8

Singular Value Decomposition

What is the Singular Value Decomposition?

The SVD is a factorization of an m×n matrix into: $A = U\Sigma V^T$

- An m×m orthogonal matrix U (columns: "left singular vectors")
- An m×n diagonal matrix Σ ("Sigma") with the singular values on the diagonal

$$\Sigma = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \sigma_n \\ 0 & & \end{pmatrix}$$

Convention: $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$

- An n×n orthogonal matrix V^T (columns: "right singular vectors")

How do I compute it?

(1) Compute the eigenvalues and eigenvectors of A^TA.

(for example using orthogonal iteration)

$$A^TA v_1 = \lambda_1 v_1 \quad \cdots \quad A^TA v_n = \lambda_n v_n$$

(2) Make a matrix V from the vectors v_i $\qquad V = \begin{pmatrix} | & & | \\ v_1 & \cdots & v_n \\ | & & | \end{pmatrix}$

(3) Make a diagonal matrix Σ from the square roots of the eigenvalues:

$$\Sigma = \begin{pmatrix} \sqrt{\lambda_1} & & & 0 \\ & \ddots & & \\ 0 & & \sqrt{\lambda_n} & \end{pmatrix}$$

(4) Find U from $A = U\Sigma V^T \iff U\Sigma = AV$

(While being careful about non-squareness and zero sing.values)

Observe U is orthogonal.

Demo: Computing the SVD

What's another way of writing the SVD?

Starting from $A = U\Sigma V^T = \begin{pmatrix} | & & | \\ u_1 & \cdots & u_m \\ | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \sigma_n \\ 0 & & \end{pmatrix} \begin{pmatrix} - & v_1 & - \\ & \vdots & \\ - & v_n & - \end{pmatrix}$

we find that: (assuming m>n for simplicity)

$$A = \begin{pmatrix} | & & | \\ \sigma_1 u_1 & \cdots & \sigma_n u_n \\ | & & | \end{pmatrix} \begin{pmatrix} - & v_1 & - \\ & \vdots & \\ - & v_n & - \end{pmatrix}$$

$$= \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_n u_n v_n^T$$

That means:

The SVD writes the matrix A as a sum of outer products

(of left/right singular vectors).

What do the singular values mean? (in particular the first/largest one)

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

U orth.

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 = \max_{\|x\|_2=1} \|U\Sigma V^T x\|_2 = \max_{\|x\|_2=1} \|\Sigma V^T x\|_2$$

$$= \max_{\|V^T x\|_2=1} \|\Sigma (V^T x)\|_2 \qquad = \max_{\|y\|_2=1} \|\Sigma y\|_2 \qquad = \sigma_1 .$$

V orthogonal, so $\|V^T x\|_2 = \|x\|_2$ Let $y = V^T x$. Σ diag.

So the SVD (finally) provides a way to find the 2-norm.

Entertainingly, it does so by reducing the problem to finding the 2-norm

of a diagonal matrix.

How expensive is it to compute the SVD?

Demo: Relative cost of matrix factorizations

So why bother with the SVD if it is so expensive? I.e. what makes the SVD special?

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_n u_n v_n^*$$

Assume $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$.

Next, define $A_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_k u_k v_k^*$. $(k \leq n)$

Observe that A_k has rank k. (And A has rank n.)

Then $\|A - B\|_2$ among all rank-k (or lower) matrices B is minimized by A_k.

("Eckart-Young theorem")

Even better:

$$\min_{\operatorname{rank} B \leq k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}.$$

A_k is called the best rank-k approximation to A.

(where k can be any number)

This best-approximation property is what makes the SVD extremely

useful in applications and ultimately justifies its high cost.

What is the Frobenius norm of a matrix?

It's the same as gluing all the rows (or columns) together into one gigantic vector and then taking the 2-(vector-)norm of that.

$$\|A\|_F := \sqrt{a_{11}^2 + a_{12}^2 + \dots + a_{1n}^2 + a_{21}^2 + \dots + a_{mn}^2}$$

⚠ Although this is called a "norm" and works on matrices, it's not really a "matrix norm" in our definition. There is no vector norm whose associated matrix norm is the Frobenius norm.

How about rank-k best-approximation in the Frobenius norm?

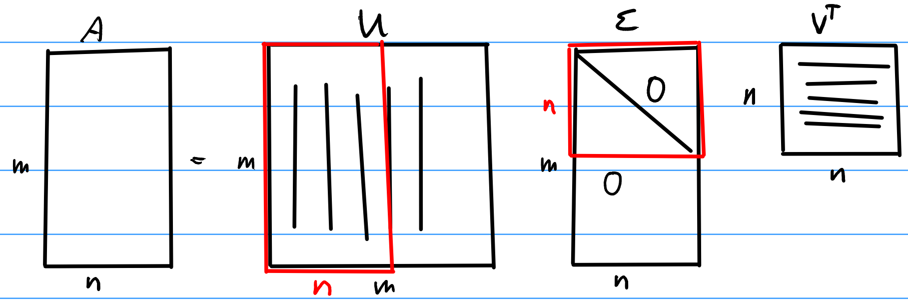
(Let A and A_k be defined as before.)

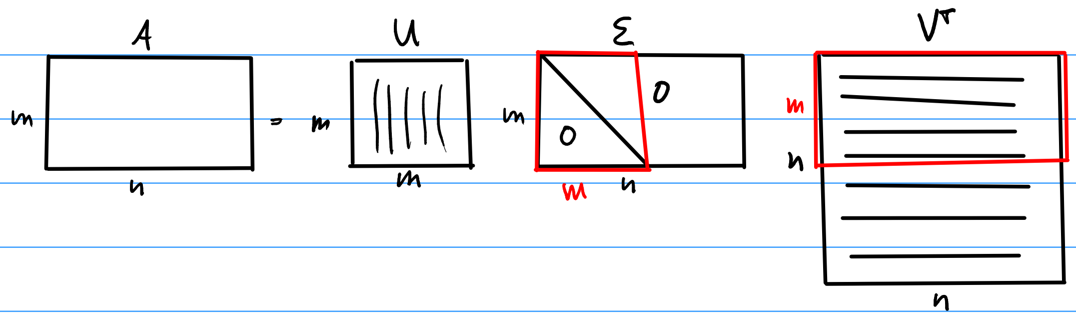
$$\min_{\operatorname{rank} B \le k} \|A - B\|_F = \|A - A_k\|_F = \sqrt{\sigma_{k+1}^2 + \dots + \sigma_n^2}.$$

I.e. A_k also minimizes the Frobenius norm among all rank-k (or lower) matrices.

Is there a "reduced" transform for non-square matrices?

Yes:





"Full" version shown in black

"Reduced" version shown in red