

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 $\text{diag}(S)$ — zeros at the end are due to duplicated rows/columns (likely at the top).

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
```

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: $\text{meanface} + \alpha_1 u_1 + \alpha_2 u_2 + \dots$

α s for the j th face are given by $U^T M_j$ (why? Linear algebra...)

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

Variance around “mean face”

Sample variance (from statistics):

$$\frac{1}{11 \cdot 15} \sum_{j=1}^{11 \cdot 15} (m_j - \text{mean})(m_j - \text{mean})^T = \widehat{MM}^T.$$

$$\widehat{MM}^T = USV^T(USV^T)^T = US^2U^T. \leftarrow \text{an SVD of } \widehat{MM}^T.$$

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.

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.

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.)