TAF MCE

UE Introduction to Machine Learning

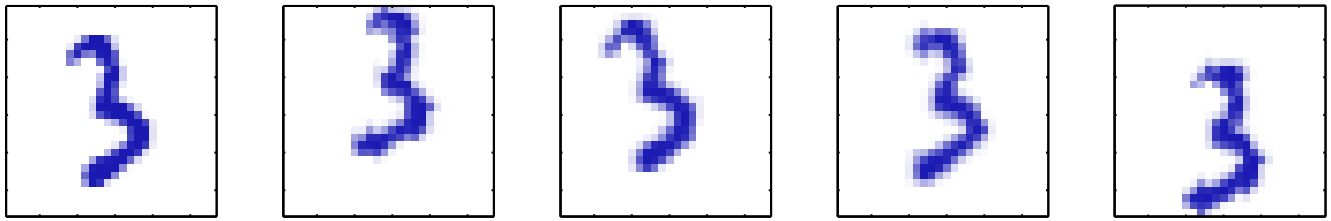
Principal Component Analysis

Lucas Drumetz
lucas.drumetz@imt-atlantique.fr

Outline

Outline

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?

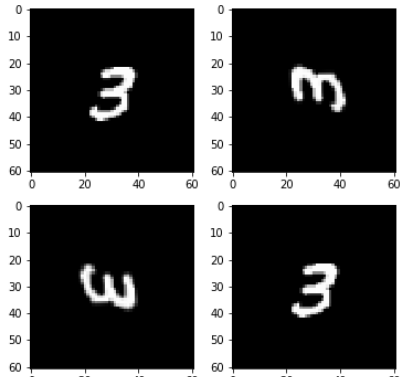
- Two Translations
- One rotation
- (With real instances of "3" images, several additional degrees from the variability of the writing of "3")

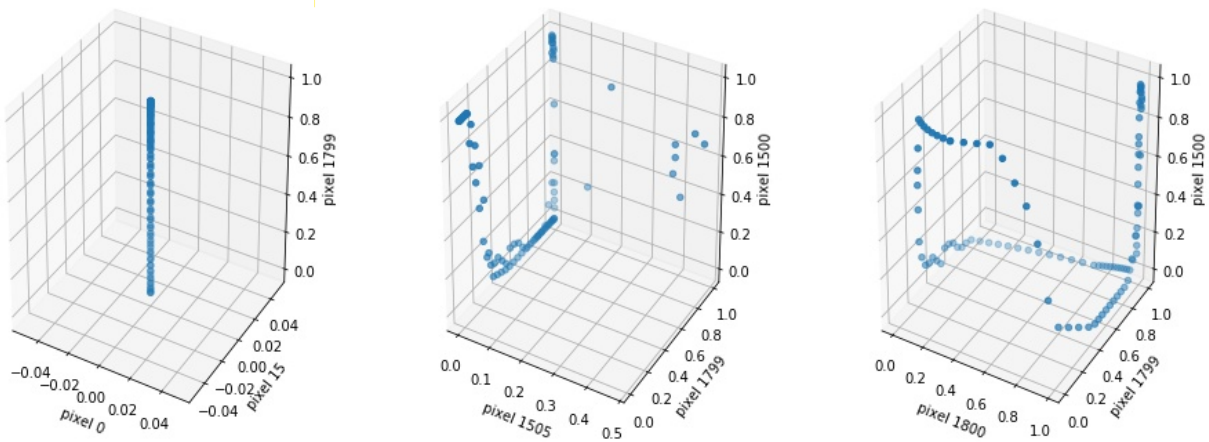
What is the *intrinsic dimensionality* of the data?

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?





→ We need a way to summarize the information in only a few variables to visualize the data efficiently

The Manifold Hypothesis

Any dataset in a high dimensions actually lives in a lower dimensional space:

- In some cases, a simple linear space if there are linear relationships between variables
- In others, in more complex spaces called "manifolds": curves (1D manifold), surfaces (2D manifolds), etc.
  For example, the data may be modeled in some cases as

$$\mathbf{x} = f(\theta_1, \theta_2, ...)$$

  where the θ_i are "degrees of freedom", and f is a potentially nonlinear function

$\rightarrow$ This means there is a way to summarize the information of a dataset in less variables than the actual dimensionality of the data

Outline

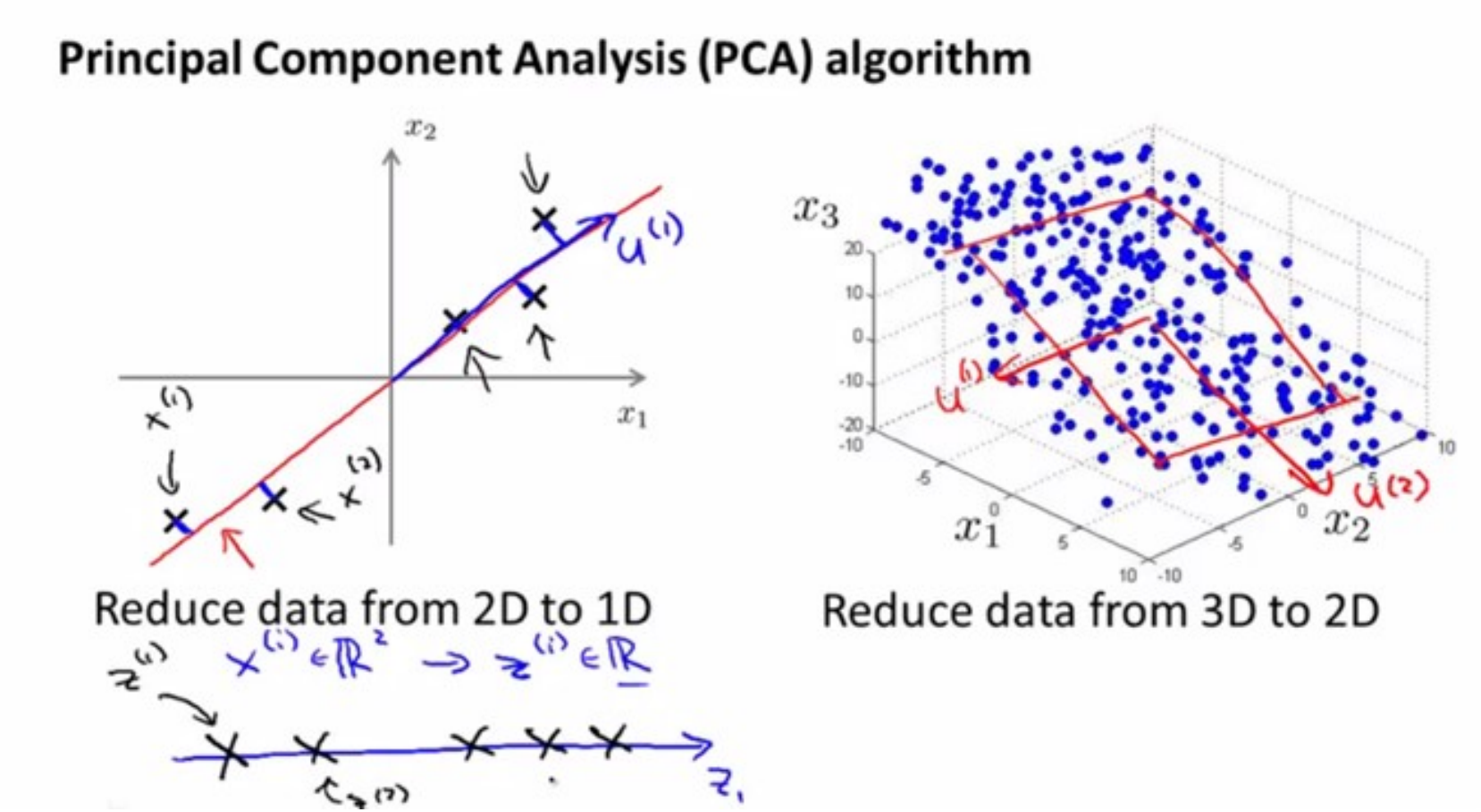
1. Visualizing High dimensional data

2. Principal Component Analysis
   - Motivation
   - Algorithm
   - Examples

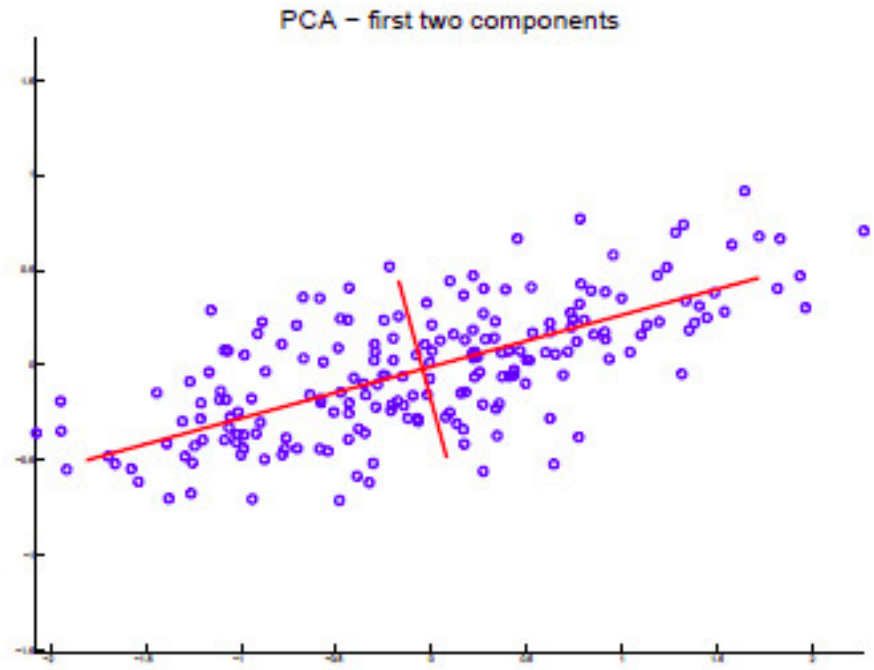
Principal Component Analysis

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



PCA (cont'd)



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:

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

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

PCA: optimization

Now we need to solve:

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

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

We write the Lagrangian (with a minus sign before the multiplier wich does not change anything):

$$\mathcal{L}(\mathbf{u_1}, \lambda_1) = \mathbf{u}_1^\top \mathbf{S}\mathbf{u}_1 - \lambda_1(||\mathbf{u}_1||_2^2 - 1) = 0$$

So necessary optimality conditions write:

$$\mathbf{S}\mathbf{u}_1 - \lambda_1 \mathbf{u}_1 = \mathbf{0}$$

PCA: optimization

There exists λ_1 such that the solutions verify

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

$\rightarrow$ This is an eigenvalue equation.

S is a symmetric positive semidefinite matrix:

There are D eigenvalues and they are all positive. The eigenvectors are also orthogonal (spectral theorem).

There are several possibilities (one for each eigenvector of **S**) and the corresponding variance is equal to:

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

$\rightarrow$ the one maximizing the variance is the eigenvector associated to the *largest* eigenvalue. And that eigenvalue is equal to the variance along the selected direction.

PCA: optimization

We look for another direction with the largest possible variance. This direction should be complementary to the one already identified: we impose that $\mathbf{u}_1 \perp \mathbf{u}_2$, i.e. $\mathbf{u}_1^T \mathbf{u}_2 = 0$
Now we need to solve:

$$\begin{aligned} &\arg\max \quad \mathbf{u}_2^\top \mathbf{S}\mathbf{u}_2 \\ &\text{s.t.} \quad ||\mathbf{u}_2||_2^2 = 1, \ \mathbf{u}_2^T \mathbf{u}_1 = 0 \end{aligned}$$

The Lagrangian writes:

$$\mathcal{L}(\mathbf{u_2}, \lambda_2, \nu) = \mathbf{u}_2^T \mathbf{S}\mathbf{u}_2 - \lambda_2(\mathbf{u}_2 - 1) + \nu \mathbf{u}_2^T \mathbf{u}_1$$

The optimality conditions write:

$$2\mathbf{S}\mathbf{u}_2 - 2\lambda_2 \mathbf{u}_2 + \nu \mathbf{u}_1 = 0$$

Multiplying by $\mathbf{u}_1$:

$$2\mathbf{u}_1^T \mathbf{S}\mathbf{u}_2 - 2\lambda_2 \mathbf{u}_1^T \mathbf{u}_2 + \nu \mathbf{u}_1^T \mathbf{u}_1 = \mathbf{0}$$

Using the constraints and the fact that $\mathbf{u}_1^T \mathbf{S}\mathbf{u}_2 = \mathbf{u}_2^T \mathbf{S}\mathbf{u}_1 = \lambda_1 \mathbf{u}_2^T \mathbf{u}_1$, we get $\nu = 0$. Then:

$$\mathbf{S}\mathbf{u}_2 = \lambda_2 \mathbf{u}_2$$

We see that $\mathbf{u}_2$ is another eigenvector of the covariance matrix, associated to the second largest eigenvalue...

We repeat this process until we get M components that are orthogonal to one another.

Practical algorithm

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).

$\mathbf{U}_M = [\mathbf{u}_1, \cdots, \mathbf{u}_M] \in \mathbb{R}^{D \times M}$. Note that if we keep D components, the now square matrix is $\mathbf{U}_D$ orthogonal, meaning that $\mathbf{U}_D^{-1} = \mathbf{U}_D^T$

The projection coefficients for a given data point i can be obtained by computing:

$$\mathbf{a}_i = \mathbf{U}_D^T \mathbf{x}_i \in \mathbb{R}^M$$

PCA as a matrix factorization

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

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

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:

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

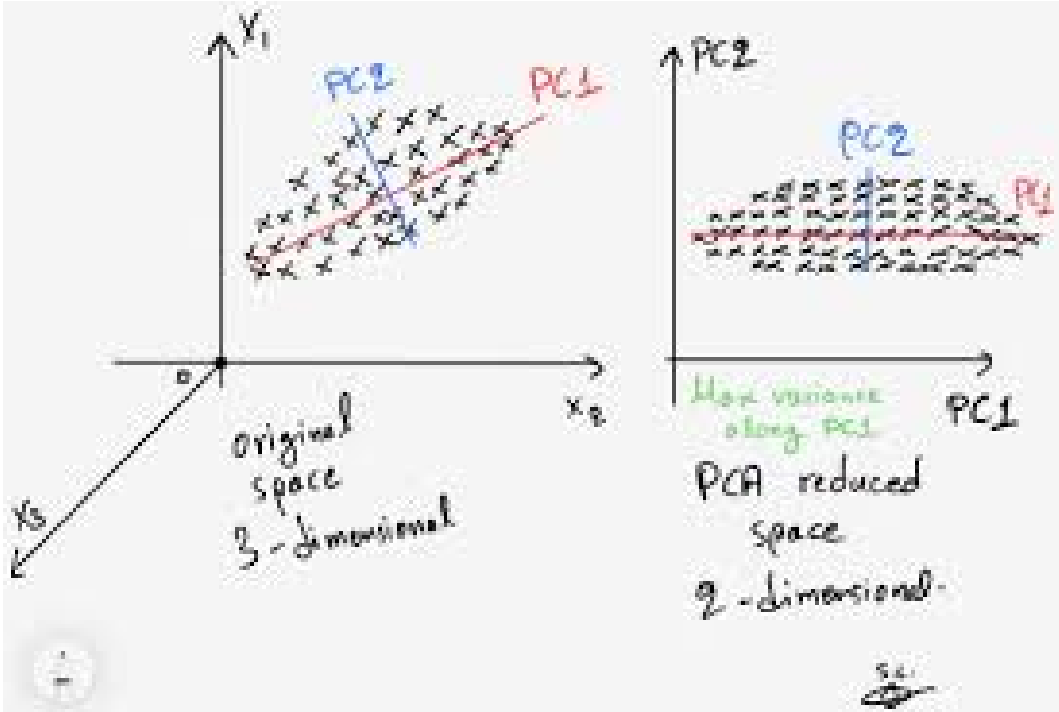
with $\mathbf{X} \in \mathbb{R}^{D\times N}$, and $\mathbf{A} \in \mathbb{R}^{D\times N}$ gathers all the coefficients for all data points

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

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

Coming back to the applications

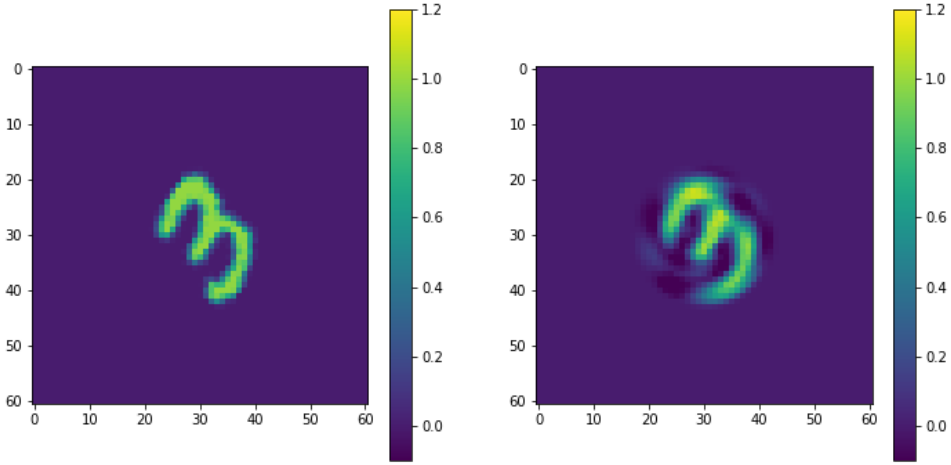
- PCA decorrelates data! The covariance of the transformed data is diagonal (diagonal elements equal to the eigenvalues)



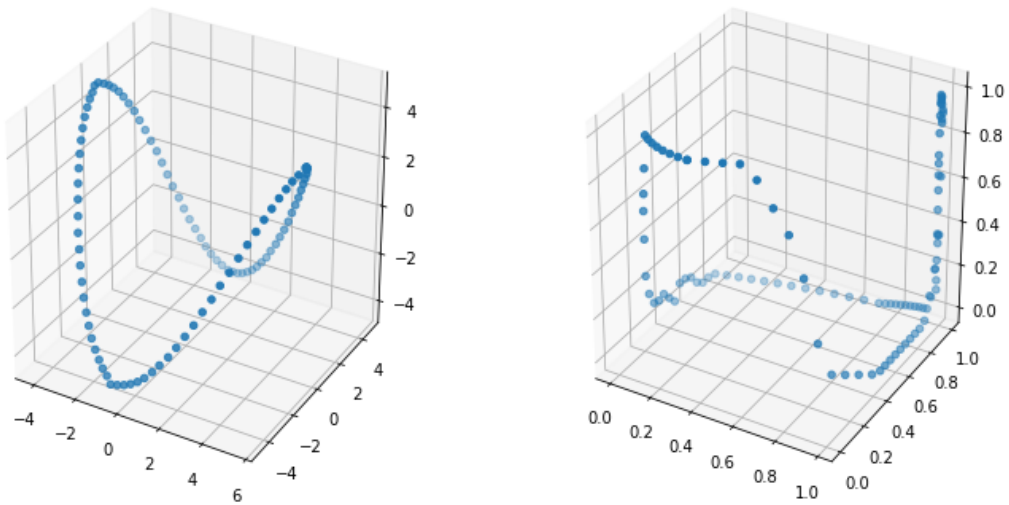
- Allows to represent high-dimensional data in 2D or 3D, minimizing the distorsion (using a linear mapping)
- Helps in denoising the data: noise is high-dimensional, its power is reduced when projected on smaller subspaces:
- If only M components are used to reconstruct the data, the data can be efficiently compressed

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



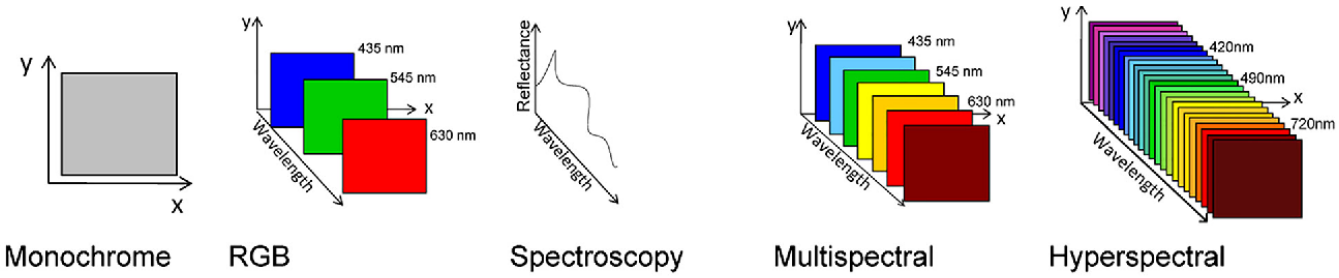
One random sample and its reconstruction using 8 PCs

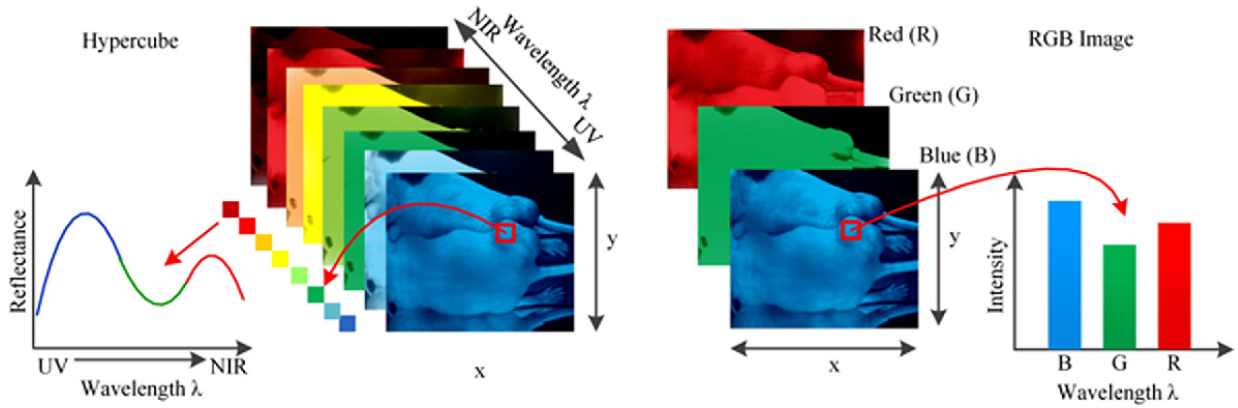


Scatterplot of the data in the first 3 PCs (left) or and a few well chosen pixels (right)

Examples in geosciences

Multi/Hyperspectral Images

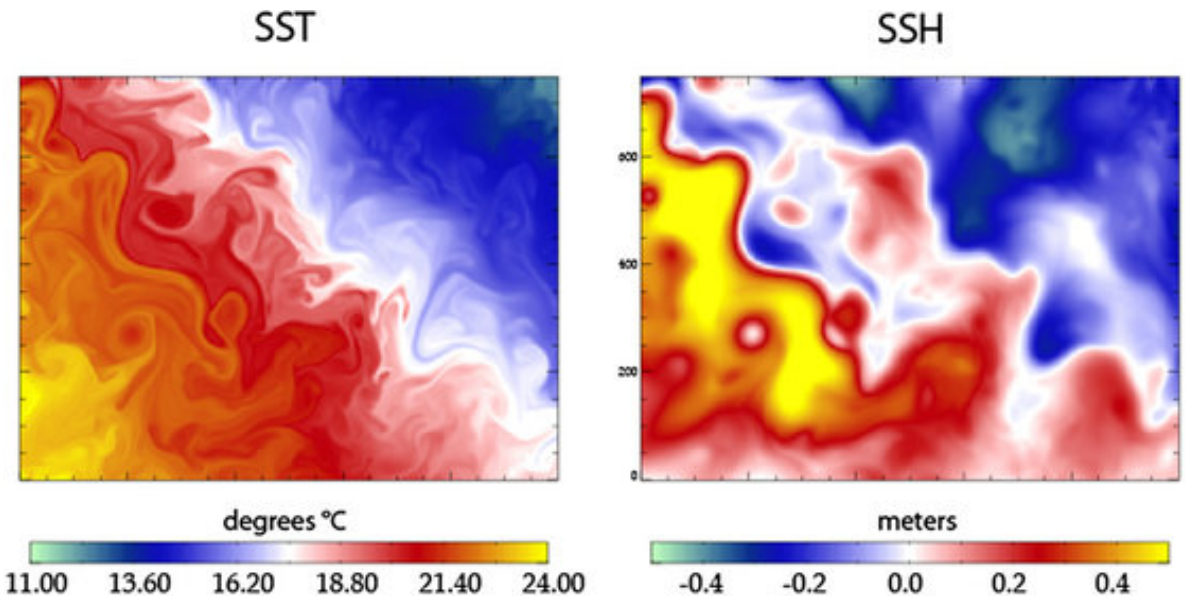




- images are matrices $\mathbf{X} \in \mathbb{R}^{D \times N}$, D is the number of bands, N is the number of pixels
- Principal components are vectors $\mathbf{u}_k \in \mathbb{R}^D$, and coefficients are vectors $\boldsymbol{a}_k \in \mathbb{R}^N$ (can be displayed as images)
- $\mathbf{X} = \mathbf{UA}$,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