Week 7b:

Dimensionality reduction

G6061: Fundamentals of Machine Learning [23/24]

Dr. Johanna Senk

UNIVERSITY
OF SUSSEX

Pre-processing data

- **Input normalisation**: Normalise (rescale) features to lie on a sensible scale.
  Useful for network/model training, and for visual inspection of data.
- **Imputing**: I.e., filling in missing data.
- **Feature construction and selection**: Incorporate your **domain knowledge** to select features that are likely to be useful.
  Mindfully build features from the data based on **visual inspection and consideration of the task** to be done.

- **Today: Dimensionality reduction**: Use maths to efficiently reduce the number of dimensions.
  Define a few new variables that are combinations of the raw variables, and account for most of the variance.
  - Principal Components Analysis (PCA)
  - Aside on eigenvalues and eigenvectors

UNIVERSITY OF SUSSEX

Dimensionality reduction

- **What is it?**
  - Reducing the number of dimensions (i.e., features) that are used to represent an object of interest

- **Why?**
  - Computationally expensive to store and work with high dimensional spaces
  - Overfitting less likely if there are fewer parameters to learn
  - One feature that combines several noisy, yet correlated, features means modelling accuracy improves
  - Impossible to visualize high-dimensional spaces - humans can typically only visualize 2 or 3 dimensions!

UNIVERSITY OF SUSSEX

Curse of dimensionality

- E.g., K-means clustering. Algorithm works based on distances between the things we want to classify. For 2 members of our population, $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$\text{Distance}(\boldsymbol{x}, \boldsymbol{y}) = [(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_D - y_D)^2]$$

- K-means clustering relies on things in the same class being closer together.
  - Suppose feature 1 and 2 will take very different values when $\boldsymbol{x}$ and $\boldsymbol{y}$ are in different classes and very similar values when $\boldsymbol{x}$ and $\boldsymbol{y}$ are in the same class.
  - Suppose features 3, 4, . . . , D vary a lot, and don't actually tell you anything about the class of x and y.
  - Then, if the number of features d is large, k-means gets messed up:
    the distances between things in the same class and things in different classes will both vary a lot, and on average won't be very different for things in the same class vs things in different classes.

UNIVERSITY OF SUSSEX

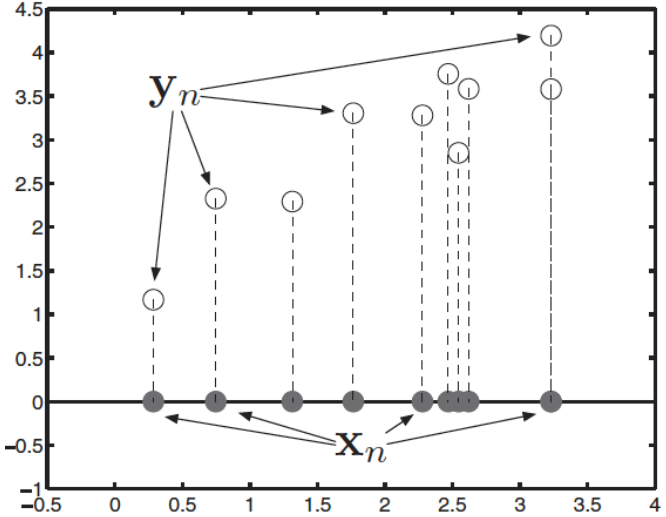
Example: Trajectory of depression symptoms

24 features → reduced to 3 features for clustering → visualize just 2 of these features

Question no.	Question
1	Do you often have back ache?
2	Do you feel tired most of the time?
3	Do you often feel miserable or depressed?
4	Do you often have bad headaches?
5	Do you often get worried about things?
6	Do you usually have great difficulty in falling or staying asleep?
7	Do you usually wake unnecessarily early in the morning?
8	Do you wear yourself our worrying about your health?
9	Do you often get into a violent rage?
10	Do people often annoy and irritate you?
11	Have you at time had a twitching of the face, head or shoulders?
12	Do you often suddenly become scared for no good reason?
13	Are you scared to be alone when there are no friends near you?
14	Are you easily upset of irritated?
15	Are you firghtened of going out alone or of meeting people?
16	Are you constantly keyed up and jittery?
17	Do you suffer from indigestions?
18	Do you often suffer from an upset stomach?
19	Is your appetite poor?
20	Does every little thing get on your nerves and wear you out?
21	Does your heart often race like mad?
22	Do you often have bad pains in your eyes?
23	Are you troubled with rhematism or fibrotics?
24	Have you ever had a nervous breakdown?

UNIVERSITY OF SUSSEX

Illustration: 2 features down to 1 feature

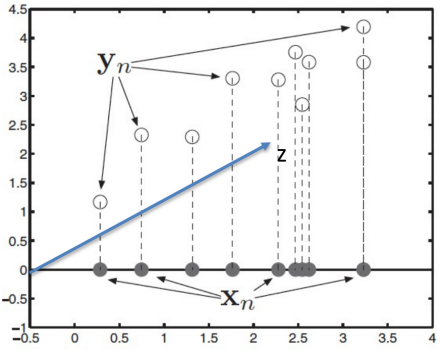


UNIVERSITY OF SUSSEX

Lower dimensional projections

Generally, rather than picking a subset of the features, we construct new features z_f that are (linear) combinations of existing features $\{x_1, x_2, ..., x_D\}$:

$$z_f = \sum_{i=1}^{D} w_{f,i} x_i$$



UNIVERSITY OF SUSSEX

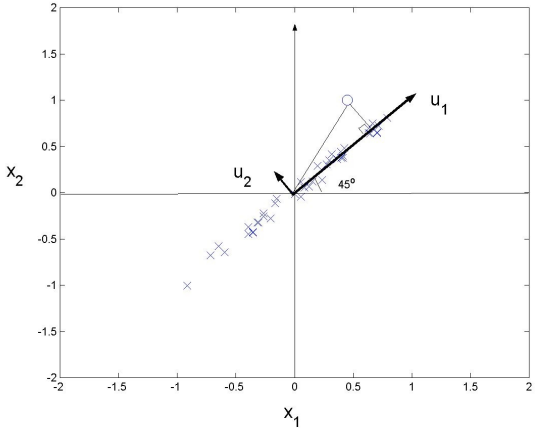
Principal Components Analysis - Key idea

- Intuitively, for a given component, if the data has large variability between examples, this component tends to be important, in the sense that it more likely determines the output value.
- On the other hand, if the data takes roughly the same or similar values for all examples, then this component is unlikely to have a significant influence on outputs, and hence can be neglected.
- In other words, it is sensible **to choose those components having large variability as important features** if we do not have other prior knowledge.

How do we measure variability? We compute the variance!

UNIVERSITY OF SUSSEX

Example: Principal component



The most important feature of the data is given by the projection of the data onto the principal component, which in this example is the direction:

$$u_1 = \frac{x_1 + x_2}{\sqrt{2}}$$

The projection on the other direction u_2 can be ignored.

UNIVERSITY OF SUSSEX

PCA: the covariance matrix holds all the information

$$\Sigma = \begin{pmatrix} var(x_1) & cov(x_1, x_2) & \dots & cov(x_1, x_D) \\ cov(x_1, x2) & var(x_2) & \dots & cov(x_2, x_D) \\ \vdots & \vdots & \dots & \dots \\ cov(x_1, x_D) & cov(x_2, x_D) & \dots & var(x_D) \end{pmatrix}$$

The idea is to:

1. Change coordinates (rotate axes), so that the off-diagonal covariance terms are zero (new features are uncorrelated)
2. Rank the variances of the newly defined features in the new coordinate system
3. The selected features are those corresponding to the largest d variances, where d is how many features are to be kept.

UNIVERSITY OF SUSSEX

Eigenvalues and eigenvectors

- Let M be a square matrix. Let λ be a constant and e a non-zero vector. λ is an eigenvalue of M and e is the corresponding eigenvector if:

$$Me = \lambda e$$

- In geometry, if M represents a transformation of the space (e.g., a rotation, reflection or enlargement) then an eigenvector is a vector which is unchanged under the transformation (e.g., the axis of rotation or reflection) and the eigenvalue is the scaling factor.

UNIVERSITY OF SUSSEX

Example

$$\boldsymbol{M} = \begin{pmatrix} 1 & \frac{3}{4} \\ \frac{3}{4} & 1 \end{pmatrix}$$

Is $\boldsymbol{v} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ an eigenvector?

$$\boldsymbol{M}\boldsymbol{v} = \begin{pmatrix} 1 & \frac{3}{4} \\ \frac{3}{4} & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \cdot 1 + \frac{3}{4} \cdot (-1) \\ \frac{3}{4} \cdot 1 + 1 \cdot (-1) \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ -\frac{1}{4} \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{4}\boldsymbol{v} = \lambda \boldsymbol{v}$$

Yes, and the corresponding eigenvalue is $\lambda = \frac{1}{4}$.

UNIVERSITY OF SUSSEX

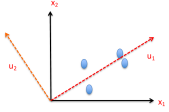
Orthogonal eigenvectors, diagonal matrix

If $\boldsymbol{A}$ is symmetric (true if A is the covariance matrix), the eigenvectors u will be orthogonal (all perpendicular to each other).
Then, the eigenvectors can be used to define new features.

Expressed in the new coordinate system, the matrix becomes diagonal, with the eigenvalues down the diagonal:

$$\boldsymbol{A}' = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \lambda_D \end{pmatrix}$$

The data is rewritten: Instead of features x_1 and x_2, create a table of new features u_1 and u_2.



UNIVERSITY OF SUSSEX

Covariance matrix

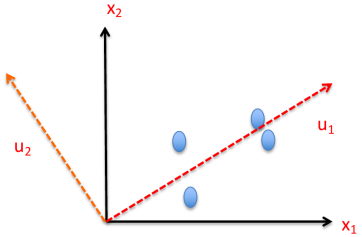
$$\Sigma = \begin{pmatrix} var(x_1) & cov(x_1, x_2) & \dots & cov(x_1, x_D) \\ cov(x_1, x2) & var(x_2) & \dots & cov(x_2, x_D) \\ \vdots & \vdots & \dots & \dots \\ cov(x_1, x_D) & cov(x_2, x_D) & \dots & var(x_D) \end{pmatrix}$$

$$\Sigma_{ii} = \frac{1}{N} \sum_{n=1}^{N} (x_{i,n} - \overline{x}_i)^2 \qquad \Sigma_{ij} = \frac{1}{N} \sum_{n=1}^{N} (x_{i,n} - \overline{x}_i)(x_{j,n} - \overline{x}_j)$$

Diagonal terms are the variance of each feature.
Off-diagonal terms are the covariances between pairs of features.
Covariance is a measure of correlation.

UNIVERSITY OF SUSSEX

Rotated feature space with no correlations



Eigenvalues of the covariance matrix are equal to the variances of redefined set of features that are uncorrelated

$$\Sigma = \begin{pmatrix} var(x_1) & cov(x_1, x_2) & \dots & cov(x_1, x_D) \\ cov(x_1, x2) & var(x_2) & \dots & cov(x_2, x_D) \\ \vdots & \vdots & \dots & \dots \\ cov(x_1, x_D) & cov(x_2, x_D) & \dots & var(x_D) \end{pmatrix}$$

$$\downarrow$$

$$\Sigma' = \begin{pmatrix} var(u_1) & 0 & \dots & 0 \\ 0 & var(u_2) & \dots & 0 \\ \vdots & \vdots & \dots & \dots \\ 0 & 0 & \dots & var(u_D) \end{pmatrix}$$

UNIVERSITY OF SUSSEX

Steps of PCA

1. Compute the covariance matrix of the input data.
2. Compute the eigenvalues and eigenvectors of the covariance matrix.
3. Arrange eigenvectors in the order of magnitude of their eigenvalues.
   - Take the first d eigenvectors as principal components if the input dimensionality is to be reduced to d.
   - Or look at the spectrum of eigenvalues to decide how many components to keep. Each eigenvalue is the variance of the corresponding eigenvector (component).
4. Project the input data onto the principal components, which forms the representation of the input data.

UNIVERSITY OF SUSSEX

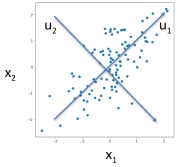
Example

Covariance matrix:

$$\begin{pmatrix} 1 & \frac{3}{4} \\ \frac{3}{4} & 1 \end{pmatrix}$$

We saw that one eigenvector was $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ with eigenvalue $\frac{1}{4}$.

Exercise:
Show that the other eigenvector is $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ with eigenvalue $\frac{7}{4}$.



Use these eigenvectors to define new features (scaled to length 1):

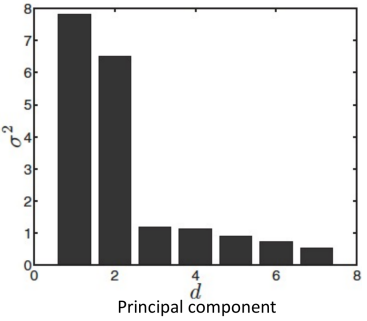
$$u_1 = \frac{x_1 + x_2}{\sqrt{2}}, \qquad u_2 = \frac{x_1 - x_2}{\sqrt{2}}$$

u_1 has a large variance $\frac{7}{4}$, whereas u_2 has a smaller variance $\frac{1}{4}$.
Therefore, could try keeping just u_1, the principal component of the data.

UNIVERSITY OF SUSSEX

How many components to keep?

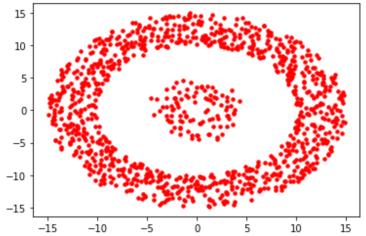
- Look at the spectrum of eigenvalues to decide how many components to keep. Each eigenvalue is the variance of the corresponding eigenvector (component).
- Or try sequential forward search.
  - Sequentially select features until there is no improvement in prediction.
  - Select 1 feature. Train and test model.
  - Add another feature. Train and test model.
  - Iterate until there is no improvement in prediction.



(Rogers & Giolami)

Does PCA always work well?

No!
When the relationship between components is very non-linear, it doesn't work well.

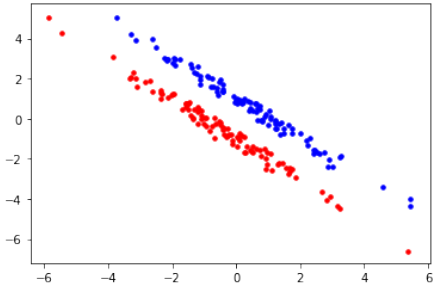


Here, PCA will overestimate the input dimensionality: there is equal variance in all directions here, so PCA would consider these data to be 2 dimensional. But in reality these are 1 dimensional data- the only actual variable here is the angle on the circle (a non-linear function of x and y coordinates).

UNIVERSITY OF SUSSEX

Other cases when PCA fails

In cases when components with small variability really matter, PCA will make mistakes due to the unsupervised nature of the process.

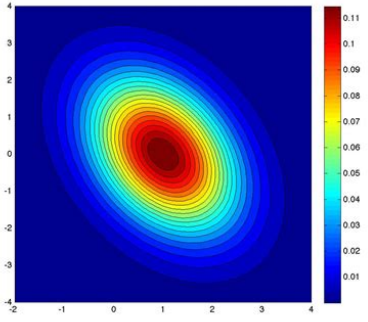


In this example, if we only consider the projections of two classes of data as input, the two classes become indistinguishable.

UNIVERSITY OF SUSSEX

When does PCA work best?

For Gaussian distributed data,
i.e. when the inputs have a normal (Gaussian) distribution.
In that case, the eigenvalues of the covariance matrix indeed quantify the variability of data along their corresponding directions. (There are no other parameters to the probability distribution.)



UNIVERSITY OF SUSSEX

Pre-processing before doing PCA

Input normalization:

- Must subtract the mean as the theory requires that the data are centred at the origin
- Also, we divide by the standard deviation as results shouldn't depend on the units of the inputs.

Summary and outlook

- **Input normalisation**: aids visualisation, generally improves performance of model training.
- **Dimensionality reduction / Feature selection / Feature extraction**:
  - **Reduces overfitting**: less redundant data means less opportunity to make decisions based on noise.
  - **Improves accuracy**: one feature that combines several noisy, yet correlated, features means modelling accuracy improves.
  - **Reduces training time**: reducing number of features reduces algorithm complexity and algorithms train faster.
  - PCA also good for visualising data.

Next lecture:

- Multi-layer perceptron

UNIVERSITY OF SUSSEX

EXTRA SLIDES: Example: Face recognition

The dimensionality of a face image is extremely high.

Suppose we describe a face image by a *MxM* two-dimensional grid. The dimensionality of the input vector is then *M^2.* Consider, for example, *M=256* is needed to achieve a reasonable precision of the image, the dimensionality of the input vector is then 65,536!

Face images have structure.

If we put all face images in a *M^2* dimensional space, they will not fully fill the whole space, but will instead only cover a very limited volume. This implies many input components are correlated, and the dimensionality can be significantly reduced.

Eigenface: PCA for face images

Use PCA to identify non-trivial, global features of face images.

Each face image is represented as a *Mx1* column vector Φ, (M=6400 in our example).

Calculate the average face $\overline{\Phi} = \frac{1}{N}\sum_{n=1}^{N}\Phi_n$

Calculate the covariance matrix of data (6400x6400)

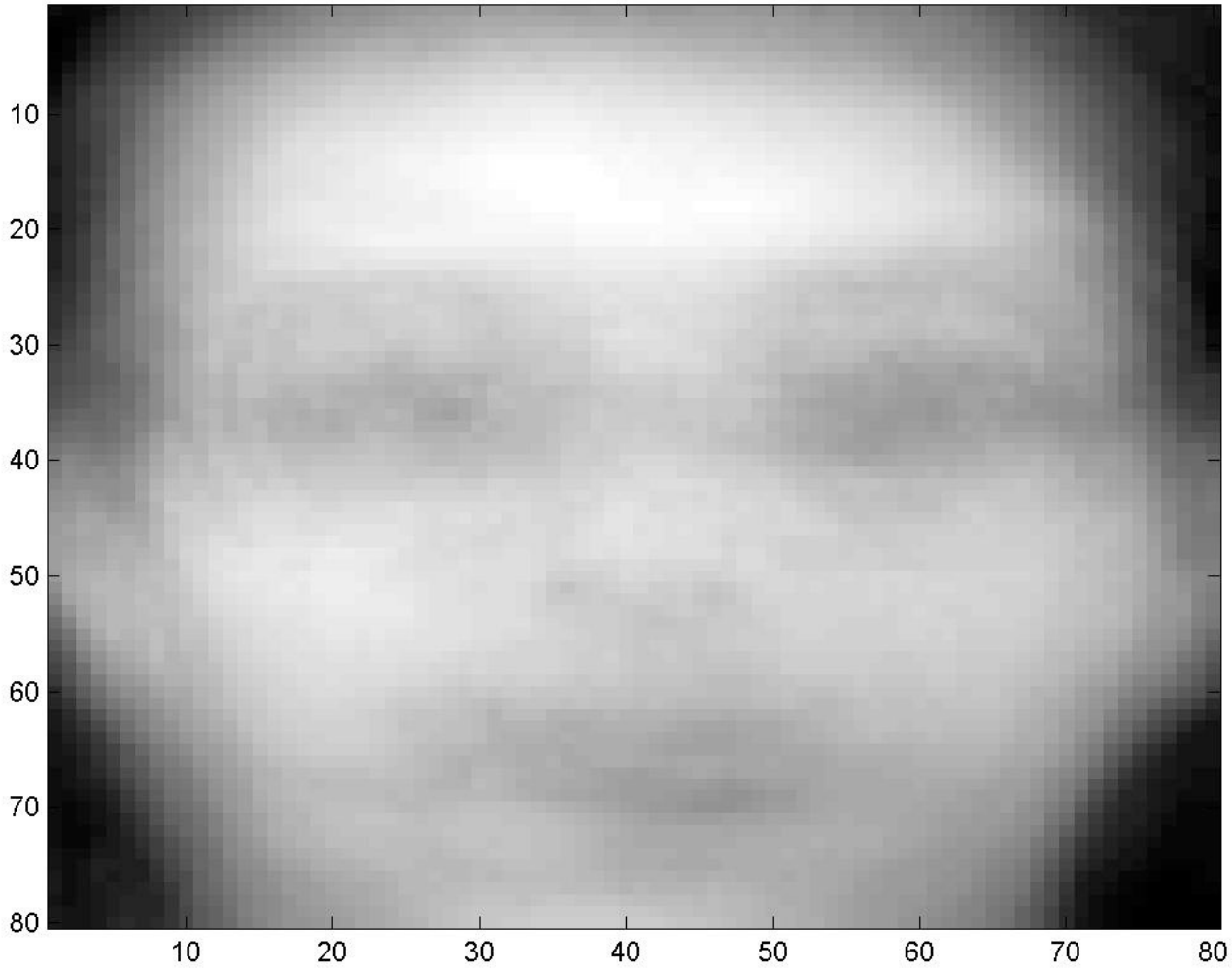
$$\Sigma = \frac{1}{N}\sum_{n=1}^{N}(\Phi_n - \overline{\Phi})(\Phi_n - \overline{\Phi})^T$$

Choose *d* eigenvectors with the first *d* largest eigenvalues as the principal components, which we call the eigenfaces.

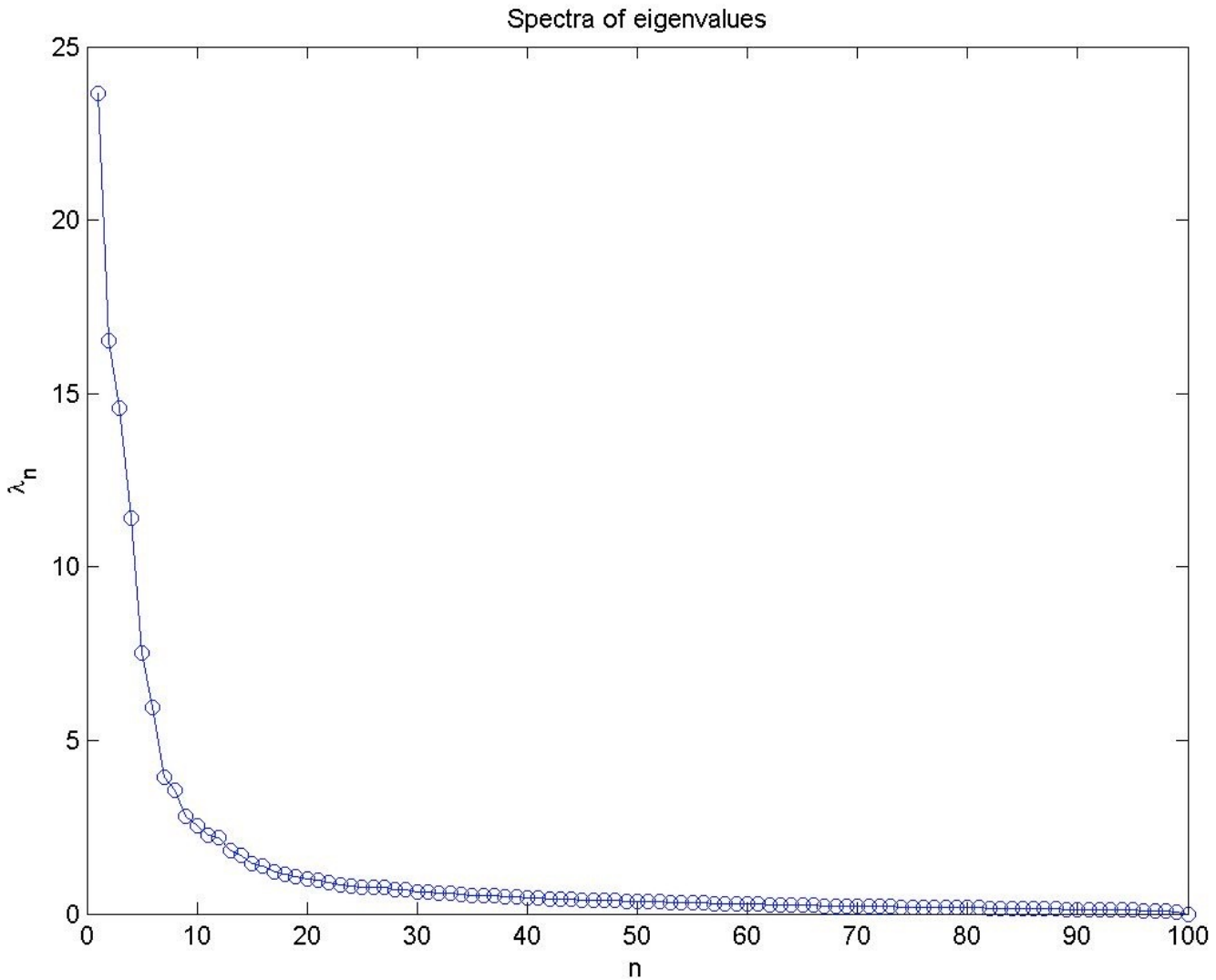
Project face images on the eigenfaces, getting new representations of data with the reduced dimensionality.

Examples of face images

The average face



The spectrum of eigenvalues



The first nine eigenfaces