

economy size SVD

$$A = \tilde{U} \tilde{\Sigma} \tilde{V}^H = \tilde{U} \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{pmatrix} \tilde{V}^H$$

economy size SVD enthält **wesentliche** Information über eine Matrix:

- $\tilde{U} \in \mathbb{K}^{m \times r}$ ist eine ONB von $\text{Bild}(A)$
- $\tilde{V} \in \mathbb{K}^{n \times r}$ ist eine ONB von $(\text{Ker}(A))^\perp$

Speicherbedarf:

A	$m \times n$
economy	$m \times r + r + n \times r = (m + n + 1)r \ll m \times n$ falls $r \ll \min\{m, n\}$

Idee für Matrixkompression:

- approximiere $A \in \mathbb{K}^{m \times n}$ durch eine Rang- r -Matrix (mit einem $r \ll \min\{m, n\}$)
- speichere diese Rang- r -Matrix in ihrer economy SVD-Darstellung

Beispiel: Datenkompression mittels SVD

$$A = \tilde{U}\tilde{\Sigma}\tilde{V}^H = \tilde{U} \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{pmatrix} \tilde{V}^H$$

Speicherbedarf:

A	$m \times n$
economy	$m \times r + r + n \times r < mn$ falls $r \ll \min\{m, n\}$

Datenkompression mittels SVD:

1. bestimme SVD von A
2. behalte nur Info für die $\hat{r} \leq r$ größten Singulärwerte \rightarrow approximiere A durch

$$A' = \tilde{U}(:, [1 : \hat{r}])\tilde{\Sigma}([1 : \hat{r}], [1 : \hat{r}])\tilde{V}(:, [1 : \hat{r}])^H$$

3. Speicherbedarf: $m \times \hat{r} + \hat{r} + n \times \hat{r}$.

Wann funktioniert das?

Übung: Für die Frobeniusnorm $\|A\|_F^2 = \sum_{i,j} |a_{ij}|^2$ gilt:

$$\|A\|_F^2 = \sum_{i=1}^r \sigma_i^2.$$

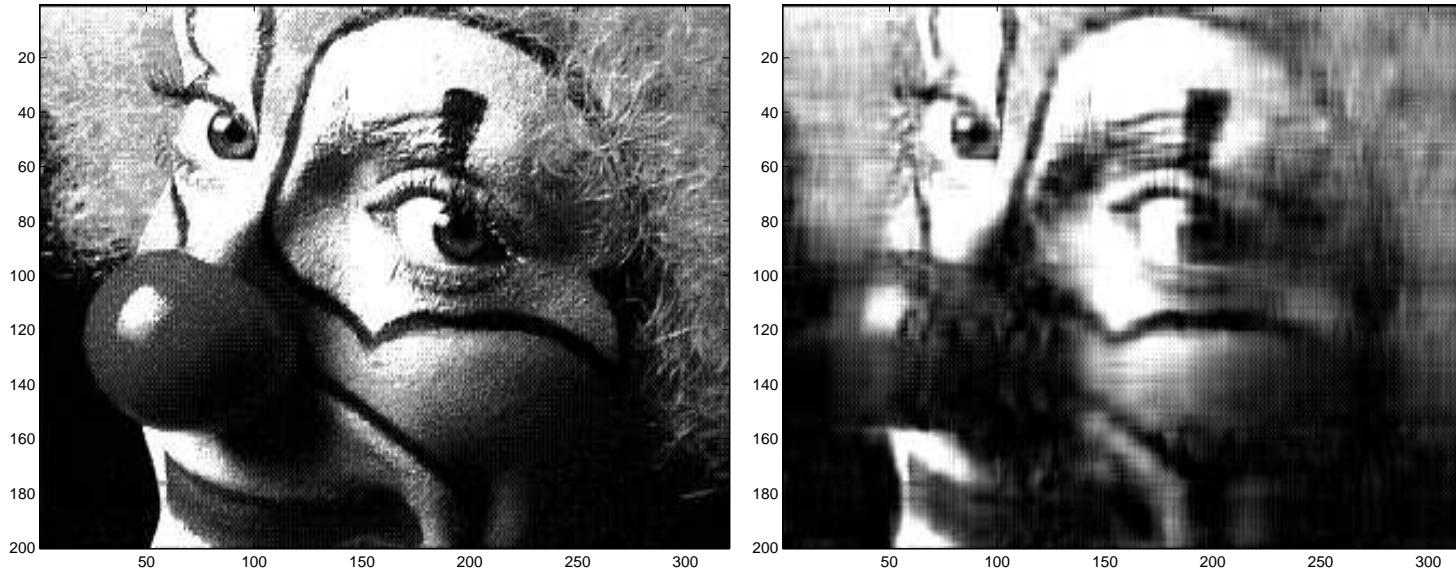
Damit ist für den Fehler $A - A'$:

$$A - A' = \tilde{U} \begin{pmatrix} 0 & & & & & & & & \\ & \dots & & & & & & & \\ & & 0 & & & & & & \\ & & & \sigma_{\hat{r}+1} & & & & & \\ & & & & \dots & & & & \\ & & & & & & \sigma_r & & \\ & & & & & & & & \end{pmatrix} \tilde{V}^H$$

d.h. $\|A - A'\|_F^2 = \sum_{i=\hat{r}+1}^r \sigma_i^2.$

Datenkompression mittels SVD funktioniert dann gut, wenn die Singulärwerte σ_i der zu komprimierende Matrix A schnell (für wachsendes i) klein werden.

Beispiel: Datenkompression mittels SVD



```
load clown.mat
figure(1);    image(X);    colormap('gray');
[U,S,V] = svd(X);
k = 15
figure(2);    image(U(:,1:k)*S(1:k,1:k)*V(:,1:k)');    colormap('gray')
```

Speicherbedarf:

X	$200 \times 320 = 64,000$
komprimiert	$200 \times k + 320 \times k = 7,800$ (für $k = 15$)

speichere lediglich $U(:, 1 : k)$ und $(S(1 : k, 1 : k) * V(:, 1 : k))'$