Example: SVD of an image

```
>> F = imread('cameraman.tif');
>> F = double(F) / 255; %normalizes values into [0,1]
>> imshow(F);
>> [U, S, V] = svd(F);
>> imshow(U(:,1)*S(1,1)*V(:,1)')
>> k = 10; imshow(U(:,1:k)*S(1:k,1:k)*V(:,1:k)')
```

Rank-1 $\iff$ a small entry in u_1 makes the whole row small.

Higher ranks still give a 'blocky' behaviour.

Still fewer data than the full image.

$\|A - U_1 S_1 V_1^T\|_F =$ sum of squares of neglected singular values.

Examine `diag(S)` — zeros at the end are due to duplicated rows/columns (likely at the top).

$$A = USV^T = \begin{bmatrix} u_1 \; u_2 \cdots u_m \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \\ & 0 & & \end{bmatrix} \begin{bmatrix} v_1 \; v_2 \; v_3 \cdots v_n \end{bmatrix}^T$$

How can we find the matrix X with $\operatorname{rk} X = 1$ that minimizes $\|A - X\|$? For $\|\cdot\|_2$, $\|\cdot\|_F$

$$X = yz^T = \begin{bmatrix} y_1 \cdot z_1 & y_1 \cdot z_2 & \cdots & y_1 \cdot z_n \\ y_2 \cdot z_1 & y_2 \cdot z_2 & & y_2 \cdot z_n \\ \vdots & \vdots & & \vdots \\ y_m \cdot z_1 & y_m \cdot z_2 & \cdots & y_m \cdot z_n \end{bmatrix}$$

$X, A \in \mathbb{R}^{m \times n}$ To represent X, we can store $y, z \sim m+n$ entries rather than $m \cdot n$

For $\|\cdot\|_2$, $\|\cdot\|_F$

$$\|A-X\|_F = \sum_{i,j=1}^{m,n} (a_{ij}-x_{ij})^2$$

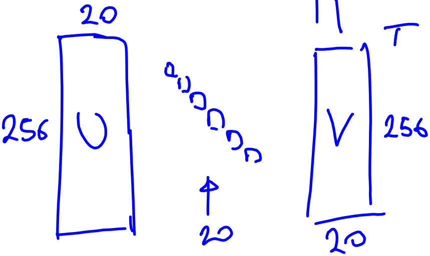
the solution is truncated SVD:

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

($u_1 \sigma_1 v_1^T$ ↑) best rank-1 approximant of A

↓ best rank-2 approximant of A

Best rank-20 approximant:

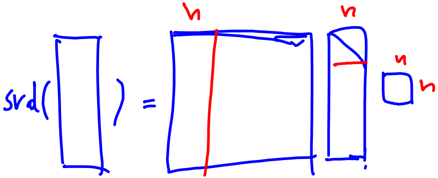


$$\sqrt{\frac{\sum_{i,j=1}^{m,n}\left(f_{ij}-x_{ij}\right)^2}{m\cdot n}} := \frac{\|F-X\|}{\sqrt{mn}}$$

Example: image classification

(With the provided files — Yale faces dataset.)

```
>> F = readyalefaces_to_tensor();
>> showyalefaces
>> size(F)
ans =
   243 320 11 15
>> imshow(F(:,:,1,1));
>> v = reshape(F(:,:,1,1), 243*320, 1)
>> M = reshape(F, 243*320, 11*15);
>> [U,S,V] = svd(M);
Error using svd
Requested 77760x77760 (45.1GB) array exceeds maximum array
become unresponsive. See array size limit or preference par
>> [U,S,V] = svd(M, 0); %thin SVD
```



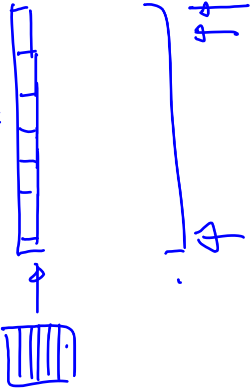
$m \times n$

$m = 243 * 320$

$n = 11 * 15$

$m^2 + n + n^2$

$mn + n + n^2$

$M =$ 

$$\hat{M} = M - \begin{bmatrix} m & m & m & m & \cdots & m \end{bmatrix}$$

$$= M - \text{meanface} \times [1, 1, \ldots, 1]$$

Variant: subtract "mean face"

```
>> meanface = mean(M,2);
>> imagesc(reshape(M(:,52) - meanface, 243, 320));
>> [U, S, V] = svd(M - meanface, 0);
>> imagesc(reshape(U(:,1), 243, 320)); % examine them
>> diag(S) % what do the zeros at the end mean?
```

(Let $\widehat{M} = M - \text{meanface} \cdot e^T$)

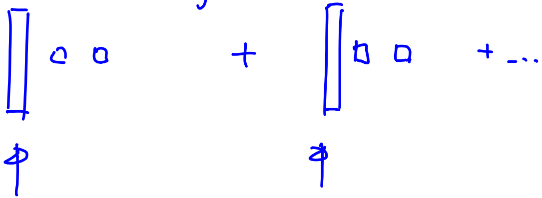
Each image is: meanface $+ \alpha_1 u_1 + \alpha_2 u_2 + \dots$

αs for the jth face are given by $U^T M_j$ (why? Linear algebra...)

```
>> interactiverec(M(:,1))
```

$$M = u_1 s_1 v_1^T + u_2 s_2 v_2^T + \cdots$$

$$M(:,j) = u_1 \cdot s_1 \cdot (v_1)_j^T + u_2 s_2 (v_2)_j^T$$



Variance around "mean face"

Sample variance (from statistics):
$\frac{1}{11\cdot 15}\sum_{j=1}^{11\cdot 15}(m_j - mean)(m_j - mean)^T = \widehat{M}\widehat{M}^T$.
$\widehat{M}\widehat{M}^T = USV^T(USV^T)^T = US^2U^T$. $\leftarrow$ an SVD of $\widehat{M}\widehat{M}^T$.

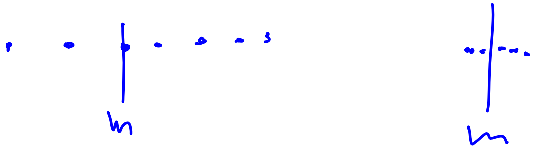
Larger singular components $\iff$ directions (in the "feature space") in which the variance is higher.

Geometric idea in a space of dimension $243 \cdot 320$, faces = points arranged like an ellipsoid.

Dimensionality reduction: we observe this ellipsoid in the directions of maximum variance, i.e., the leading components in the basis U of the 'face space' (α_1, α_2).

```
>> eigenfaces_scatter(F, [1,2])
```

Try other choices for the α indices.



$$\frac{1}{K}\sum_{i=1}^{K}(X_i - m)^2$$

more then 1 component:

$$V = \frac{1}{K}\sum_{i=1}^{K}(v_i - \text{mean})(v_i - \text{mean})^T$$

$$V = \begin{bmatrix} D & & \\ & D & \\ & & \ddots \end{bmatrix}$$

V_{ii} = variance of i-th component

V_{ij} = covariance of i-th and j-th component

Recognizing new images

Training set: remove an image for each individual.

```
>> T = F(:, :, 1:10, :);
>> T = reshape(T, 243*320, 10*15);
>> meanface = mean(T, 2);
>> [U, S, V] = svd(T - meanface, 0);
```

Test set: new image to be recognized.

```
>> S = F(:, :, 11, 1); %outside training test
>> S = reshape(S, 243*320, 1);
```

Look for "most similar" image along first 5 α scores.

```
>> training_scores = U(:,1:5)'*T;
>> test_scores = U(:,1:5)'*S;
>> distances = sum((training_scores - test_scores).^2, 1);
>> [value, position] = min(distances)
```

Limitations

Distance used: Frobenius distance among images $\implies$ not necessarily pictures of the same individual.

Need good dataset — if images are misaligned, it is a problem.

We are not trying to infer which ‘features’ are good / bad for recognition.
We never used the fact that we have several poses of the same individual.

Images outside of the “face space”

```
B = imread('bart.png'); B = double(B) / 255;
B = reshape(B,243*320,1);
test_scores = U(:,1:5)'*B;
interactiverec(B)
distances = sum((training_scores - test_scores).^2, 1);
[value, position] = min(distances)
```

Try also `B = imread('car.png');`

Example: text mining

1. Breakfast is the most important meal of the day.
2. I had a peanut butter sandwich for breakfast.
3. I like to eat almonds and peanuts.
4. People normally eat three meals a day.
5. My neighbor got a little dog the other day.
6. Cats and dogs are mortal enemies.
7. You mustn't feed peanuts to your dog.
8. My dog chased a cat in the garden.

Term-document matrix

	1	2	3	4	5	6	7	8
Breakfast	1	1						
Meal	1			1				
Day	1			1	1			
Peanut		1	1				1	
Sandwich		1						
Eat			1					
Almonds			1					
My					1			1
Neighbor					1			
Dog					1	1	1	1
Cat						1		1
Garden								1

Meaning of U and V

Columns of U = concepts (with positive/negative 'scores' for each word).

Columns of V = occurrences of concepts in each sentence.

(Think about "perfectly split" topics.)