Singular Value Decomposition (SVD)

CS 663

Ajit Rajwade

Singular value Decomposition

- For any *m* x *n* matrix **A**, the following decomposition **always** exists:

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \mathbf{A} \in R^{m \times n},$$
$$\mathbf{U}^\mathbf{T}\mathbf{U} = \mathbf{U}\mathbf{U}^\mathbf{T} = \mathbf{I}_\mathbf{m}, \mathbf{U} \in R^{m \times m},$$
$$\mathbf{V}^\mathbf{T}\mathbf{V} = \mathbf{V}\mathbf{V}^\mathbf{T} = \mathbf{I}_\mathbf{n}, \mathbf{V} \in R^{n \times n},$$
$$\mathbf{S} \in R^{m \times n}$$

Diagonal matrix with non-negative entries on the diagonal – called **singular values.**

Columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{A}\mathbf{A}^T$ (called the left singular vectors).

Columns of $\mathbf{V}$ are the eigenvectors of $\mathbf{A}^T\mathbf{A}$ (called the right singular vectors).

The non-zero singular values are the positive square roots of

non-zero eigenvalues of $\mathbf{A}\mathbf{A}^T$ or $\mathbf{A}^T\mathbf{A}$.

Singular value Decomposition

- For any m x n real matrix **A**, the SVD consists of matrices **U**,**S**,**V** which are always real – this is unlike eigenvectors and eigenvalues of **A** which may be complex even if **A** is real.

- The singular values are always non-negative, even though the eigenvalues may be negative.

- While writing the SVD, the following convention is assumed, and the left and right singular vectors are also arranged accordingly:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{m-1} \geq \sigma_m$$

Singular value Decomposition

- If only r < min(m,n) singular values are non-zero, the SVD can be represented in **reduced form** as follows:

$$A = USV^T, A \in R^{m \times n},$$

$$U \in R^{m \times r},$$

$$V \in R^{n \times r},$$

$$S \in R^{r \times r}$$

Singular value Decomposition

$$\mathbf{A} = \mathbf{USV}^T = \sum_{i=1}^{r} \mathbf{S}_{ii} \boxed{\mathbf{u}_i \mathbf{v}_i^t}$$

This *m* by *n* matrix $\mathbf{u_i}\,\mathbf{v^T}_\mathbf{i}$ is the product of a column vector $\mathbf{u_i}$ and the transpose of column vector $\mathbf{v_i}$. It has rank 1. Thus **A** is a weighted summation of *r* rank-1 matrices.

Note: $\mathbf{u_i}$ and $\mathbf{v_i}$ are the i-th column of matrix **U** and **V** respectively.

Singular value decomposition

$$\mathbf{A} = \mathbf{USV}^T$$

$$\mathbf{AA}^T = (\mathbf{USV}^T)(\mathbf{USV}^T)^T = \mathbf{USV}^T\mathbf{VSU}^T = \mathbf{US}^2\mathbf{U}^T$$

Thus, the left singular vectors of $\mathbf{A}$ (i.e. columns of $\mathbf{U}$) are the eigenvectors of $\mathbf{AA}^T$.

The singular values of $\mathbf{A}$ (i.e. diagonal elements of $\mathbf{S}$) are square-roots of the eigenvalues of $\mathbf{AA}^T$.

$$\mathbf{A}^T\mathbf{A} = (\mathbf{USV}^T)^T(\mathbf{USV}^T) = \mathbf{VSU}^T\mathbf{USV}^T = \mathbf{VS}^2\mathbf{V}^T$$

Thus, the right singular vectors of $\mathbf{A}$ (i.e. columns of $\mathbf{V}$) are the eigenvectors of $\mathbf{A}^T\mathbf{A}$.

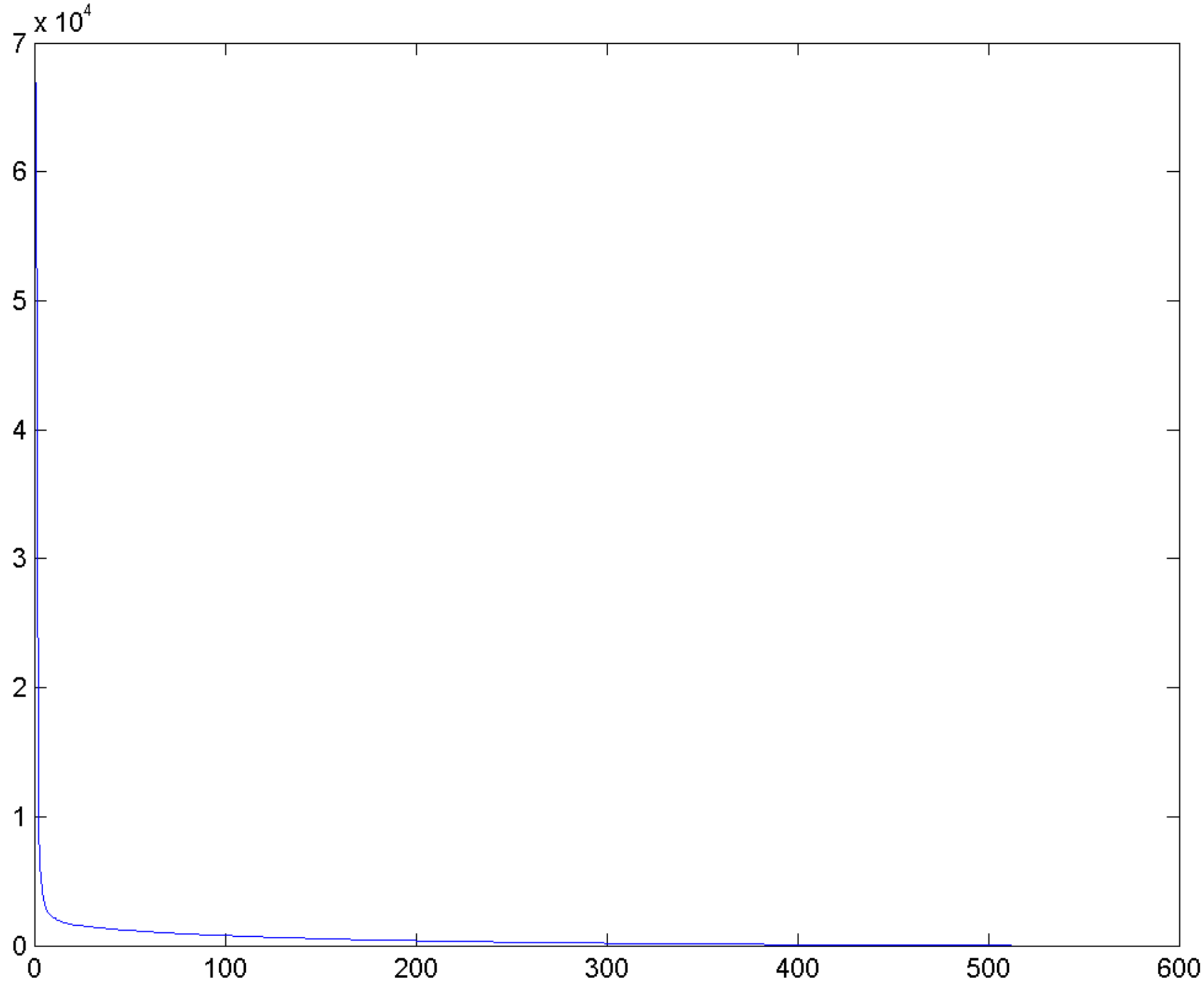
The singular values of $\mathbf{A}$ (i.e. diagonal elements of $\mathbf{S}$) are square-roots of the eigenvalues of $\mathbf{A}^T\mathbf{A}$.

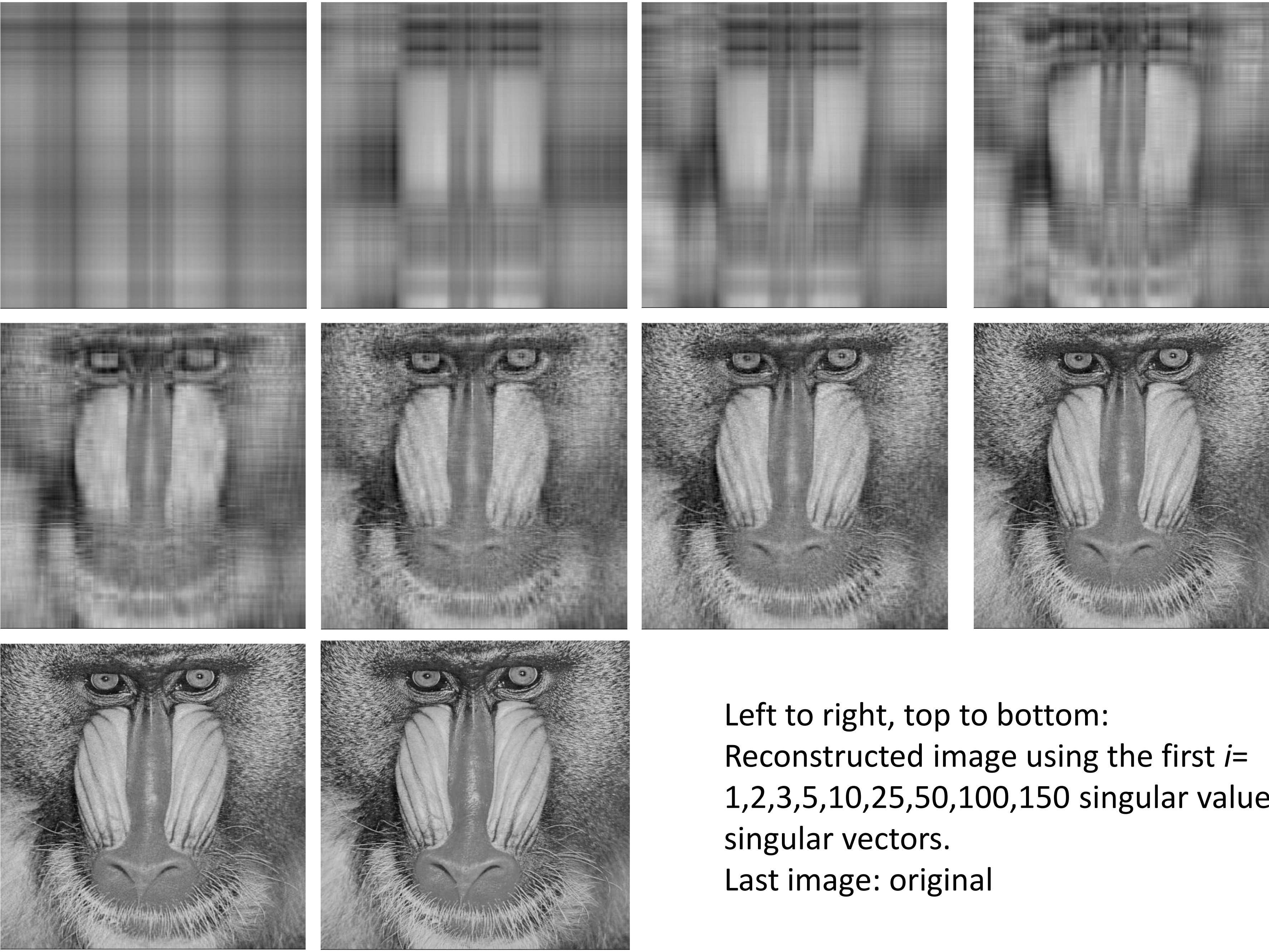
Application: SVD of Natural Images

- An image is a 2D array – each entry contains a grayscale value. The image can be treated as a matrix.
- It has been observed that for many image matrices, the singular values undergo rapid decay (note: they are always non-negative).
- An image can be approximated with the *k* largest singular values and their corresponding singular vectors:

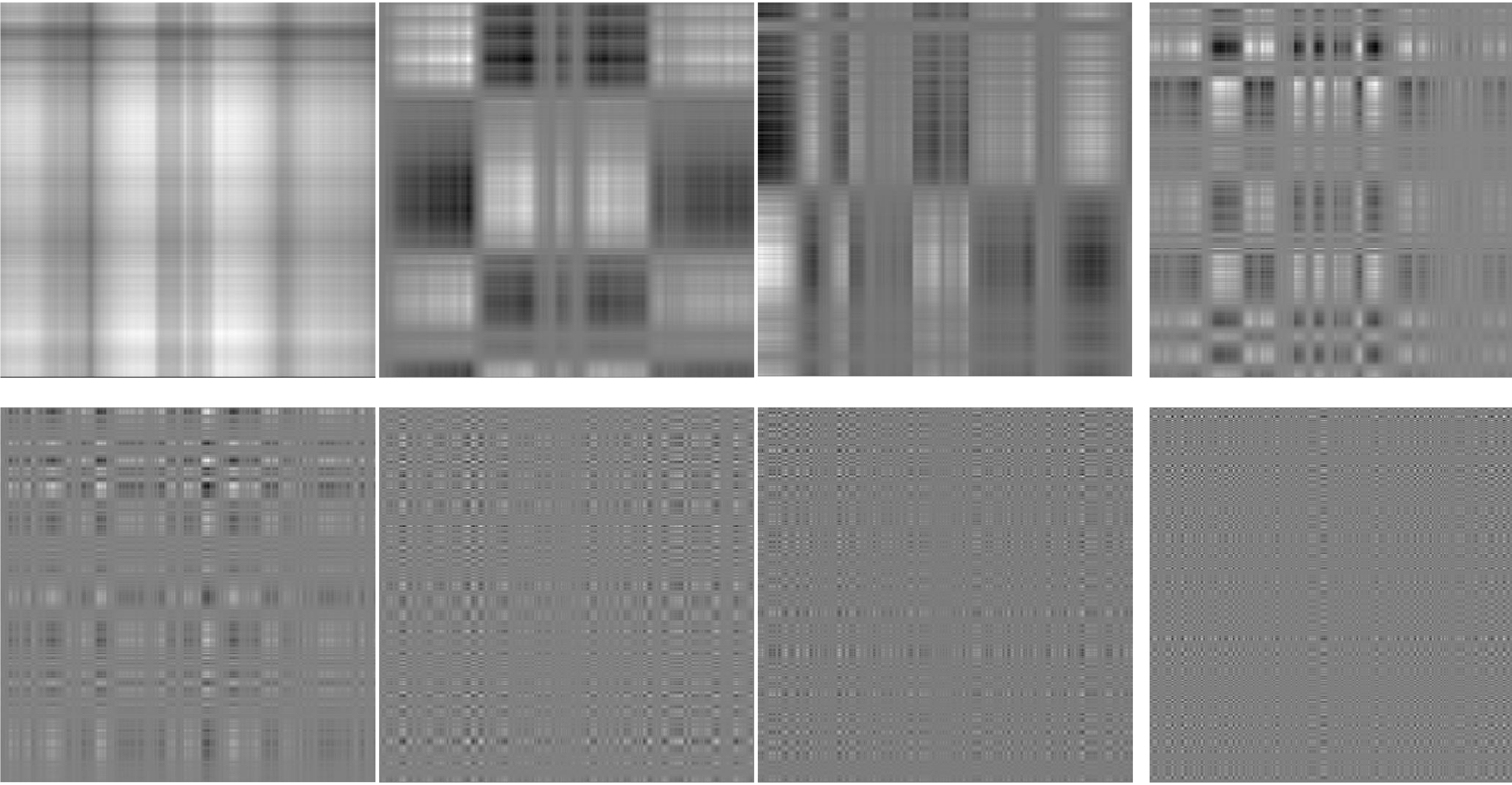
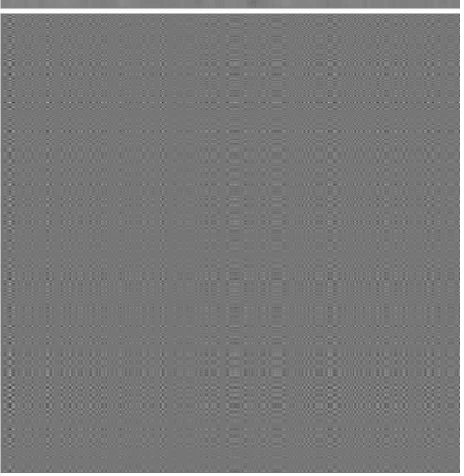
$$\mathbf{A} \approx \sum_{i=1}^{k} \mathbf{S}_{ii}\mathbf{u}_i\mathbf{v}_i^t, k < \min(m,n)$$

Singular values of the Mandrill Image: notice the rapid exponential decay of the values! This is characteristic of MOST natural images.





Left to right, top to bottom:
Reconstructed image using the first *i*= 1,2,3,5,10,25,50,100,150 singular values and singular vectors.
Last image: original

Left to right, top to bottom, we display:

$$\mathbf{u}_i \mathbf{v}_i^T$$

where i = 1,2,3,5,10,25,50,100,150. Note each image is independently re-scaled to the 0-1 range for display purpose.

Note: the spatial frequencies increase as the singular values decrease

SVD: Use in Image Compression

- Instead of storing *mn* intensity values, we store *(n+m+1)r* intensity values where *r* is the number of stored singular values (or singular vectors). The remaining *m-r* singular values (and hence their singular vectors) are effectively set to 0.

- This is called as storing a low-rank (rank *r*) approximation for an image.

Properties of SVD: Best low-rank reconstruction

- SVD gives us the **best possible** rank-*r* approximation to *any* matrix (it may or may not be a natural image matrix).
- In other words, the solution to the following optimization problem:

$$\min_{\hat{\mathbf{A}}} \left\| \hat{\mathbf{A}} - \mathbf{A} \right\|_F^2 \text{ where } \operatorname{rank}(\hat{\mathbf{A}}) = r, r \leq \min(m,n)$$

is given using the SVD of **A** as follows:

$$\hat{\mathbf{A}} = \sum_{i=1}^{r} \mathbf{S}_{ii} \mathbf{u}_i \mathbf{v}_i^t, \text{ where } \mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

Note: We are using the singular vectors corresponding to the *r* **largest** singular values.

This property of the SVD is called the **Eckart Young Theorem**.

Properties of SVD: Best low-rank reconstruction

$$\min_{\hat{\mathbf{A}}} \left\| \hat{\mathbf{A}} - \mathbf{A} \right\|_F^2 \text{ where } \operatorname{rank}(\hat{\mathbf{A}}) = r, r \le \min(m, n)$$

$$\left\| \mathbf{A} \right\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{A}_{ij}^2}$$

Frobenius norm of the matrix (fancy way of saying you square all matrix values, add them up, and then take the square root!)

$$Note: \left\| \hat{\mathbf{A}} - \mathbf{A} \right\|_F^2 = \sigma_{\mathrm{r+1}}^2 + \sigma_{\mathrm{r+2}}^2 + \ldots + \sigma_n^2$$

Why?

Geometric interpretation: Eckart-Young theorem

- The best linear approximation to an ellipse is its longest axis.
- The best 2D approximation to an ellipsoid in 3D is the ellipse spanned by the longest and second-longest axes.
- And so on!

Properties of SVD: Singularity

- A square matrix **A** is non-singular (i.e. invertible or full-rank) if and only if all its singular values are non-zero.
- The ratio σ_1/σ_n tells you how close **A** is to being singular. This ratio is called **condition number** of **A**. The larger the condition number, the closer the matrix is to being singular.

Properties of SVD: Rank, Inverse, Determinant

- The rank of a rectangular matrix **A** is equal to the number of non-zero singular values. Note that rank(**A**) = rank(**S**).
- SVD can be used to compute inverse of a square matrix:

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \mathbf{A} \in R^{n \times n},$$
$$\mathbf{A}^{-1} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T$$

- Absolute value of the determinant of square matrix **A** is equal to the product of its singular values.

$$|\det(\mathbf{A})| = |\det(\mathbf{U}\mathbf{S}\mathbf{V}^T)| = |\det(\mathbf{U})\ \det(\mathbf{S})\ \det(\mathbf{V}^T)| = \det(\mathbf{S}) = \prod_{i=1}^{n} \sigma_i$$

Properties of SVD: Pseudo-inverse

- SVD can be used to compute pseudo-inverse of a rectangular matrix:

$$\mathbf{A} = \mathbf{USV}^T, \mathbf{A} \in R^{m \times n},$$

$$\mathbf{A}^+ = \mathbf{VS}_0^{-1}\mathbf{U}^T, \text{where } \mathbf{S}_0^{-1}(i,i) = \mathbf{S}^{-1}(i,i) = \frac{1}{\mathbf{S}(i,i)} \text{ for all}$$

$$\text{non-zero singular values and } \mathbf{S}_0^{-1}(i,i) = 0 \text{ otherwise.}$$

Properties of SVD: Frobenius norm

- The Frobenius norm of a matrix is equal to the square-root of the sum of the squares of its singular values:

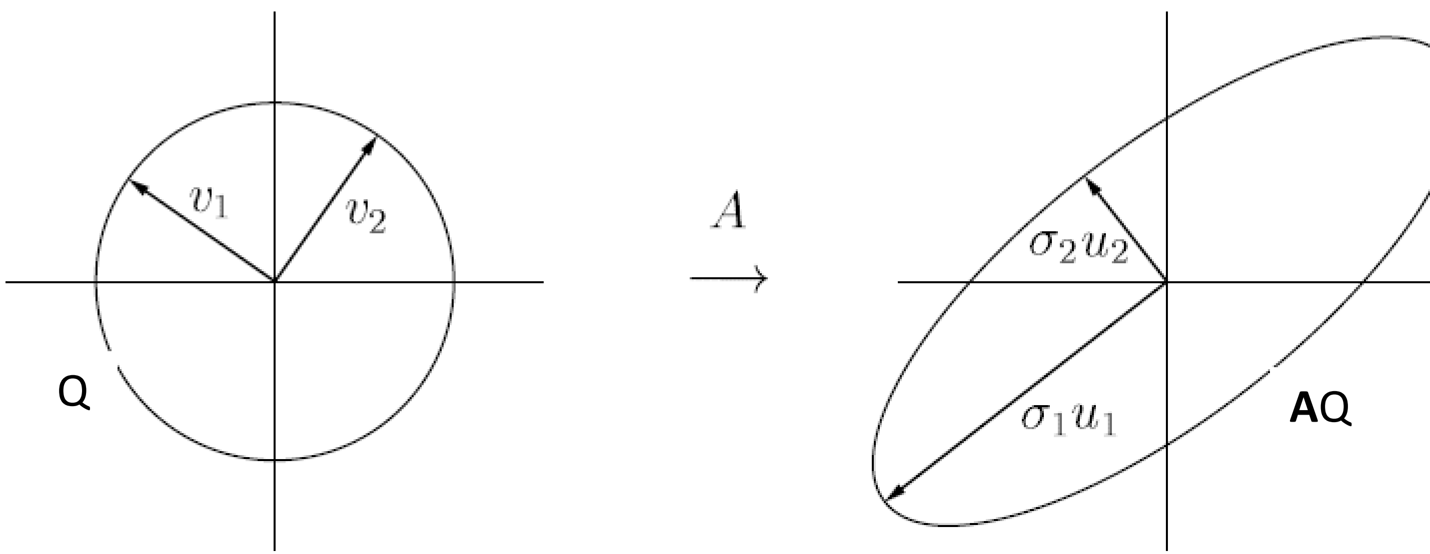
$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{A}_{ij}^2} = \sqrt{trace(\mathbf{A}^T\mathbf{A})} = \sqrt{trace((\mathbf{USV}^T)^T(\mathbf{USV}^T))}$$

$$= \sqrt{trace(\mathbf{V}^T\mathbf{S}^2\mathbf{V})} = \sqrt{trace(\mathbf{V}\mathbf{V}^T\mathbf{S}^2)} = \sqrt{trace(\mathbf{S}^2)}$$

$$= \sqrt{\sum_i \sigma_i^2}$$

Geometric interpretation of the SVD

- Any *m* x *n* matrix **A** transforms a hypersphere Q of unit radius (called as unit sphere) in $\mathcal{R}^n$ into a hyperellipsoid in $\mathcal{R}^m$ (assume *m* >= *n*).



Geometric interpretation of the SVD

- But why does **A** transform the hypersphere into a *hyperellipsoid*?
- This is because **A = USV**^{T}.
- **V**^{T} transforms the hypersphere into another (rotated/reflected) hypersphere.
- **S** stretches the sphere into a hyperellipsoid whose semi-axes coincide with the coordinate axes as per **V**.
- **U** rotates/reflects the hyperellipsoid without affecting its shape.
- As any matrix **A** has an SVD decomposition, it will always transform the hypersphere into a hyperellipsoid.
- If **A** does not have full rank, then some of the semi-axes of the hyperellipsoid will have length 0!

Geometric interpretation of the SVD

- Assume **A** has full rank for now.
- The singular values of **A** are the lengths of the *n* principal semi-axes of the hyperellipsoid. The lengths are thus $\sigma_1, \sigma_2, \ldots, \sigma_n$.
- The *n* left singular vectors of **A** are the directions $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ (all unit-vectors) aligned with the *n* semi-axes of the hyperellipsoid.
- The *n* right singular vectors of **A** are the directions $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ (all unit-vectors) in hypersphere Q, which the matrix **A** transforms into the semi-axes of the hyperellipsoid, i.e.

$$\forall i, \mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i$$

Geometric interpretation of the SVD

- Expanding on the previous equations, we get the reduced form of the SVD

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \left[\begin{array}{c|c|c|c} & & & \\ v_1 & v_2 & \cdots & v_n \\ & & & \end{array}\right] = \left[\begin{array}{c|c|c|c} & & & \\ u_1 & u_2 & \cdots & u_n \\ & & & \end{array}\right] \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}$$

n x *n* orthonormal matrix **V**

m x *n* matrix (*m* >> *n*) with orthonormal columns - **U**

n x *n* diagonal matrix - **S**

Computation of the SVD

- We will not explore algorithms to compute the SVD of a matrix, in this course.
- SVD routines exist in the LAPACK library and are interfaced through the following MATLAB functions:

s = svd(X) returns a vector of singular values.

[U,S,V] = svd(X) produces a diagonal matrix S of the same dimension as X, with nonnegative diagonal elements in decreasing order, and unitary matrices U and V so that X = U*S*V'.

[U,S,V] = svd(X,0) produces the "economy size" decomposition. If X is m-by-n with m > n, then svd computes only the first n columns of U and S is n-by-n.

[U,S,V] = svd(X,'econ') also produces the "economy size" decomposition. If X is m-by-n with m >= n, it is equivalent to svd(X,0). For m < n, only the first m columns of V are computed and S is m-by-m.

s = svds(A,k) computes the k largest singular values and associated singular vectors of matrix A.

SVD Uniqueness

- If the singular values of a matrix are all distinct, the SVD is unique – up to a multiplication of the corresponding columns of **U** and **V** by a sign factor.
- Why?

$$\mathbf{A} = \sum_{i=1}^{r} \mathbf{S}_{ii}\mathbf{u}_i\mathbf{v}_i^t = \mathbf{S}_{11}\mathbf{u}_1\mathbf{v}_1^t + \mathbf{S}_{22}\mathbf{u}_2\mathbf{v}_2^t + \ldots + \mathbf{S}_{rr}\mathbf{u}_r\mathbf{v}_r^t$$

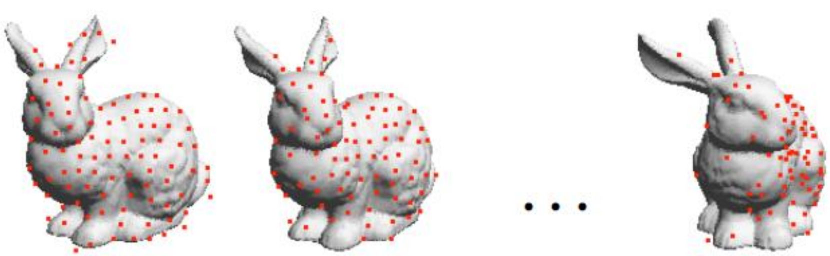
$$= \mathbf{S}_{11}(\;\mathbf{-u}_1)\;(\;\mathbf{-v}_1^t) + \mathbf{S}_{22}\mathbf{u}_2\mathbf{v}_2^t + \ldots + \mathbf{S}_{rr}(\;\mathbf{-u}_r)\;(\;\mathbf{-v}_r^t)$$

SVD Uniqueness

- A matrix is said to have *degenerate* singular values, if it has the same singular value for 2 or more pairs of left and right singular vectors.
- In such a case any normalized linear combination of the left (right) singular vectors is a valid left (right) singular vector for that singular value.

Any other applications of SVD?

- **Face recognition** – the popular eigenfaces algorithm can be implemented using SVD too!
- **Point matching:** Consider two sets of points, such that one point set is obtained by an unknown rotation of the other. Determine the rotation!
- **Structure from motion:** Given a sequence of images of a object undergoing rotational motion, determine the 3D shape of the object as well as the 3D rotation at every time instant!



PCA Algorithm using SVD

1. Compute the mean of the given points:

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i, \mathbf{x}_i \in R^d, \bar{\mathbf{x}} \in R^d$$

2. Deduct the mean from each point:

$$\bar{\mathbf{x}}_{\mathbf{i}} = \mathbf{x}_{\mathbf{i}} - \bar{\mathbf{x}}$$

3. Compute the covariance matrix of these mean-deducted points:

$$\mathbf{C} = \frac{1}{N-1}\sum_{i=1}^{N}\bar{\mathbf{x}}_i\bar{\mathbf{x}}_i^T = \frac{1}{N-1}\mathbf{X}\mathbf{X}^T, Note: \mathbf{C} \in R^{d\times d}$$

$$\mathbf{X} = [\bar{\mathbf{x}}_{\mathbf{1}} \,|\, \bar{\mathbf{x}}_{\mathbf{2}} \,|\, \ldots \,|\, \bar{\mathbf{x}}_{\mathbf{N}}] \in R^{d\times N}$$

PCA Algorithm using SVD

4. Instead of finding the eigenvectors of **C**, we find the left singular vectors of **X** and its singular values

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \mathbf{U} \in R^{d \times d}$$

$\mathbf{U}$ contains the eigenvectors of $\mathbf{X}\mathbf{X}^T$.

U,**S**,**V** are obtained by computing the SVD of **X**.

5. Extract the *k* eigenvectors in **U** corresponding to the *k* largest singular values to form the extracted **eigenspace:**

$$\hat{\mathbf{U}}_{\mathbf{k}} = \mathbf{U}(:,1:k)$$

There is an implicit assumption here that the first *k* indices indeed correspond to the *k* largest eigenvalues. If that is not true, you would need to pick the appropriate indices.

References

- Scientific Computing, Michael Heath
- Numerical Linear Algebra, Treftehen and Bau
- http://en.wikipedia.org/wiki/Singular_value_decomposition