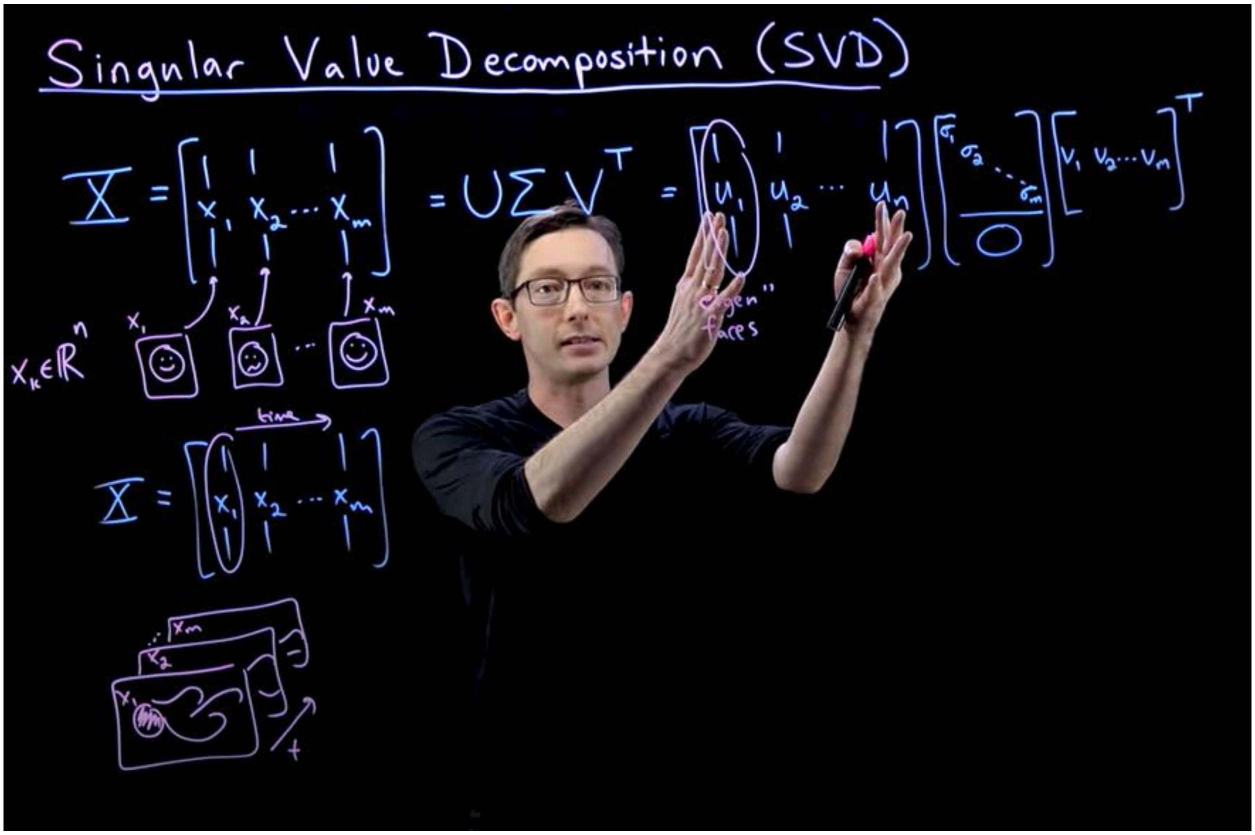
SVD with Steven Brunton, University of Washington



The matrix X of *n*m* dimentions is decomposed into the product of three matrices

$X = U \sum V^T$

U are the *left singular vectors*
$\sum$ are the *singular values*
V^T are the *right singular values*

U and V^T are unitary, orthogonal matrices

$\sum$ is a diagonal matrix, because there are *m* columns in X, there will be *m* singular values (nonzero)

U: the cols of U have the same shape as the cols of X, so if X has 1M-by-1 vectors, so does U

The cols of U are, however, hierarchically arranged in their ability to describe the variance of the cols of X, they are "eigen" cols of X, whatever the cols of X represent

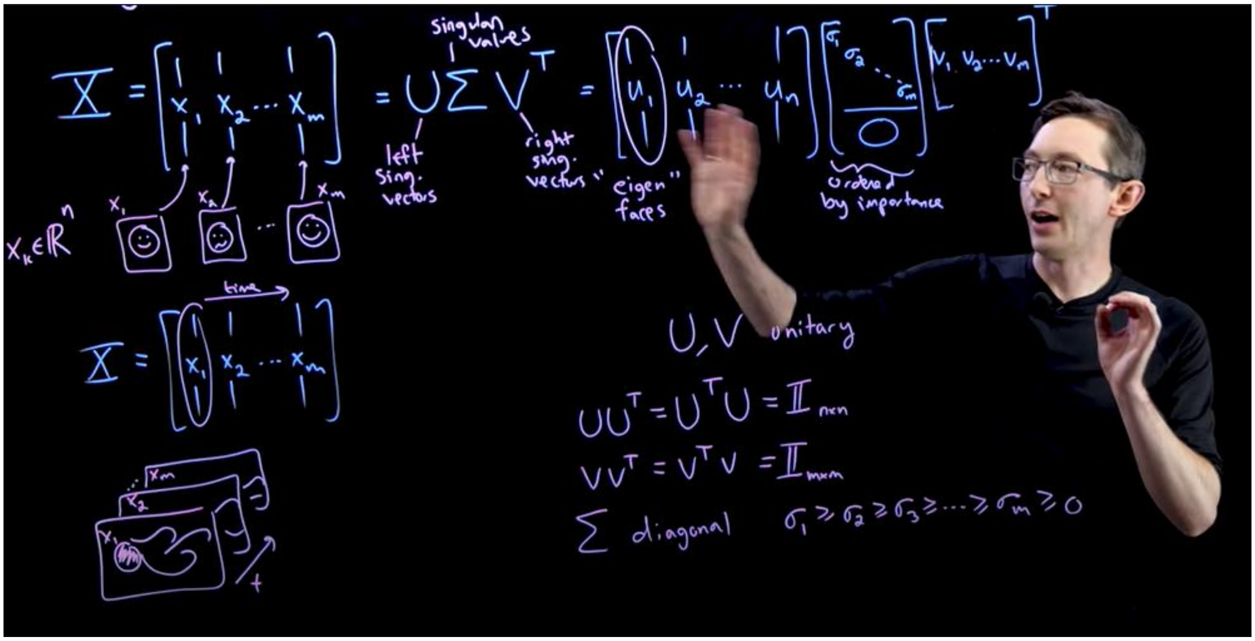
Ex. faces, flow fields, tokens → eigen faces, eigen flow fields, eigen tokens

The cols of U are orthonormal, they are orthogonal and have unit length

$UU^T = U^T U = I_{n*n}$ AND $VV^T = V^T V = I_{m*m}$

U provides the basis for the n-dimentional vector space where the cols of X live

$\sum$ is nonnegative and in decreasing order of magnitude



V^T specifies the mixture of U cols needed to add up to a given col in X; V1 makes X1, and so forth.

Because the X matrix only has *m* cols, there are only *m* linearly independent cols in this *m*-dimensional vector space that can be spanned by these, so only the first *m* cols of U are important in representing this data.

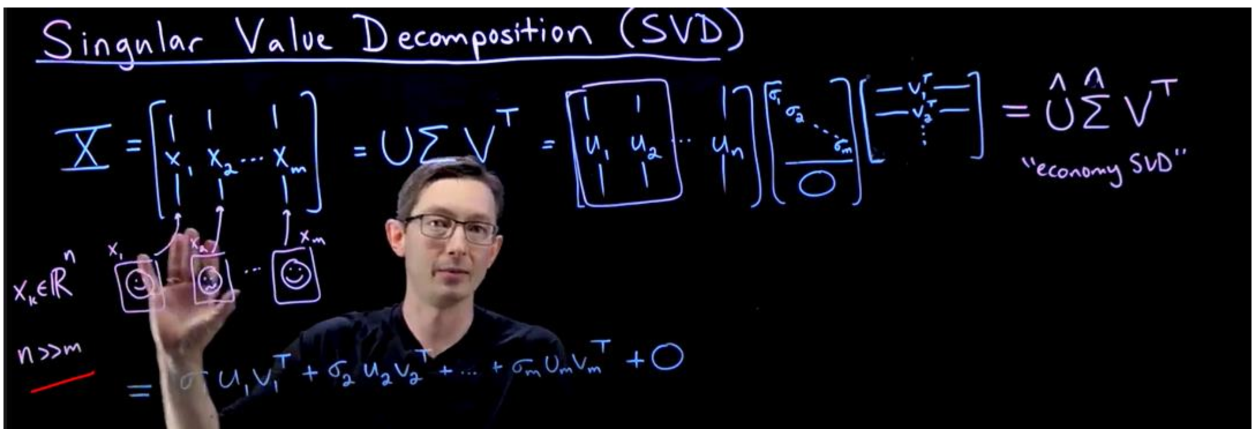
Representing expansion as a sum of rank 1 matrices

- Since $\sum$ is diagonal, the first product U $\sum$ means $U_1 * \sum_1$, $U_2 * \sum_2$, etc.
- Similarly, the product U V^T means $U_1 * V^T_1$, $U_2 * V^T_2$, etc.


4


Even though the expansion of U $\sum$ V^T can be fully fleshed out to the *n* sum of products $\sum_n U_n V^T_n$ terms, since there are only *m* nonnegative singular values, only *m* terms are necessary, the rest amount to 0.

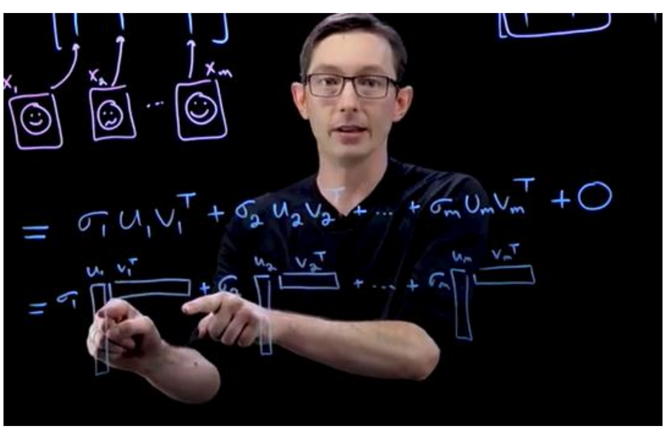
This is the cornerstone of SVD, since *n* can be massive (remember, U is an *n*n* matrix) and generally *m* is not – in fact Brunton is assuming this the entire time, that *n* >> *m* (say, *n*=1M, *m*=1k):



X is exactly this "economy SVD" which is the first *m* cols of U (or Û) * *m* singular vectors (Sigma-hat) * V^T

Each term in this sum of products is a rank 1 matrix

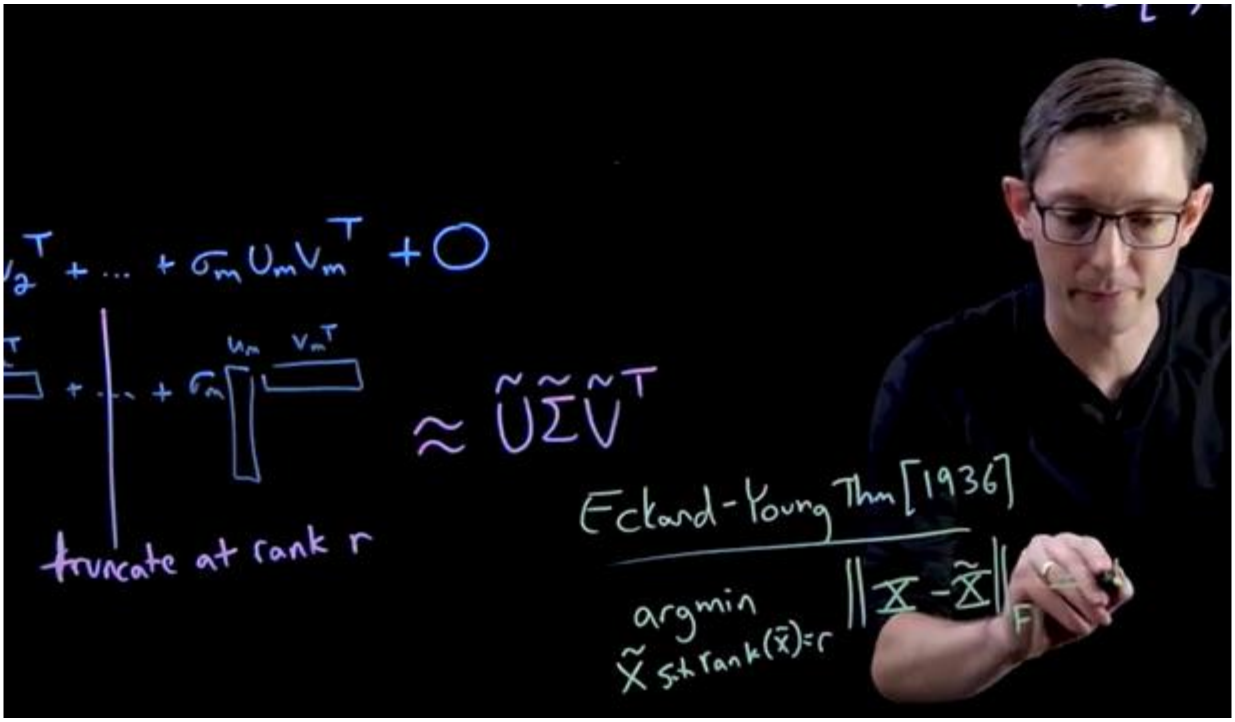
The $U_1 * V_1^T$ outer product is performed by multiplying each element in U_1 by each row in V^T to create



the rank 1 matrix – so called because it has exactly one linearly independent column and linearly independent row.

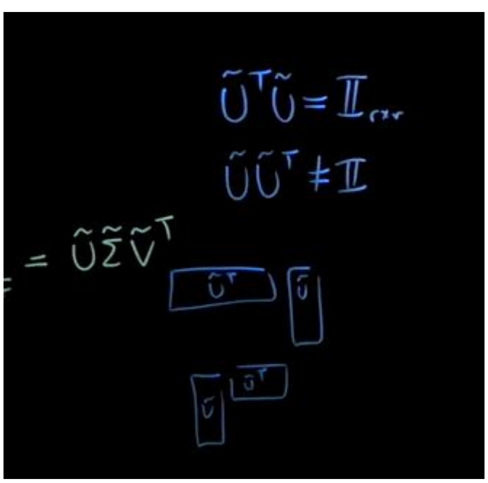
In sum, SVD decomposes X into orthognal bases U and V^T by a sum of rank 1 matrices that increasingly approximate X.

Truncating at rank r means examning the decrease of singular values in ∑ so as to throw away the low-signal singular values, creating an U-tilde Sigma-tilde V-transpose-tilde "best approximation of X" that is encapsulated in the Eckart-Young-Mirsky theorem.



Note how truncated SVD means the identity matrix cannot be achieved by U-tilde*U-tilde transpose.

The main point is that you can approximate a tall-skinny data matrix X by a lower-rank approximate matrix X-tilde given by a truncated SVD.



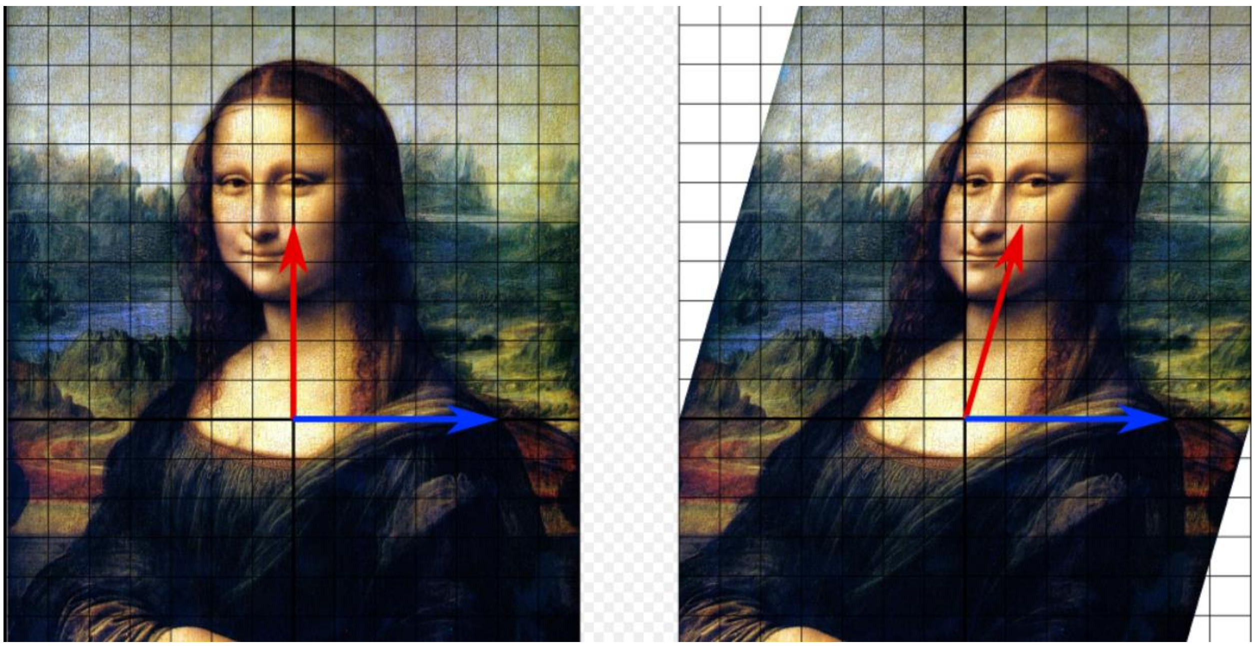
Computing the expanded matrices and interpreting them as encoding correlation structures in X

One of the most important interpretations of SVD take into account the correlations between the cols of X and correlations between the rows of X.

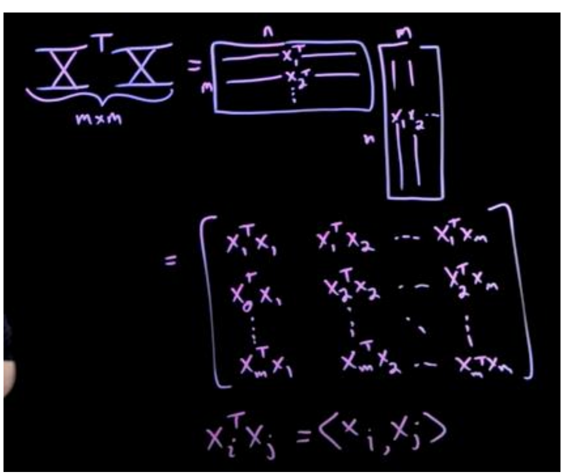
One can think of the U and V^T matrices in the SVD as *eigenvectors* of a correlation matrix given by XX^T or X^TX.

An *eigenvector* is a vector in a linear transformation *T* that doesn't change direction when *T* is applied, it only scales up (when its *eigenvalue,* a scalaer deonted by λ is > 1), down (when $0 > \lambda > 1$), remains the same (when $\lambda = 1$), or changes direction (when $\lambda < 0$).

Wikipedia's example is a *shear mapping* in which the blue arrow is the eigenvector with $\lambda = 1$.



Keeping in mind that X is a tall, skinny matrix of n row and m cols, in the case of X^TX, the resulting correlation matrix is a much smaller matrix of size *m*m*. Every entry in this m-by-m correlation matrix is an inner product between two columns of the matrix X:

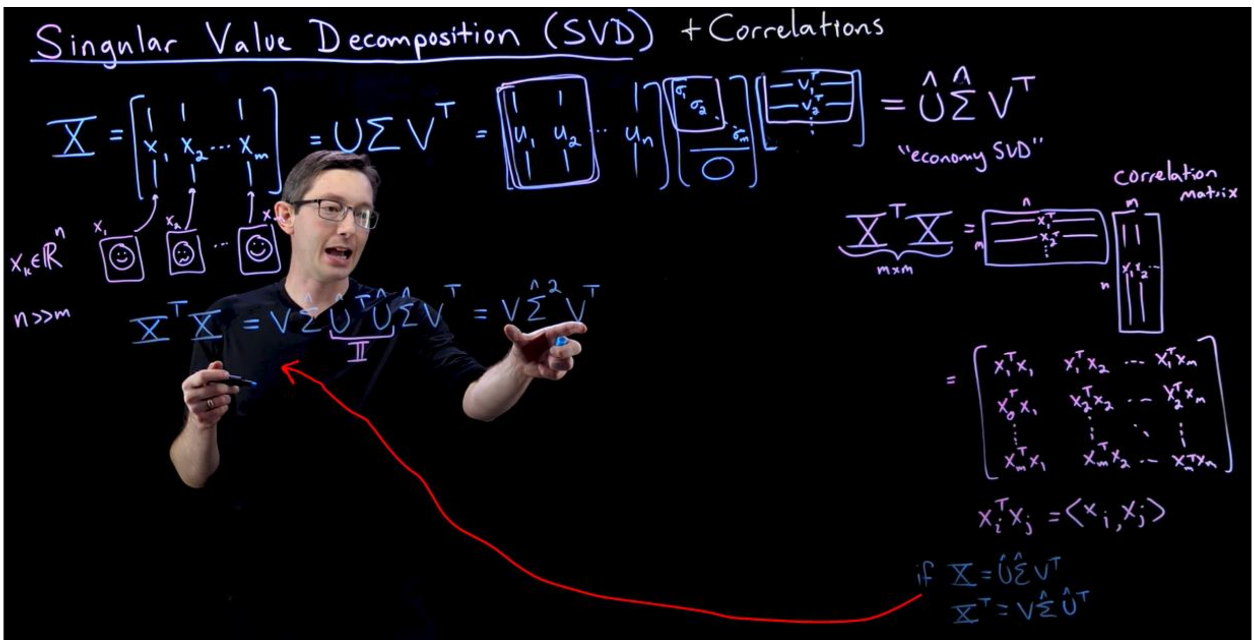


In the case of X being a matrix of “faces” (say, a million pixels elongated into a tall super skinny 1M-by-1 vector column of X), then the inner product of the ith vector and jth vector basically means that if the product of two people’s faces was large, their faces are similar, if the product has a small value, that means their faces are nearly orthogonal, they have very dissimilar faces.

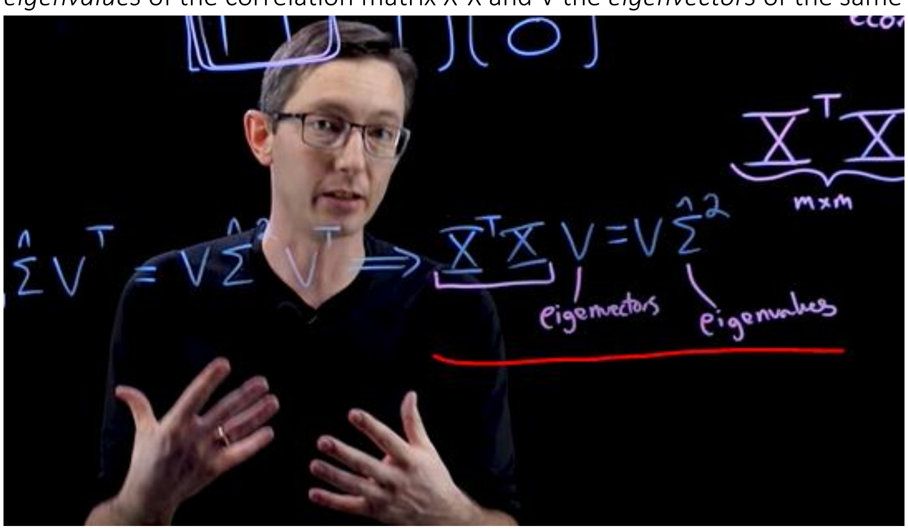
So this smaller m-by-m matrix contains all the information about the similarity-dissimilarity (correlations) between every col and every other col of X.

Computing X^TX in terms of the economy SVD (assuming it exists), one notices:

- X^T is just the reverse product of each SVD matrix’s transpose (see bottom-right below)
- The product starts with a cancellation as U^TU is the identity matrix
- The resulting expression $V\sum V^T$ is the *eigenvalue decomposition* of X^TX



When multiplying both sides of the equation by V, we get $X^TXV = V\sum^2$ where sigma-hat-squared are the *eigenvalues* of the correlation matrix X^TX and V the *eigenvectors* of the same X^TX.



So the real intuitive interpretation of SVD is that the right singular vectors in V are just *eigenvectors* of the columnwise correlation matrix, and $\sum$ are the square roots of the *eigenvalues* of that correlation matrix.

One can similarly compute the left singular values as *eigenvectos* of the rowwise correlation matrix XX^T which is a huge *n-by-n* matrix and arrive at the same *eigenvalues* – however, this would be too computationally expensive. In fact, computing the SVD via correlation matrices is not advisable, there are many other faster and more accurate methods (ex. based on the QR factorization); correlation matrices were only used because in this lecture because they are the most intuitive way to understand SVD.