Machine Learning course - Chapter 7

# Dimensionality and its Reduction

*(First part)*

09-03-16
Laura Hangard

# The curse of dimensionality

- Purchase of a car -> all the cars match : 1
- Criterion n°1 : fast -> fraction of matching cars : $r < 1$
- Add criteria for the perfect car (hyp. same proba for each):
    - red -> r
    - 8 cylinders -> r
    - Leather interior -> r
- => Proba of finding your ideal car is : $r*r*r*r = r^4 << 1$
- The more selection conditions, the tinier is the chance of finding your ideal car.

# The curse of dimensionality

- Selection conditions of the car ~ the dimensions in your dataset. Effect known as “curse of dimensionality”.

- Impacts of this effect:

    - size of data required to constrain a model

    - complexity of the model

    - search time to optimise the model

# The curse of dimensionality

- Let’s quantitatively find out how much data is needed to estimate the proba of finding points within a high-dimensional space:

We consider N points from a D-dimensional *uniform distribution* (hypercube centred at the origin, edge length 2).

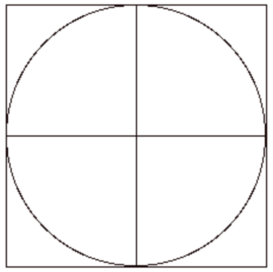
—> What proportion of points falls within a unit distance of the origin?

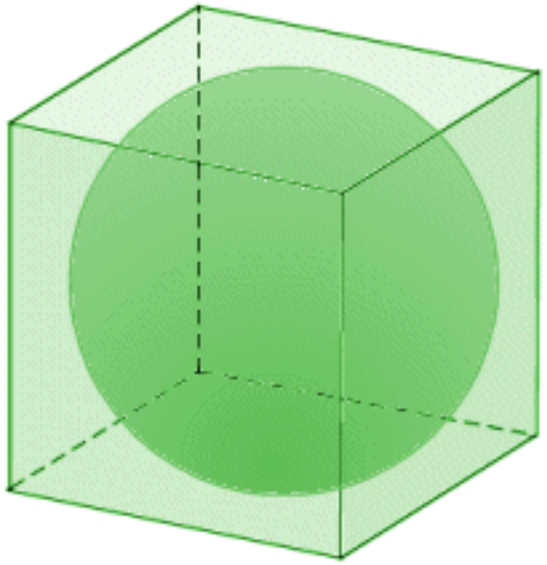
$$\text{ratio} = \frac{\text{volume of a unit hypersphere centred at the origine}}{\text{volume of the side-length=2 hypercube centred at the origine}}$$

2D : $f_2 = \frac{\pi r^2}{(2r)^2} = \pi/4 \approx 78.5\%$

3D : $f_3 = \frac{(4/3)\pi r^3}{(2r)^3} = \pi/6 \approx 52.3\%$

D dimensions : $f_D = \frac{V_D(r)}{(2r)^D} = \frac{\pi^{D/2}}{D 2^{D-1}\Gamma(D/2)} \longrightarrow 0$





The number of points in a data set required to evenly distribute points in the hypervolume will grow exponentially with dimension.

# The curse of dimensionality

- Context of astronomy : SDSS dataset = 357 million sources, each having 448 measured parameters (flux, size, position…)

- With only 30 of those parameters, the proba of 1 source residing in our previous hypersphere is ~$10^{-6}$ !

- It is difficult to characterize structures in high dimensional datasets.

- Adding to that, so far we have assumed that all dimensions are equally weighted. **Not true** in real life.

—> It is possible to find projections within the data that capture the principal physical and statistical correlations between measured quantities *(intrinsic dimensionality)*.

# The curse of dimensionality

<u>This is the challenge of this chapter :</u>

- finding the accurate dimensions

- and thereby reducing the dimensionality of the data.

# The data sets used in this chapter

- Spectrum = x(λ) : sampled at D discrete flux values, so it can be written as a D-dimensional vector. So a spectrum can be represented as a point in a high dimensional space.

- But this also works for a D = N * K image, which can be represented as a vector with D elements, so a point in a D-dimensional space.

- More straight forward, it is the same also for catalogues of data.

# The data sets used in this chapter

- Here we will use SDSS galaxy spectra : 3200 to 7800 Å divided in 1000 wavelength bins

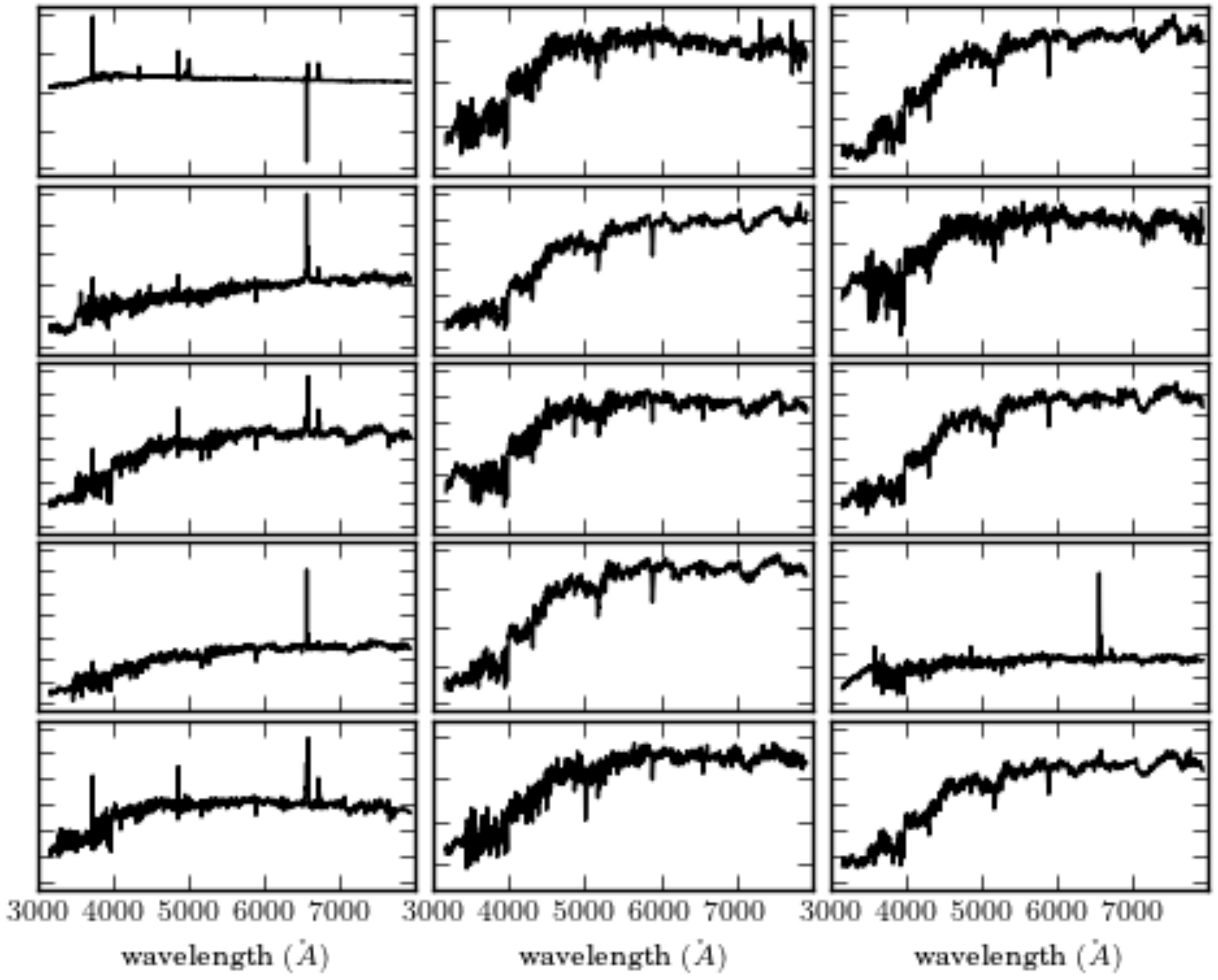


*Figure 7.1:*
*A sample of 15 galaxy spectra selected from the SDSS spectroscopic data set. These spectra span a range of galaxy types, from star-forming to passive galaxies. Each spectrum has been shifted to its rest frame and covers the wavelength interval 3000-8000 Angstroms. The specific fluxes, on the ordinate axes have an arbitrary scaling.*

# Principal Component Analysis

- Example: distribution of points from bivariate Gaussian centred at the origine of x and y.
- Points clearly correlated along a particular direction but this direction does not align with the axes.
- To reduce the nb of features (axes) and have a more compact representation describing the data, we should rotate the axes to align with the correlation.
- We choose a rotation that will maximise the variance along the new axes:
    - 1st axis = direction with maximal variance
    - 2nd axis = orthogonal to the 1st, that maximises the residual variance
    - (…and so on for more dimensions)

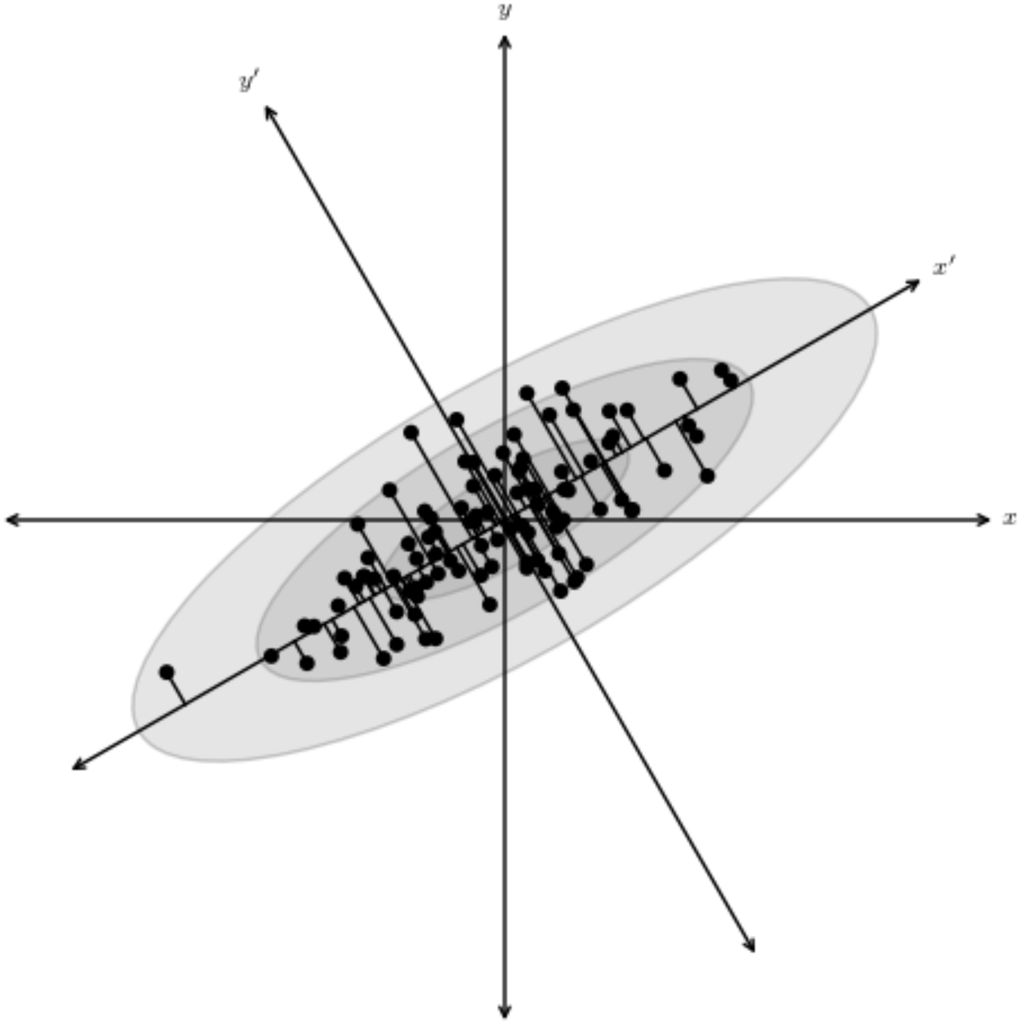


*Figure 7.2:*
*A distribution of points drawn from a bivariate Gaussian and centered on the origin of x and y. PCA defines a rotation such that the new axes (x' and y') are aligned along the directions of maximal variance (the principal components) with zero covariance. This is equivalent to minimizing the square of the perpendicular distances between the points and the principal components.*

# Principal Component Analysis

This dimensional reduction technique is what we call PCA, and is the most applied dimensionality reduction technique:

**A linear transform applied to multivariate data, that defines a set of uncorrelated axes (=the principal components), ordered by variance.**

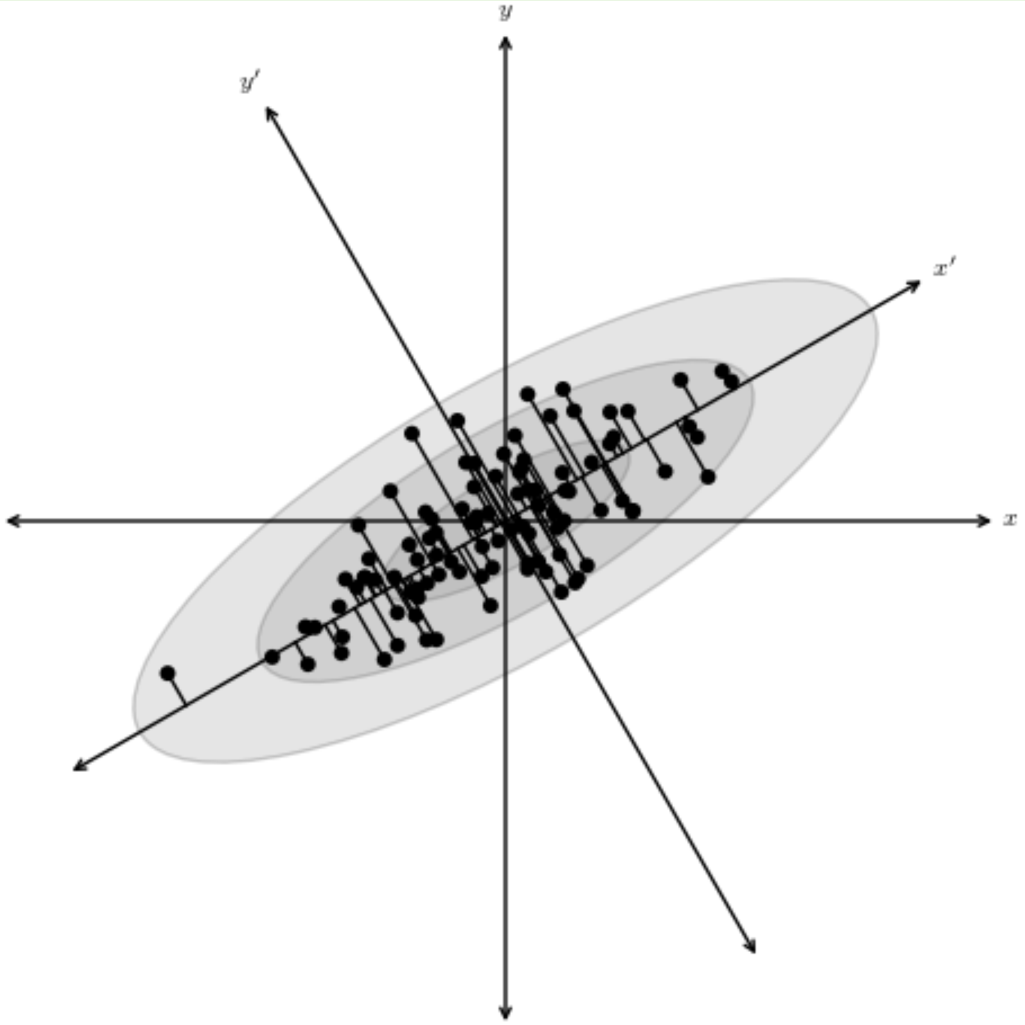
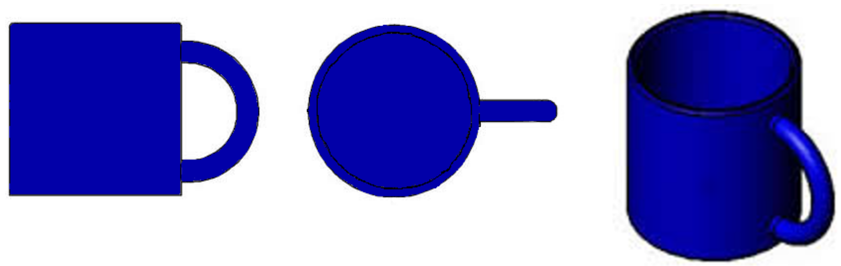


*Figure 7.2:*
*A distribution of points drawn from a bivariate Gaussian and centered on the origin of x and y. PCA defines a rotation such that the new axes (x' and y') are aligned along the directions of maximal variance (the principal components) with zero covariance. This is equivalent to minimizing the square of the perpendicular distances between the points and the principal components.*

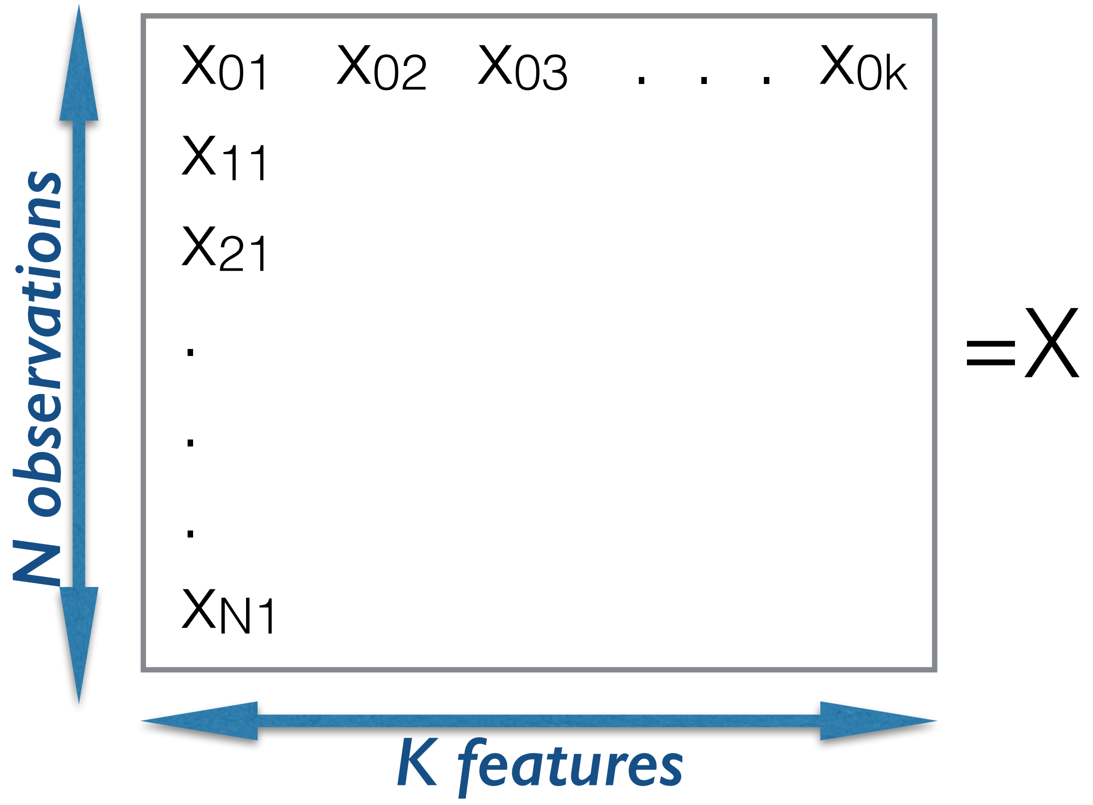


# Principal Component Analysis

## How to derive PCA?

- Set of data $\{x_i\}$ of N observations, each observations has K measured features.
- Initially center the data by subtracting the mean of each feature in $\{x_i\}$, and then write this N * K matrix as X.
- The covariance of the centred data is:

$$C_X = \frac{1}{N-1} X^T X$$



$$\begin{bmatrix} x_{01} & x_{02} & x_{03} & . \; . \; . & x_{0k} \\ x_{11} & & & & \\ x_{21} & & & & \\ . & & & & \\ . & & & & \\ . & & & & \\ x_{N1} & & & & \end{bmatrix} = X$$

Nonzero off-diagonal components in this covariance matrix arise if there exist correlations between the measured features.

# Principal Component Analysis

## How to derive PCA?

PCA wants to identify R, projection of $\{x_i\}$ aligned with the directions of maximal variance.

This projection is written Y = X R, and its covariance is:

$$C_Y = R^T X^T X R = R^T C_X R$$

Let's consider $r_1$ the first component of R, ie. the projection with maximum variance (constraint : $r_1^T r_1 = 1$ *(rotation?)*). We derive it by using Lagrange multipliers and defining the cost function $\phi(r_1, \lambda)$ as:

$$\phi(r_1, \lambda_1) = r_1^T C_X r_1 - \lambda_1 (r_1^T r_1 - 1)$$

# Principal Component Analysis

## How to derive PCA?

- Then the derivative of ϕ($r_1$, λ) with respect to r1 is set to zero, which gives:

$$C_X r_1 - \lambda_1 r_1 = 0$$

- From that we deduce that $\lambda_1$ is the root of the equation :

$$\det(C_X - \lambda_1 \mathbf{I}) = 0$$

which means that $\lambda_1$ is an eigenvalue of the covariance matrix $C_X$.

- The variance for the principal component $r_1$ is maximised when $\lambda_1$ is the largest eigenvalue of the covariance matrix, with :

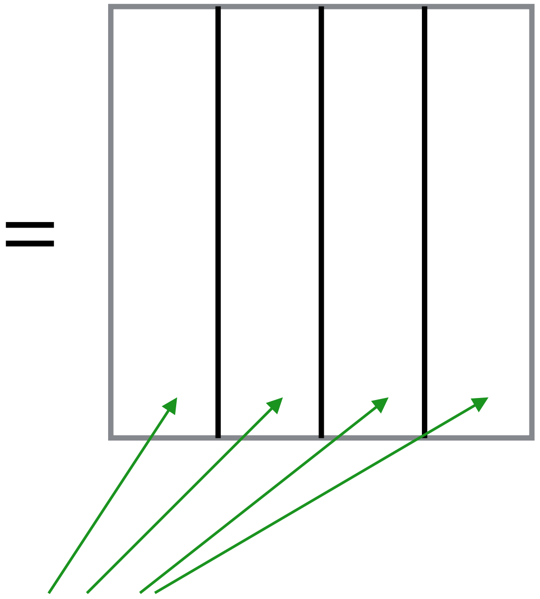
$$\lambda_1 = r_1^T C_X r_1$$

- Second (and further) principal component similarly derived but with additional constraint that it is **uncorrelated** with $r_1$: $r_2^T C_X r_1 = 0$

# Principal Component Analysis

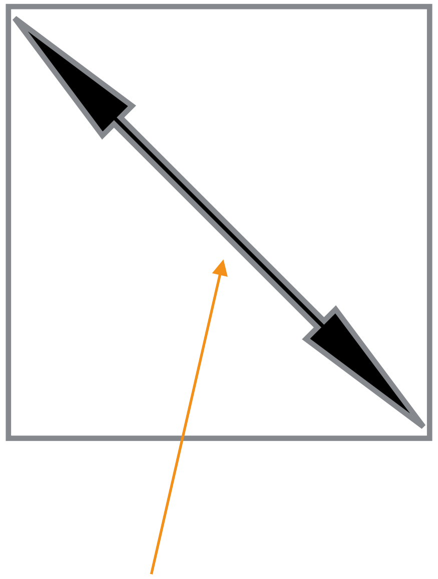
## How to derive PCA?

$R =$



eigenvectors, or principal components

$C_Y =$



diagonal values are the amount of variance in each component

We have: $C_X = R C_Y R^T$

and if we order the eigenvectors by their eigenvalue, we can define the set principal components for X.

# Principal Component Analysis

## The application of PCA

- To form the data matrix X, the data vectors are centred by subtracting the mean of each dimension.
- In addition, the data are often preprocessed in order to ensure that PCA is maximally informative.

*Example: galaxy shape and flux = heterogeneous data -> columns preprocessed by dividing by their variance.*

This ensures that the variance of each feature is comparable, leading to a more physically meaningful set of principal components.

*(Note: for images or spectra, common processing step is to normalise each row to have the integrated flux of each object being 1. This helps removing uninteresting correlations based on the overall brightness of the spectrum or image.)*

# Principal Component Analysis

## The application of PCA

Case of our spectra : normalised to a constant total flux + centred to have zero mean (upper left panel).

Left panel : first four “eigenspectra”, ordered by their eigenvalues.

*(As a vector can be represented by the sum of its components, a spectrum will be represented by the sum of its eigenspectra)*

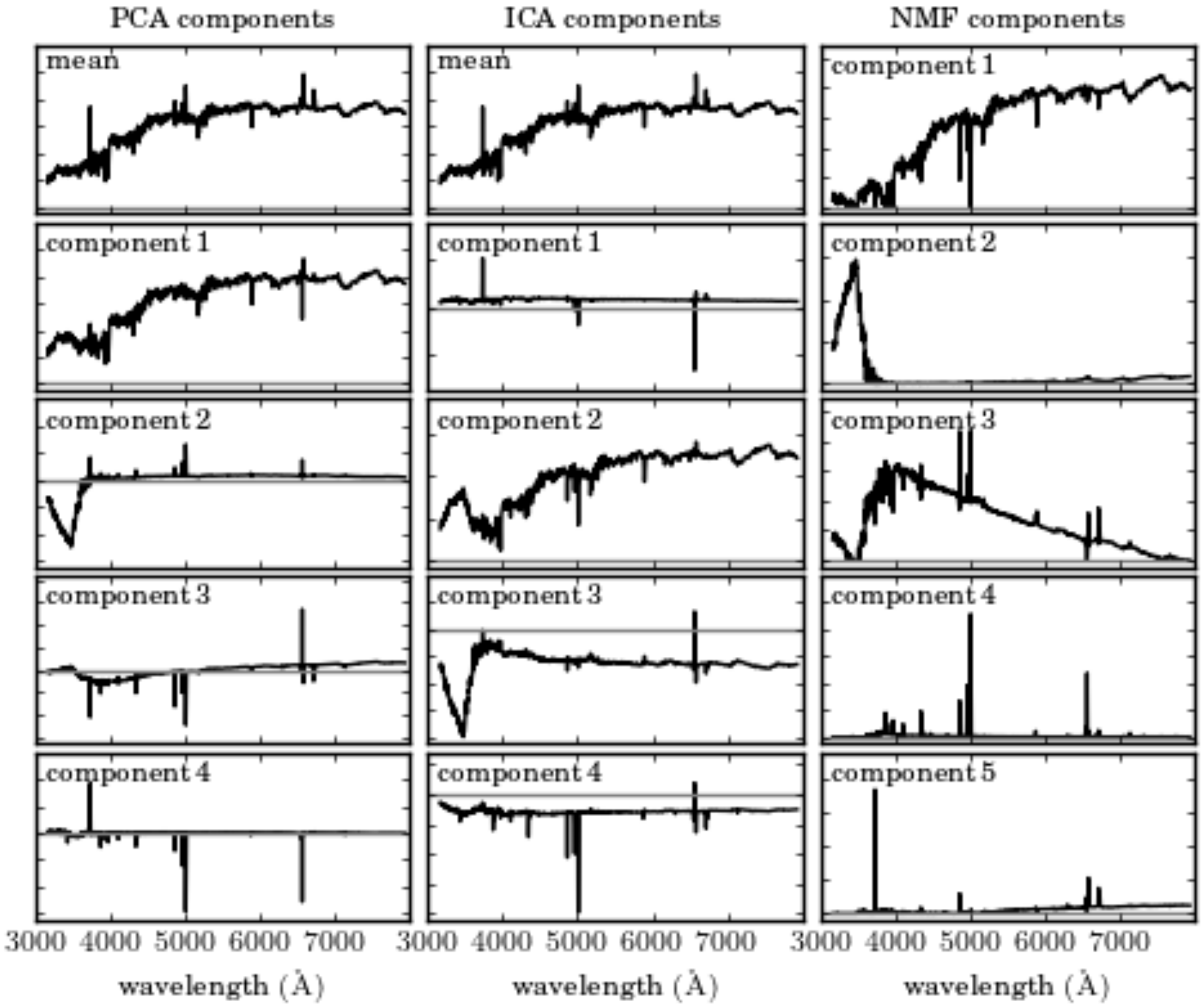


*Figure 7.4:*
*A comparison of the decomposition of SDSS spectra using PCA (left panel), ICA (middle panel) and NMF (right panel). The rank of the component increases from top to bottom. For the ICA and PCA the first component is the mean spectrum (NMF does not require mean subtraction). All of these techniques isolate a common set of spectral features (identifying features associated with the continuum and line emission). The ordering of the spectral components is technique dependent.*

# Principal Component Analysis

## The application of PCA

The “scree plot” shows the amount of variance contained within each eigenspectra (by showing the associated eigenvalues).

With constraint: sum of the eigenvalues = total variance of the system.

- 94% of the variance in the SDSS spectra is captured by the first 10 eigenvectors!

➡ If we project each spectrum onto those 10 eigenspectra, 94% of the information in each spectrum is conserved.

- 6% loss of information, but dimensionality reduced from 1000 to 10 !

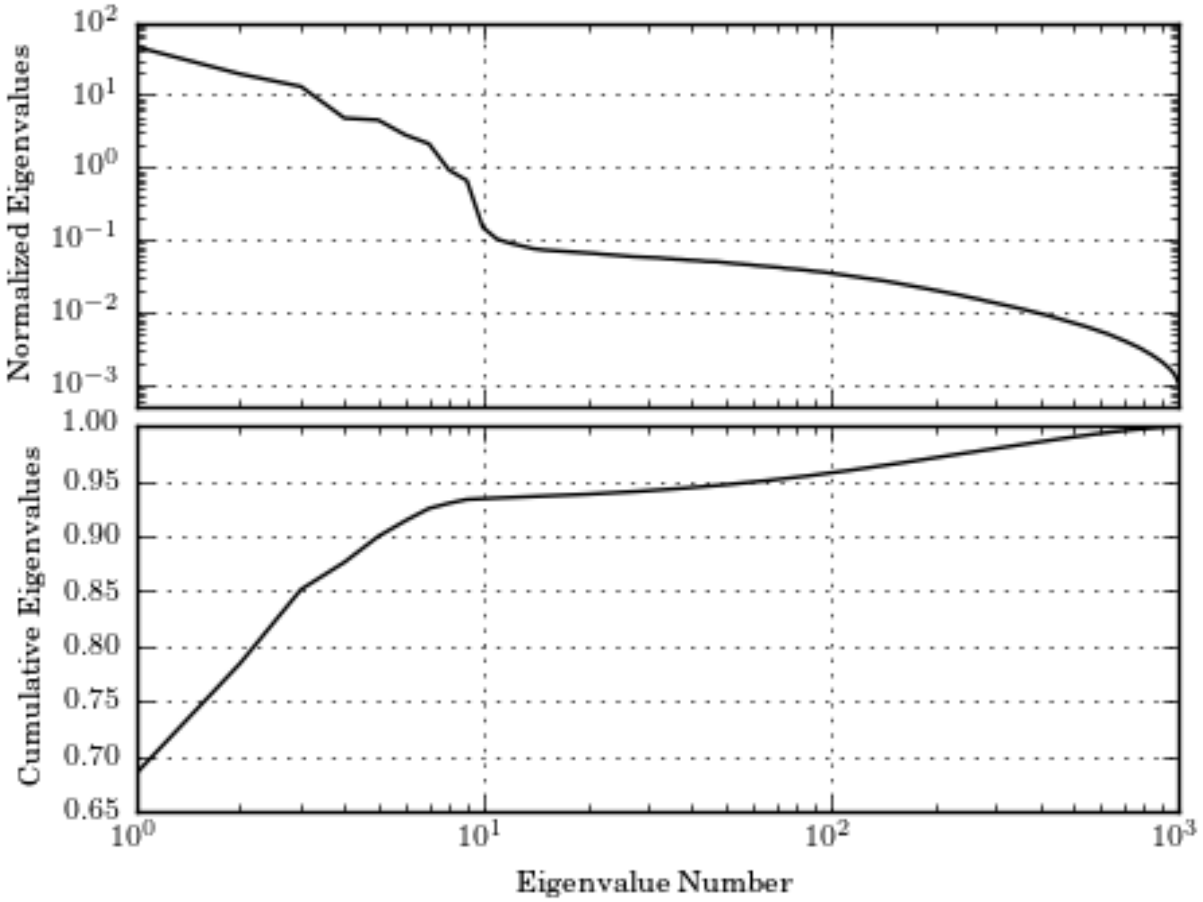


*Figure 7.5:*
*The eigenvalues for the PCA decomposition of the SDSS spectra. The top panel shows the decrease in eigenvalue as a function of the number of eigenvectors, with a break in the distribution at ten eigenvectors. The lower panel shows the cumulative sum of eigenvalues normalized to unity. 94% of the variance in the SDSS spectra can be captured using the first ten eigenvectors.*

# Principal Component Analysis

## The application of PCA

Reconstruction of a spectrum x(k) from the eigenbasis e$_i$(k). Each spectrum x$_i$(k) is described by:

$$\boldsymbol{x}_i(k) = \boldsymbol{\mu}(k) + \sum_{j}^{R} \theta_{ij}\,\boldsymbol{e}_j(k)$$

total nb of eigenspectra

eigenvector

mean spectrum

the nb of the eigenspectrum

linear expansion coefficients:

$$\theta_{ij} = \sum_{k} \boldsymbol{e}_j(k)(\boldsymbol{x}_i(k) - \boldsymbol{\mu}(k))$$

- If you take R as **all** of the eigenvectors, the input spectrum is fully described with no loss of information.

- If you sum over **r < R** then you exclude the eigenvectors with smaller eigenvalues (mostly noise).

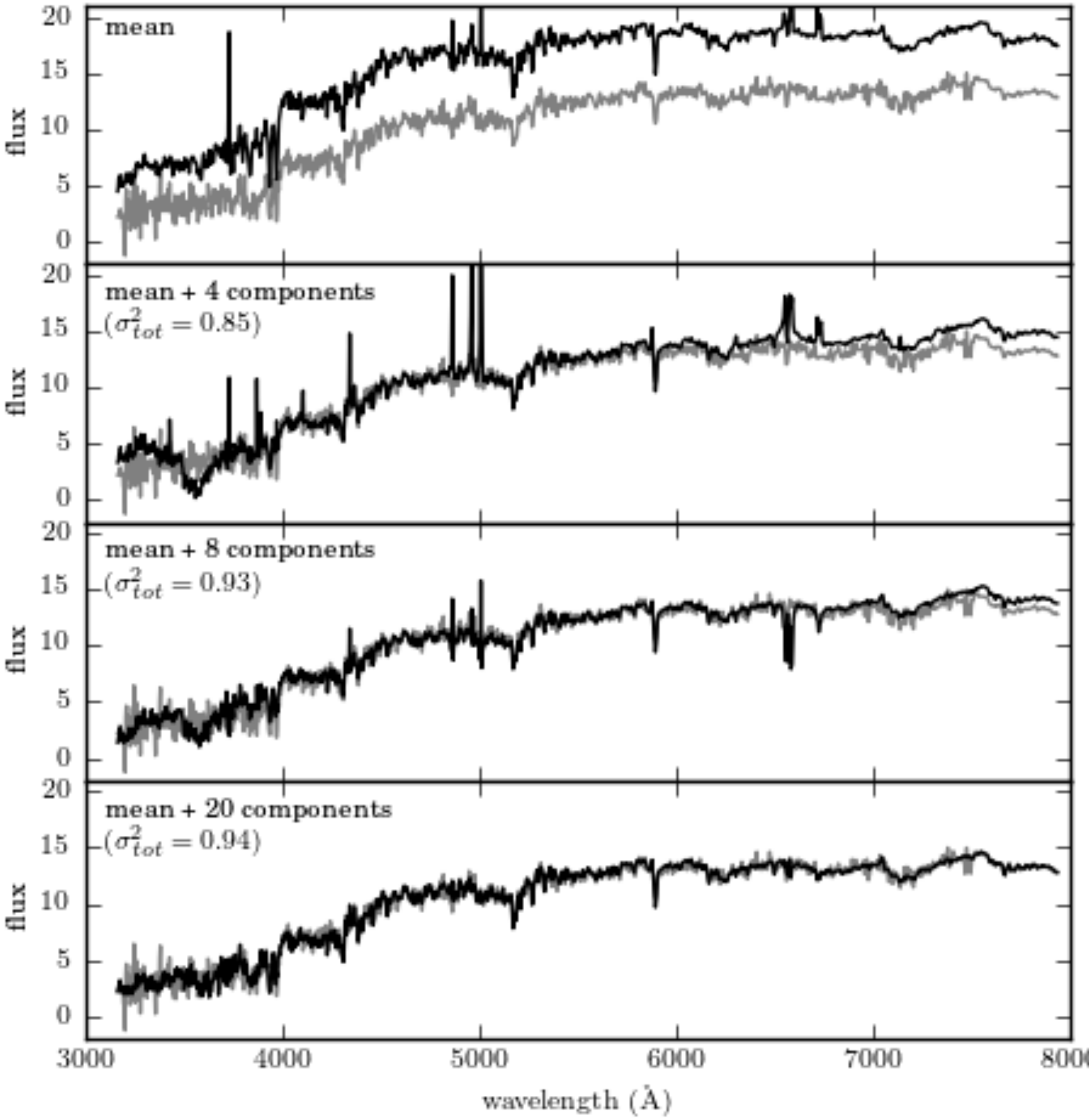


*Figure 7.6:*

*The reconstruction of a particular spectrum from its eigenvectors. The input spectrum is shown in gray, and the partial reconstruction for progressively more terms is shown in black. The top panel shows only the mean of the set of spectra. By the time 20 PCA components are added, the reconstruction is very close to the input, as indicated by the expected total variance of 94%.*

# Principal Component Analysis

## The application of PCA

/!\ You **must** be **careful** when truncating at a small number of components! It’s a very important point of the reconstruction of the data set:

- Too many components —> introducing noise
- Too few —> missing physical correlation within the data

Let’s define $\alpha$, a bound on the fraction of the variance we want to capture (and $\sigma_i$ being the eigenvalues):

$$\frac{\sum_i^{i=r} \sigma_i}{\sum_i^{i=R} \sigma_i} < \alpha$$

Typical values for $\alpha$ range from 70% to 95%, but you should choose the threshold depending on the “scree plot”. No generic solution, it will depend for each case.

# Hands-on

- Apply PCA to new datasets *(www.astroml.org/examples/datasets/index.html)*

- Make the different plots seen in the chapter for the new datasets, and interpret *(figures 7.1 - 7.4 - 7.5 - 7.6)*