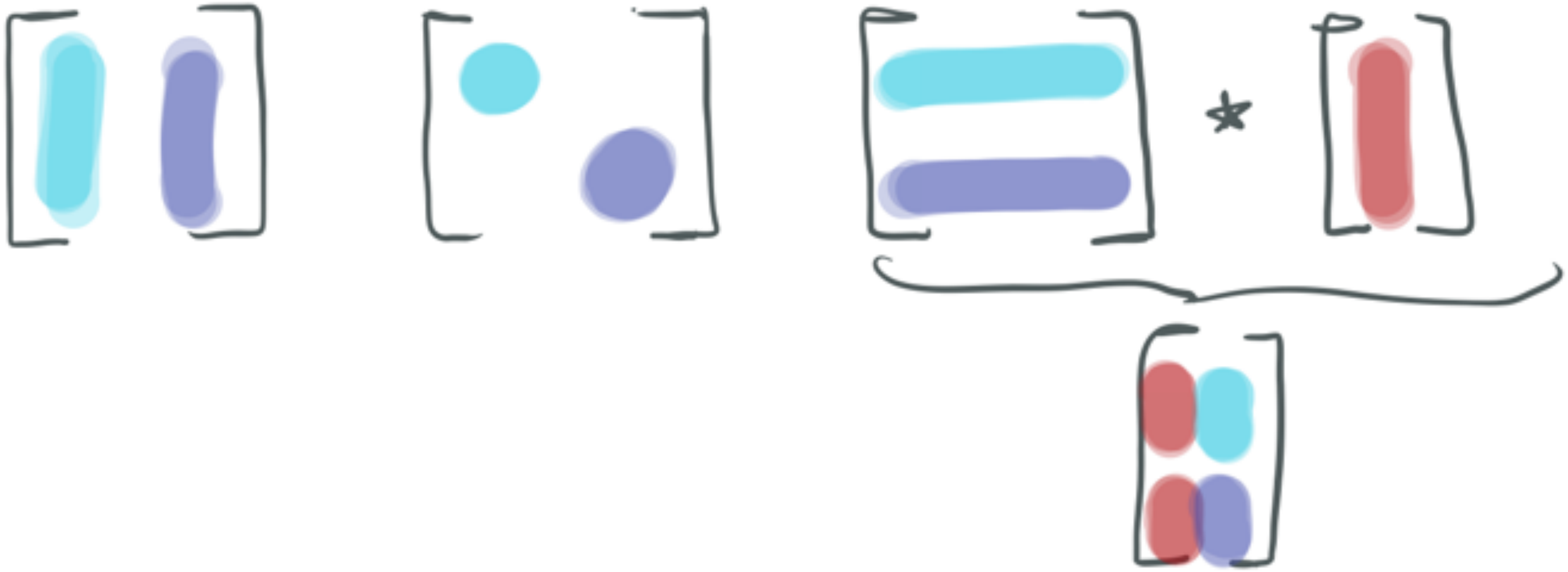
svd(M) — the V

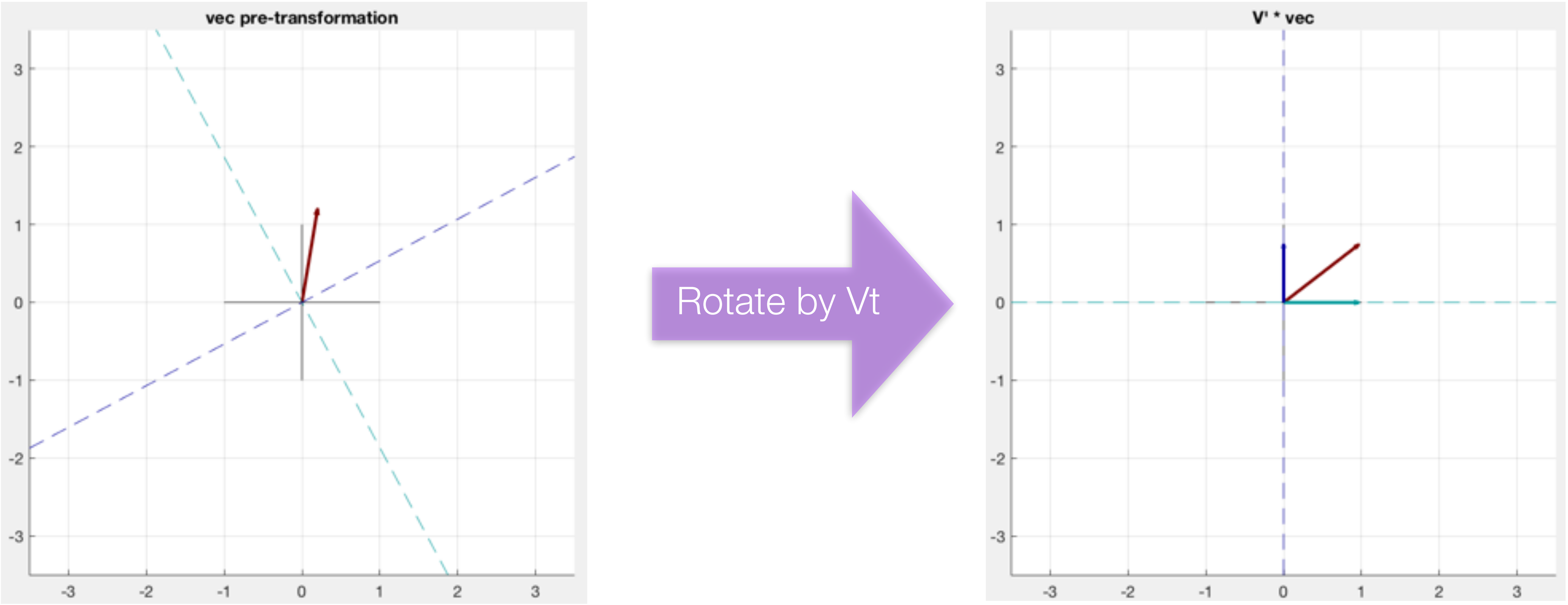


Why these vectors? What's special about them?

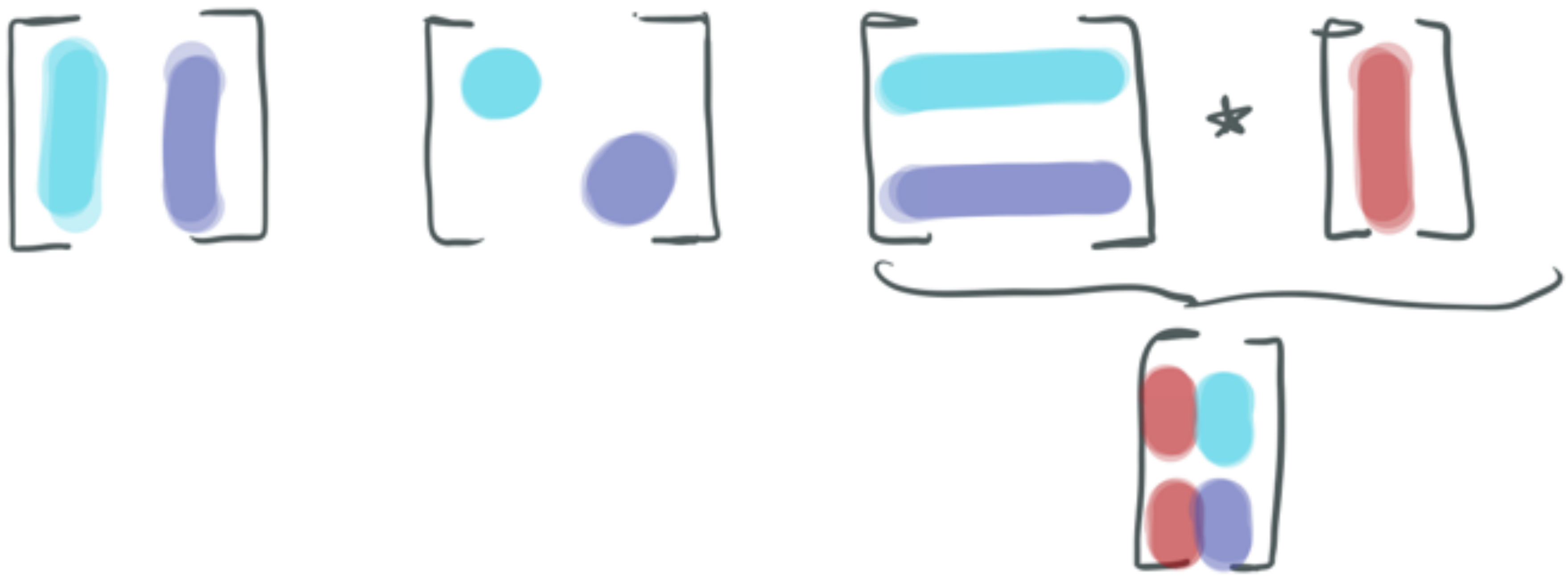
- Of all possible input vectors, the ones lying on the span of v1 (dashed cyan line on plot) get "stretched" the most
- Conversely, the input vectors lying on the span of v2 get "stretched" the least (in 2d).

Vt * vec — ROTATE!



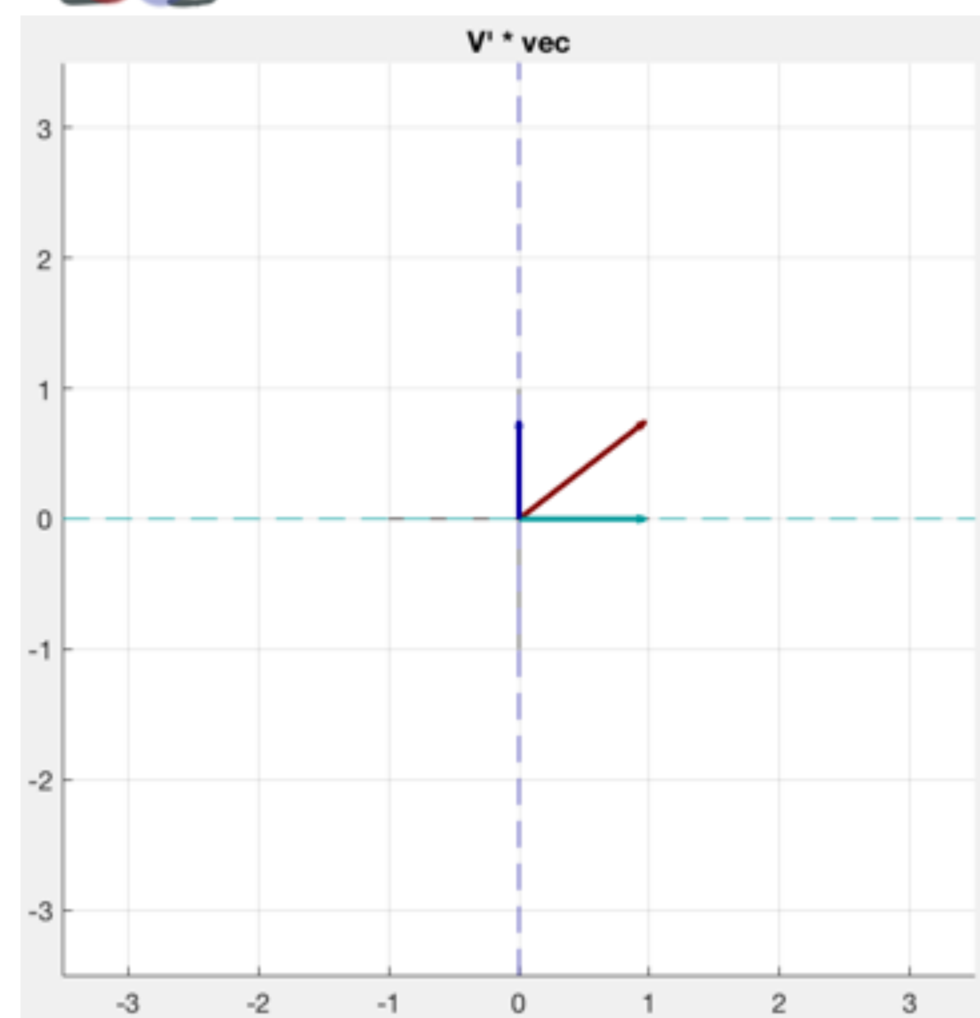


Preparing for S

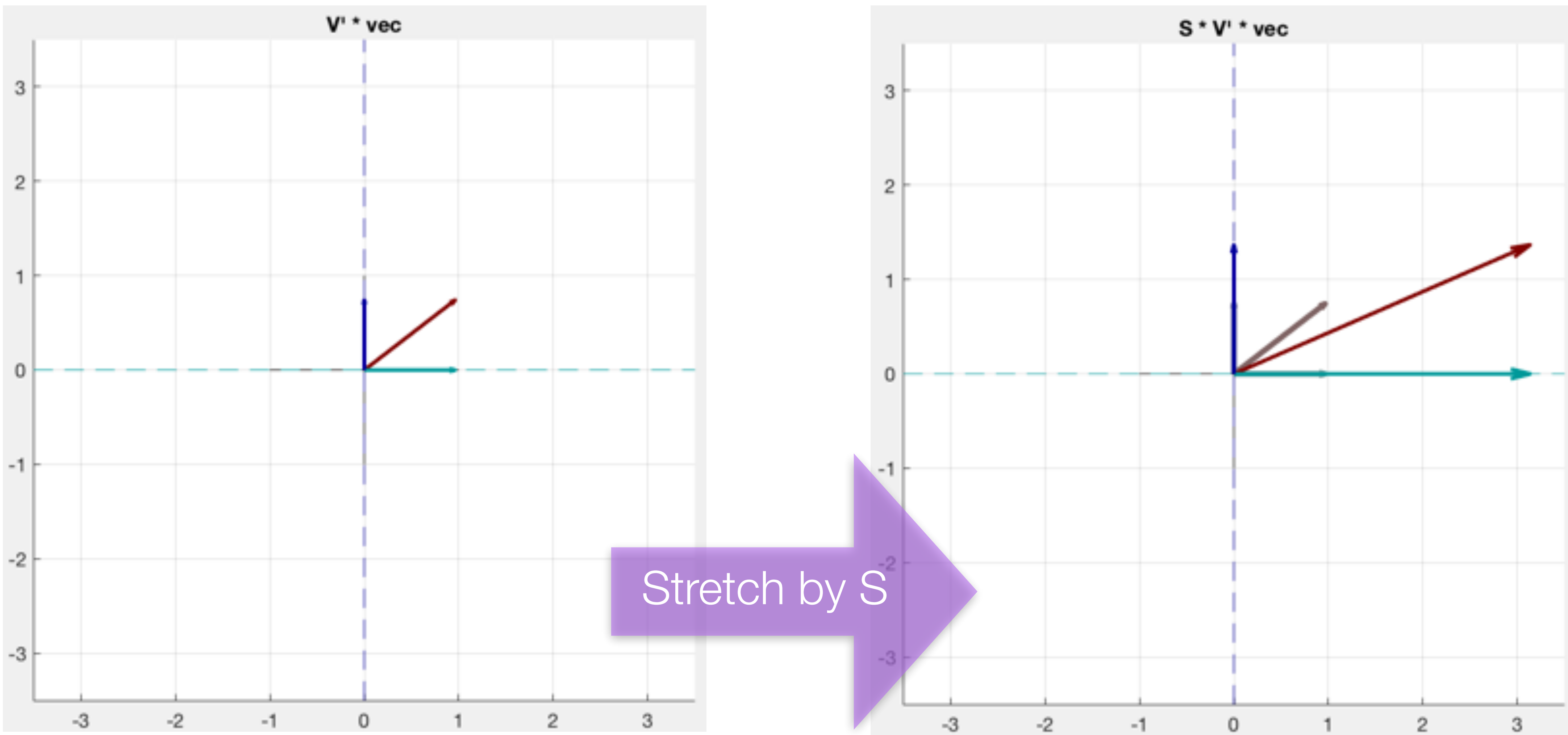


Notice how we can decompose the rotated vector into a v1 component and a v2 component.

- vec's v1 component gets stretched by s1
- Vec's v2 component gets stretched by s2

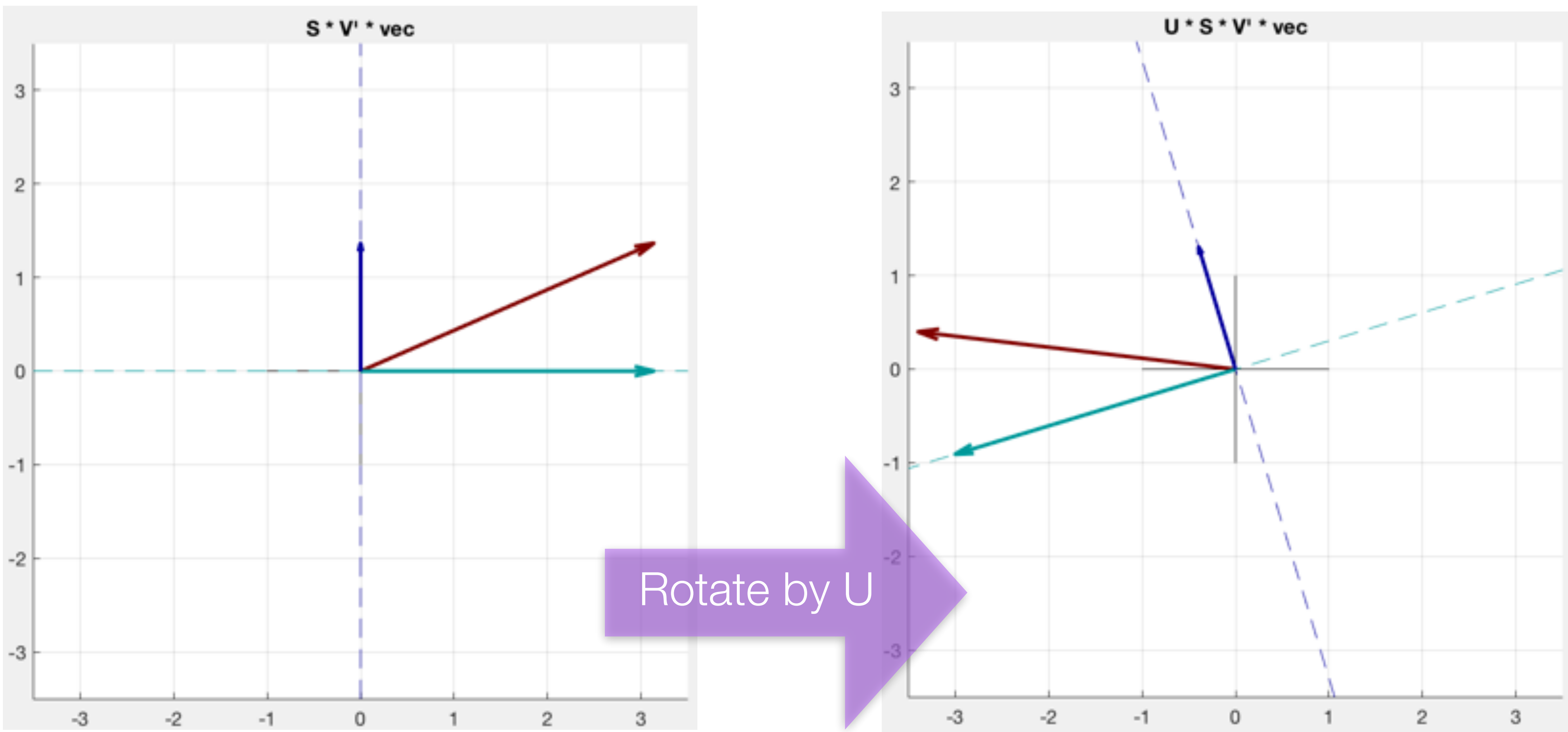


S * Vt * vec — STRETCH!



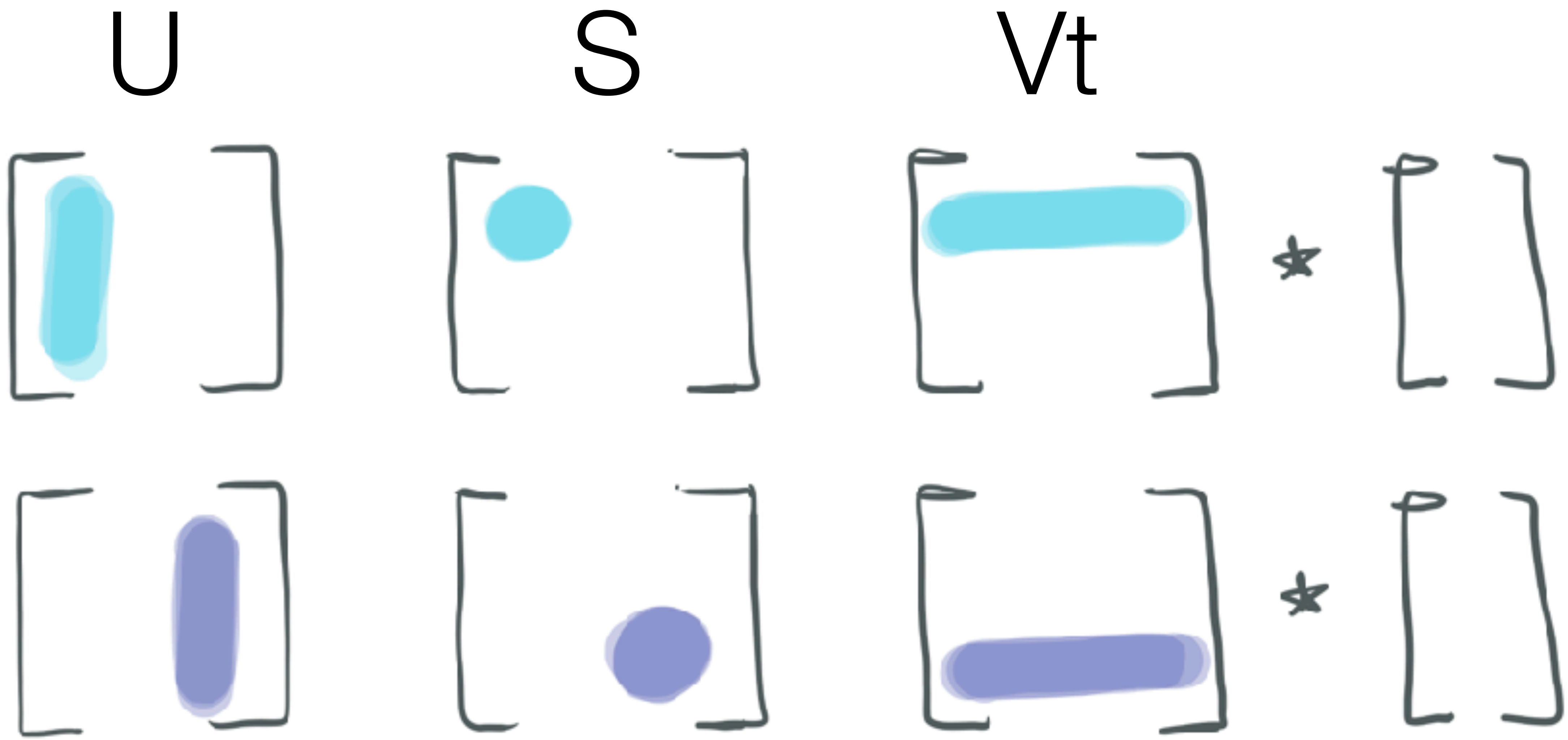
Vec’s v1 component gets stretched by S1
Vec’s v2 component gets stretched by S2

U * S * Vt * vec — ROTATE!



U rotates the vector to its final position. Notice how the v1 component ends up lying in the span of u1, where u1 = the first column of U.

U * S * Vt * vec



Breaking up the input vector into components allows us to easily visualize what happens to each component as the vector is successively rotated, stretched, and rotated by Vt, S, and U

Now go practice/visualize some examples in matlab!

Convince yourself that Vt and U actually rotate, ask what happens if your input vector is v1, try this out in higher dimensions, etc.