Machine Learning and Artificial Intelligence

Lab 04 – Principal Component Analysis

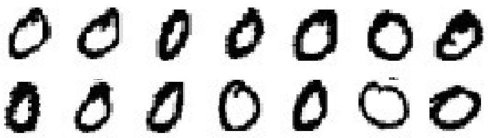
30/03/2021

The problem

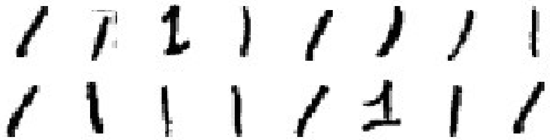
We want to recognise and classify images of handwritten figures
https://en.wikipedia.org/wiki/MNIST_database

For this lesson, we only consider images of “0” and “1”→ the training set size is reduced to about 12000 images

«0»



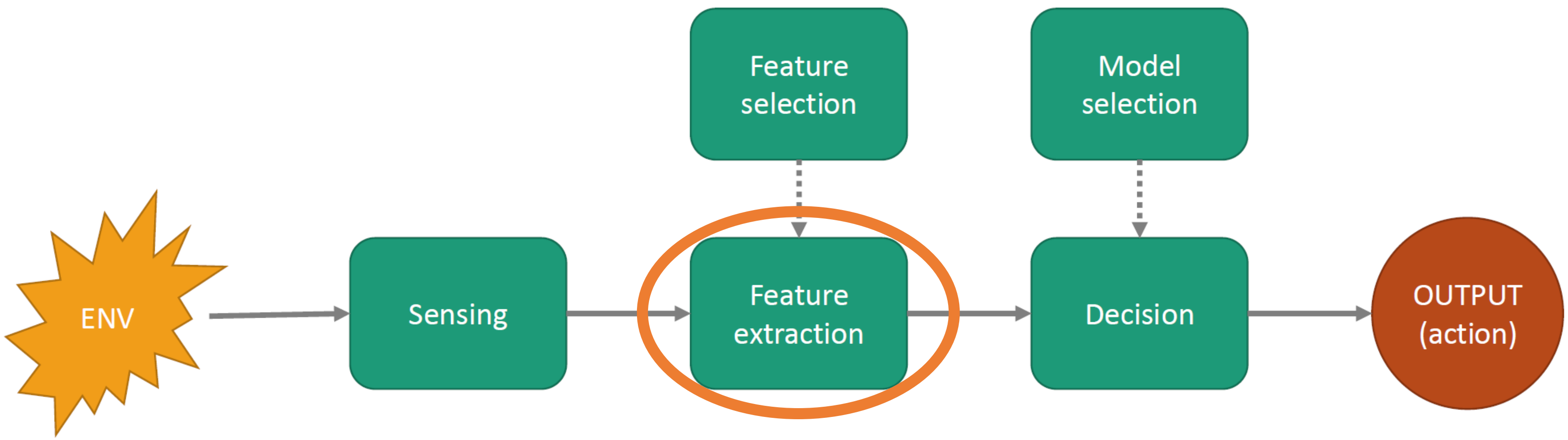
«1»



Feature Extraction

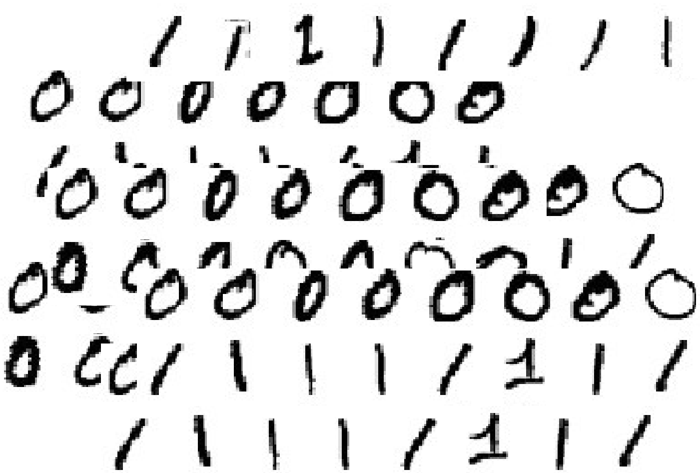
Problem: each image is a 28x28 matrix.

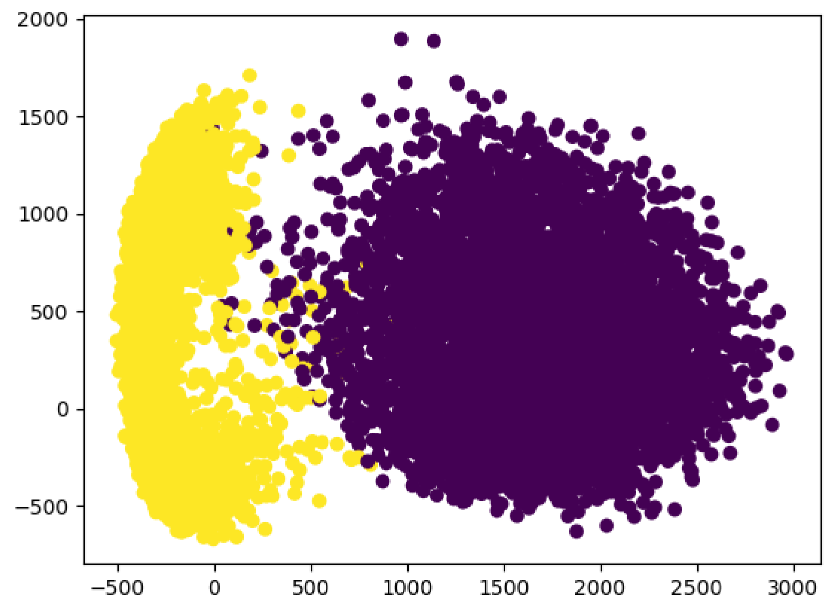
Feature extraction: How can I describe the points of the training set in a compact way?



Idea

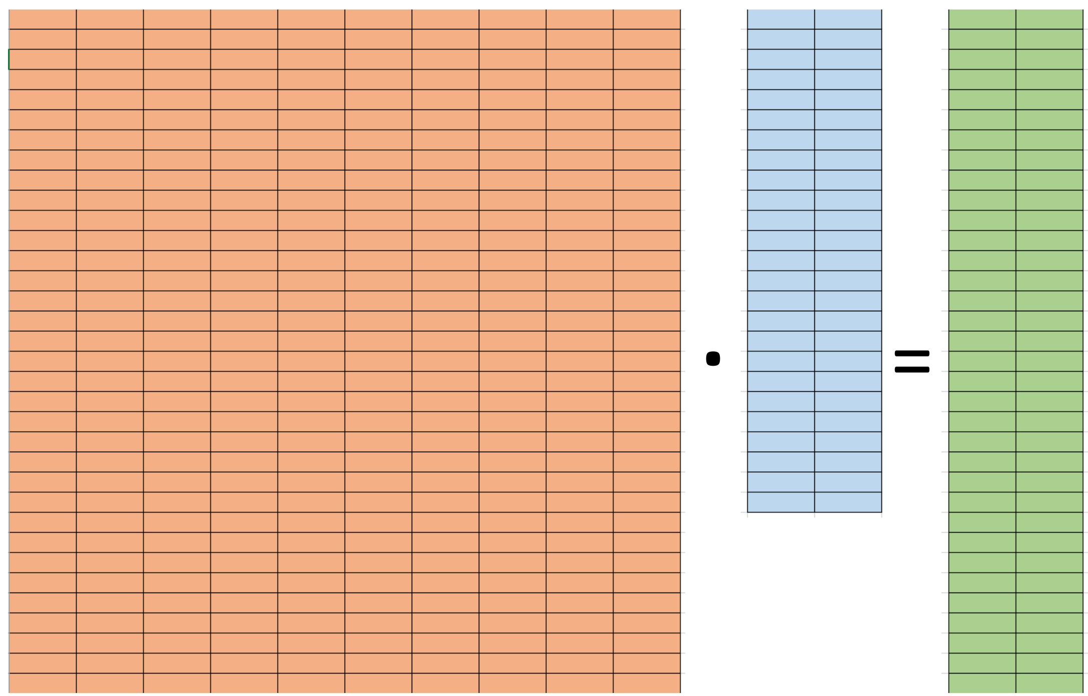
Reduce the dimensionality of the space, while retaining as much information as possible.





Idea

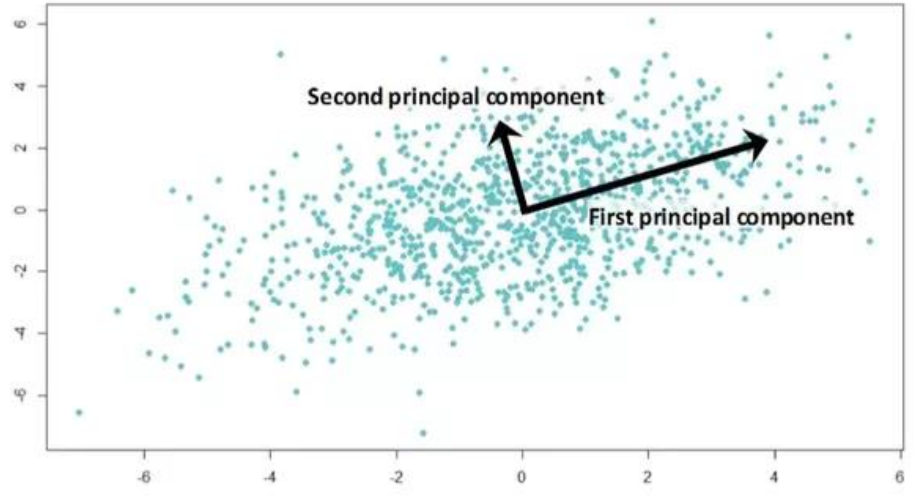
Dimensionality reduction:



X (12000x784) A (784x2) X’ (12000x2)

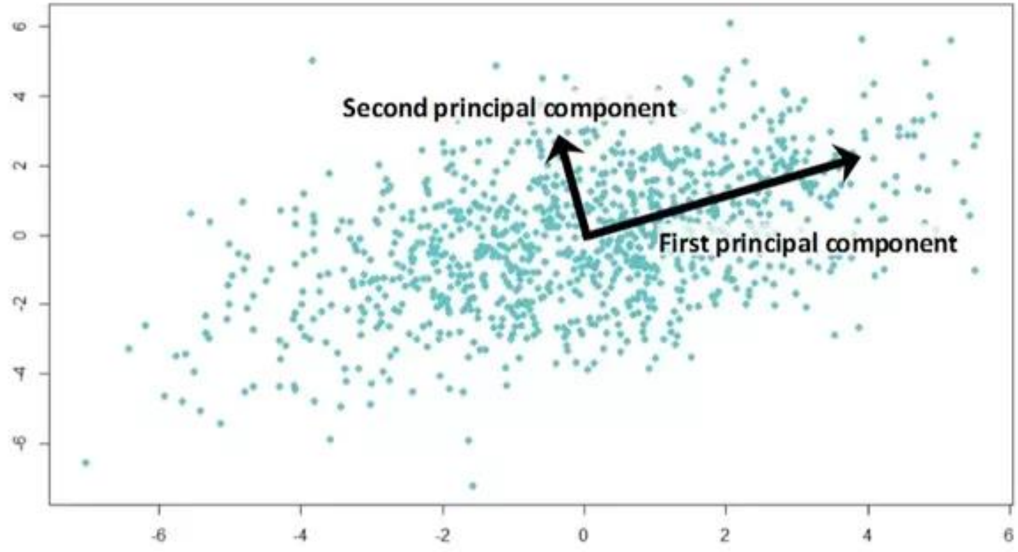
Principal Component Analysis

- Project the data into a space such that:
  - The first direction is the direction of maximum variance.
  - The second direction is the one of maximum variance, orthogonal to the first one.
  - And so on....
- Information → Variance



Principal Component Analysis

- The eigenvectors of the covariance matrix encode the principal directions.
- The largest eigenvalues occur in the directions of maximum dispersion of the data.



Principal Component Analysis

How to apply it:

1. Subtract from each $x^k, k = 1 \dots M$, the mean $m = \frac{1}{M} \sum_{k=1}^{M} x^k$, obtaining in this way the centered data $\{{x_c}^k\}$
2. Calculate the covariance matrix C from the centered data.
3. Calculate the eigenvalues and eigenvectors of the covariance matrix.

```
>> D,V = np.linalg.eigh(C)
>> # D vector of eigenvalues, V matrix with eigenvectors in its columns
```

https://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.eigh.html

Principal Component Analysis

How to apply it (2):

4. Sort the eigenvalues from largest to smallest (the eigenvectors corresponding to the largest eigenvalues encode the main directions).

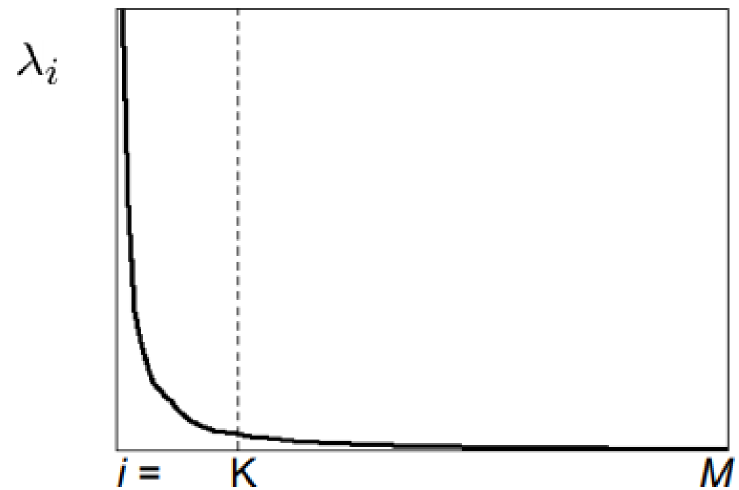
5. The transformation matrix T, that is used to reduce the data into N dimensions, is made of the first N eigenvectors (columns of V), corresponding to the N largest eigenvalues.

6. Compute ω^k, i.e., the data projected into the desired dimensions by multiplying ${x_c}^k, k = 1 \dots M$ by T. Each ω^k is composed of single components a_i^k , $i = 1 \dots N$ (the new features).

How many eigenvectors to use?

- Each eigenvector “carries” a certain variance, which can be seen as the amount of information with respect to the data.

- “Good” data, i.e., tractable, have low dimensions and high variance.

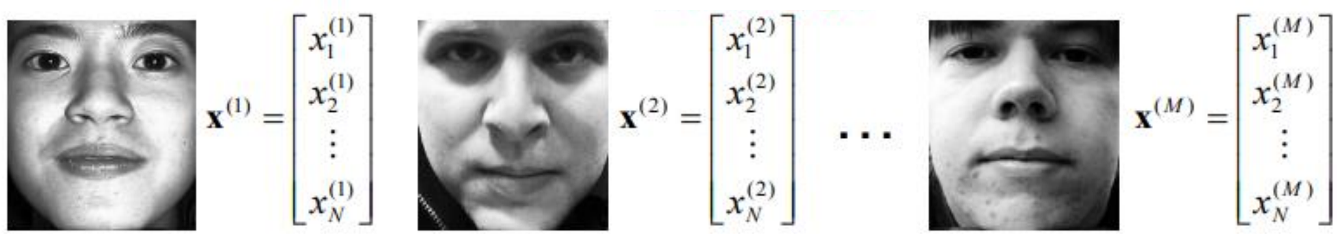


Exercises

Practical problem

- Appearance-based recognition of faces.
- Consider each point $x^k, k = 1 \dots M$ as the image of a face.

Gray levels

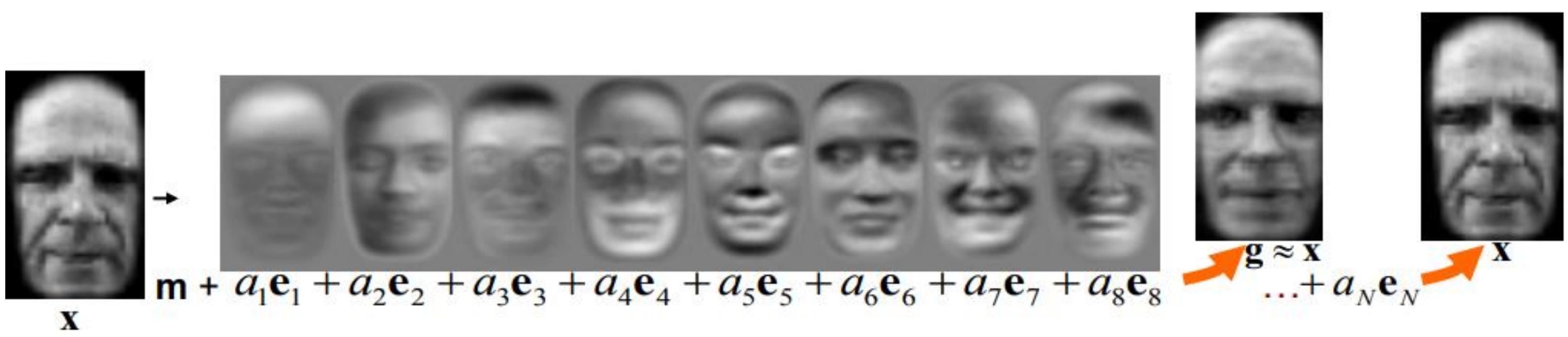


Usually,
M << N

EigenFaces

- The image of a face x can be projected in the same way in a new space obtaining $\omega^k = [a_1^k, a_2^k, \ldots a_N^k]$.

- The reconstruction g is given from the sum of the multiplications of the components a_i^k of the image x, with the corresponding eigenfaces $e_i, i = 1 \ldots N$, used for the projection.



Recognition with eigenfaces

TRAINING

1. Data analysis
   a) Apply PCA, obtaining the eigenfaces $e_i, i = 1 \dots N \quad \equiv$ V and the components $a_i^k, i = 1 \dots N$ for each image.
   b) Calculate the threshold

$$\Theta = \max\left\{\left\|\omega^j - \omega^k\right\|_2\right\} \; for \; j, k = 1, \dots M$$

TESTING

2. Given a new image to recognise x^{te}
   a) Subtract the mean of the training dataset m, obtaining x_c^{te}
   b) Project x_c^{te} into the new space and obtain $\omega^{te} = [a_1, a_2, \dots, a_N]$
   c) Calculate the set of distances

$$(\epsilon^k) \quad = \left\|\omega^{te} - \omega^k\right\|_2 \; for \; k = 1, .., M$$

Recognition with eigenfaces

TESTING

3. Reconstruct the face using eigenfaces and components

$$g = \sum_{i=1}^{K} a_i e_i \quad or \;\; g = V\omega^{te}$$

4. Calculate the distance between the "unknown" starting face and the reconstructed face.

$$\xi = \|g - x_c^{te}\|_2$$

5. If
   - $\xi \geq \Theta$ → it's not a face
   - $\xi < \Theta$ and $\epsilon^k \geq \Theta\,, (\mathrm{k} = 1, \ldots, \mathrm{M})$ → it's a new face
   - $\xi < \Theta$ and $\min\{\epsilon^k\} < \Theta$ it's a known face, more precicely the $k_{best} - th$, where:

$$k_{best} = argmin_k \{\epsilon^k\}$$

Exercises