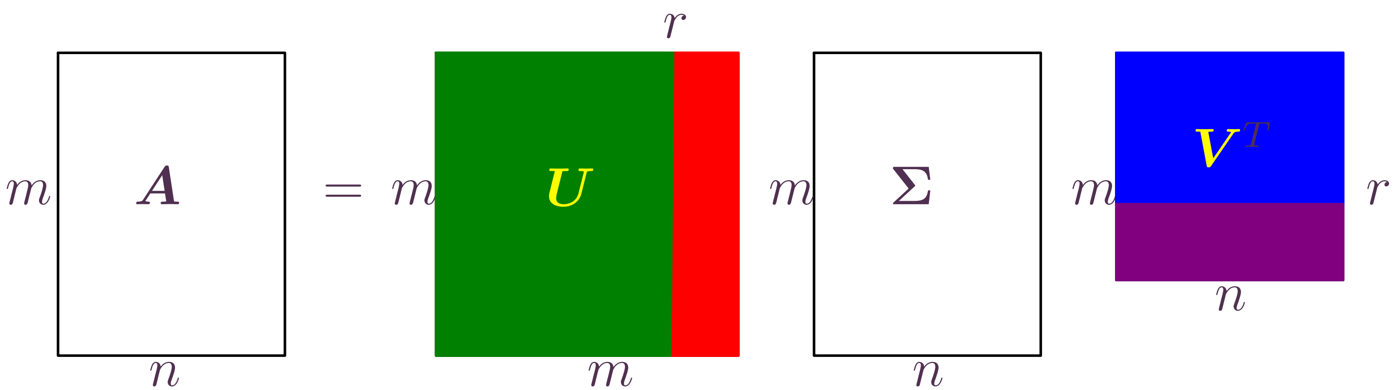
- New concepts:
  - SVD computation
  - Matrix norm
  - Low-rank approximations
  - Image compression

- SVD of $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ reveals: $\operatorname{rank}(\boldsymbol{A})$, bases for $C(\boldsymbol{A}), N(\boldsymbol{A}^T), C(\boldsymbol{A}^T), N(\boldsymbol{A})$



$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{u}_1 & \dots & \boldsymbol{u}_r & \boldsymbol{u}_{r+1} & \dots & \boldsymbol{u}_m \end{pmatrix} \begin{pmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & & \\ & & & 0 & \\ & & & & \ddots \end{pmatrix} \begin{pmatrix} \boldsymbol{v}_1^T \\ \vdots \\ \boldsymbol{v}_r^T \\ \boldsymbol{v}_{r+1}^T \\ \vdots \\ \boldsymbol{v}_n^T \end{pmatrix}$$

- From $A = U\Sigma V^T$ deduce $AA^T = U\Sigma^2 U^T$, $A^TA = V\Sigma^2 V^T$, hence U is the eigenvector matrix of AA^T, and V is the eigenvector matrix of A^TA
- SVD computation is carried out by solving eigenvalue problems

```
matlab>>A=[2 -1; -3 1]; [U S2]=eig(A*A'); [V S2]=eig(A'*A); S=sqrt(S2); disp([ U
        S V']);
```

The above is *not* an SVD since the singular values on the diagonal are out of order. The matlab `svd` function returns the correct ordering.

```
matlab>>[U S V]=svd(A); disp([U S V']);
```

- Hand computation of the SVD is a direct application of eigenvalue computation. Note that eigenvalues of AA^T and A^TA are identical, but the eigenvectors differ.

- From $A = U\Sigma V^T$ deduce $AA^T = U\Sigma^2 U^T$, $A^T A = V\Sigma^2 V^T$, hence U is the eigenvector matrix of AA^T, and V is the eigenvector matrix of $A^T A$
- SVD computation is carried out by solving eigenvalue problems

```
matlab>>A=[2 -1; -3 1]; [U S2]=eig(A*A'); [V S2]=eig(A'*A); S=sqrt(S2); disp([ U
        S V']);
```

```
>>    -0.8174   -0.5760    0.2588         0   -0.3606   -0.9327
      -0.5760    0.8174         0    3.8643   -0.9327    0.3606
```

The above is *not* an SVD since the singular values on the diagonal are out of order. The matlab `svd` function returns the correct ordering.

```
matlab>>[U S V]=svd(A); disp([U S V']);
```

- Hand computation of the SVD is a direct application of eigenvalue computation. Note that eigenvalues of AA^T and $A^T A$ are identical, but the eigenvectors differ.

- From $A = U\Sigma V^T$ deduce $AA^T = U\Sigma^2 U^T$, $A^T A = V\Sigma^2 V^T$, hence U is the eigenvector matrix of AA^T, and V is the eigenvector matrix of $A^T A$
- SVD computation is carried out by solving eigenvalue problems

```
matlab>>A=[2 -1; -3 1]; [U S2]=eig(A*A'); [V S2]=eig(A'*A); S=sqrt(S2); disp([ U
        S V']);
```

```
>>    -0.8174   -0.5760    0.2588         0   -0.3606   -0.9327
      -0.5760    0.8174         0    3.8643   -0.9327    0.3606
```

The above is *not* an SVD since the singular values on the diagonal are out of order. The matlab `svd` function returns the correct ordering.

```
matlab>>[U S V]=svd(A); disp([U S V']);
```

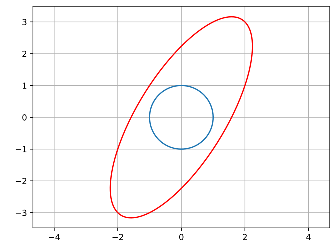
```
>>    -0.5760    0.8174    3.8643         0   -0.9327    0.3606
       0.8174    0.5760         0    0.2588   -0.3606   -0.9327
```

- Hand computation of the SVD is a direct application of eigenvalue computation. Note that eigenvalues of AA^T and $A^T A$ are identical, but the eigenvectors differ.

- Construct a diagram of the SVD of $\boldsymbol{A}=\begin{bmatrix} 2 & -1 \\ 3 & 1 \end{bmatrix}$, with $\boldsymbol{f}(\boldsymbol{x})=\boldsymbol{A}\boldsymbol{x}$ the associated linear mapping by taking $\theta\in[0,2\pi]$ and

$$\boldsymbol{x}=\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, \|\boldsymbol{x}\|_2=1$$

traversing the unit circle in the domain of $\boldsymbol{f}\colon \mathbb{R}^2\to\mathbb{R}^2$. The image of the unit circle is an ellipse. The length of the semiaxes are the singular values of $\boldsymbol{A}$, the orientation of the semiaxes are given by the right singular vectors $\boldsymbol{U}$.



- The above offers a way to think about the "size" of a matrix as defined by the maximal amplification factor among all directions with the domain

Definition. *Given the vector norms* $\|\|_{(n)}: \mathbb{R}^n \to \mathbb{R}_+$. $\|\|_{(m)}: \mathbb{R}^m \to \mathbb{R}_+$ *for vector spaces* $(\mathbb{R}^m, \mathbb{R}, +, \cdot)$, $(\mathbb{R}^n, \mathbb{R}, +, \cdot)$, *the induced matrix norm of* $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *is defined as*

$$\|\boldsymbol{A}\|_{(m,n)} = \max_{\boldsymbol{x}\in\mathbb{R}^n} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_{(m)}}{\|\boldsymbol{x}\|_{(n)}} = \max_{\boldsymbol{x}\in\mathbb{R}^n, \|\boldsymbol{x}\|_{(n)}=1} \|\boldsymbol{A}\boldsymbol{x}\|_{(m)}.$$

The above definition states that the "size" of a matrix can be interpreted as the maximal amplication factor among all possible orientations of a unit vector input.

- The most commonly encountered case is for both the $\|\|_{(m)}$ and the $\|\|_{(n)}$ norms to be 2-norms

$$\|\boldsymbol{b}\|_{(m)} = \left(\sum_{i=1}^{m} b_i^2\right)^{1/2}, \|\boldsymbol{x}\|_{(n)} = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2}$$

- When the vector norms are both 2-norms as above, the induced matrix norm is simply the largest singular value of $\boldsymbol{A}$

$$\|\boldsymbol{A}\| = \sigma_1$$

Low-rank matrix approximation

- Full SVD

$$\boldsymbol{A} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T, r \leqslant \min(m, n).$$

- Truncated SVD

$$\boldsymbol{A} \cong \boldsymbol{A}_p = \sum_{i=1}^{p} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T.$$

Interpret $\boldsymbol{A}_p$ as furnishing an approximation to $\boldsymbol{A}$, with $\operatorname{rank}(\boldsymbol{A}_p) = p \leqslant r$.

- Many applications, e.g., image compression

Figure 1. Successive SVD approximations of Andy Warhol's painting, *Marilyn Diptych* (~1960), with $k = 10, 20, 40$ rank-one updates.