FUNDAMENTALS - NUMERICAL METHODS AND ALGORITHMS

PhD, Andrius Kriščiūnas

- **Interpolation**
- Approximation
- Optimization problem

Time series analysis

Data that you want to explore can be given in the time different periods (by day, by month, by quarter, by year, etc.)

Data example

Staff salaries

2015 K1	2015 K2	2015 K3	2015 K4	2016 K1	2016 K2	2016 K3	2016 K4	2017 K1	2017 K2	2017 K3	2017 K4
686,4	700,9	722,3	744,2	737	761,2	783,3	812,8	808,7	830	842,7	876,4

Born

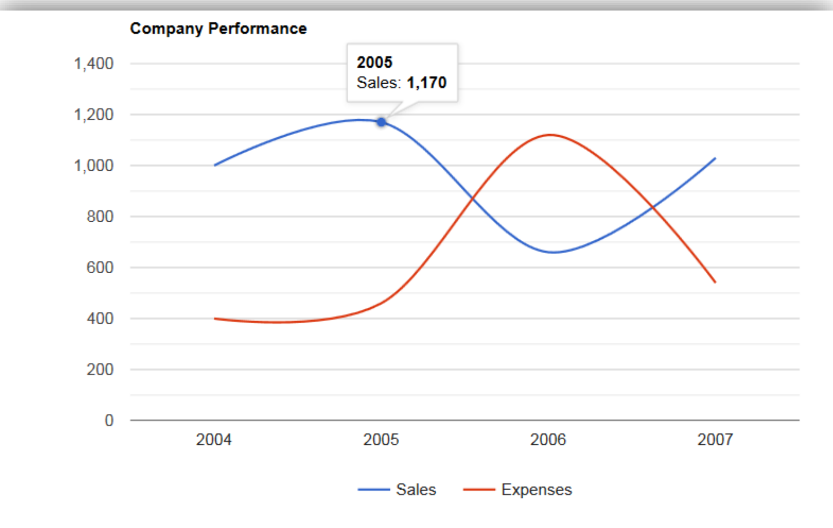
2015	2016	2017
31475	30623	28696

How to analyze the data in the context of **2016 February 25**?

Source: Statistics Lithuania
https://www.stat.gov.lt/home

What is the interpolation?

Wiki: **interpolation** is a method of constructing new [data points](#) within the range of a [discrete set](#) of known data points.




```
function drawChart() {
  var data = google.visualization.arrayToDataTable([
    ['Year', 'Sales', 'Expenses'],
    ['2004',  1000,      400],
    ['2005',  1170,      460],
    ['2006',  660,       1120],
    ['2007',  1030,      540]
  ]);
```

https://developers.google.com/chart/interactive/docs/gallery/linechart

Target: find the function which goes through the given points

Polynomial interpolation

Polynomial is a function of the form:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-2} x^{n-2} + a_{n-1} x^{n-1}$$

Given function can go through $\boldsymbol{n}$ given point!

$f(x) = a_0$ (1 point)

$f(x) = a_0 + a_1 x$ (2 point)

$\boldsymbol{f(x) = a_0 + a_1 x + a_2 x^2}$ **(3 point)**

…..

Born $\boldsymbol{n = 3}$

2015	2016	2017
31475	30623	28696

Unknown: a_1, a_2, a_3

$$f(x) = a_0 + a_1 x + a_2 x^2$$

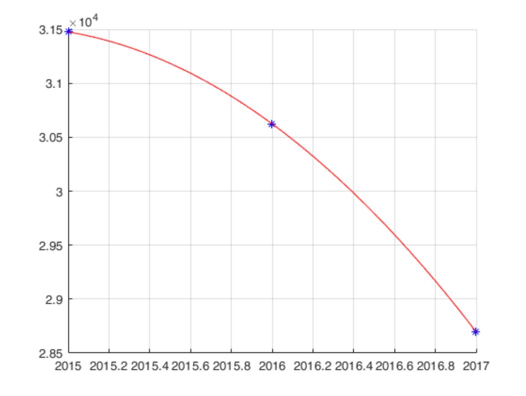
Born $\quad n = 3$

x	2015	2016	2017
$f(x)$	31475	30623	28696

$$\begin{cases} a_0 + a_1 * 2015 + a_2 * 2015^2 = 31475 \\ a_0 + a_1 * 2016 + a_2 * 2016^2 = 30623 \\ a_0 + a_1 * 2017 + a_2 * 2017^2 = 28696 \end{cases}$$

Solving linear equation system by any method (i.e. Gaussian elimination, LU decomposition, etc.)

Here we can take **any x,** and get the value!



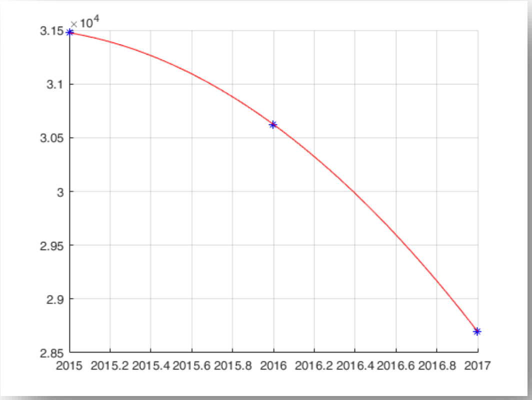
$$f_b(x) = a_0 + a_1 x + a_2 x^2$$

Born $\quad n = 3$

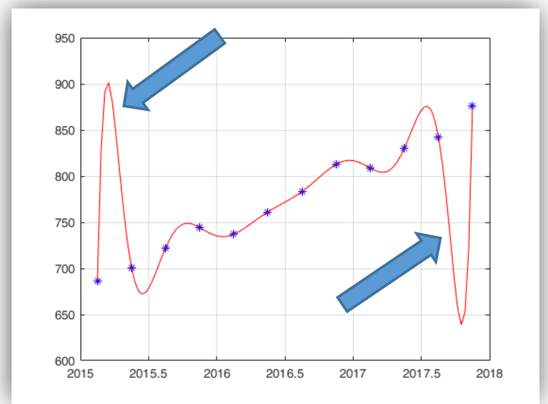
x	2015	2016	2017
$f(x)$	31475	30623	28696

Staff salaries $\quad n = 12$

2015 K1	2015 K2	2015 K3	2015 K4	2016 K1	2016 K2	2016 K3	2016 K4	2017 K1	2017 K2	2017 K3	2017 K4
686,4	700,9	722,3	744,2	737	761,2	783,3	812,8	808,7	830	842,7	876,4



$$f_b(x) = a_0 + a_1 x + a_2 x^2$$

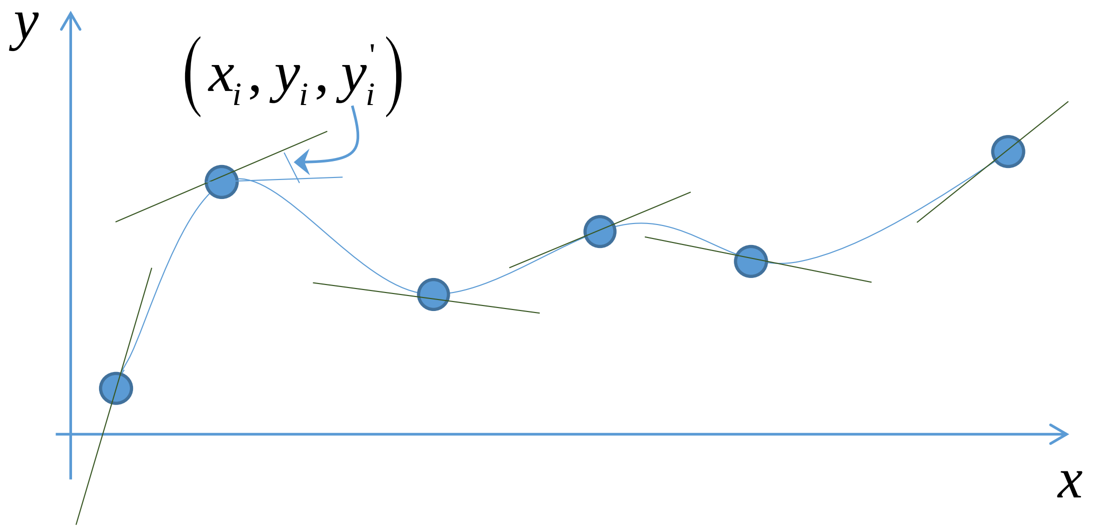


$$f_s(x) = a_0 + a_1 x + \cdots + a_{11} x^{11}$$

Large degree of polynomial causes the **waviness** of interpolating curve!

Spline interpolation

$$(x_i, y_i, y_i'), \qquad y_i = (x), \qquad y_i' = \frac{df}{dx}\bigg|_{x_i}, \qquad i = 1{:}n$$



Obtaining derivative value numerically

$$f(x)=\frac{(x-x_i)(x-x_{i+1})}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})}f_{i-1}+\frac{(x-x_{i-1})(x-x_{i+1})}{(x_i-x_{i-1})(x_i-x_{i+1})}f_i+\frac{(x-x_{i-1})(x-x_i)}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)}f_{i+1}$$

$$f^{'}(x)=\frac{(x-x_i)+(x-x_{i+1})}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})}f_{i-1}+\frac{(x-x_{i-1})+(x-x_{i+1})}{(x_i-x_{i-1})(x_i-x_{i+1})}f_i+\frac{(x-x_{i-1})+(x-x_i)}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)}f_{i+1}$$

$$f^{'}(x_{i-1}),\ f^{'}(x_i),\ f^{'}(x_{i+1})$$

Left point Middle point Right point

Hermitian spline

Input: $\left(x_i, y_i, y_i^{'}\right),\ i=1:n$

Interpolation function

$$\boxed{f(x)=\sum_{j=1}^{n}\left(U_j(x)y_j+V_j(x)y_j^{'}\right)}$$

$$U_j(x)=(1-2L_j^{'}(x_j)(x-x_j))L_j^2(x);$$

$$V_j(x)=(x-x_j)L_j^2(x), \qquad j=1:n$$

$$L_j(x)=\prod_{\substack{i=1\\ i\neq j}}^{n}\frac{x-x_i}{x_j-x_i}$$

Hermitian spline

Input: $\left(x_i, y_i, y_i^{'} \right),\ i = 1:n$

Interpolation function

$$\boxed{f(x) = \sum_{j=1}^{n} \left(U_j(x) y_j + V_j(x) y_j^{'} \right)}$$

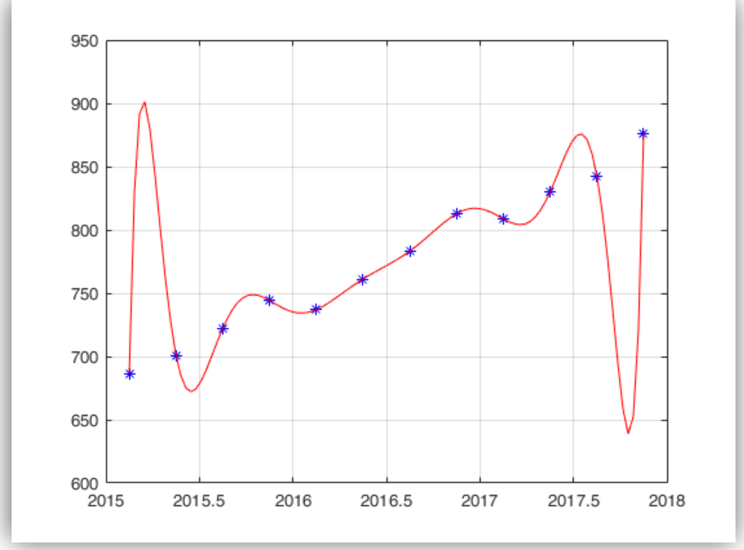
Most commonly used cubic **Hermitian splines**, then interpolation is performed between two adjacent points **(n=2)**

In this case, two different polynomials, defined at adjacent intervals, merge at each interpolation point. However, the **junction is smooth**, as the meanings of multiple derivatives at the interpolation point coincide.

Staff salaries $n = 12$

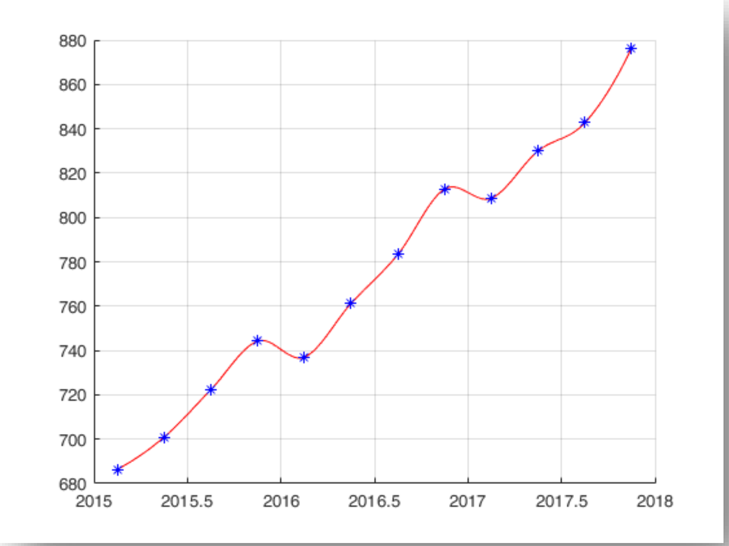
2015 K1	2015 K2	2015 K3	2015 K4	2016 K1	2016 K2	2016 K3	2016 K4	2017 K1	2017 K2	2017 K3	2017 K4
686,4	700,9	722,3	744,2	737	761,2	783,3	812,8	808,7	830	842,7	876,4

Polynomial interpolation



$$f(x) = a_0 + a_1 x + \cdots + a_{11} x^{11}$$

Hermitian spline



$$f_1(x) = a_{1\_0} + a_{1\_1}x + a_{1\_2}x^2 + a_{1\_3}x^3$$
$$f_2(x) = a_{2\_0} + a_{2\_1}x + a_{2\_2}x^2 + a_{2\_3}x^3$$
$$\ldots$$
$$f_{11}(x) = a_{11\_0} + a_{11\_1}x + a_{11\_2}x^2 + a_{11\_3}x^3$$

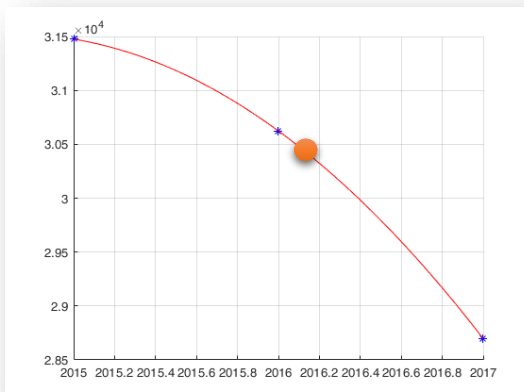
Staff salaries

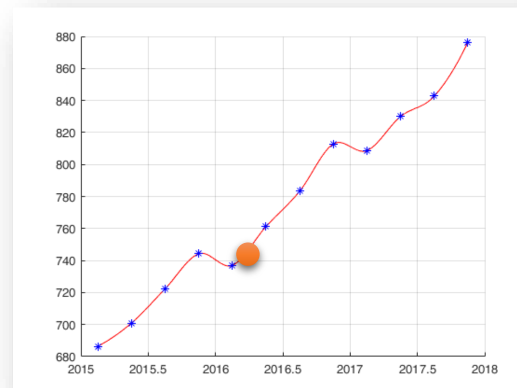
2015 K1	2015 K2	2015 K3	2015 K4	2016 K1	2016 K2	2016 K3	2016 K4	2017 K1	2017 K2	2017 K3	2017 K4
686,4	700,9	722,3	744,2	737	761,2	783,3	812,8	808,7	830	842,7	876,4

Born

2015	2016	2017
31475	30623	28696

How to analyze the data be the context in **2016 February 25**?





$$b \approx f_b(2016.157) \approx \mathbf{3039}$$

$$s \approx f_s(2016.157) \approx \mathbf{739}$$

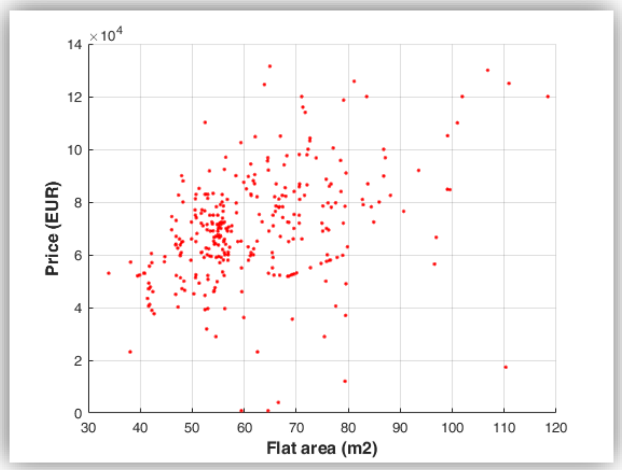
- Interpolation
- **Approximation**
- Optimization problem

Approximation

$$\left(x_i, y_i\right), \qquad i = 1, \dots, n$$

$\boldsymbol{f}(\mathbf{x}) = ?$

Data examples:



The approximation quality estimation function

$$\Psi = \frac{1}{2}\sum_{j=1}^{n}\left(f\left(x_j\right) - y_j\right)^2$$

finding function

$$\min_f \Psi$$

Approximation

Given points: $(x_i, y_i)\ ,\quad i = 1, \dots, n$

Linear combination of selected base functions with weighted coefficients

$$f(x) = \begin{bmatrix} g_1(x) & g_2(x) & \dots & g_{m-1}(x) & g_m(x) \end{bmatrix} \begin{Bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{m-1} \\ c_m \end{Bmatrix} = [\mathbf{g}(x)]\{\mathbf{c}\}$$

coefficients

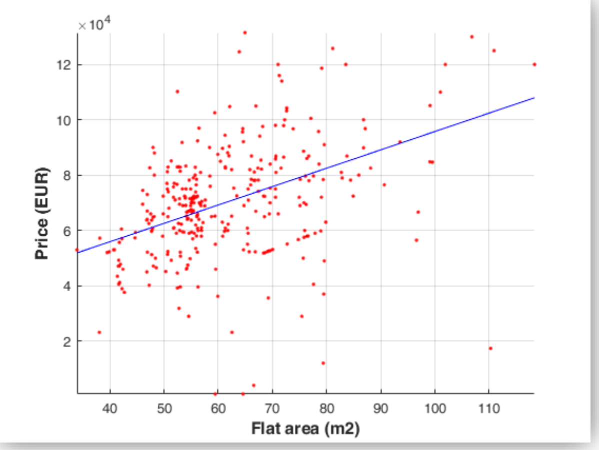
Solving linear equation system by any method (i.e. Gaussian elimination, LU decomposition, etc.)

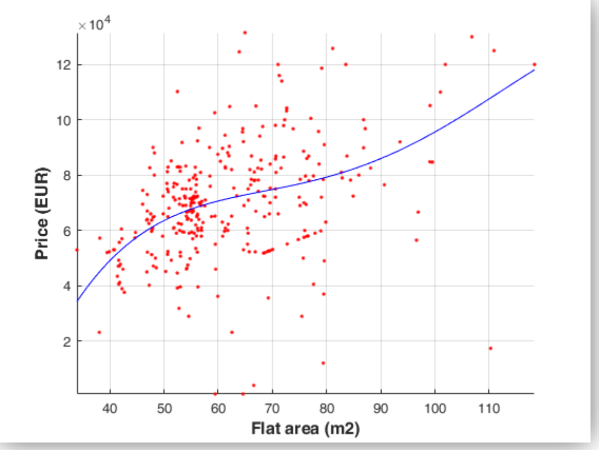
$$\left(\left(\mathbf{G}^T\right)_{m\times n} \mathbf{G}_{n\times m}\right)_{m\times m} \mathbf{c}_{m\times 1} = \left(\mathbf{G}^T\right)_{m\times n} \mathbf{y}_{n\times 1}$$

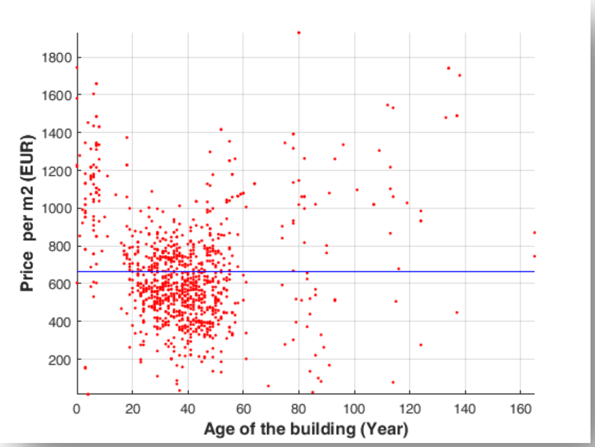
Approximation

$$f(x) = a_0 + a_1 x$$

$$f(x) = a_0 + a_1 x + \cdots + a_5 x^5$$







What form of approximating function should be selected?

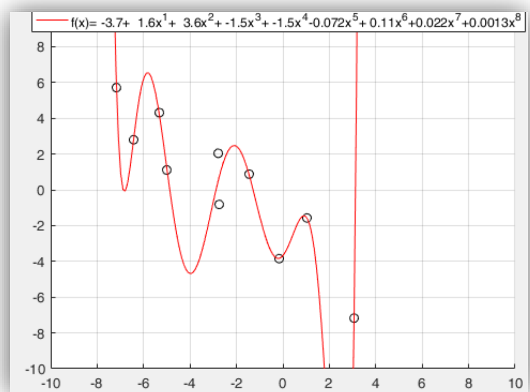
The approximation quality estimation function

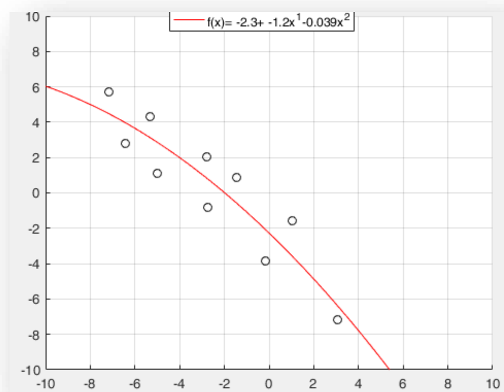
$$\Psi = \frac{1}{2}\sum_{j=1}^{n}\left(f\left(x_j\right) - y_j \right)^2$$

In order to avoid "<u>overfitting</u>", data should be spited in to the different datasets:

"Training data" – to obtain the approximating function

"Validation data" - to validate the approximating function





- Interpolation
- Approximation
- **Optimization problem**

Optimization problem

Scalar target function

$$\min_{\{\mathbf{x}\}} \Psi(\mathbf{x})$$

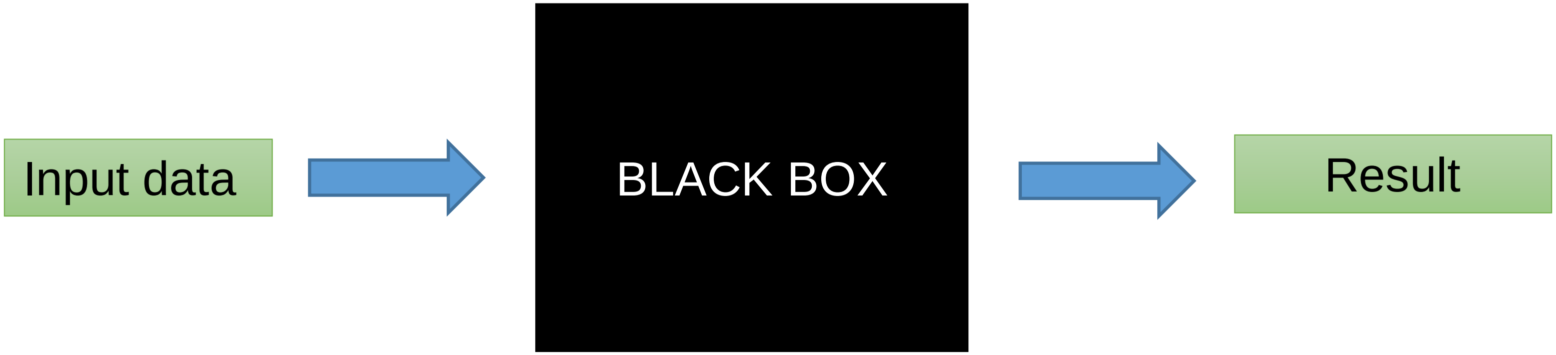
Vector argument

Minimum of value function Ψ is the minimum value of all possible!

The task of minimizing the function Ψ is to find the argument vector $\mathbf{x}$ at which the target function value is minimal;

i.e. constructed decision tree model (x – is the parameters of the tree) for classification problem, where target function Ψ returns the value of classification mistakes of "Validation dataset"

Idea of the Machine Lerning



Basically the “black box” hide the mathematics where model consists of the different algorithms based on approximation functions, where coefficients of this functions are adopted in the “learning” process.

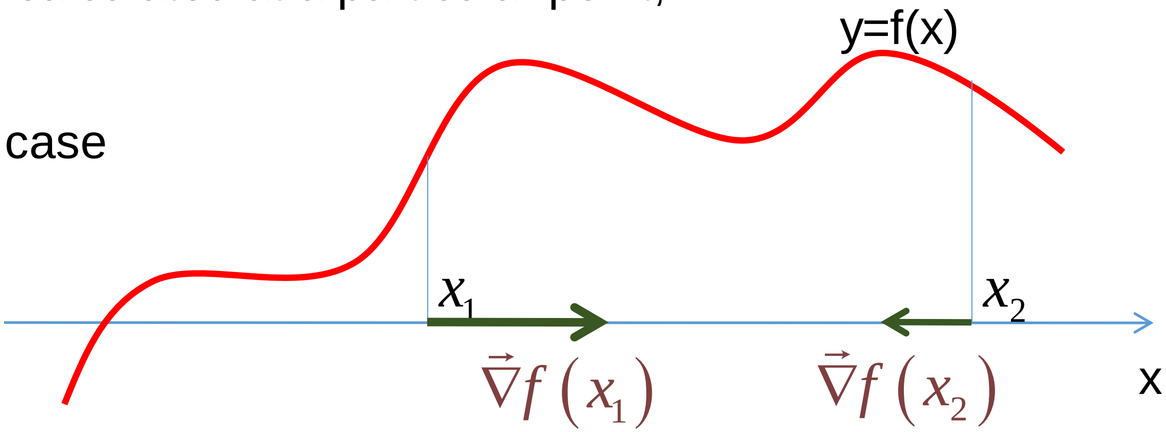
Basically, the parameters of the approximating function are obtained by using gradient techniques.

Why the different algorithms are necessary?

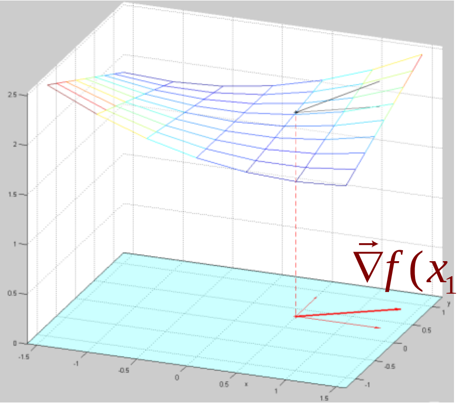
Gradient explanation

The **function gradient** is a vector representing a function derivative calculated at a particular point;

1D case



2D case



$$\vec{\nabla} f\left(x_1, y_1\right)=\left(\frac{\partial f}{\partial x}\ ,\ \frac{\partial f}{\partial y}\right)_{\left(x_1, y_1\right)}$$

Gradient descend and Conjugate gradient methods

Gradient descend: in each step, changes in the arguments obtained in the opposite direction of the gradient

1. Obtain gradient $$\nabla\Psi = \left(\frac{\partial\Psi}{\partial x_1}, \frac{\partial\Psi}{\partial x_2}, \dots, \frac{\partial\Psi}{\partial x_n}\right)$$

2. Obtain new parameter values $$\boldsymbol{X}_{i+1} = \boldsymbol{X}_i - \Delta s \cdot \nabla\Psi$$

If value of target function grows, decrease step or ending the optimization

Conjugate gradient: after calculating the gradient vector, it is moved in the opposite direction until the function continues to decrease

Obtaining derivatives numerically

$$\nabla\Psi = \left(\frac{\partial\Psi}{\partial x_1}, \frac{\partial\Psi}{\partial x_2}, \dots, \frac{\partial\Psi}{\partial x_i}, \dots, \frac{\partial\Psi}{\partial x_n}\right)$$

$$\frac{\partial\Psi}{\partial x_i} = \frac{\Psi(x_1, x_2, \dots, x_i + h, \dots, x_n) - \Psi(x_1, x_2, \dots, x_i, \dots, x_n)}{h}$$

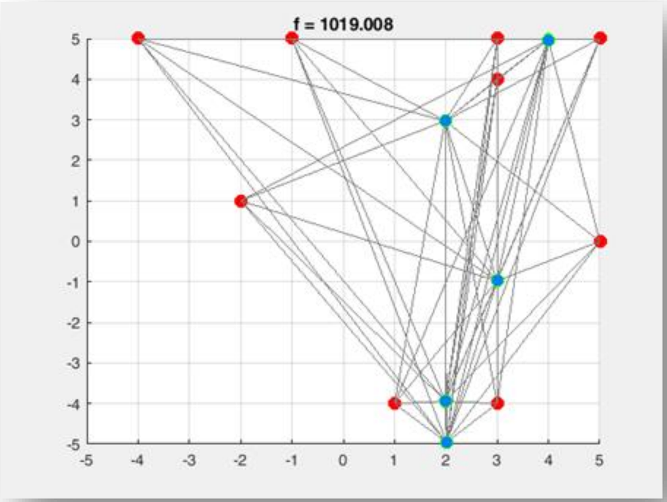
Given:

Coordinates of the fixed nodes:

$$\{(x_{M+1}, y_{M+1}), \ldots, (x_N, y_N)\}$$

Target:

Find positions $\{(x_1, y_1), \ldots, (x_M, y_M)\}$ of M nodes so that the distances between the points are as close to the average distance as possible



Gradient optimization

$$\min_{x_1,\dots,x_M,y_1,\dots,y_M} \Psi = \sum_{i=1}^{M} \sum_{j=i+1}^{N} \left(\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 - \bar{d} \right)^2$$

M – number of nodes with unknown positions

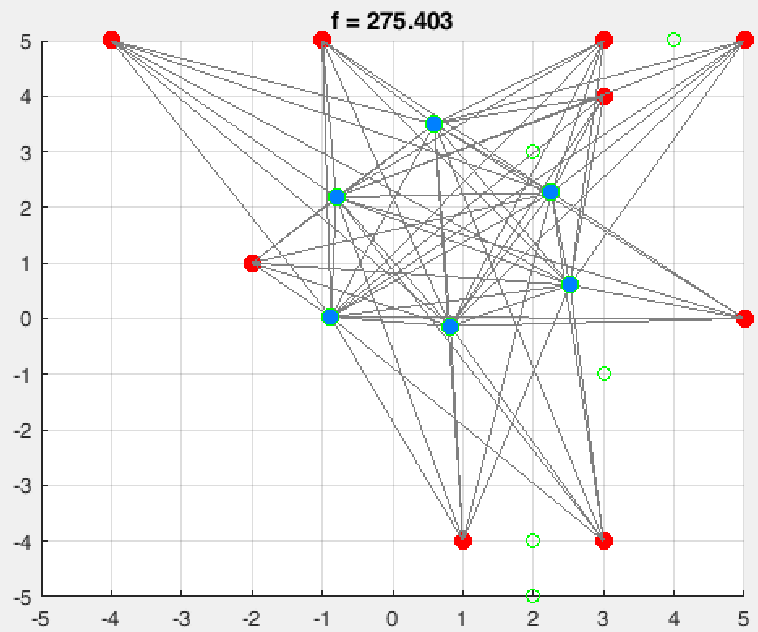
N – total number of nodes

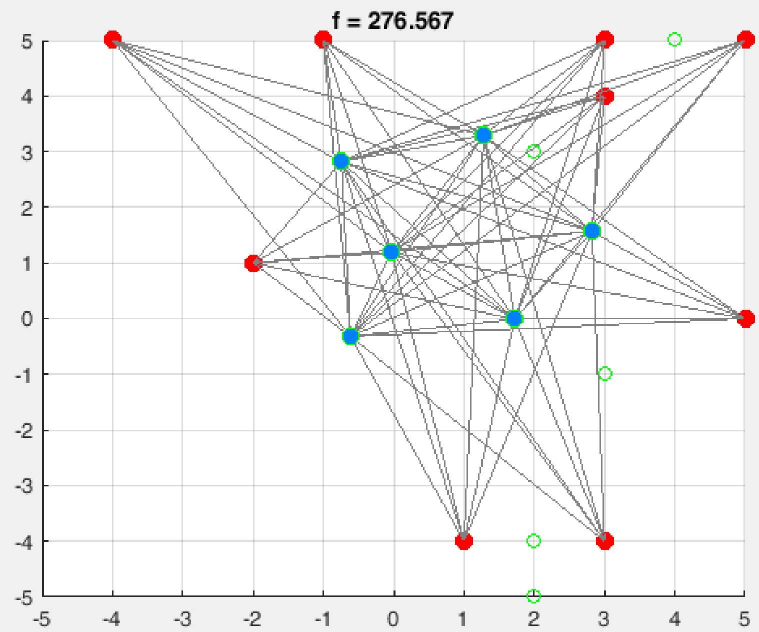
$\bar{d}$ – average distance between the nodes

Continuous optimization

Gradient descent method

Conjugate gradient method





$$\nabla\Psi = \left(\frac{\partial\Psi}{\partial x_1}, \frac{\partial\Psi}{\partial x_2}, \dots, \frac{\partial\Psi}{\partial x_i}, \dots, \frac{\partial\Psi}{\partial x_n}\right)$$

Why the different algorithms are necessary?

$$\frac{\partial\Psi}{\partial x_i} = \frac{\Psi(x_1, x_2, \dots, x_i + h, \dots, x_n) - \Psi(x_1, x_2, \dots, x_i, \dots, x_n)}{h}$$

For each element of the gradient the target function should be calculated. It is very calculation expensive!

Gradient boosting

Thea same idea as in **gradient descend** when numerical approximation of gradient is used. Here the approximation is performed by **decision tree.**