Data Science for Geoscience 2020

Lucas Drumetz
lucas.drumetz@imt-atlantique.fr

Data Science for Geoscience 2020, Toulouse
Introductory lecture: Statistics and probability, Data exploration

Outline

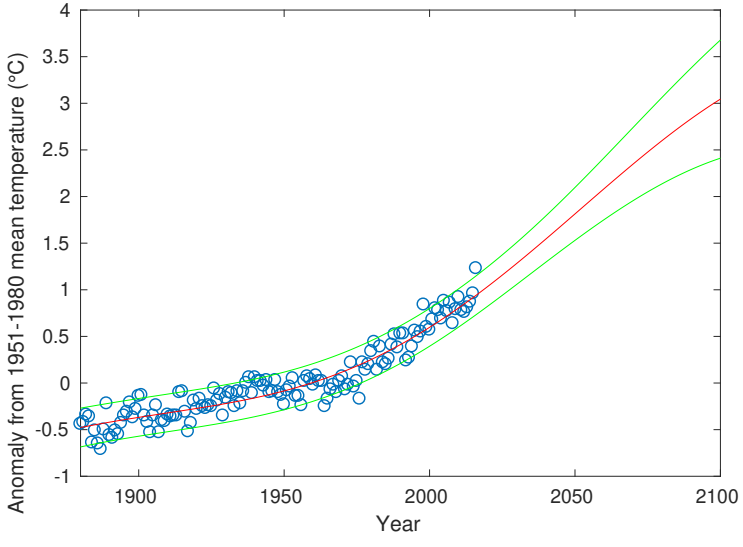
Datasets

A dataset can be seen as a collection of N realizations of a D-dimensional random variable

$$\mathbf{X} \in \mathbb{R}^{D \times N}$$

Examples:

- Collection of grades among a population of N students
- The N pixels of an image (gray-level $D = 1$ or color $D = 3$)
- A time series of data (e.g. the average temperature at a given location over time, N is the number of time samples)
- A time series of images (here D can be the number of pixels and N the number of time samples)



Data description

How can we describe univariate data (D=1) other than by a collection of values?

- Are certain values which are more probable than others?
- Are the values centered on a particular value?
- What is the spread around that value ?
- Is the data symmetric around the central value?
- Are there values that are very different from most others?

Ideally, we should know the probability density function (pdf)

$$p(x)$$

$\rightarrow$ provides all the necessary information

Descriptive Statistics

- Estimate the pdf from the data
- Compute quantities to try and answer the previous questions

Statistical Independence

Two random variables are independent if $p(x, y) = p(x)p(y)$

Expected value of a (continuous) random variable

$$\mu = \mathbb{E}[x] = \int_{x \in \mathbb{R}} x p(x) \mathrm{d}x$$

Variance of a random variable

$$\sigma^2 = \mathrm{Var}(x) = \mathbb{E}[(x - \mathbb{E}[x])^2] = \mathbb{E}[x^2] - \mathbb{E}[x]^2$$

We also often use the standard deviation $\sigma = \sqrt{\sigma^2}$

Mean and variance estimation

The expectation is approximated by the empirical mean

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

such that if the x_i all follow the same distribution and are drawn independently:

$$\mathbb{E}[\hat{\mu}] = \mu \qquad \mathrm{Var}(\hat{\mu}) = \frac{\sigma^2}{N} \underset{N \to +\infty}{\longrightarrow} 0$$

$\rightarrow$ The empirical mean is an estimator of the mean which is *unbiaised* and *consistent*

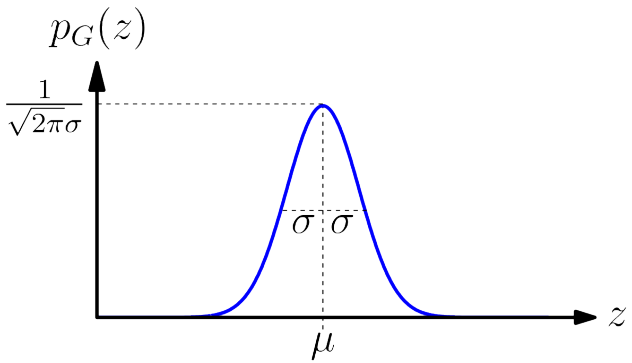
The variance of the data is approximated by the empirical variance:

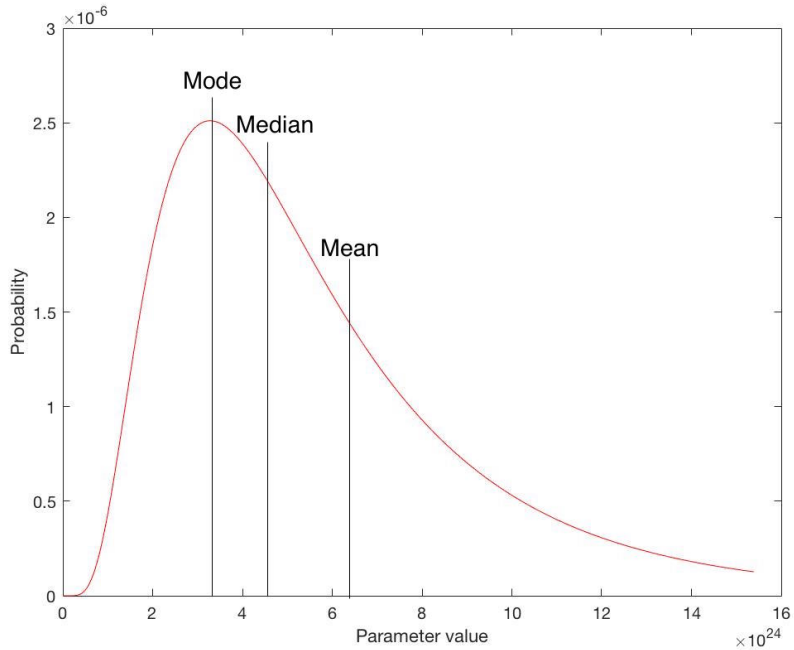
$$\hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \hat{\mu})^2$$

Other statistical descriptors

- Mode of the distribution: $\arg\max_x p(x)$
- Median: value that separates the data into two subsets that are equiprobable
  $\rightarrow$ empirical estimation: value that splits the datasets into two subsets with the same number of samples

  How is that different from the mean?

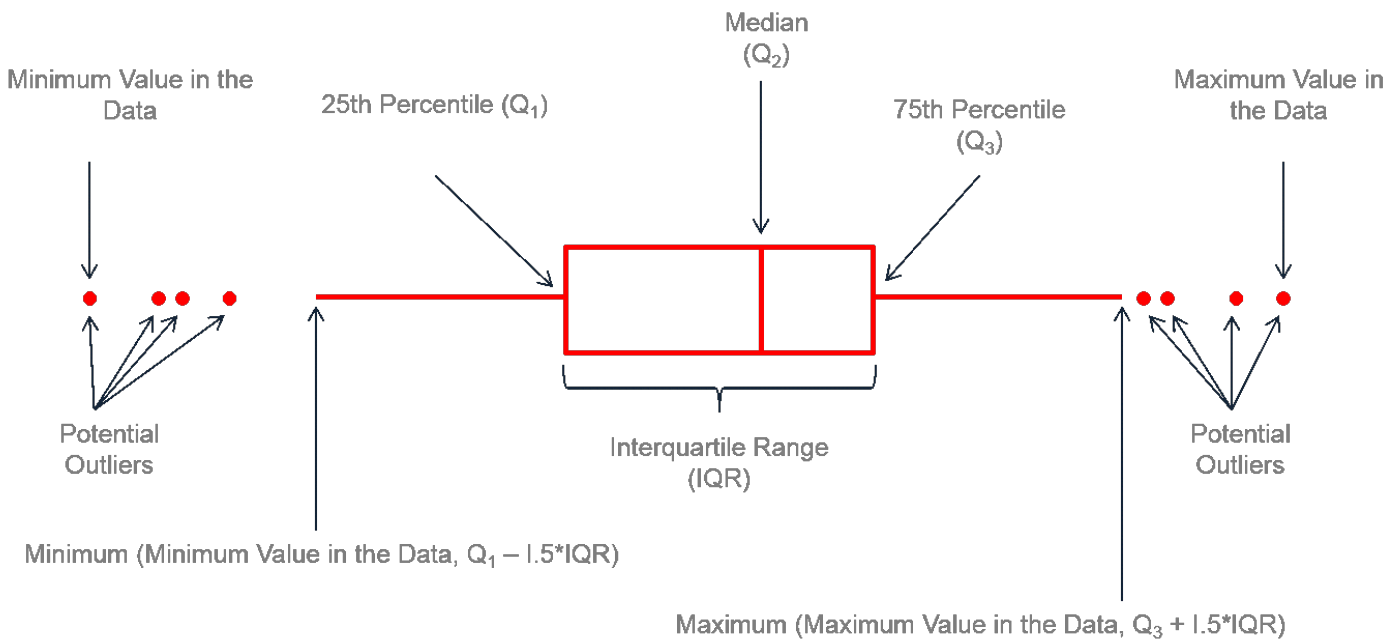




Other statistical descriptors (cont'd)

- Quantiles: generalization of the median
  $\rightarrow$ the q-quantiles divide the domain into q regions, all with probability $1/q$ (or $\mathrm{floor}(N/q)$ samples)

  Box plots use the median and the other quartiles (4-quantiles)



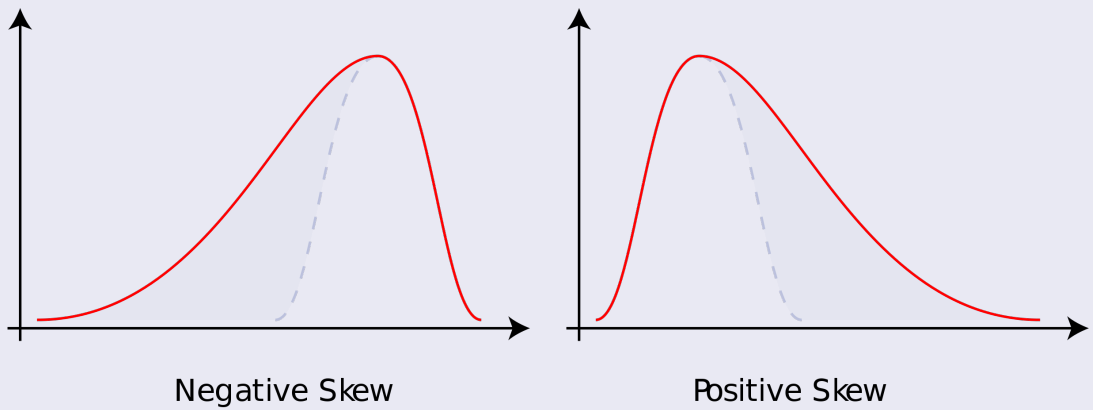
Other statistical descriptors (cont'd)

- Higher order moments: related to $\mathbb{E}[x^k]$, $k > 2$
  - 3rd order: Related to the assymetry of the distribution

Skewness

Skewness is a measure of symmetry of the distribution:

$$\gamma = \mathbb{E}\left[\left(\frac{x-\mu}{\sigma}\right)^3\right]$$



$\gamma = 0$: symmetric distribution, $\gamma < 0$ (resp. $\gamma > 0$) distribution with a longer left tail (resp. right tail)

  - 4th order: Quantifies how heavy the tails of the distribution are

Multivariate data

Multivariate data: $D > 1$, $\mathbf{x}_i$ is now a realization of a random vector.

Expectation:

$$\mathbb{E}[\mathbf{x}] = \mathbb{E}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} = \begin{bmatrix} \mathbb{E}[x_1] \\ \mathbb{E}[x_2] \\ \vdots \\ \mathbb{E}[x_D] \end{bmatrix}$$

Notion of covariance:

$$\mathrm{Cov}(x_i, x_j) = \mathbb{E}[(x_i - \mathbb{E}[x_i])(x_j - \mathbb{E}[x_j])] \in \mathbb{R}$$

Covariance matrix:

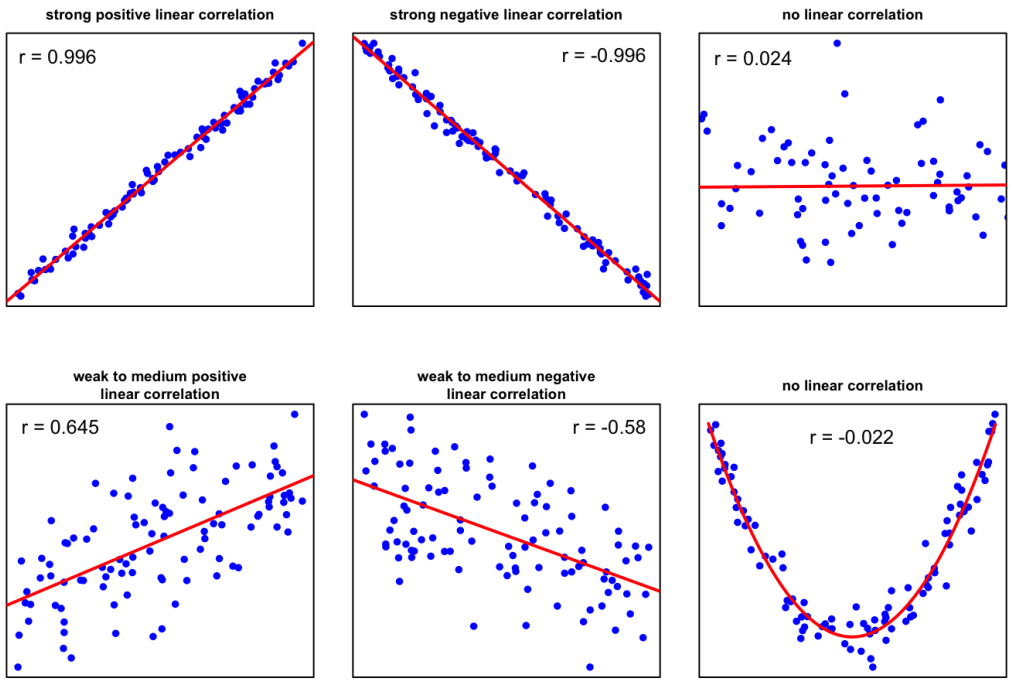
$$\mathrm{Cov}(\mathbf{x}) = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^\top] \in \mathbb{R}^{D \times D}$$

Correlation and independence

Two variables X and Y are uncorrelated if their covariance $\mathrm{Cov}(X, Y)$ is zero

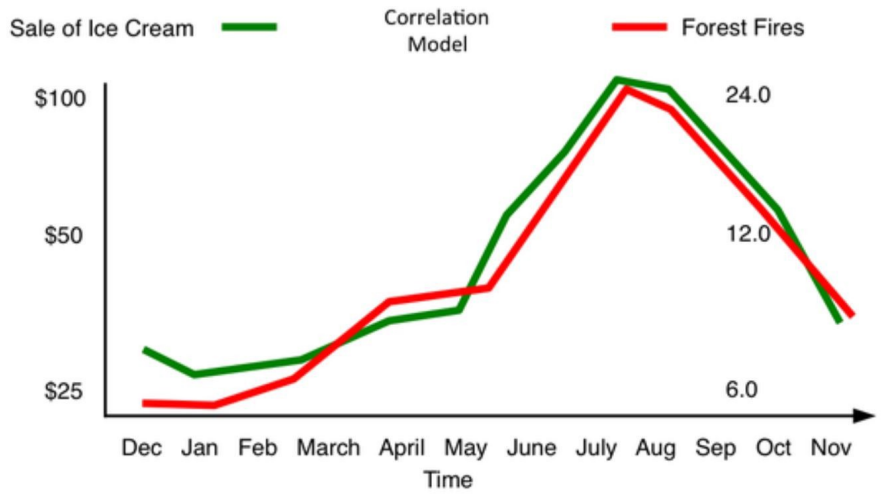
Correlation is defined as

$$\mathrm{Corr}(X, Y) = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \sigma_Y} \in [-1, 1]$$

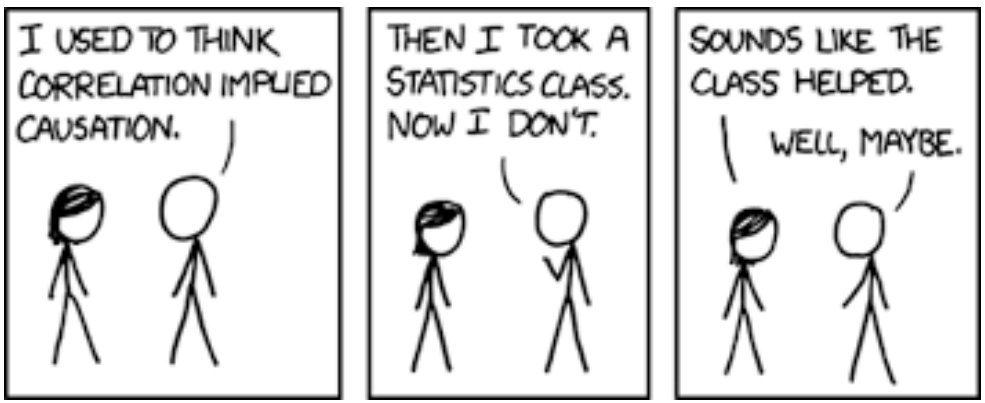


Two uncorrelated variables can be dependent!
Example : $Y = X^2$, where X is uniform on [-1,1].

Correlation does not imply causation



$\rightarrow$ In this case, both variables are caused by another common variable (i.e. the high temperatures in summer)



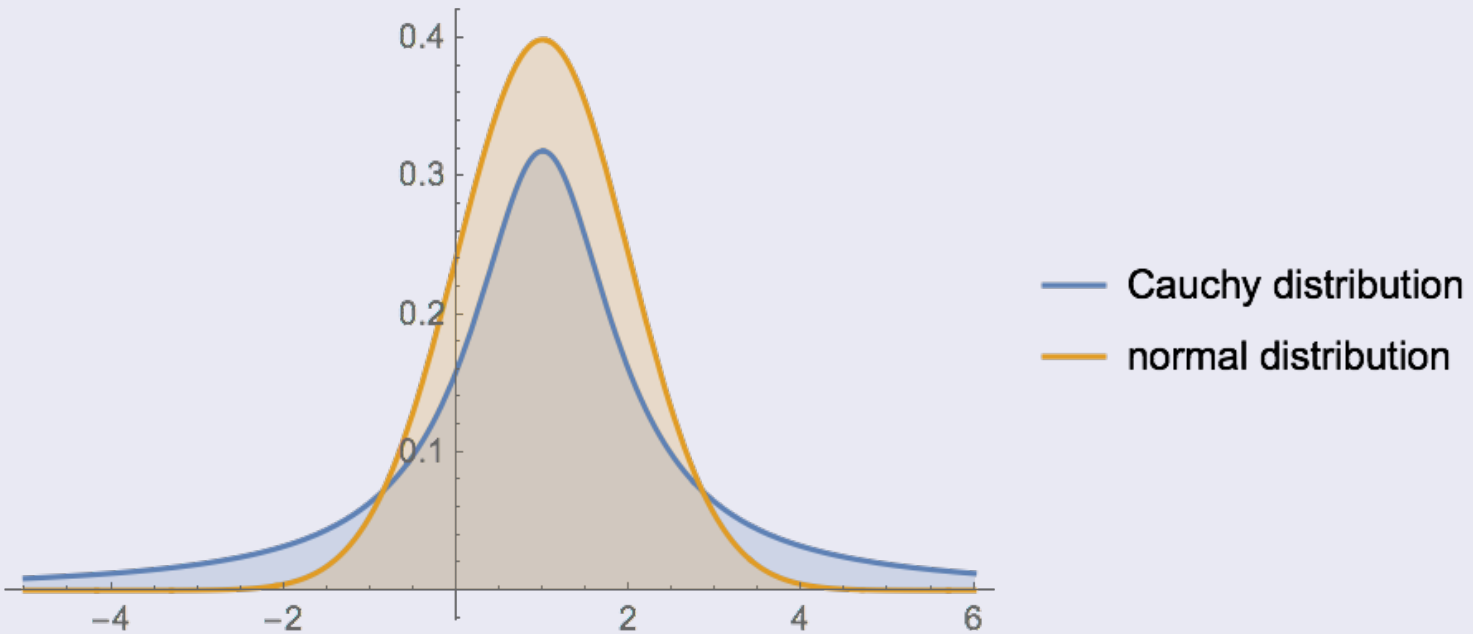
Outline

Outliers

There is no clear definition of what an outlier is, it is simply described as a data point which "significantly" differs from the rest.

Possible causes

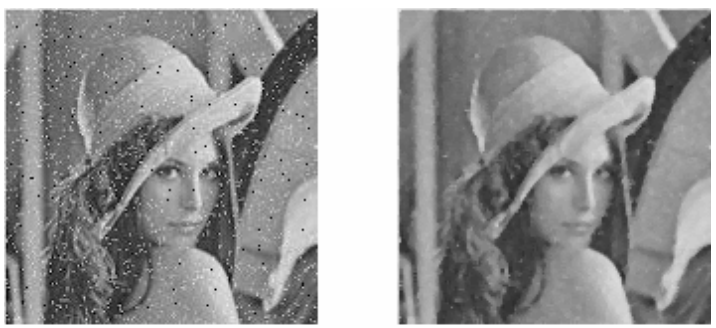
- Data point which does not follow the same distribution as the others (e.g. measurement error)
- The data has a heavy tailed distribution



Outliers

Examples of outliers in geoscience:

- Optical images: sun glint saturating the sensor in a few pixels
- Extreme values of geophysical fields (e.g. extreme wind gusts, rogue waves for sea surface heights...)
- Sensor issues: dead pixels on a CCD device, numerical "salt and pepper" noise...



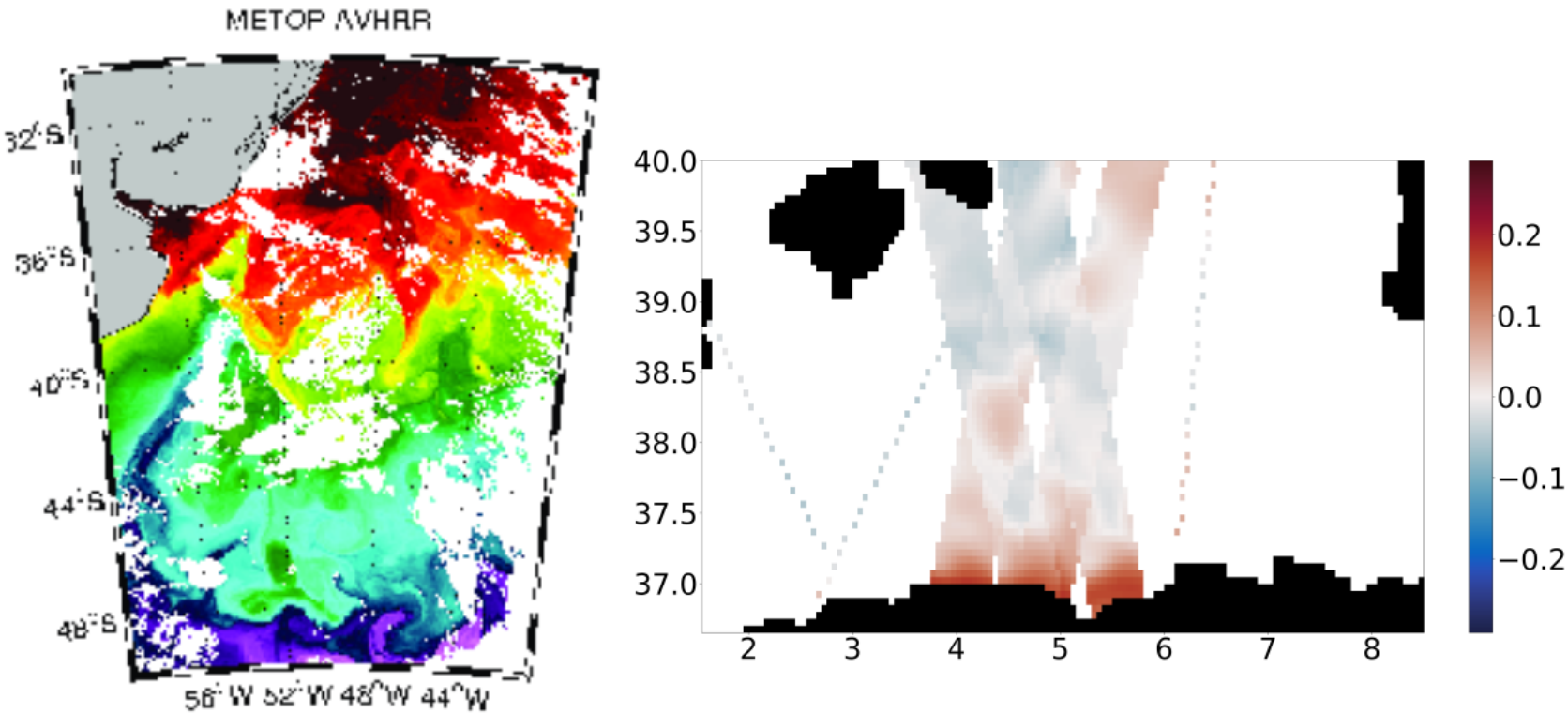
Original Image with Median Filter

- A possible strategy for outlier removal: Remove all data points such that $x \notin [median - 3\sigma, median + 3\sigma]$
  Rationale: for the normal (Gaussian) distribution, a drawn sample has 99% probability to be in this interval.

Missing Values

Missing values are frequent in geoscientific data. Ex:

- Cloud cover in optical imaging
- "Along track" satellite acquisitions
- Sensor issues
- Boundaries: e.g. land pixels in oceanic variables



Missing data in microwave Sea Surface Temperature (SST) data (left) and in altimetric Sea Surface Height (SSH) along track data (right)

Handling missing values

There are different cases:

- The data points are just used in a statistical way (no use of the topology): then analyses can still be performed with a reduced number of points
- The topology is used, and we need to fill in values: interpolation

Interpolation

- "Brutal" way: fill everything with zeros
- Statistical way: try to combine and diffuse the information in observed points to unobserved ones
- "Assimilation" way: Combine statistics and a priori physical information (a dynamical model for space time processes, coarse resolution auxiliary data...)

Statistical interpolation techniques

Basic idea

- Reconstruct the value of a desired grid point using a linear combination of all observations:

$$\mathbf{x} = \sum_{i=1}^{N} w_i \mathbf{d}_i$$

- The weights can be calculated in several ways, and are typically higher for observed points close to the desired point and decay with the distance.

We define $r_i = ||\mathbf{d}_i - \mathbf{x}||_2$ for all grid points $\mathbf{x}$ and a parameter R.
Gaussian weights:

$$\tilde{w}_i = \exp\left(\frac{-r_i^2}{2R^2}\right)$$

The weights are normalized so they sum to one over all the data:

$$w_i = \frac{\tilde{w}_i}{\sum_{j=1}^{N} \tilde{w}_j}$$

Optimal Interpolation

Idea

Define a way to perform the interpolation which accounts for spatial/temporal correlations in a flexible way and that is statistically optimal under some hypotheses

State-space formulation: we define a state vector $\mathbf{x} \in \mathbb{R}^P$ containing all of our P field values to estimate, and an observation vector $\mathbf{y} \in \mathbb{R}^N$ containing the values at the N observed points.

$$\mathbf{x} = \mathbf{x}_b + \boldsymbol{\eta}$$
$$\mathbf{y} = \mathcal{H}(\mathbf{x}) + \boldsymbol{\epsilon}$$

- $\mathbf{x}_b$ is a background estimation
- $\boldsymbol{\eta}$ is the background error
- $\mathcal{H}$ is an operator mapping the full state to the observations
- $\boldsymbol{\epsilon}$ is the observation error

$\rightarrow$ OI finds the Best Linear Unbiaised Estimator (BLUE) of $\mathbf{x}$ given the uncertainties on the model and the observations.

Outline

Probability density estimation

Knowledge of $p(\mathbf{x})$ provides all the info on the variable
$\rightarrow$ Estimate it from data!

There are two ways of doing this:

- Assume the data follows a parametric density and estimate those parameters from data
  $\rightarrow$ Maximum Likelihood estimation
- Use non-parametric approaches to fit a pdf to the data
  $\rightarrow$ Histograms and kernel density estimation

Parametric density estimation

- Assume that the data follows some distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ where $\boldsymbol{\theta}$ are the parameters of the distribution.
- Find the values of the parameters $\boldsymbol{\theta}$ such that their value makes $p_{\boldsymbol{\theta}}(\mathbf{x})$ maximal:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\mathbf{x})$$

  This procedure is called *Maximum Likelihood* estimation
- In many cases, we have assume that the samples $\mathbf{x}_i$ are independent and identically distributed (i.i.d.), so that

$$p_{\boldsymbol{\theta}}(\mathbf{X}) = \prod_{i=1}^{N} p_{\boldsymbol{\theta}}(\mathbf{x}_i)$$

- Main advantage:the distribution is summarized by a small number of parameters

Parametric density estimation (cont'd)

Example of the Gaussian

- Assume that $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})\ \forall i = 1, ..., N$. Here $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ (the pdf is completely specified by those two parameters) and

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$

- Then, we can show that

$$\hat{\boldsymbol{\mu}}_{ML} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i \text{ and } \hat{\boldsymbol{\Sigma}}_{ML} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x} - \hat{\boldsymbol{\mu}})(\mathbf{x} - \hat{\boldsymbol{\mu}})^\top$$

Issues with parametric density estimation:

- Assumptions on the distribution are not always realistic
- Estimation of the parameters can be cumbersome

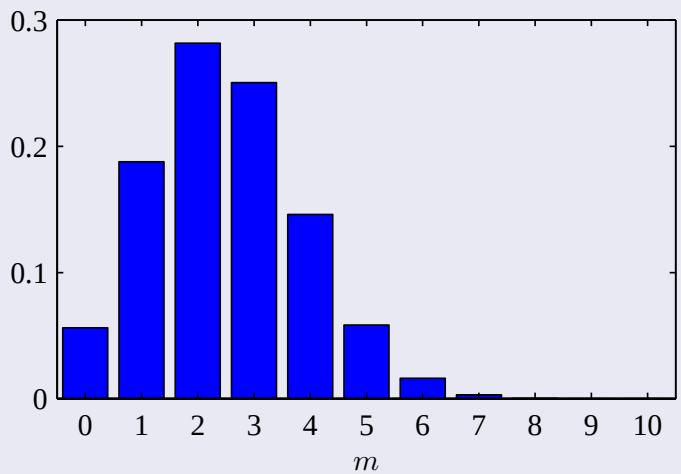
Histograms

Divide the domain into bins of width Δ_i and count the number of occurrences n_i of the data values in each bin.

Then we say that the probability that a data point falls into bin $\#i$ is given by

$$p_i = \frac{n_i}{N\Delta_i}$$

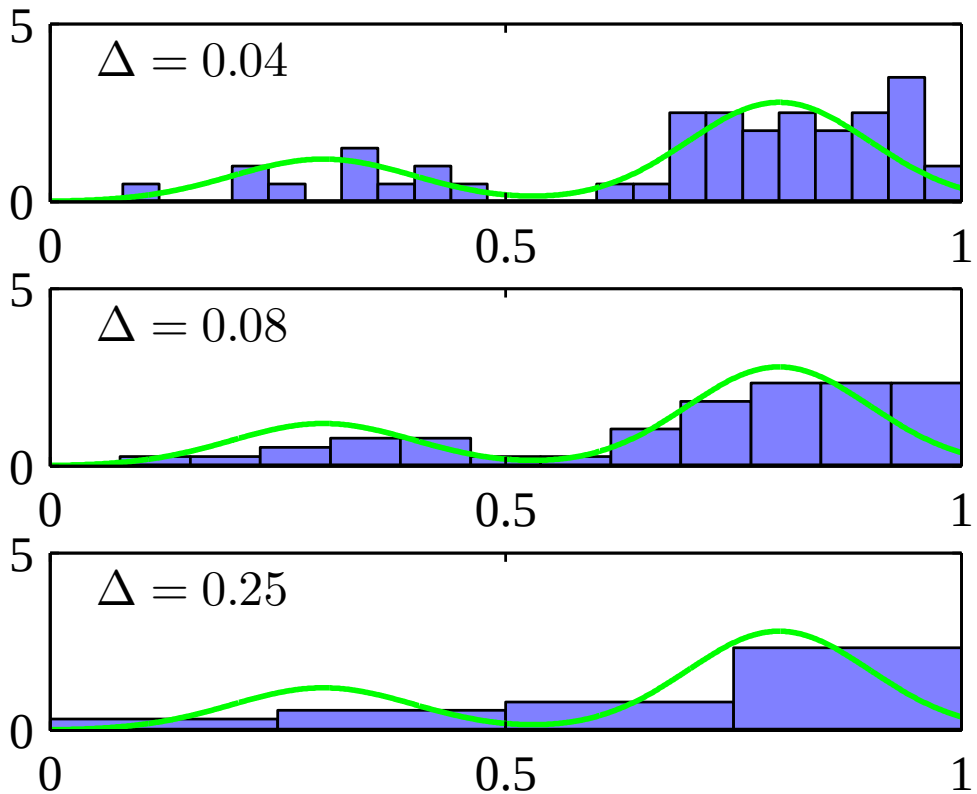
This assumes that the pdf is piecewise constant



We can show that the p_i define a valid pdf.

Histograms (cont'd)

The number of bins (or equivalently their width if it is the same for each bin) has to be tuned carefully.



Histograms for density estimation. 50 points from the true (green) distribution have been sampled, and the pdf is approximated by histograms with different bin widths.

Kernel density estimation

Consider a possible value of a vector RV $\mathbf{x}$, and a small region $\mathcal{R}$ such that $\mathbf{x} \in \mathcal{R}$. Then the probability mass of $\mathcal{R}$ is:

$$P = \int_{\mathcal{R}} p(\mathbf{x}) d\mathbf{x}$$

We observe N realizations of the data. $P \in \mathcal{R}$ with probability P. Then, the probability that there are K observations in $\mathcal{R}$ out of N data points follows a binomial distribution $\mathrm{Bin}(K|N, P)$.

$$\mathbb{E}[K/N] = P, \text{ and } \mathrm{var}(K/N) = P(1-P)/N \underset{N \to \infty}{\to} 0$$

So, for large N, $K \approx NP$.

Kernel Density Estimation

In addition, if region $\mathcal{R}$ is small, $p(\mathbf{x})$ is almost constant on the support of the region (volume V):

$$P \approx p(\mathbf{x})V$$

Finally:

$$p(\mathbf{x}) \approx \frac{K}{NV}$$

Our goal is to estimate $p(\mathbf{x})$

- Choose K and try to estimate V locally:
  K-nearest neighbor approach
- Choose V and try to estimate K:
  kernel density estimator

Kernel density estimator

Let's choose the region $\mathcal{R}$ to be a small hypercube centered at $\mathbf{x} \in \mathbb{R}^D$. We define the *indicator function* (an example of kernel) of a hypercube centered on the origin and of "radius" $\frac{1}{2}$:

$$k(\mathbf{u}) = \begin{cases} 1 & \text{if } |u_i| \leq 1/2, \ \forall i = 1, ..., D \\ 0 & \text{otherwise} \end{cases}$$

Then $k(\frac{\mathbf{x}-\mathbf{x}_n}{h})$ is the indicator function of a hypercube centered at $\mathbf{x}$ of side h.

$$K = \sum_{n=1}^{N} k\left(\frac{\mathbf{x} - \mathbf{x}_n}{h}\right)$$

And finally:

$$p(\mathbf{x}) \approx \frac{K}{NV} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{h^D} k\left(\frac{\mathbf{x} - \mathbf{x}_n}{h}\right)$$

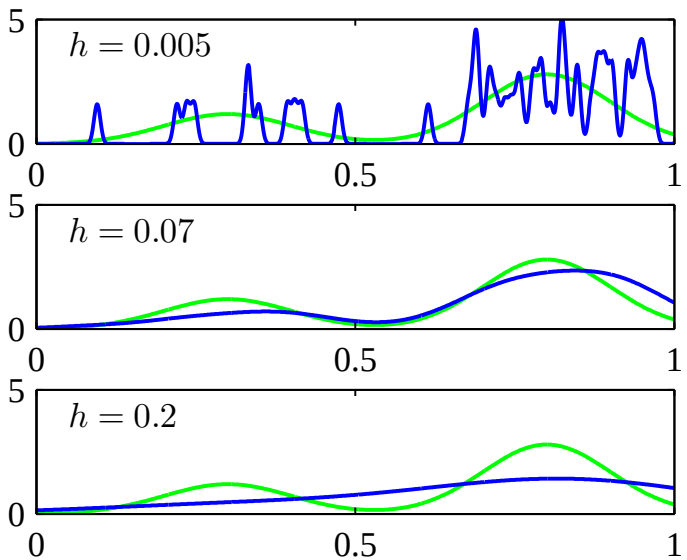
Note: This generalizes the framework of histograms!

Kernel density estimator

Trick: changing the function k into a smoother one, e.g. a Gaussian of variance h^2 for each dimension (diagonal covariance matrix) :

$$p(\mathbf{x}) \approx \frac{K}{NV} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{(2\pi h^2)^{D/2}} \exp\left(-\frac{||\mathbf{x} - \mathbf{x}_n||^2}{2h^2}\right)$$

Any function satisfying the axioms of a pdf can be chosen: $k(\mathbf{u}) \geq 0$ and $\int k(\mathbf{u}) d\mathbf{u} = 1$



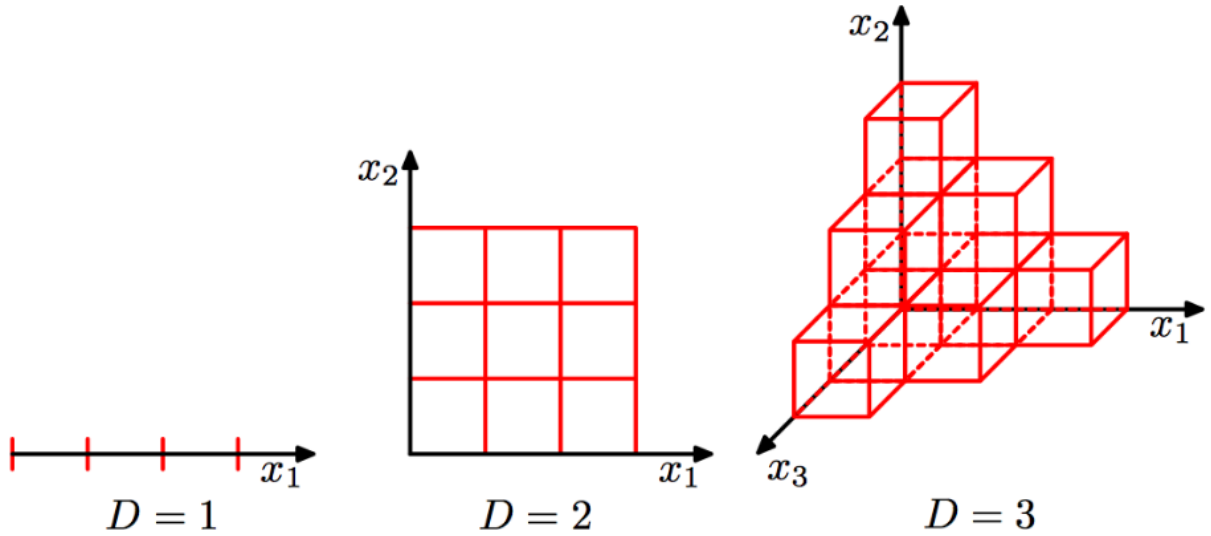
Gaussian Kernel density estimator on the same data as before

Note: KDE can also be seen as an interpolation technique!

Outline

Problems in High Dimensions

- Computing histograms in large dimensions is tricky because it requires to grid the domain
  $\rightarrow$ The number of needed hypercubes to grid $[0,1]^D$ scales exponentially with the dimension D of the data!



The volume of a single hypercube (h^D, $h < 1$) also becomes smaller and smaller...
This means that data points are *isolated* in a high dimensional space: you need exponentially more points for an "equivalent" sampling

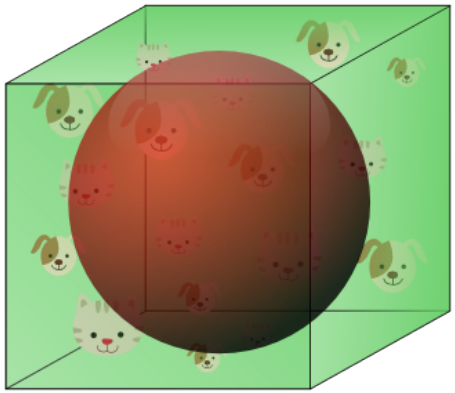
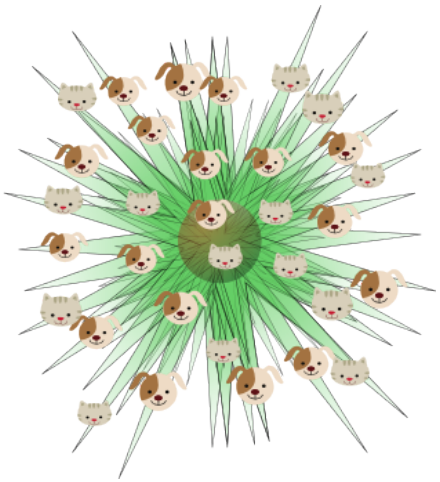
The curse of dimensionality

- Visualizing data or probability distributions in more than 2-3D is hard
- Parametric estimation also has issues: e.g. for a multivariate Gaussian, the number of parameters scales as D^2 (full covariance matrix): need to
  - estimate them from data (requires a large number of samples)
  - invert or find the eigenvalues of those matrices... (computing time scales in $\mathcal{O}(D^3)$)

The curse of dimensionality (cont'd)

The performance of learning tasks increases with the dimension of the features (more info) but only *up to a point*

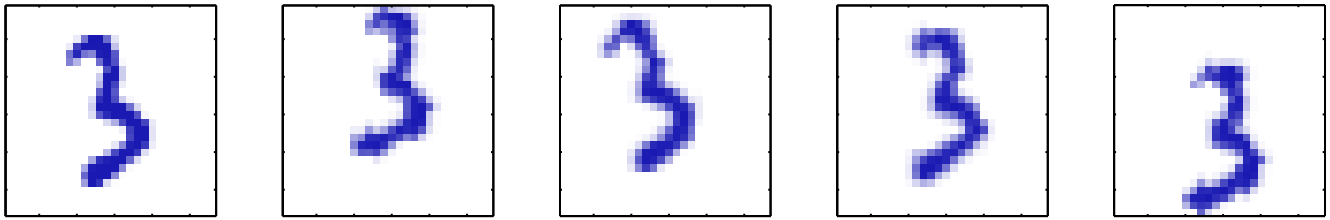
In high dimensions, the Euclidean distance does not separate data points well enough anymore

$$\frac{\mathrm{Vol}_{Sphere}(D)}{\mathrm{Vol}_{Cube}(D)} \underset{D\to\infty}{\to} 0$$

This means that randomly sampled data points get further and further away from the mean and from eachother as D grows. Every point is an outlier!

Visualizing High dimensional data



Several transformed instances of the same "3" in the MNIST Digit dataset (100×100 images).

What is the actual dimensionality of the data?
$\rightarrow$ Each image can be seen as a point in $\mathbb{R}^{10^4}$

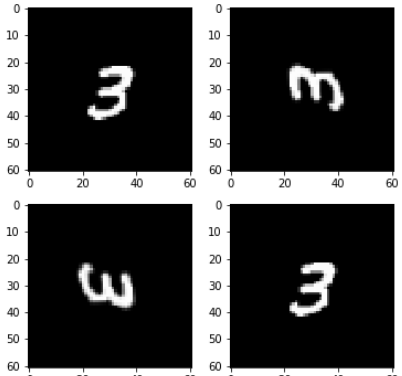
What is the number of degrees of freedom in all these data?

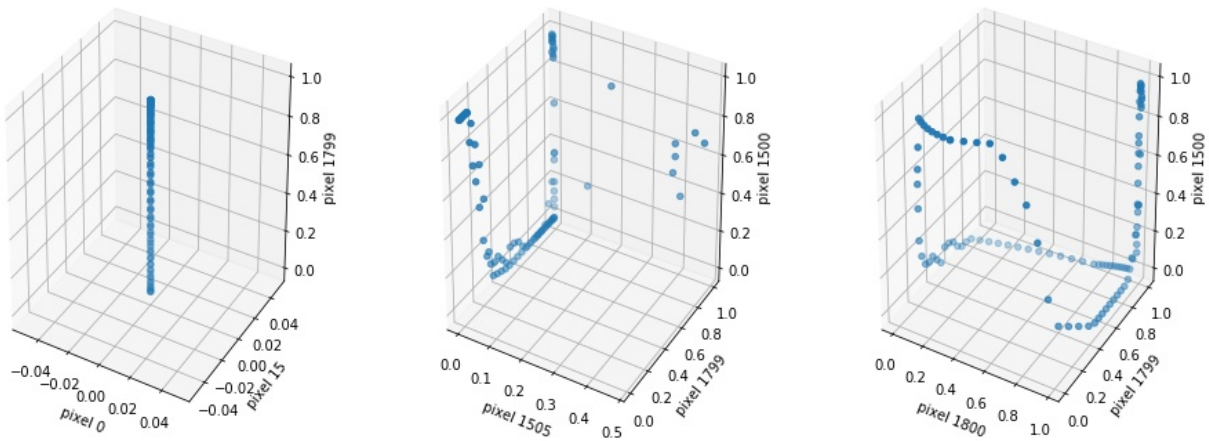
- Two Translations
- One rotation
- (With real instances of "3" images, several additional degrees from the variability of the writing of "3")

What is the *intrinsic dimensionality* of the data?

Visualizing rotated images in 3D

Only one degree of freedom here (one rotation).
Data still lives in $\mathbb{R}^{10^4}$. How can we visualize the feature space?





$\rightarrow$ We need a way to summarize the information in only a few variables to visualize the data efficiently

Outline

Motivation

High Dimensional datasets are often intrisically *low-dimensional*.
$\rightarrow$ much fewer degrees of freedom than the dimensionality of the data.

This means that fewer variables than D can summarize the information efficiently

The goal is to be able to recover *latent variables* from the data.

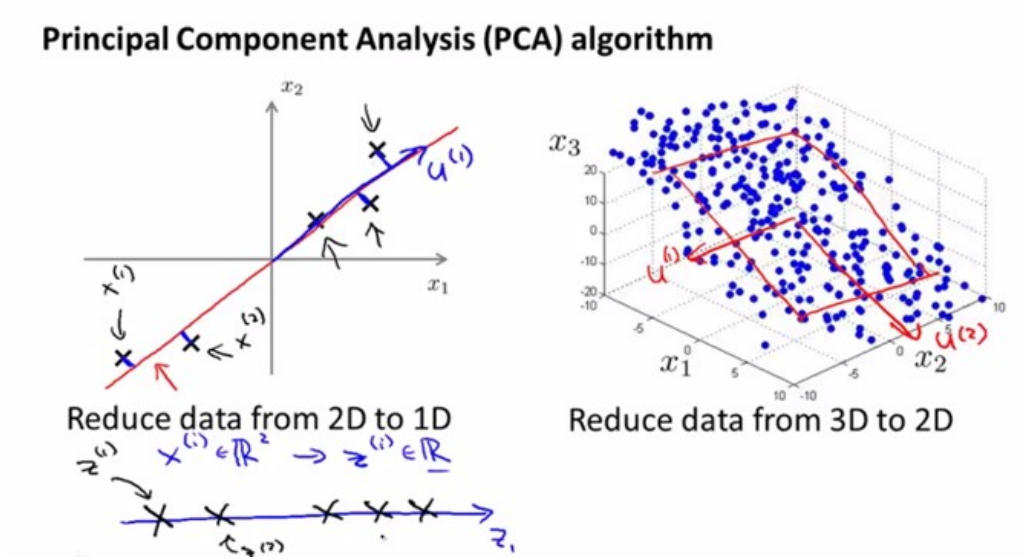
Applications:

- Dimensionality Reduction
- Data compression
- Data visualization
- Feature Extraction
- Denoising

Principal Component Analysis

Basic idea: try to find a P-dimensional linear subspace which best represents the data in some sense.

Chosen criterion: Find the subspace such that the the variance of the orthogonal projection of the data on it is maximal



Alternative criterion: find an orthogonal projection of the data on a subspace such that the projection error is minimal

$\rightarrow$ both formulations lead to the same solution and algorithm, called *Principal Component Analysis*.

Maximum Variance Formulation

Goal: Define an *orthogonal projection of the data* on a M-dimensional subspace, such that:

- the *variance* of the data on the first component is maximal
- the variance of the residual on the second component (orthogonal to the first) is maximal
- and so on.

How to derive the first component? $\rightarrow$ projection of the data on a line, directed by $\mathbf{u}_1$ (we choose $||\mathbf{u}_1||_2^2 = \mathbf{u}_1^\top \mathbf{u}_1 = 1$)

The projection of a data point $\mathbf{x}_n$ on the line is:

$$p(\mathbf{x}_n) = (\mathbf{u}_1^\top \mathbf{x}_n)\mathbf{u}_1$$

The sample covariance of the *projected* data is:

$$\frac{1}{N}\sum_{n=1}^{N}(\mathbf{u}_1^\top \mathbf{x}_n - \mathbf{u}_1^\top \bar{\mathbf{x}})^2 = \mathbf{u}_1^\top \mathbf{S}\mathbf{u}_1$$

with $\bar{\mathbf{x}}$ is the sample mean of the data, and $\mathbf{S}$ its sample covariance.

PCA: practical algorithm

Now we need to solve:

$$\begin{aligned} \arg\max \quad & \mathbf{u}_1^\top \mathbf{S}\mathbf{u}_1 \\ \text{s.t.} \quad & ||\mathbf{u}_1||_2^2 = 1 \end{aligned}$$

The constraint removes degenerate solutions $||\mathbf{u}_1|| \to +\infty$

From this, we can show that there exists λ_1 such that the solutions verify

$$\mathbf{S}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$

There are several possibilities (one for each eigenvector of $\mathbf{S}$, symmetric and positive semidefinite matrix) and the corresponding variance is equal to:

$$\mathbf{u}_1^\top \mathbf{S}\mathbf{u}_1 = \lambda_1$$

$\to$ the one maximizing the variance is the eigenvector associated to the *largest* eigenvalue.

Optimization for Maximum Variance

To obtain the remaining components, we simply have to repeat the process on the residual data:

$$\hat{\mathbf{x}}_n = \mathbf{x} - (\mathbf{u}_1^\top \mathbf{x}_n)\mathbf{u}_1$$

(orthogonal to the line directed by $\mathbf{u}_1$).

And we can repeat the process until we get M components.

Algorithm

1. Compute the sample covariance Matrix $\mathbf{S}$ of the data
2. Perfom its eigenvalue decomposition
3. The projection matrix $\mathbf{U}_M$ is given by the M eigenvectors, associated to the M largest eigenvalues (in decreasing order).

PCA as a matrix factorization

With D (all) components, we can write any data point as a linear combination of the principal components $\mathbf{u}_i$:

$$\mathbf{x}_i = \sum_{j=1}^{D} (\mathbf{u}_j^\top \mathbf{x}_i)\mathbf{u}_j \triangleq \sum_{j=1}^{D} a_{ij}\mathbf{u}_j = \mathbf{U}_D \mathbf{a}_i$$

where $\mathbf{U}_D \in \mathbb{R}^{D \times D}$ gathers all the eigenvectors, and $\mathbf{a}_i \in \mathbb{R}^D$ all the projection coefficients for data point $\mathbf{x}_i$

This rewrites globally for all data points as:

$$\mathbf{X} = \mathbf{U}_D \mathbf{A}$$

with $\mathbf{X} \in \mathbb{R}^{D \times N}$, and $\mathbf{A} \in \mathbb{R}^{D \times N}$ gathers all the coefficients for all data points

$\rightarrow$ keeping only M components amounts to approximate $\mathbf{X}$ by:

$$\mathbf{X} \approx \mathbf{U}_M \mathbf{A}_M$$

Coming back to the applications

- PCA decorrelates data! The covariance of the transformed data is diagonal
- Allows to represent high-dimensional data in 2D or 3D, minimizing the distorsion (using a linear mapping)
- Helps in denoising the data: noise is high-dimensional, its power is reduced when projecting on smaller subspaces:

$$||\mathbf{n}||^2 = \left\| \sum_{i=1}^{D} (\mathbf{u}_i^\top \mathbf{n}) \mathbf{u}_i \right\|^2 = \sum_{i=1}^{D} \left(\mathbf{u}_i^\top \mathbf{n} \right)^2$$

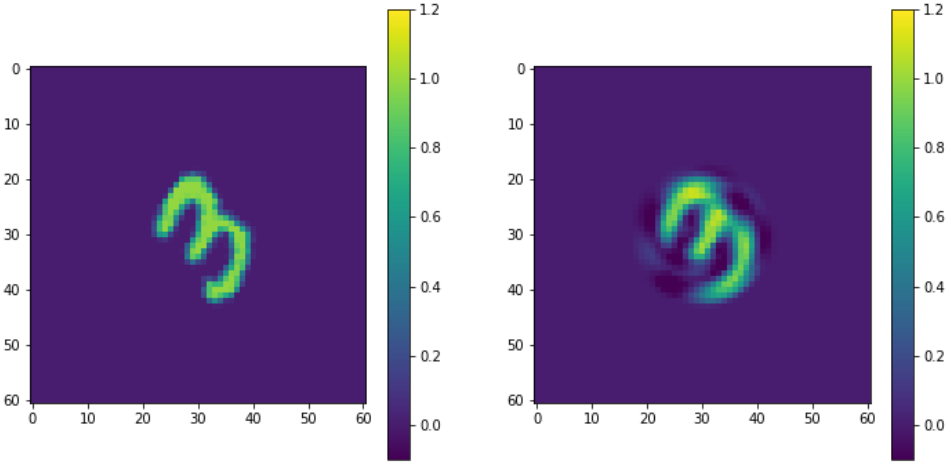
whereas

$$||\tilde{\mathbf{n}}||^2 = \left\| \sum_{i=1}^{M} (\mathbf{u}_i^\top \mathbf{n}) \mathbf{u}_i \right\|^2 = \sum_{i=1}^{M} \left(\mathbf{u}_i^\top \mathbf{n} \right)^2$$

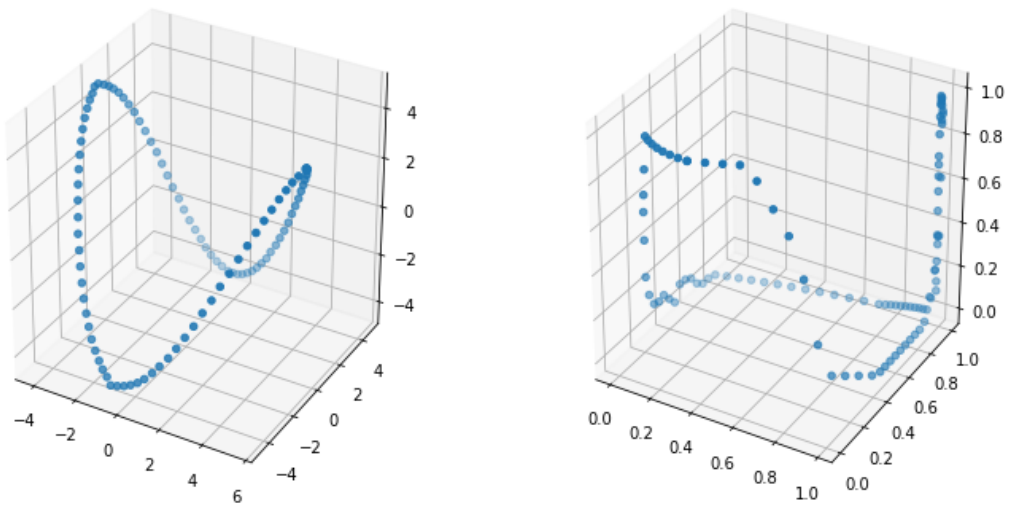
(this assumes that the noise has zero-mean—often a reasonable assumption)

PCA in practice

Most of the variance in the data is contained in the first components: we can obtain good approximations of the data with a drastically reduced number of variables

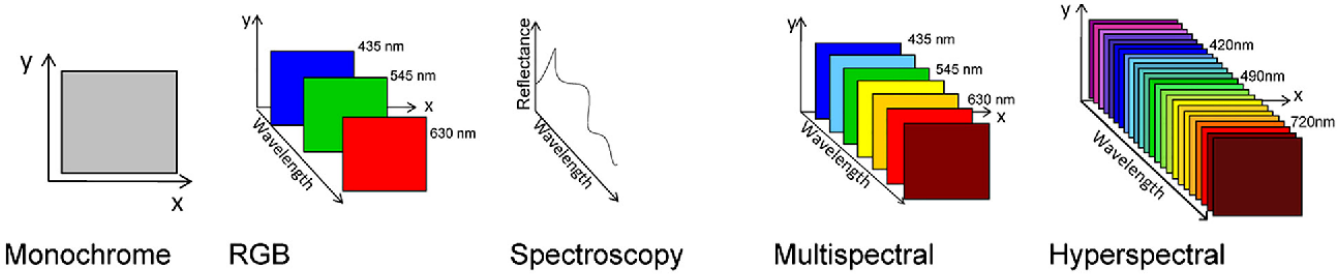


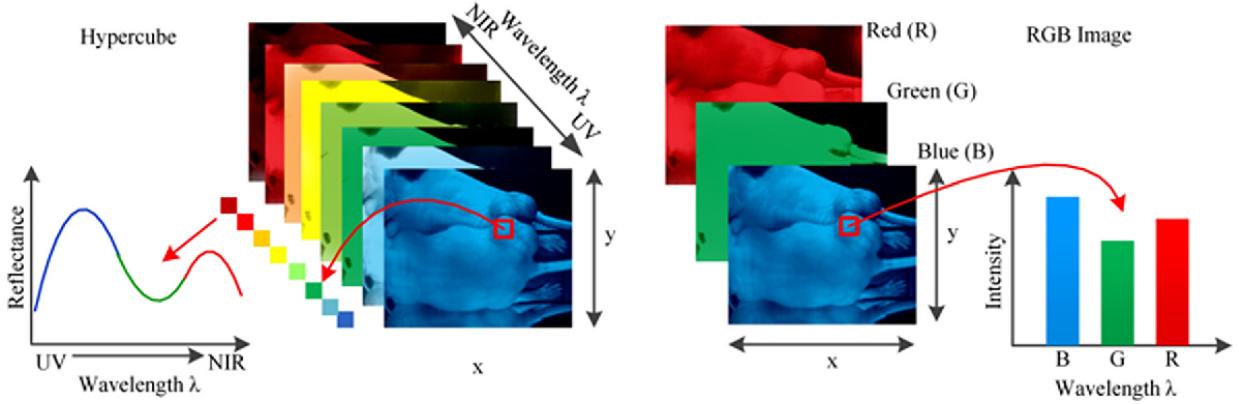
One random sample and its reconstruction using 8 PCs



Scatterplot of the data in the first 3 PCs (left) or and a few well chosen pixels (right)

Multi/Hyperspectral Images

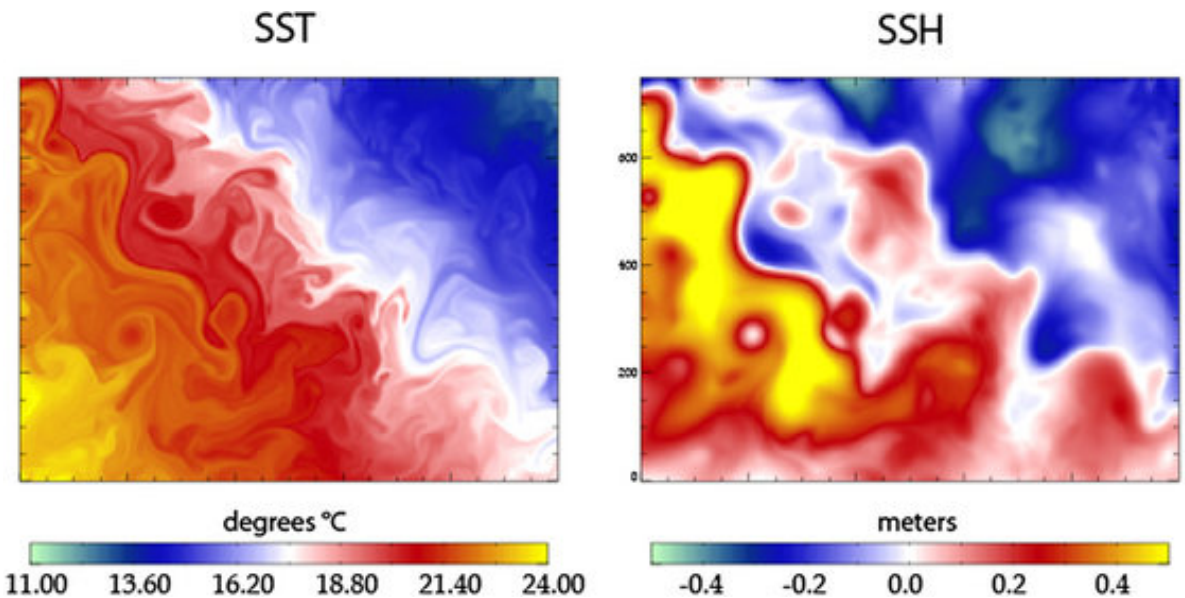




- images are matrices $\mathbf{X} \in \mathbb{R}^{D \times N}$, D is the number of bands, N is the number of pixels
- Principal components are vectors $\mathbf{u}_k \in \mathbb{R}^D$, and coefficients are vectors $\boldsymbol{a}_k \in \mathbb{R}^N$ (can be displayed as images)
- $\mathbf{X} = \mathbf{UA}$, $\mathbf{U} \in \mathbb{R}^{D \times D}$, and $\mathbf{A} \in \mathbb{R}^{D \times N}$

Examples in geosciences (cont'd)

Sea Surface Height/Temperature time series



- data are matrices $\mathbf{X} \in \mathbb{R}^{D \times N}$, N is the number time samples, D is the number of pixels
- Principal components are vectors $\mathbf{u}_k \in \mathbb{R}^D$, and coefficients are vectors $\boldsymbol{a}_k \in \mathbb{R}^N$ (can be displayed as time series)
- $\mathbf{X} = \mathbf{UA}$, $\mathbf{U} \in \mathbb{R}^{D \times D}$, and $\mathbf{A} \in \mathbb{R}^{D \times N}$