Geometrical Methods of Machine Learning

prof. Alexander Bernstein

Course introduction: motivation and content

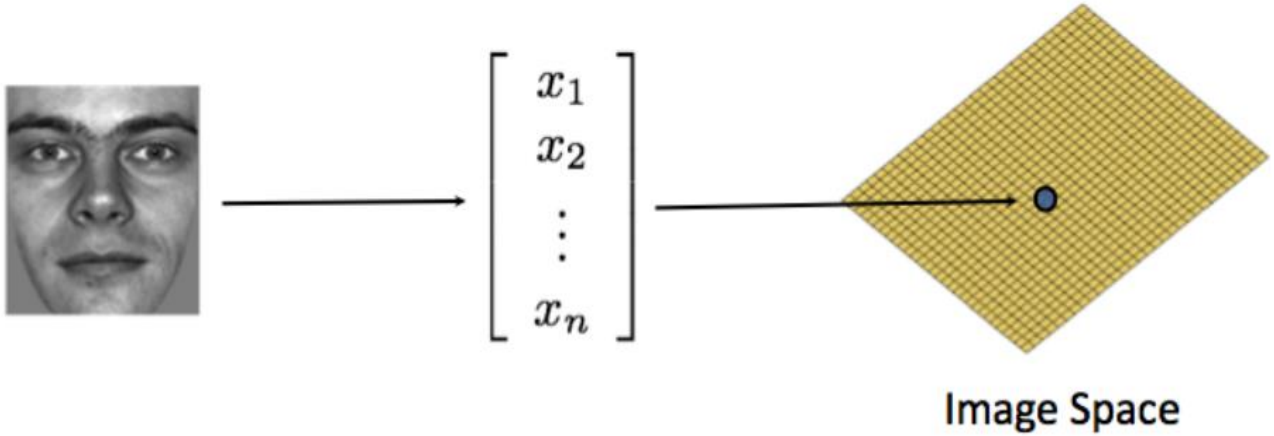
1. **The world is multidimensional**
2. **Multidimensional data are difficult to use**
3. **Machine Learning/Data Analysis tasks can be solved for real multidimensional data: why, when and how**
4. **Machine Learning problems are fundamentally geometric in nature**
5. **Course: short information**

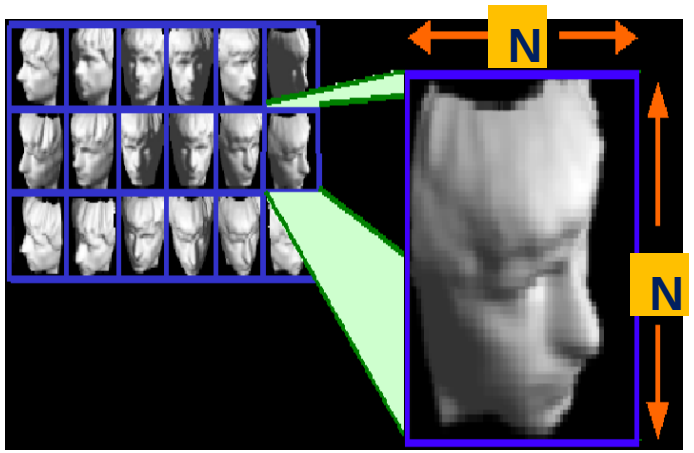
The world is multidimensional: Machine Learning/Data Analysis tasks deal with real-world data which are presented in multidimensional spaces

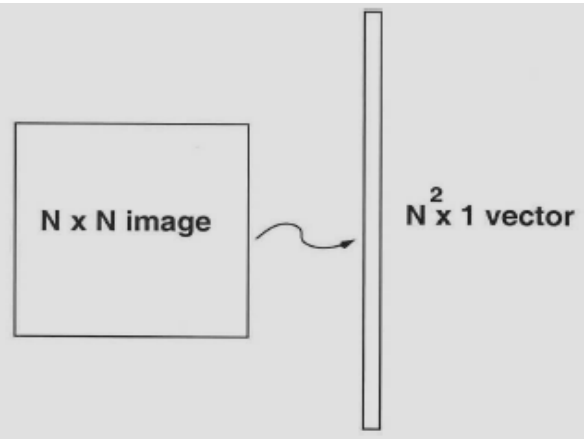
Example 1: Images as the outputs of a digital camera:

vectors in Image space with components specifying light intensities at an image pixels

1.1. Faces as face-vectors in high-dimensional Face space

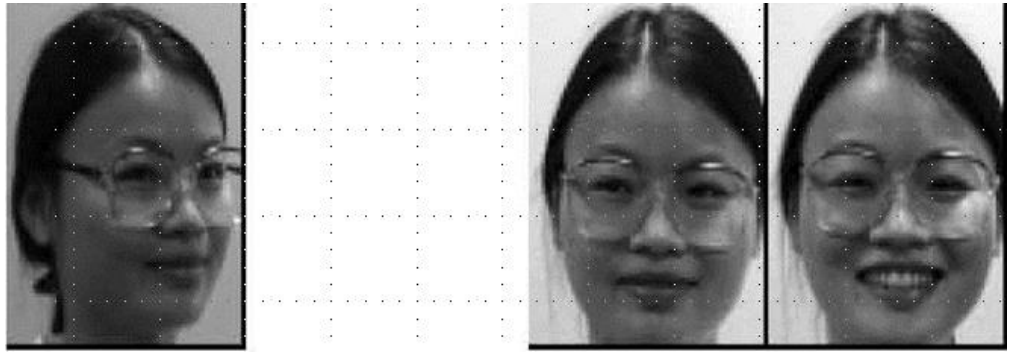






Gray-scale face description at 1024×1024 pixels:

face-vector with dimension $p = 2^{20} \approx 10^6 = 1\,000\,000$

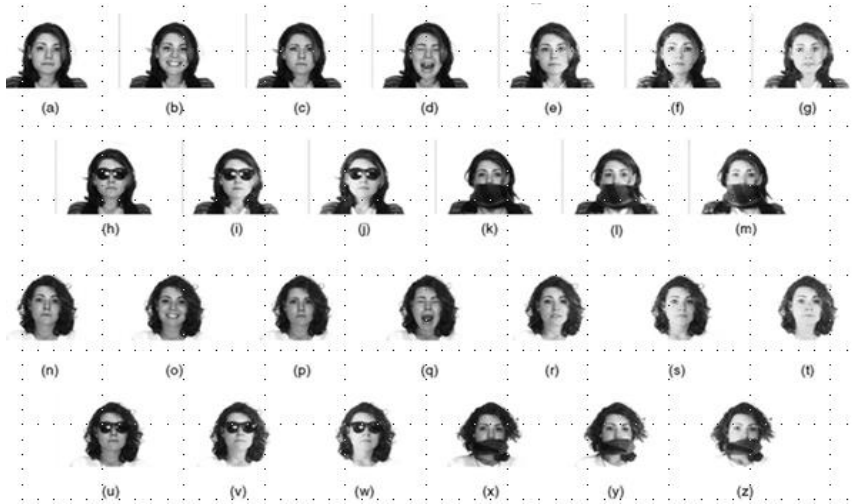


Face authentication

Emotion recognition

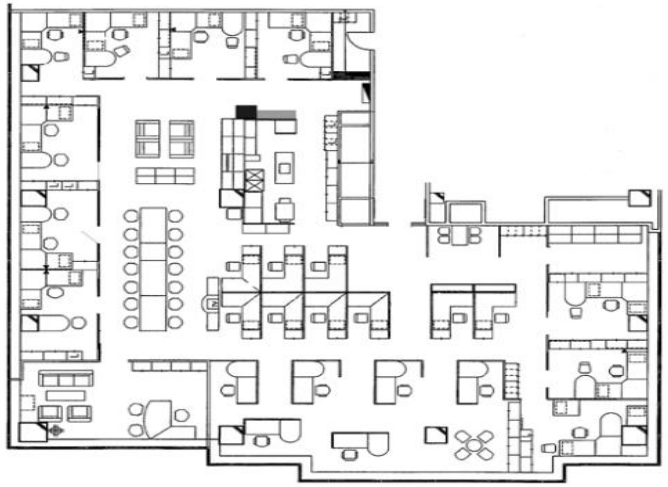
Face detection

Search

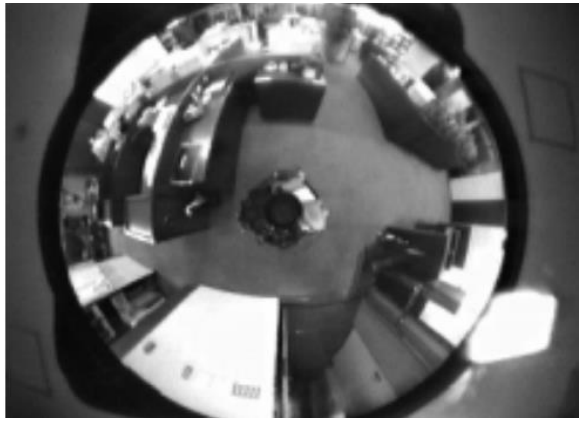


1.2: Mobile robot self-localization from captured images

Mobile robot with omnidirectional camera



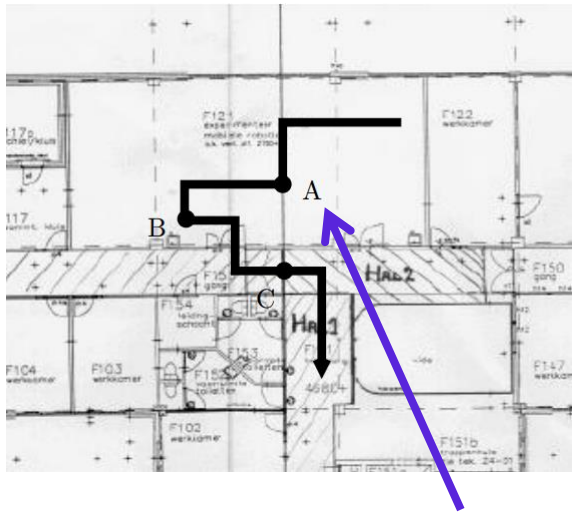
Workspace



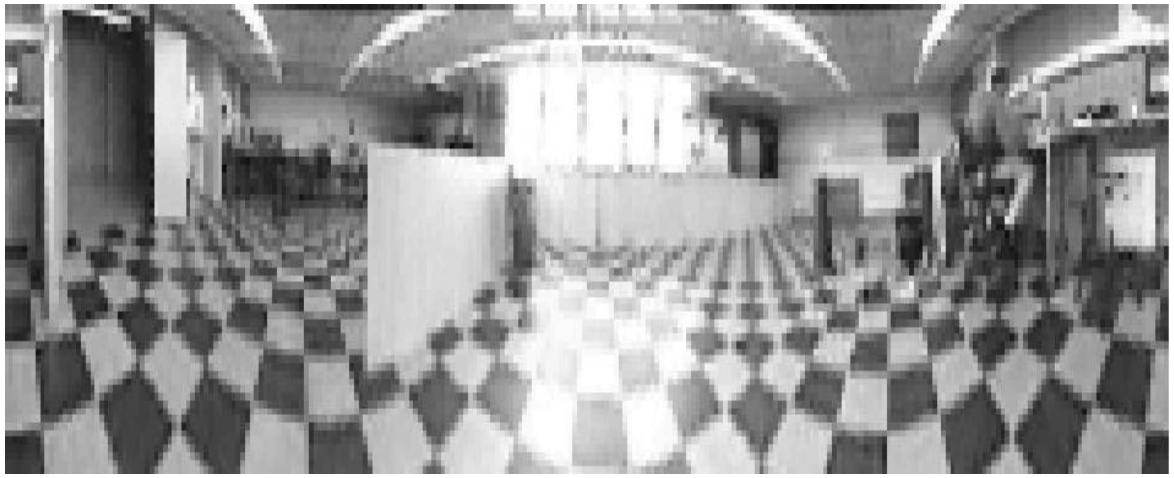
Captured omnidirectional image

360 degrees panoramic image derived from omnidirectional image

Task: Estimate robot position from the image captured in this position



Robot **position (A)**



Panoramic image (X) from position A in robot trajectory

Robot Localization problem: to find Robot position **(A)** from image **X** captured in **position A**

Dimensionality p for panoramic image-vector **X**:

- omnidirectional camera: 640×256 pixel images - p = 163840
- camera with steerable orientation: 96×72 pixel images - p = 6912

Formal problem statement:

Robot localization: $\theta = (\theta_{RC}, \theta_{RO}) \in \Theta \subset R^3$ - **Localization space**

$\theta_{RP} \in \mathbf{Y} \subset R^2$ (**R**obot **C**oordinates on **workspac**e **Y**) + $\theta_{RO} \in R^1$ (**R**obot **O**rientation)

Captured images: p**-pixel image (image-vector)** X captured by visual system with localization $\theta \in \Theta$ and modeled by unknown **Image modeling function** $X = \varphi(\theta)$ defined on Localization space Θ

Appearance space: $\mathbf{M} = \{X = \varphi(\theta), \theta \in \Theta\} \subset R^p$ consists of images which may be **captured under all possible localizations** $\theta \in \Theta$

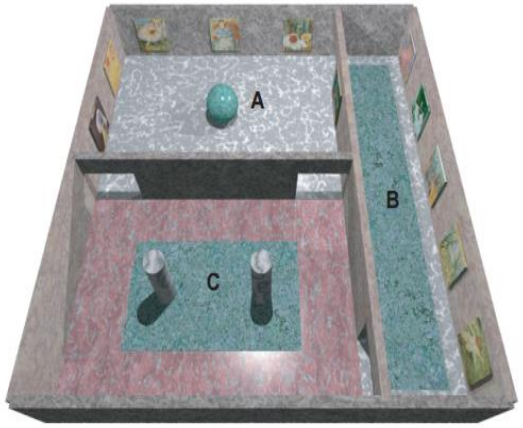
Assumption: modeling function $\varphi: \Theta \rightarrow \mathbf{M}$ is **one-to-one.**

Robot Localization problem: to construct **inverse function** called **Localization function**

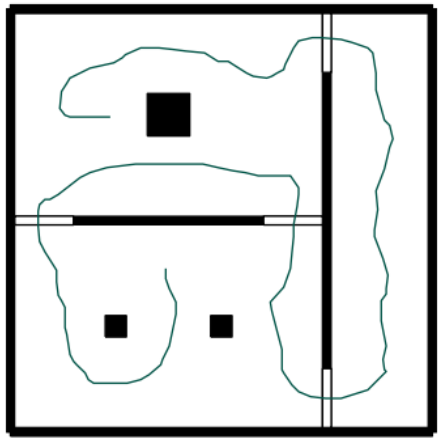
$$\psi = \varphi^{-1}: \mathbf{M} \rightarrow \Theta$$

from Appearance space to Localization space

Machine learning framework



Workspace



Robot trajectory

Panoramic images taken from chosen robot positions

Using training data - captured images $\{X_i\}$ in known positions $\{\theta_i\}$ -

to estimate an unknown robot position $\theta = \psi(X)$ from a newly acquired image X

Regression Image modeling problem:

given **Training data**

$$\mathbf{S_n} = \{(\theta_i, X_i = \varphi(\theta_i)), i = 1, 2, \dots, n\}$$

consisting of images $\{X_i = \varphi(\theta_i) \in \mathbf{M}\}$ captured in chosen known localizations $\{\theta_i \in \mathbf{\Theta}\}$**,**

to estimate the **Image modeling function** $X = \varphi(\theta)$ **- high-dimensional output**

Regression Robot localization problem:

given **Training data**

$$\mathbf{S_n} = \{(X_i, \theta_i = \psi(X_i)), i = 1, 2, \dots, n\}$$

consisting of known values $\{\theta_i = \psi(X_i) \in \mathbf{\Theta}\}$ of unknown function ψ in known input values $\{X_i \in \mathbf{M}\}$:

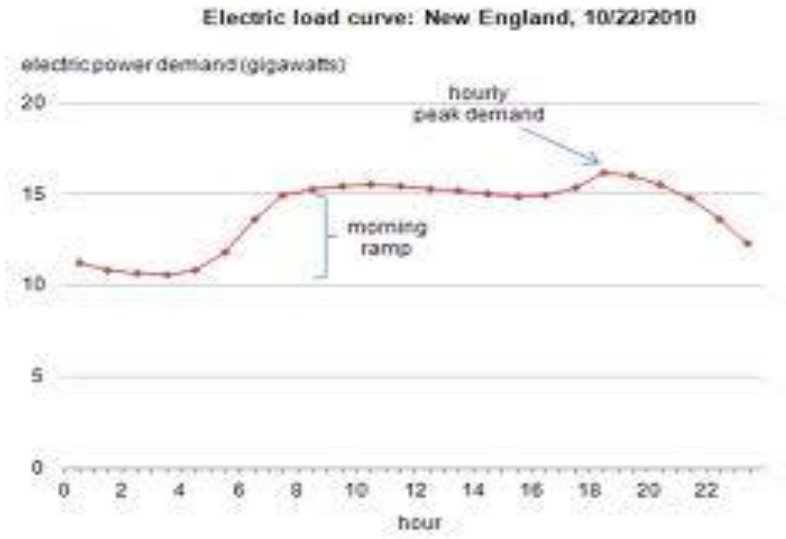
to estimate the **Localization function** $\theta = \psi(X)$ **- high-dimensional input**

Example 2: Electricity price forecasting

Electricity “daily prices”: a multidimensional ($p = 24$) time series (called ‘electricity price curve’)

$$\mathbf{X}_t = (X_{t1}, X_{t2}, \dots, X_{t,24})^T \in R^{24}, \quad t = 1, 2, \dots, T$$

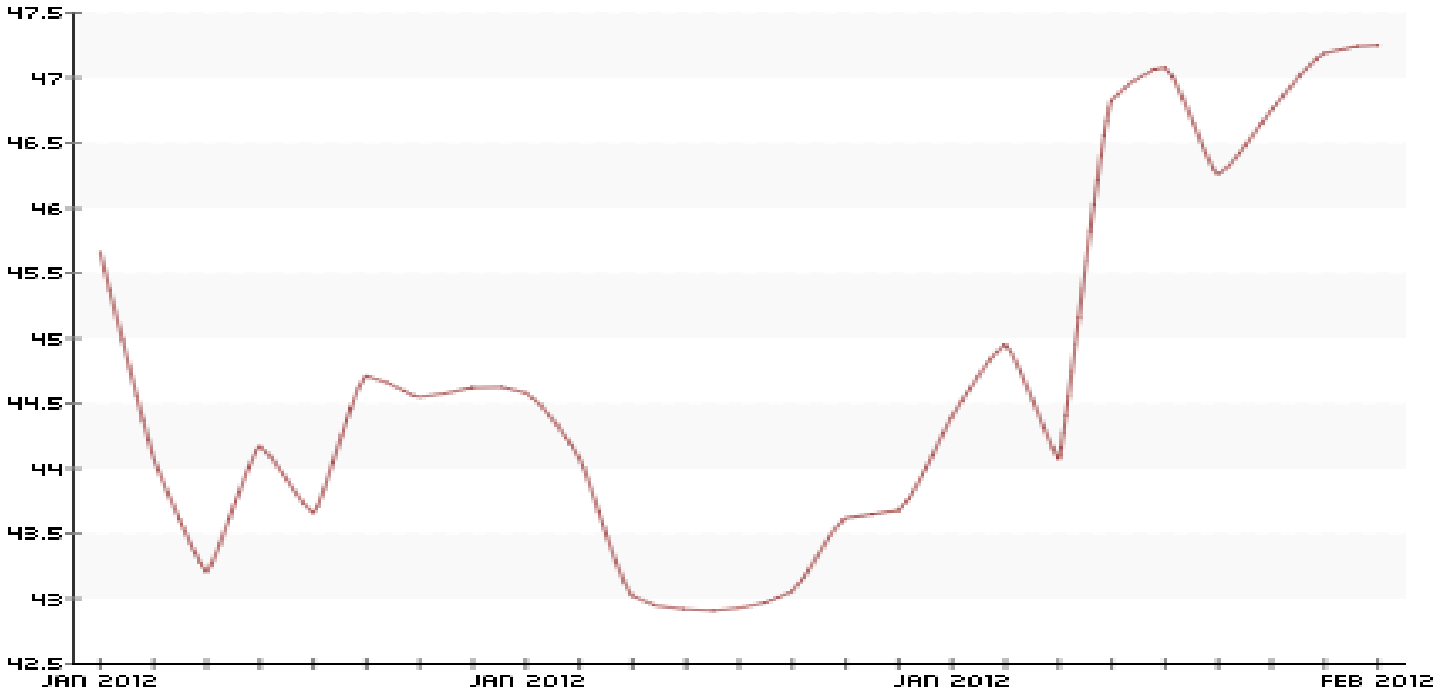
consisting of “hour-prices” in the course of day t.



Based on the ‘daily prices’ vectors

$$X_{1:T} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T)$$

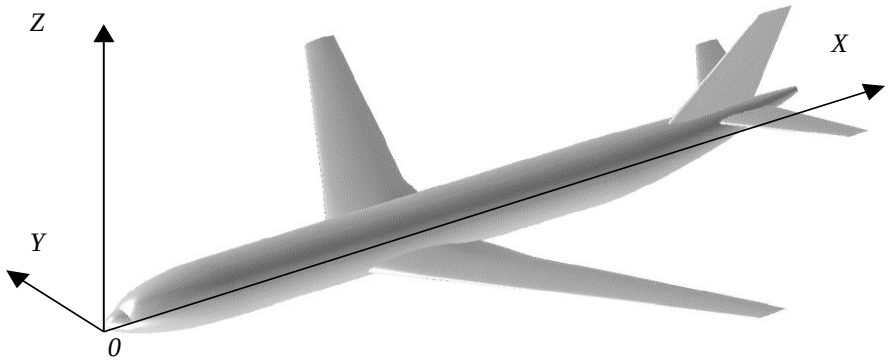
up to day T, to construct a forecast $\widehat{\mathbf{X}}_{T+1}$ for $\mathbf{X}_{T+1}$

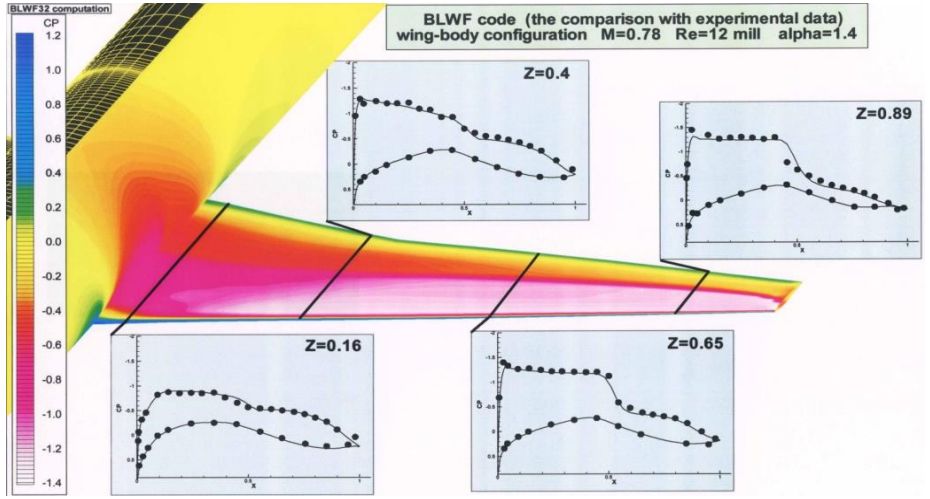


Example 3: wing shape optimization

- Design variable: wing geometry
- Optimized function: lifting force,

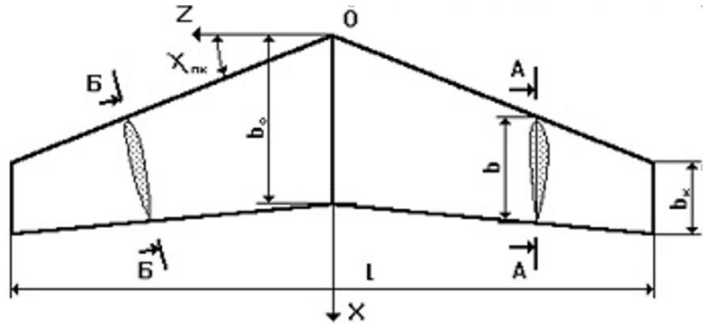
under given flight regime (Max number, angle of attack) and constrained aircraft's drag



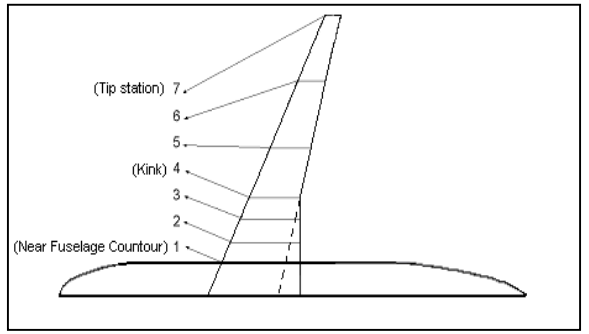


Wing geometry description (shape design variables):

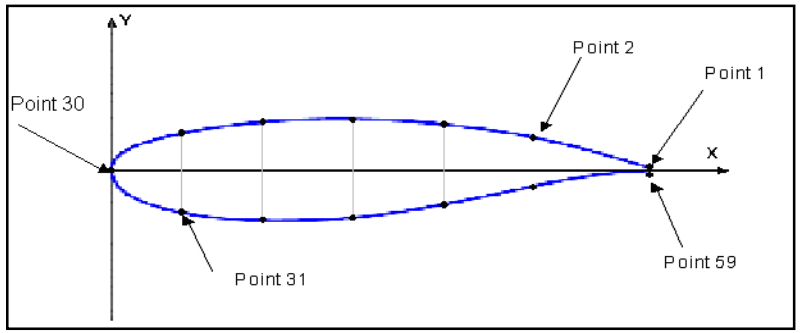
- wing in plan description (a few parameters)



- a number of **p**-dimensional **detailed descriptions** of wing airfoils in various wing cross-sections



Airfoil description: coordinates of points lying densely on the airfoils' contours



The dimension p varies usually in the range from 50 to 200; a specific value of p is selected depending on the required accuracy of airfoil description.

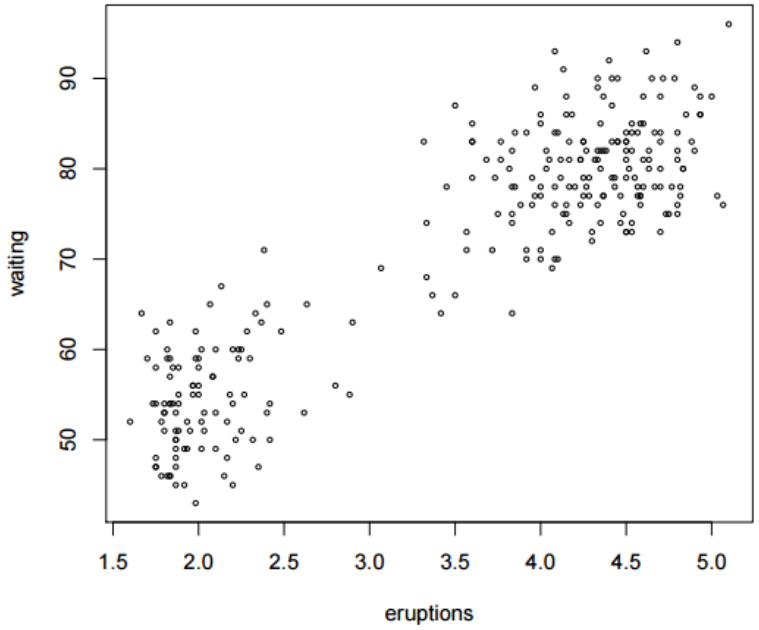
Wing description on Figures: 4 + 7× 59 = 417

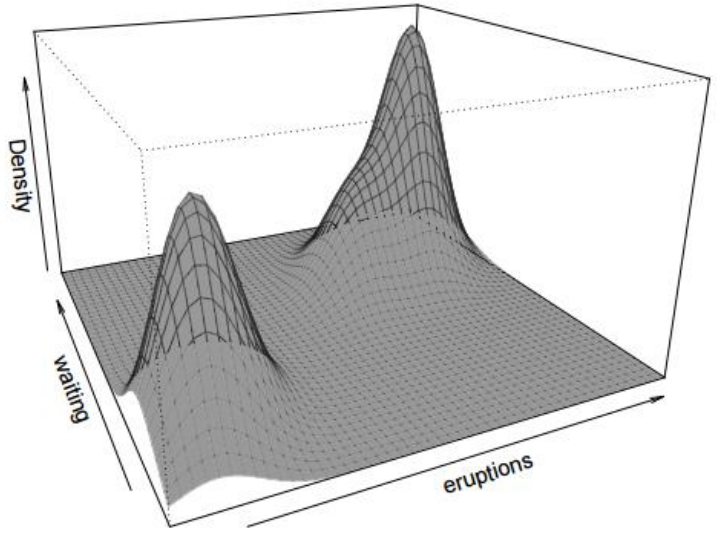
Example 4: Clustering and multimodal density estimation in Data analysis (p = 2)

'Old Faithful' data set:

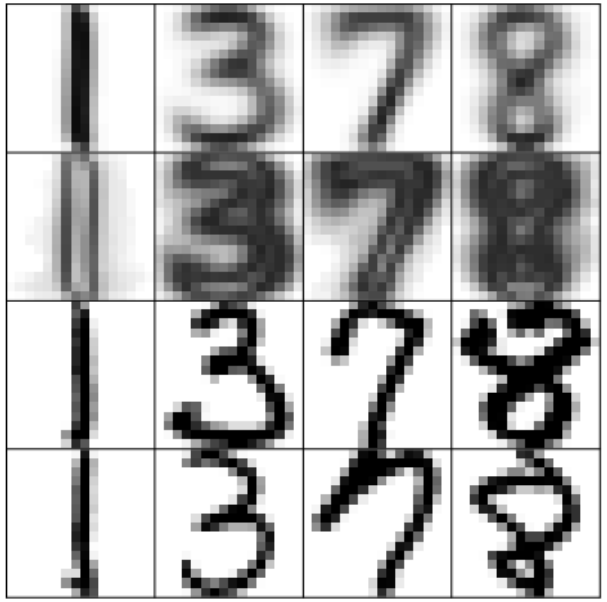
- the waiting time between eruptions (waiting)
- the duration of the eruptions (eruptions)

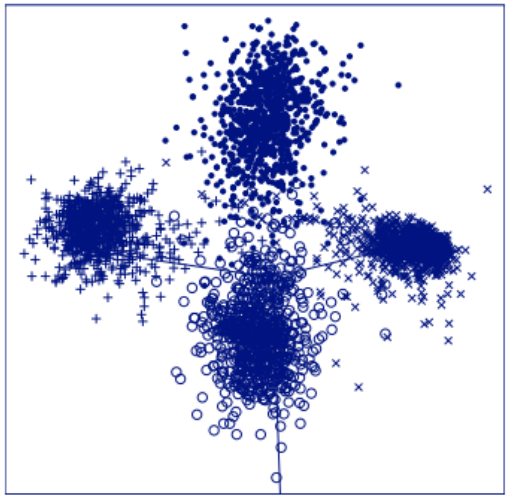
from Old Faithful geyser in Yellowstone National Park, Wyoming, USA

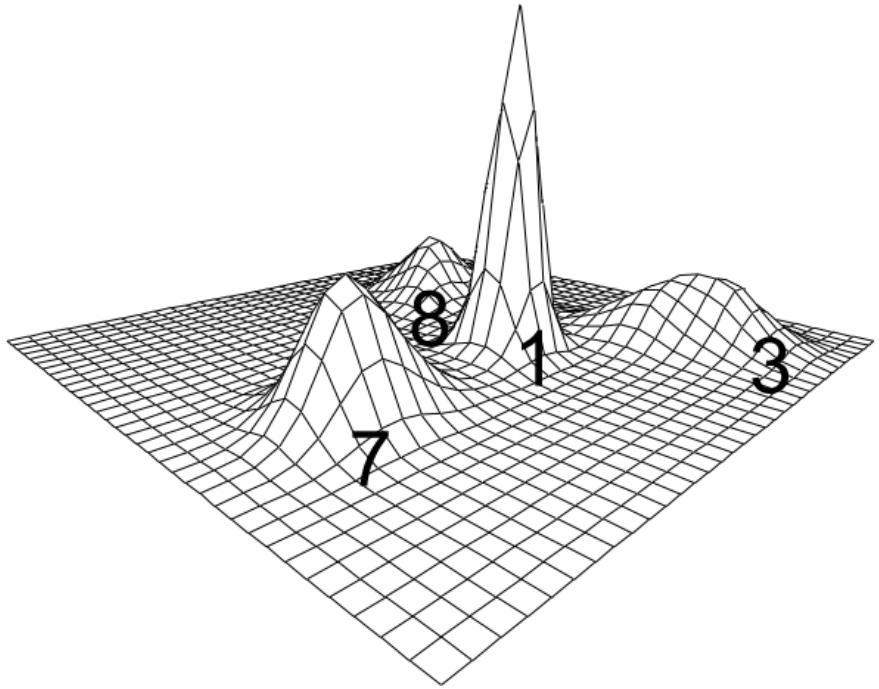




Classification and multimodal density estimation in Data analysis







Multidimensional data are difficult to use: *Computational complexity*, *Curse of dimensionality*, *Empty space phenomenon*, etc.:

- an obstacle to the use of many Data Analysis techniques for solving various Machine Learning tasks for multidimensional data

Computational complexity (number of calculations, numerical errors, etc.).

Example 1.1: Faces classification

Usage of a multilayer perceptron in the classification problem for 1024×1024 face-images results in multiple calculation the weights for components of ~ 1 000 000-dimensional face-vector (for each sigmoid!) whose total number is exceedingly large.

Example 3: wing shape optimization

A building a surrogate model (the solution of nonlinear regression problem) for lifting force, drag, etc., with its subsequent optimization for nonlinear functions of least ~ 500 variables doesn't have satisfactory numerical solution.

The ‘curse of dimensionality’

Bellman (1961): in approximation (optimization, numerical integration, etc.) tasks for a function of p variables, **in the absence of simplifying assumptions** (we know only that it is Lipschitz, say), we need order $(1/\varepsilon)^p$ evaluations of the function on a grid to obtain an approximation (respectively, optimization, numerical integration, etc.) with uniform approximation error ε

Known theoretical result (Ibragimov, Khasminskii (1979); Stone (1982)):

If $\mathbf{F} = \{\psi: [0, 1]^p \to R^1, \psi \text{ is Lipschitz}\}$, then for any estimator $\hat{\psi}$ of any kind for $\psi(X)$ from n known measurements $\{(X_i, \psi(X_i))\}$:

$$\sup_{\psi \in \mathbf{F}} E(\psi(X) - \hat{\psi}(X))^2 \geq \text{Const} \times n^{-2/(2+p)}, \quad n \to \infty.$$

The lower bound is nonasymptotic!

The error cannot achieve a convergence rate faster than $n^{-1/(2+p)}$

Numerical example:

$p = 10$: $n = 10\,000$ measurements to achieve given accuracy

$p = 20$: $n \sim 10\,000\,000$ measurements to achieve the same accuracy

Example 1.2: mobile robot self-localization from captured images

The solution to Regression robot localization problem: estimating the Localization function

$$\psi: \mathbf{M} \subset R^p \rightarrow \mathbf{\Theta} \subset \infty\ R^3 + \text{noise}$$

from known training dataset $\mathbf{S_n} = \{(X_i, \theta_i = \psi(X_i)), i = 1, 2, \ldots, n\}$

Dimension p varies usually in the range from 2^{12} to 2^{18}

(p = 6912 and 163 840 in above examples)

Example 4: Clustering and multimodal density estimation in Data analysis

- $f(x)$ - unknown probability density function of random vector $X \in R^p$
- $\{X_1, X_2, \dots, X_n\}$ - sample (i.i.d)
- $\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h_{i,n}^n} K\left(\frac{\|X_i - x\|}{h_{i,n}}\right)$, $h_{i,n} = O(n^{-1/(p+4)})$ - standard kernel density estimation

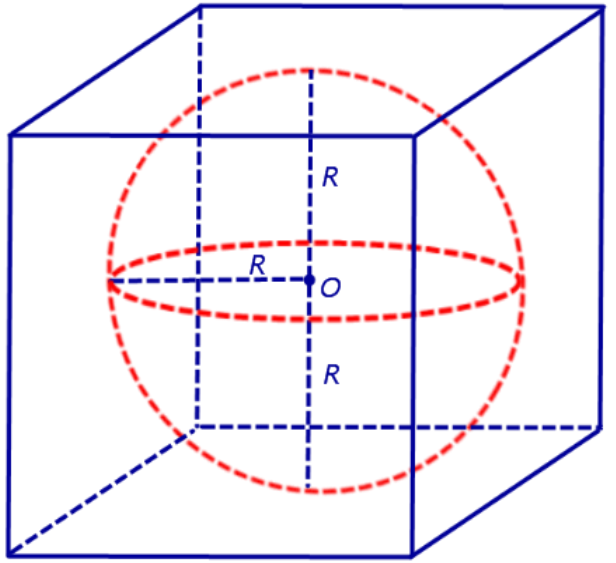
$$MSE(\hat{f}(x)) = E(\hat{f}(x) - f(x))^2 \sim O(n^{-4/(p+4)})$$

Empty space phenomenon - ‘responsible’ for the curse of the dimensionality

Scott and Thompson (1983): high-dimensional spaces are inherently sparse.

The geometry of high-dimensional spaces

1. Hypervolume of hypercubes and spheres inscribed in a hypercube



$C(R, p) = [-R, R]^p$ - p-dimensional cube, $V(C(R, p)) = (2R)^p$

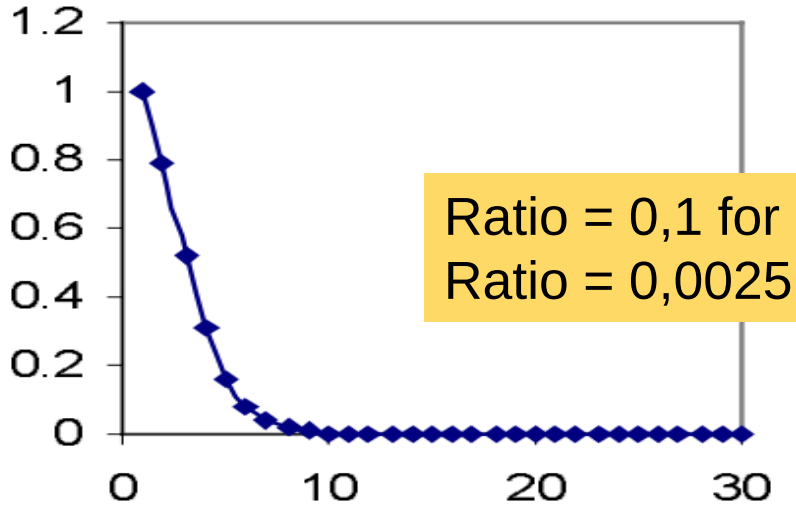
$B(R, p) = \{ x \in R^p: |x| \leq R\}$ - p-dimensional ball, $V(B(R, p)) = \frac{\pi^{p/2} \times R^p}{\Gamma\left(\frac{p}{2}+1\right)}$

With increasing dimension, volume of the hypercube concentrates on its corners:

$$\lim_{p \to \infty} \frac{V(B(R, p))}{V(C(R, p))} = 0$$

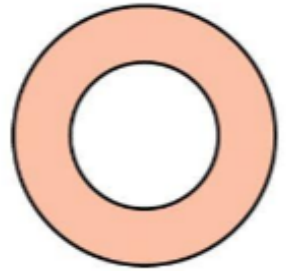
$V(C(1/2, p)) = 1$

$\lim_{p \to \infty} V(B(1/2, p)) = 0$



Ratio = 0,1 for p = 6
Ratio = 0,0025 for p = 10

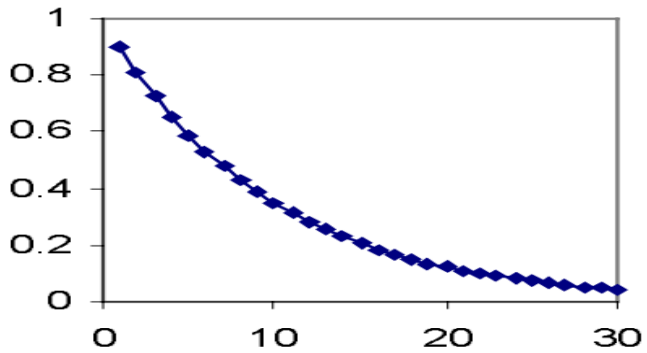
2. Hypervolume of a thin spheical shell



With increasing dimension, volume of the ball concentrates close to its surface

(on a thin ε-shell, ε is small):

$$\lim_{p\to\infty} \frac{V(B(1, p)) - V(B(1-\varepsilon, p))}{V(B(1, p))} = 0$$



3. Tail probability of the multivariate normal

- $X \in R^p \sim N(0, \sigma^2 \times I_p)$
- $R_{0,95}(\sigma, p)$: $P\{X \in B(R(\sigma), p)\} = 0,95$

The radius $R_{0,95}(\sigma, p)$ grows as the dimensionality p increases:

$$R_{0,95}(\sigma, 1) = 1,96\sigma$$

$$R_{0,95}(\sigma, 6) = 3,54\sigma$$

The sample “neighbors” of a specific point, which are used in many machine learning procedures, get far away with dimensionality grows and don’t provide expected ‘locality’

Example 4: multivariate density estimation

Standard kernel multivariate density estimators require “to reach out farther” to **provide sufficient number of points (“neighbors”)** in the neighborhood of specific point but **the locality can be lost**

4. Diagonal of a hypercube

- $C(1, p) = [-1, 1]^p$ - p-dimensional cube
- half-diagonal: any segment E from center $(0, 0 \dots, 0)^T \in R^p$ to one of its corners $(\pm1, \pm1, \dots, \pm1)^T \in R^p$
- any coordinate axis $E_k = ((0, 0 \dots, 0, 1, 0, \dots, 0)^T \in R^p$
- $\theta_k(p) = \frac{(E,E_k)}{\|E\| \times \|E_k\|} = \frac{\pm1}{\sqrt{p}}$ - angle between arbitrary half-diagonal and arbitrary axis

Half-diagonals are nearly orthogonal to all coordinate axes for large p: $\lim_{p \to \infty} \theta_k(p) = 0$

Visualization of high-dimensional data by performing projection to only a 2 or 3 coordinate axes

- can be misleading for a cluster of points lying near each diagonal line of the space: they are plotted near the origin and mislead our perception
- a cluster of points lying near a coordinate axis can be plotted is some projections as intuitively expected

5. Concentration phenomenon: concentration of norms and distances

Theorem (P. Demartines, 1994): $\mathbf{X} = (X_1, X_2, \ldots, X_p)^T \in R^p$ - i.i.d., $E(X_1)^4 < \infty$, and $\mathbf{Y} = |\mathbf{X}|$. Then

$$E\mathbf{Y} = \sqrt{ap - b} + O(p^{-1})$$

$$Var(\mathbf{Y}) = b + O(p^{-1/2})$$

a, b - constants dependent on moments of X_1 and written in explicit form.

The norm of random vectors grows proportionally to $\sqrt{p}$ (as naturally expected), but the variance remains more or less constant for a sufficiently large p

From Chebychev's inequality: $P\{|\mathbf{Y} - E\mathbf{Y}| \geq c\} \leq \frac{Var(\mathbf{Y})}{c^2}$:

- the successive sampled p-dimensional vectors $\{\mathbf{X}\}$ yield almost the same norm
- the distance $|\mathbf{X} - \mathbf{X}'|$ between two sampled vectors $\mathbf{X}$ and $\mathbf{X}'$ is approximately constant

The nearest-neighbor search problem is difficult to solve in high-dimensional spaces

Other troubles

Example 2: Electricity price forecasting

Forecasting of 24-dimensional ‘daily-prices’ time series:

- well-known **Auto-Regressive Integrated Moving Average** (ARIMA) technique is always applied to low dimensional multivariate data and often finally turns out to be too complicated to be useful in multivariate ($p = 24$) case
- another intuitive way of prediction is “to treat the electricity prices at the same hour over the days as a **univariate time series** (by ARIMA, for example), and then apply the univariate time series forecasting to 24 series” - **ignores** both the correlation among the prices at different hours in a day and **‘the integrity’ of the price curve**.

Machine Learning/Data Analysis tasks can be solved for real multidimensional data - when and why?

Properties of real-world high-dimensional data which help avoiding

the *curse of dimensionality* and *empty space phenomena: w*hich and how?

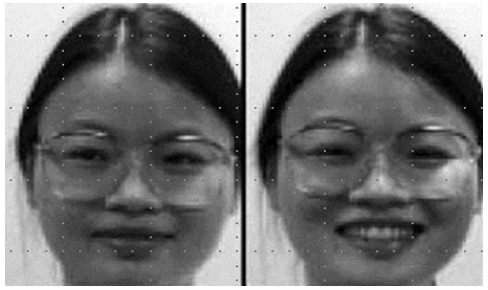
Processes data (speech signal, images, or patterns in general) – a collection of vectors

- $X = \begin{pmatrix} x_1 \\ \cdots \\ x_p \end{pmatrix} \in R^p$ - multidimensional vector
- $x_1, x_2, \ldots, x_p$ - its components (features, variables, attributes, …)
- $\{X_1, X_2, \ldots, X_n\}$ – training or learning data (samples, examples, patterns, prototypes, …)

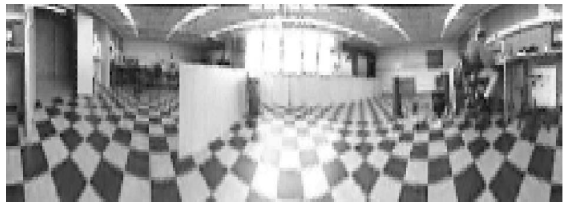
- Y – **variable 'of interest'** (information, knowledge, 'object', etc.) whose 'value' should be extracted from given vector X (input, measured, observed, available, etc., vector)
- the values $\{Y_1, Y_2, \ldots, Y_n\}$ of variable of interest Y for training data can be known (given by an "oracle" or "professor") - supervised learning

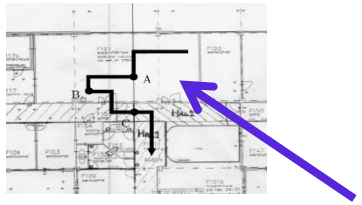
Face authentication: $X = (Z, Z')$ - two faces,

$Y = f(X)$ – 'answer': $Y = 0$ – 'YES', $Y = 1$ – 'NO'



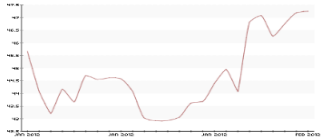
Robot localization:

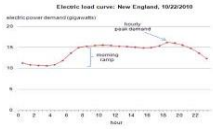




X – captured image-vector $Y = f(X)$ - (position, orientation) $\in R^3$

Electricity price forecasting:





$X = X_{1:T}$ – time series (daily prices up to day T) $Y = f(X) = \hat{X}_{T+1}$ - a forecast for $X_{T+1} \in R^{24}$

Machine Learning:

to construct 'variable of interest' $Y = f(X)$ from high-dimensional data X from training data

Model statistical task:

- U and V - random variables, $\hat{U} = f(v)$ – a forecast for U from given value $V = v$
- optimal forecast: $\hat{U}(V) = E(U \mid V)$ – conditional mathematical expectation
- optimal **linear** forecast: $\hat{U}(V) = EU + \rho \times \frac{\sqrt{Var(U)}}{\sqrt{Var(V)}} \times (V - EV)$, $\rho \in [-1, 1]$ – correlation coefficient
- Error: $E(\hat{U}(V) - U)^2 = (1 - \rho^2) \times Var(U)$

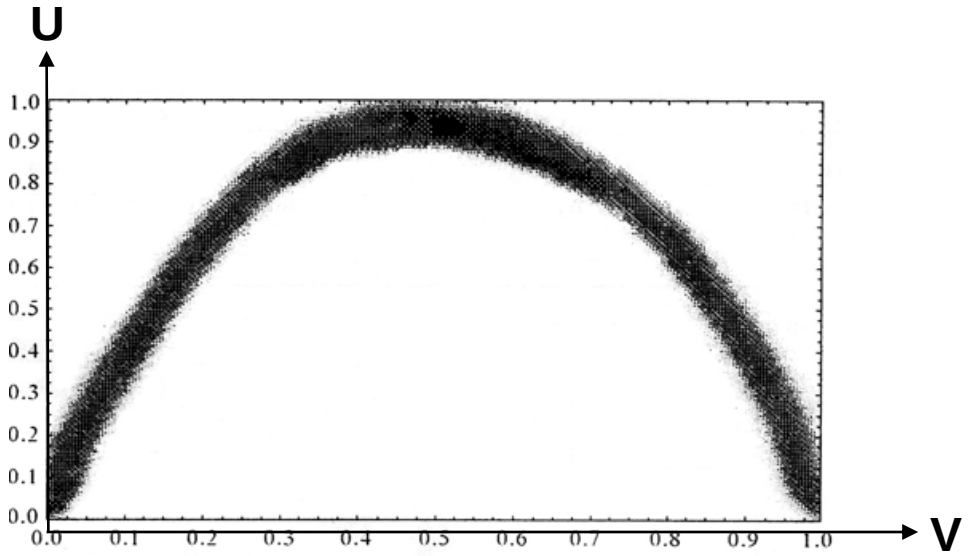
Case $\rho = 0$: $E(\hat{U}(V) - U)^2 = Var(U)$

- a known value of variable V does not reduce “the uncertainty” of the variable U
- variable V is **“irrelevant”** for variable U

Case $\rho \approx 1$ (highly correlated case)**:** $E(\hat{U}(V) - U)^2 \approx 0$

- a known value of variable V eliminates “the uncertainty” of the variable U
- variable U can be replaced (with small error) by its prognosis $\hat{U}(V)$ **with small error**
- variables U and V are related by a **non-random** linear relationship $U = aV + b$ **with small error**
- variable U is “redundant” under known variable V

- the pairs $(U, V) \in [0, 1] \times [0, 1]$ do not fill the full square $[0, 1]^2$



1. Feature selection: a removal from high-dimensional feature-vector:

- irrelevant features w.r.t. considered variable of interest using chosen ‘relevance measure’ (e.g., based on correlation with variable of interest)
- redundant features (which can be restored from the remaining features with required accuracy)
- noisy features (features have variation smaller than the measurement noise and thus are irrelevant)

Let:

- all remaining variables are relevant
- all the features, which can be restored from other (correlated) original features, are eliminated (e.g., one feature in each highly-correlated pair is removed)

2. Feature extraction:

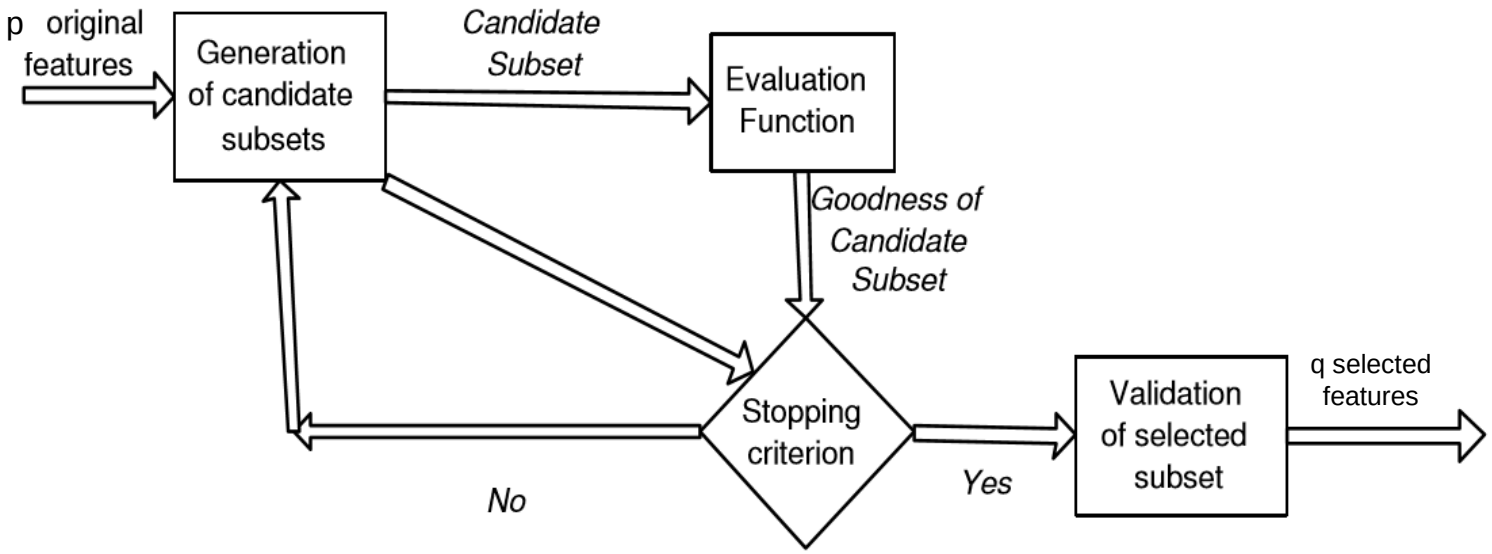
- **a construction** (after an Feature selection procedure) **NEW features** (transformed features, reduced features) **from remaining original features**
- number of new features less than number of remaining original features
- new features **preserve as much as possible information** contained in the remaining original features and meaningful for the considered variable of interest (information which User ‘wishes to catch’)

Feature selection: given a set $S = \{x_1, x_2, \dots, x_p\}$, $\#(S) = p$, consisting of p features, find a subset

$$S_0 \subset S, \#(S_0) = q$$

consisting of **the best** $q < p$ features (**by a removal** of irrelevant, redundant and noisy features) to optimize chosen **objective function** $J(S_0)$ (**evaluation function**, or **selection criterion**) over all other possible combinations of p features.

Feature selection process:



Examples of used objective functions:

1) Information measure reflects ‘information losses’; determines the information gain from a feature:

- the difference between prior uncertainty and expected posterior uncertainty using the feature

2) Dependence measure - ‘relevance measure’ of specific feature

- based on its correlation with variable of interest;
- quantifies the ability of the feature to explain (to predict) the value of the variable of interest

3) Distance Measure (is also known as separability, divergence, or discrimination measure)

4) Consistency Measure (prefers a consistent hypothesis definable over as few features as possible)

5) Classifier Error Rate Measure (the feature set giving the minimum classifier error rate is selected)

Feature selection as a linear projection

$X = \begin{pmatrix} x_1 \\ \cdots \\ x_p \end{pmatrix} \in R^p$ - multidimensional vector consisting of original features $\{x_1, x_2, \ldots, x_p\}$

$Z = \begin{pmatrix} x_{i_1} \\ \cdots \\ x_{i_q} \end{pmatrix} \in R^q$ - vector consisting of selected features $\{x_{i_1}, x_{i_2}, \ldots, x_{i_q}\}$

$$Z = P \times X$$

$q \times p$ **projection matrix** P maps the feature vector X into the q-dimensional space **spanned by coordinate axes** with selected numbers $\{i_1, i_2, \ldots, i_q\}$

Feature extraction/feature transform: given relevant features $S = \{x_1, x_2, \dots, x_p\}$, to construct $q < p$ **new** features $S_0 = \{z_1, z_2, \dots, z_q\}$ which are certain **linear or nonlinear** transform (mapping, projection)

$$Z = P(X): R^p \rightarrow R^q$$

of the original features S, to optimize certain chosen objective function $J(S_0)$ which reflects, for examples, the "errors" under restoring the original information content from the reduced features

Reduced features do not create **new information content** (w.r.t. considered variable of interest) that differs from the content contained in the original features, but only **represent it in a different form**

Original' information content can be restored from new transformed (reduced) features **with small error**

Feature extraction with linear mapping P - **Feature reduction:**

- all original features are used
- the transformed features are **linear** combinations of the original features: $Z = P \times X$
- Feature reduction is Feature selection when $q \times p$ projection matrix P maps the feature vector X into q-dimensional space spanned by **coordinate axes** corresponding to the selected features

Feature extraction (feature reduction, feature selection) is a research area at the intersection of statistics, databases, data mining, text mining, pattern recognition, machine learning, artificial intelligence, visualization, optimization, etc.

Each of these areas has its own way of looking at the problem. For example,

- **in pattern recognition** the problem of dimensionality reduction is to extract a small set of **new features** that recovers most of the variability of the data.
- **in text mining**, however, the problem is defined as selecting a small subset of words or terms (**not new features that are combination of words or terms**).

Feature extraction (feature reduction, feature selection) procedures for feature vector X can be constructed based on:

- domain knowledge, using explicit background or meaning of the features (not the subject of the course)
- "data-driven" feature extraction when projection P is learned from training data $\{X_1, X_2, \ldots, X_n\}$ using
  - ✓ training data $\{X_1, X_2, \ldots, X_n\}$ only - unsupervised learning
  - ✓ training data + values of variable of interest Y:

$$\{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$$ - supervised learning

1. All original (p-dimensional) features (components of high-dimensional vectors) are actually be governed by a few q, $q < p$, simple variables (“hidden causes”, “latent variables”, etc.)

Panoramic image $X \in R^p$ ($p = 2^{12} \div 2^{18}$) captured by mobile robot at position $\theta \in R^3$ is determined by latent variable θ: $X = \varphi(\theta)$

- all captured panoramic images (features) $\mathbf{M} = \{X\}$ captured by mobile robot at all possible positions $\Theta = \{\theta\}$ is 3-dimensional surface $\mathbf{M} = \{X = \varphi(\theta): \theta \in \Theta \subset R^3\} \subset R^p$ in high dimensional space R^p
- any 3-dimensional one-to-one parameterization $Z = F(X) \in R^3$ of the surface $\mathbf{M}$ gives the new reduced features

Feature space is q-dimensional surface in R^p ‘parameterized’ by latent variables

Original multiple features $\{X\}$ are indirect measurements of small number of an underlying sources $\{\theta\}$ which typically cannot be directly measured

From a geometrical point of view, when two or more feature depend on each other:

- their joint distribution **does not span the whole space**
- the dependence induces some **low-dimensional structure** in the distribution in the form of a geometrical locus that can be seen as a kind of object in the space
- **feature transform / dimension reduction** aims at giving a new representation of these objects while **preserving their structure**

‘Latent variables’ is not only case that results in low-dimensional intrinsic structure of the Feature space.

2. **Empirical fact:** real-world observed or measured features (obtained from real sources) **occupy only a very small part in the high-dimensional ‘observation space’** R^p -

Feature space has low-dimensional intrinsic structure

- ✓ it can be ‘described’ by small number (q) of parameter which are desired ‘reduced features’ which “are coming” from a much lower dimensional intrinsic structure
- ✓ Feature space is “low-dimensional surface” embedded in the ambient high-dimensional space R^p

In contrast to ‘latent variables’ case (Feature space is exactly q-dimensional surface in R^p parameterized by latent variables), low-dimensional structure of Feature space is, in general, an ‘empirical’ fact, sometimes without any “physical” explanations

Example 1.1. Face-vectors in high-dimensional Face space

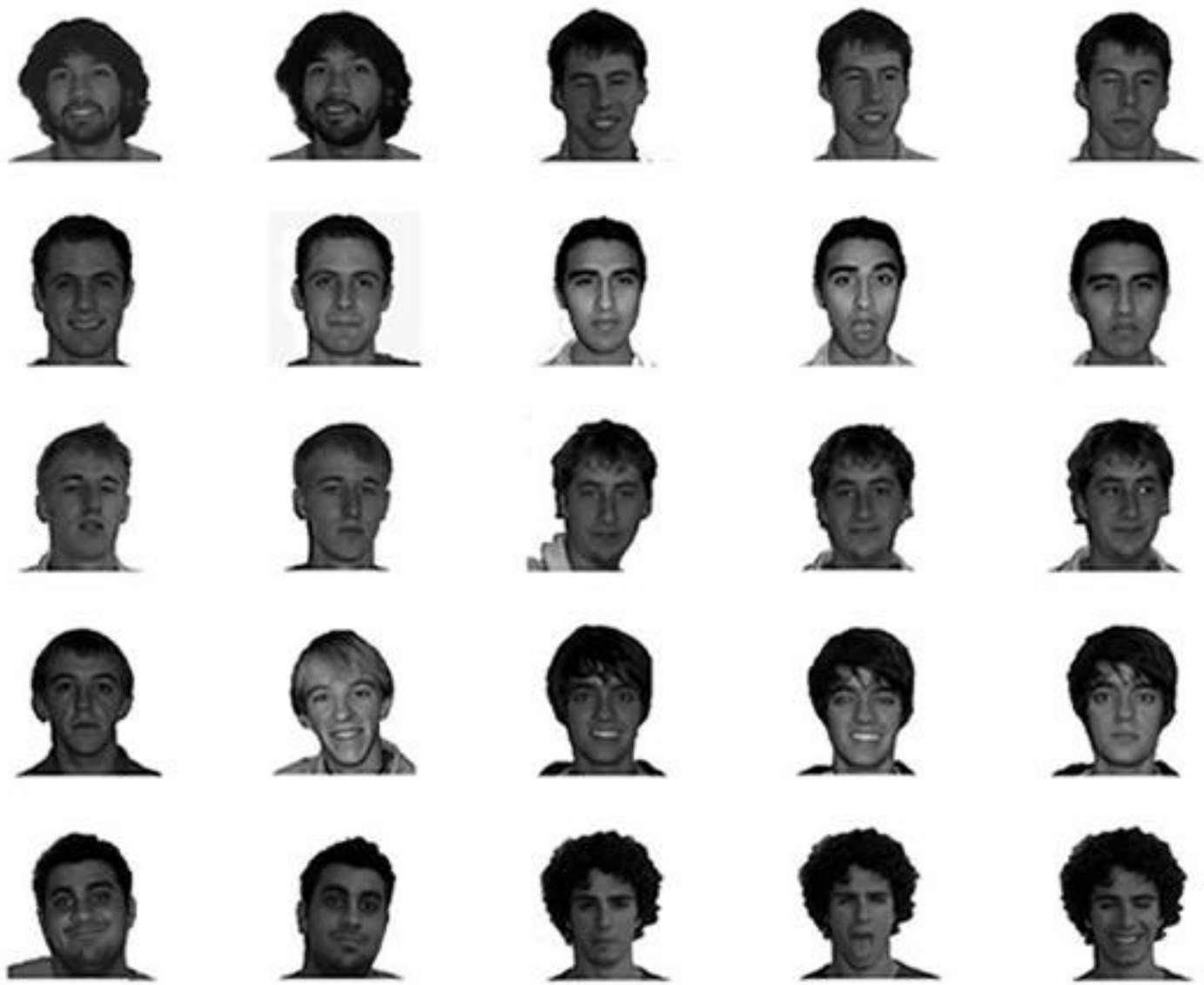
Original face described by 10^6-dimensional vector

Faces can be described no more than 100 features - ‘intrinsic dimensionality’ of the Face space is less than 100

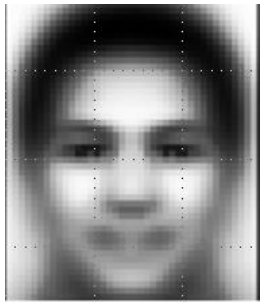
Left to right: the same face described by q reduced features

q = 84 q = 40 q = 20 q = 3 q = 2 q = 1

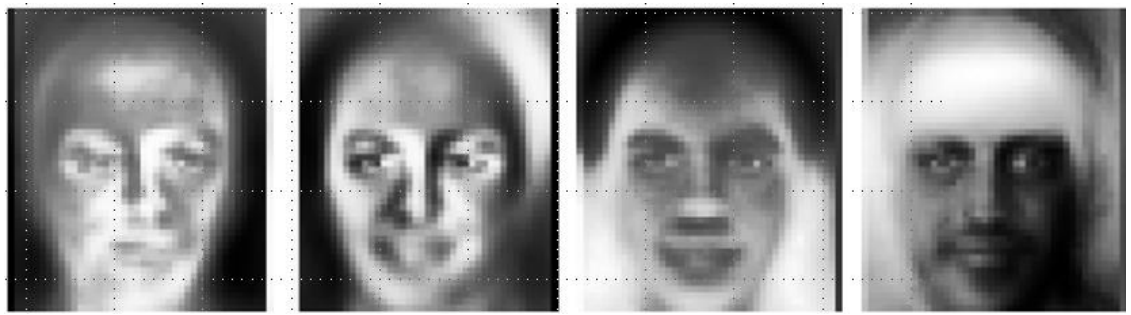
Training Face dataset: each face has dimension p = 2061



‘Mean’ face

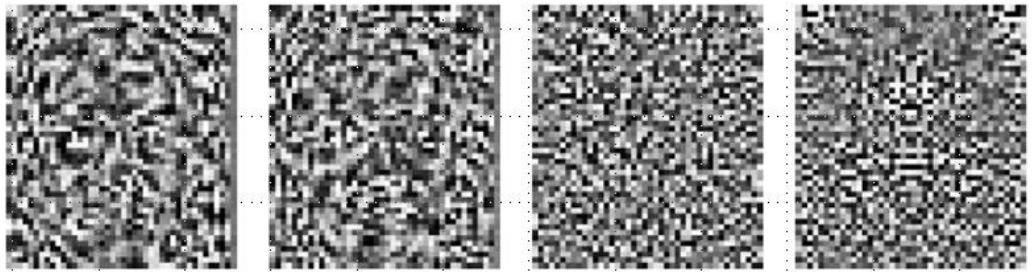


and 2061 “principal faces” (eigenfaces



first 4 ‘principal faces’

‘principal faces’
15, 100, 200, 250, 300



‘principal faces’
400, 450, 1000, 2000

Original face

Reconstructed face: $X^* = \text{mean face} + z_1 \times \mathbf{e}_1 + z_2 \times \mathbf{e}_2 + \ldots + z_8 \times \mathbf{e}_8$

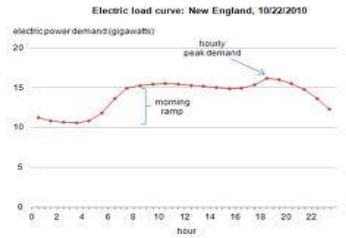
- linear combination of 8 first principal faces $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_8\}$

- projection onto 8-dimensional space in R^{2061} spanned by $\{\mathbf{e}_1, \mathbf{e}_2, \ldots \mathbf{e}_8\}$

$(z_1, z_2, \ldots, z_8)$ - 8-dimensional reduced feature vector

$y_1 \times \mathbf{e}_1$ $y_2 \times \mathbf{e}_2$ $y_3 \times \mathbf{e}_3$ $y_4 \times \mathbf{e}_4$ $y_5 \times \mathbf{e}_6$ $y_6 \times \mathbf{e}_7$ $y_7 \times \mathbf{e}_7$ $y_8 \times \mathbf{e}_8$

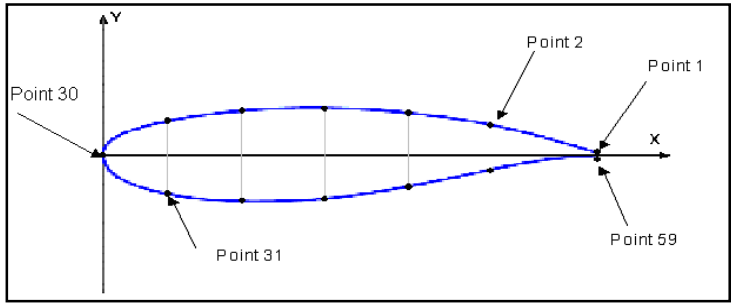
Example 2: Electricity price forecasting



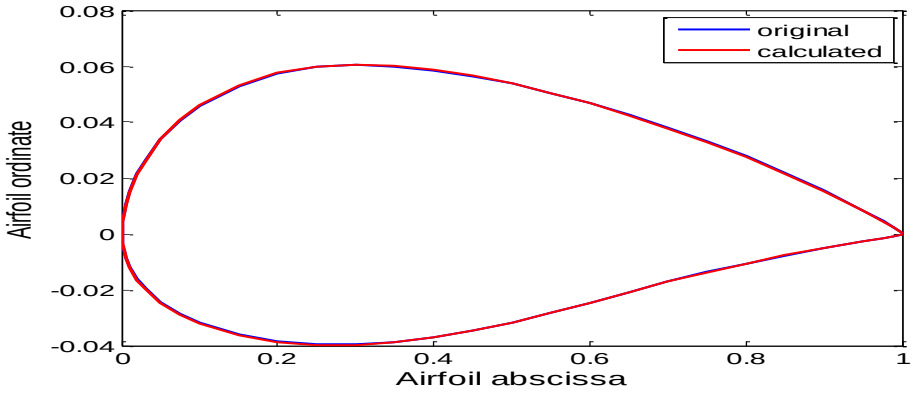
24-dimensional Electricity "daily prices" vectors $\mathbf{X}_t = (X_{t1}, X_{t2}, \ldots, X_{t,24})^T \in R^{24}$

can be described by **4-dimensional reduced features** $z_t = P(\mathbf{X}_t)$, P - **nonlinear** projection

Example 3: wing shape optimization



59-dimensional airfoil descriptions can be described by **6 reduced features**



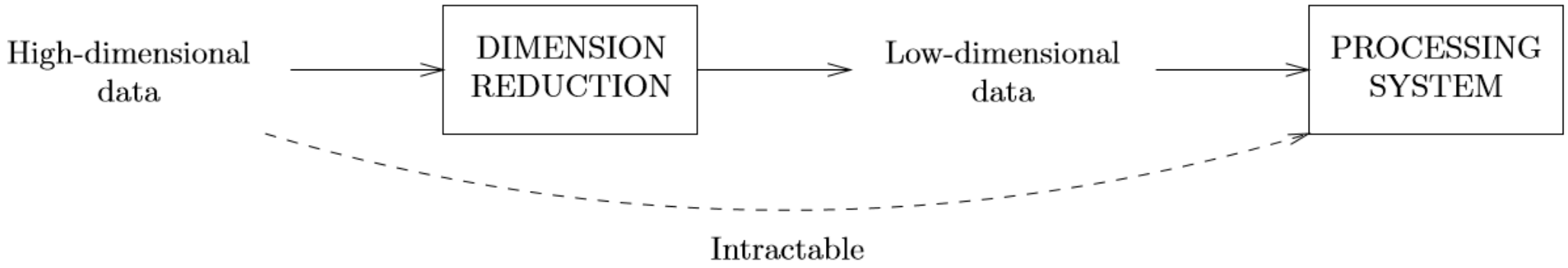
Blue – original 59-dimensional airfoil

Red – the airfoil described by 6 reduced features

General conclusion:

- high-dimensional real-world data occupy only a very small part with lower dimensional intrinsic structure in the high dimensional ‘observation space’ R^p which can be described by small number of reduced features
- the general problem is
  - ✓ to recognize the ‘low-dimensional’ structure of the Data space
  - ✓ or, **Dimensionality reduction problem**: to construct the reduced features which describe the original features (data) and preserve the most compact representation of information from it

Dimension reduction is a preprocessing stage in the whole system



Dimensionality reduction is nothing else but a mapping (linear or nonlinear) from a p-dimensional Original feature space onto an q-dimensional, q << p, Reduced Feature space, with its associated change of coordinates and preserving certain chosen subject-driven data properties

When latent variables (features that are at the origin of the observed ones but cannot be measured directly) exist, Dimensionality reduction only usually focuses on **the number of latent variables** and attempts **to give a** faithful **low-dimensional representation of data according to this number**

For this reason, dimensionality reduction does not care for the latent variables themselves: **any equivalent representation will do** though Dimensionality reduction problem can include various additional requirements to the desired reduced features – for examples, they should be independent ones

Another more difficult problem called **Latent variable separation** is not only to reduce the dimensionality but also, beyond dimensionality reduction, to retrieve the unknown latent variables as well as possible.
Signal processing: Blind source separation

Use of Dimensionality reduction techniques varies with **the application domain**. Examples of applications of dimensionality reduction techniques include:

- mining of text documents,
- gene structure discovery,
- image processing,
- statistical learning,
- and exploratory data analysis.

Many different fields are affected by the dimensionality reduction:

Statistics: it is related to multivariate density estimation, regression and smoothing techniques

Pattern recognition: dimensionality reduction is equivalent to feature extraction, where the feature vector would be the reduced-dimension one

I**nformation theory**: it is related to the problem of data compression and coding

Visualization technique: some kind of dimension reduction

Complexity reduction: if the complexity in time or memory of an algorithm depends on the dimension of its input data, as a consequence of the curse of the dimensionality, reducing this will make the algorithm more efficient

Latent-variable models - a small number of hidden causes acting in combination: the latent-variable space is the **low-dimensional representation** or **coordinate system** of the Original feature space

Geometric methods in Machine Learning

Many Machine Learning problems are fundamentally geometric in nature:

- general goal of Machine Learning/Data Mining/Data Analysis is to extract previously unknown information (patterns or regularities) from data
- target information is reflected in the **structure (underlying geometry) of the data**
- the **structure must be discovered from the data**

Understanding the shape of the data plays an important role in modern learning theory and data analytics

‘**Machine learning is about the shape of data**’ (ICML’2014)

- geometrical methods allow discovering the shape of data from given patterns: geometry - understanding the shape of the domain
- arising in part out of earlier dimensionality reduction researches, geometrical methods in machine learning has now become the central methodology for finding a structure (shape) in data and uncovering the semantics of information from the data
- if the data are intrinsically non-Euclidean, ignoring their geometrical structure can lead to non-effective results

Course ‘Geometrical Methods of Machine Learning’

- The aim of the course is to explain basic ideas and results in using the modern geometrical methods for solving main machine learning problems (classification/regression, data representation/dimensionality reduction, clustering, etc.)
- A large part of the course addresses to most popular geometrical model of high-dimensional data called manifold model (2000) and introduces modern manifold learning methods. Manifolds (Riemannian manifolds with a measure + noise) provide a natural mathematical language for thinking about high-dimensional data
- The course lets students to be involved in meaningful real-life machine learning projects to cope with challenging problems.

Course topics

- Linear methods (linear projection techniques): Principal Component Analysis (PCA), Independent Component Analysis, Projection Pursuit, …
- Intrinsic dimension of nonlinear structures: definitions and estimation
- Nonlinear projection techniques: Replicative Neural Networks, Kernel PCA, Multidimensional Scaling
- Manifold model for high-dimensional data
- Manifold learning: Locally Linear Embedding, ISOmetric MAPing (ISOMAP), Laplacian Eigenmaps, Hessian Eigenmaps, …
- Subspace learning: tangent spaces and Riemannian structure estimation, Local Tangent Space Alignment, Grassman&Stiefel Eigenmaps
- Manifold learning in Regression

and

- Elements of differential geometry and topology (short basics)

Seminars (Yury Yanovich, PhD)

- Linear and nonlinear projection techniques
- Intrinsic dimension estimation
- Manifold learning techniques

Grading policy

Grade structure:

30% - paper-based home assignments (3 home assignments, each one has the same weight)

35% - course project

35% - final exam

Grade mapping:

>80% - A

>70% - B

>60% - C

>50% - D

>40% - E

<=40% - F