Lecture 1 – Linear Regression

Bulat Ibragimov

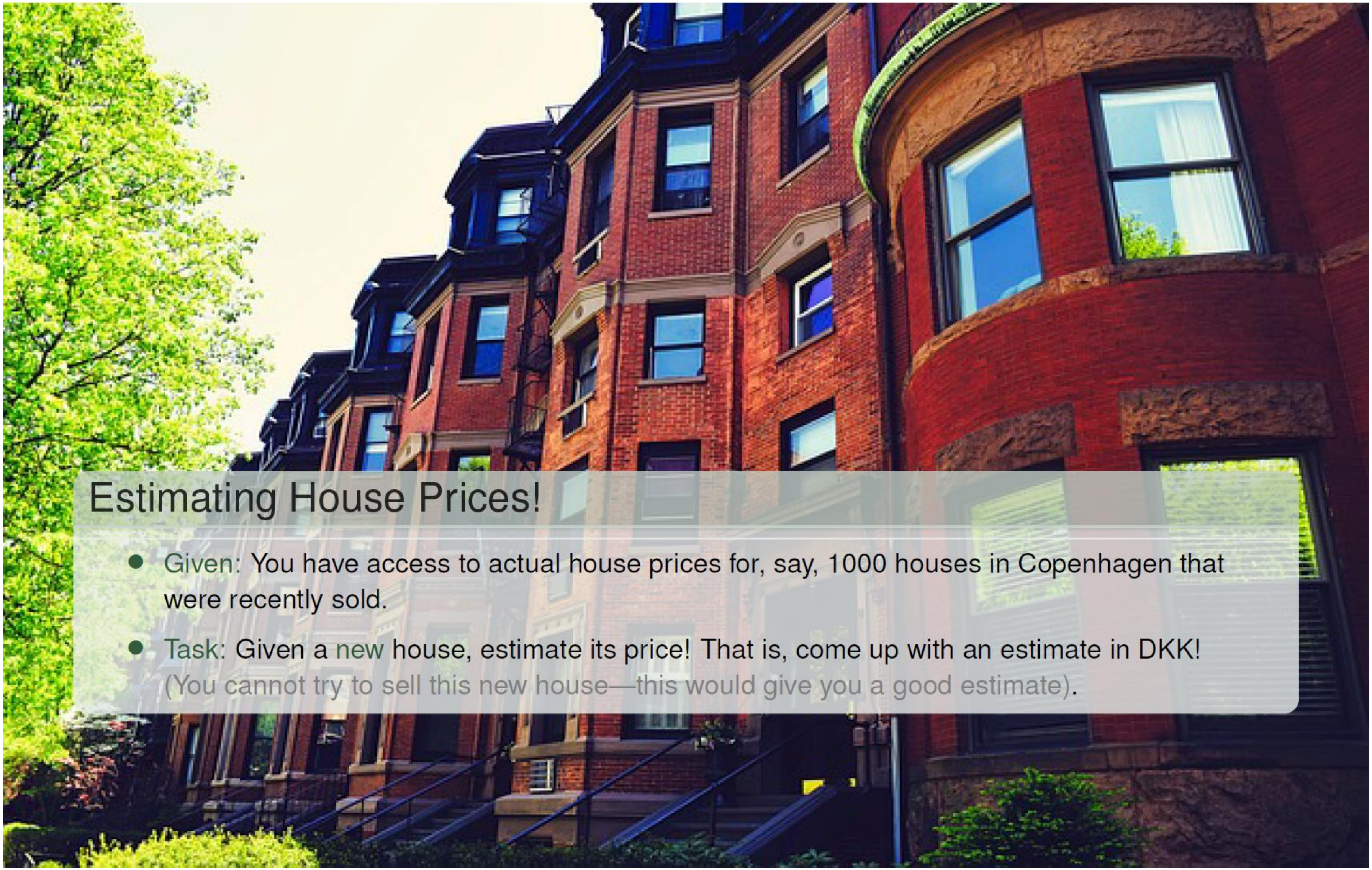
bulat@di.ku.dk

Department of Computer Science
University of Copenhagen

Outline

1. Motivation
2. Linear Regression I
3. Summary & Outline

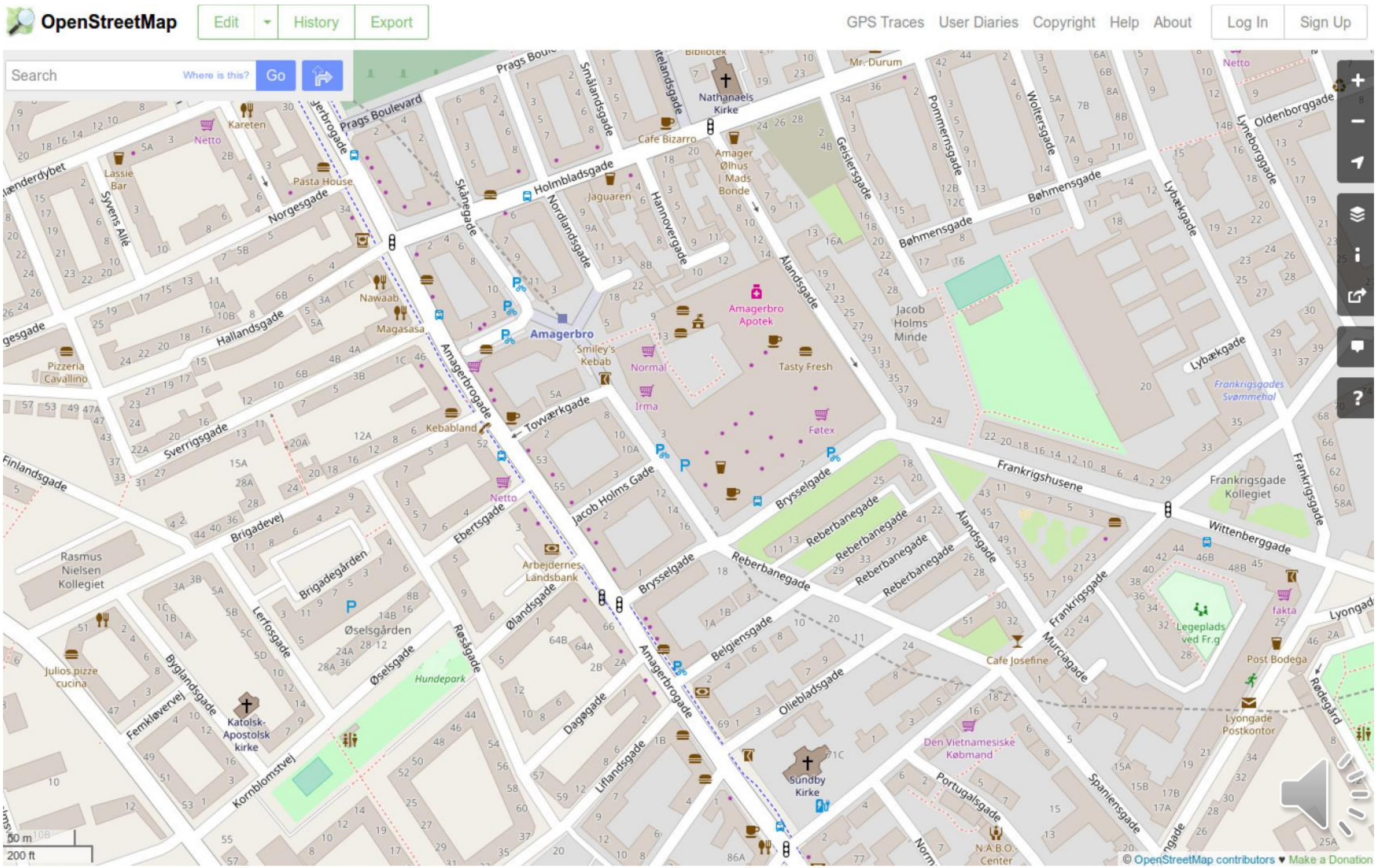
Motivation



Estimating House Prices!

- Given: You have access to actual house prices for, say, 1000 houses in Copenhagen that were recently sold.
- Task: Given a new house, estimate its price! That is, come up with an estimate in DKK! (You cannot try to sell this new house—this would give you a good estimate).

Motivation


OpenStreetMap
Edit
History
Export
GPS Traces
User Diaries
Copyright
Help
About
Log In
Sign Up
Search
Where is this?
Go
Prags Boulevard
Holmbladsgade
Norgesgade
Hallandsgade
Sverrigsgade
Brigadevej
Brigadegården
Øselsgården
Øselsgade
Amagerbrogade
Tovværkgade
Jacob Holms Gade
Ebertsgade
Ølandsgade
Brysselgade
Reberbanegade
Belgiensgade
Oliebladsgade
Frankrigshusene
Wittenberggade
Frankrigsgade
Lybækgade
Bøhmensgade
Oldenborggade
Lyneborggade
Ålandsgade
Geislersgade
Pommernsgade
Woltersgade
Smålandsgade
Hannovergade
Nordlandsgade
Skånegade
Syvens Allé
Finlandsgade
Lerfosgade
Røsågade
Dagøgade
Liflandsgade
Kornblomstvej
Femkløvervej
Byglandsgade
Murciagade
Portugalsgade
Spaniensgade
Rødegård
Lyongade
Amagerbro
Amagerbro Apotek
Nathanaels Kirke
Cafe Bizarro
Amager Ølhus | Mads Bonde
Jaguaren
Kareten
Netto
Pasta House
Lassie Bar
Nawaab
Magasasa
Pizzeria Cavallino
Kebabland
Smiley's Kebab
Normal
Irma
Føtex
Tasty Fresh
Jacob Holms Minde
Frankrigsgades Svømmehal
Frankrigsgade Kollegiet
Rasmus Nielsen Kollegiet
Arbejdernes Landsbank
Hundepark
Julios pizze cucina
Katolsk-Apostolsk kirke
Sundby Kirke
Den Vietnamesiske Købmand
Cafe Josefine
Legeplads ved Fr.g
fakta
Post Bodega
Lyongade Postkontor
N.A.B.O. Center
Mr. Durum
50 m
200 ft
© OpenStreetMap contributors ♥ Make a Donation

Regression

- Given some data related to houses, estimate the price $y \in \mathbb{R}$ in DKK for each house
- Given some stock, estimate the value $y \in \mathbb{R}$ it will have in ten days
- Given the results of biopsy, demographics and disease history predict the survival time $y \in \mathbb{R}$ of a patient

These tasks are called regression tasks since we are interested in a real value $y \in \mathbb{R}$

Classification

- Given some photos, classify them into “cats” (y = 0), “dogs” (y = 1), or “other” (y = 2)
- Given the results of biopsy, demographics and disease history predict the if the patient will survive 3-year threshold (y = 1), or not (y = 0)

These tasks are called classification tasks since we are interested in a class $y \in \{0, 1, 2, \dots\}$

Clustering

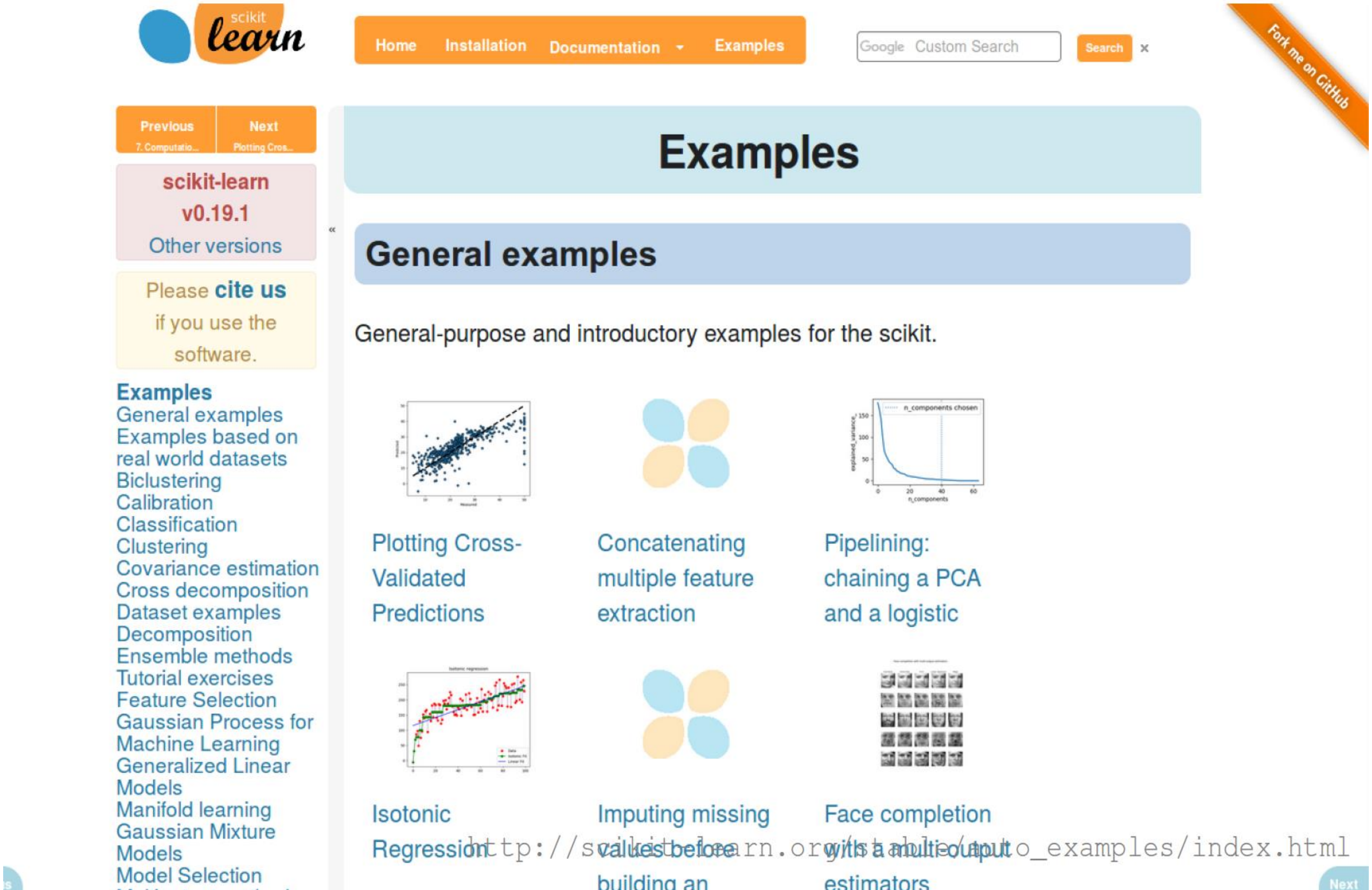
- Given some photos, automatically partition them into groups
- Given the results of biopsy, demographics and disease partition patients into groups

Classes/groups not known beforehand. These tasks are called clustering tasks.

Dimensionality reduction

- Reducing the database size by removing unnecessary data dimensions
- Simplify data interpretation

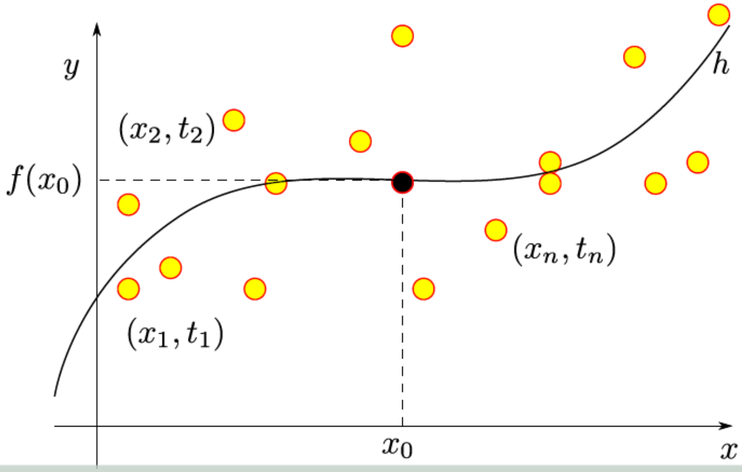
Demo: Machine Learning & Scikit-Learn



http://scikit-learn.org/stable/auto_examples/index.html

Outline

Regression



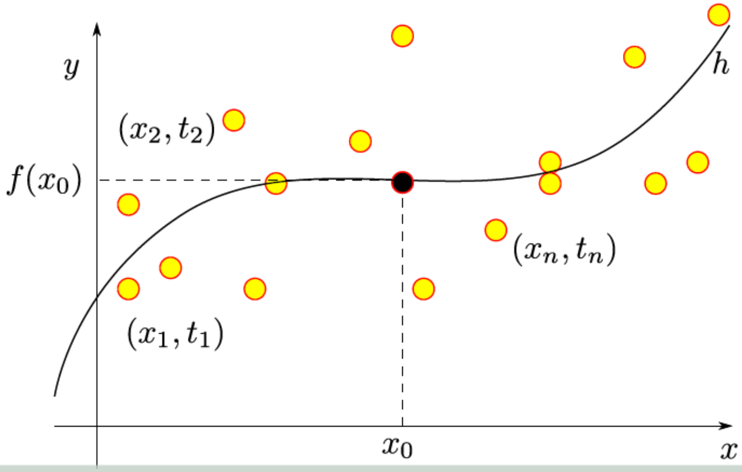
A Learning Problem

- **Input:** N pairs $(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)$ of observed
  - input variables/vectors $\mathbf{x}_n \in \mathbb{R}^D$ and
  - target variables $t_n \in \mathbb{R}$.
- **Assumption:** There is a functional relationship

$$y = f(\mathbf{x}),$$

where $f \colon \mathbb{R}^D \to \mathbb{R}$.

Regression



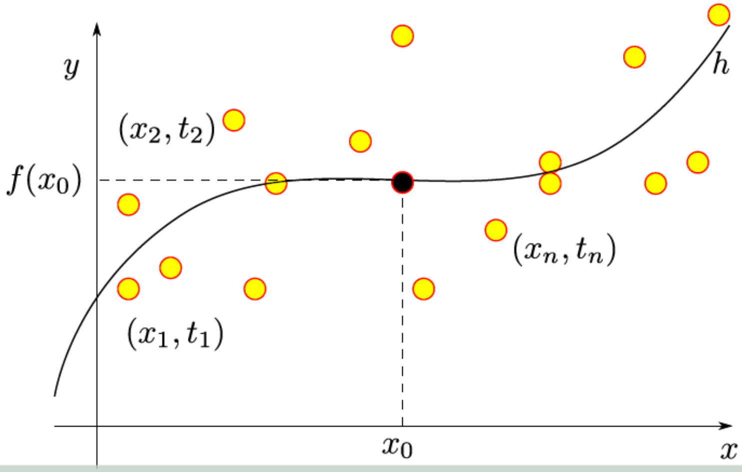
A Learning Problem

- **Input:** N pairs $(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)$ of observed
  - input variables/vectors $\mathbf{x}_n \in \mathbb{R}^D$ and
  - target variables $t_n \in \mathbb{R}$.
- **Assumption:** There is a functional relationship

$$y = f(\mathbf{x}),$$

  where $f : \mathbb{R}^D \to \mathbb{R}$.
- **Goal:** Learn the function $f(\mathbf{x})$ from the N data points!

Regression



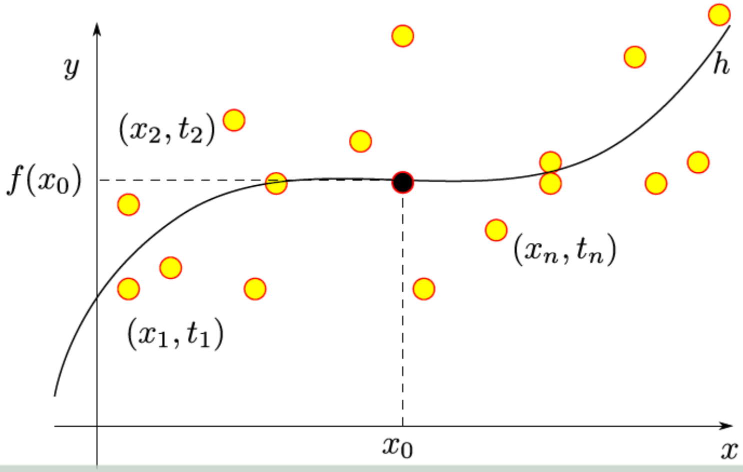
A Learning Problem

- **Input:** N pairs $(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)$ of observed
  - input variables/vectors $\mathbf{x}_n \in \mathbb{R}^D$ and
  - target variables $t_n \in \mathbb{R}$.
- **Assumption:** There is a functional relationship

$$y = f(\mathbf{x}),$$

  where $f\colon \mathbb{R}^D \to \mathbb{R}$.
- **Goal:** Learn the function $f(\mathbf{x})$ from the N data points!
- **What is this good for?**

Regression



A Learning Problem

- **Input:** N pairs $(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)$ of observed
  - input variables/vectors $\mathbf{x}_n \in \mathbb{R}^D$ and
  - target variables $t_n \in \mathbb{R}$.
- **Assumption:** There is a functional relationship
$$y = f(\mathbf{x}),$$
where $f\colon \mathbb{R}^D \to \mathbb{R}$.
- **Goal:** Learn the function $f(\mathbf{x})$ from the N data points!
- **What is this good for?** Given a new observed input variable $\mathbf{x}_0$, we can "predict" the corresponding output variable $f(\mathbf{x}_0)$!

Example: murder rates

- Unemployment rates $\rightarrow$ murder rates
- Question: What are the $\mathbf{x}_n$ and t_n?

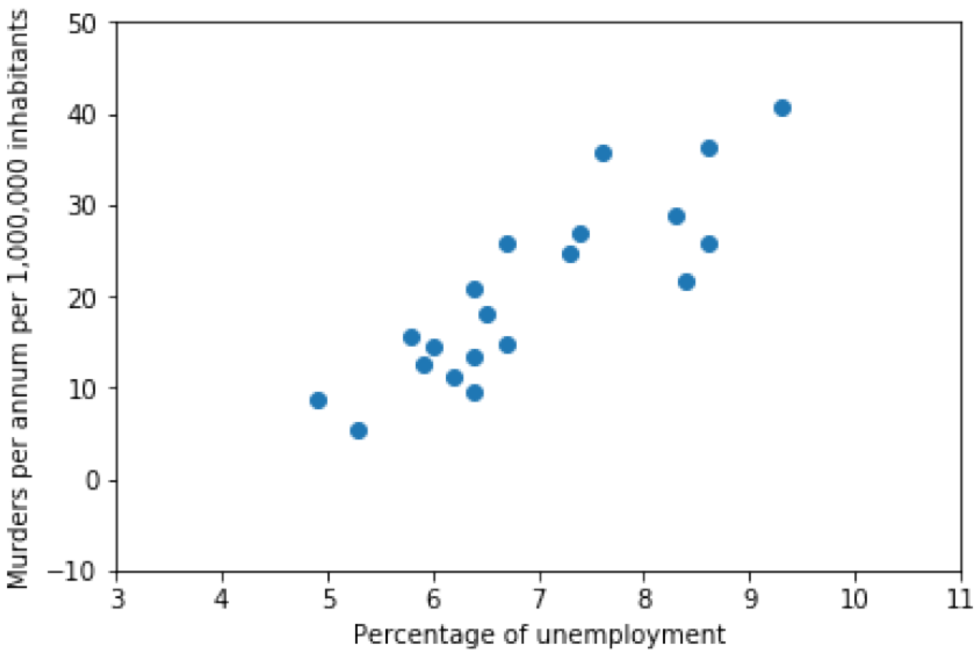
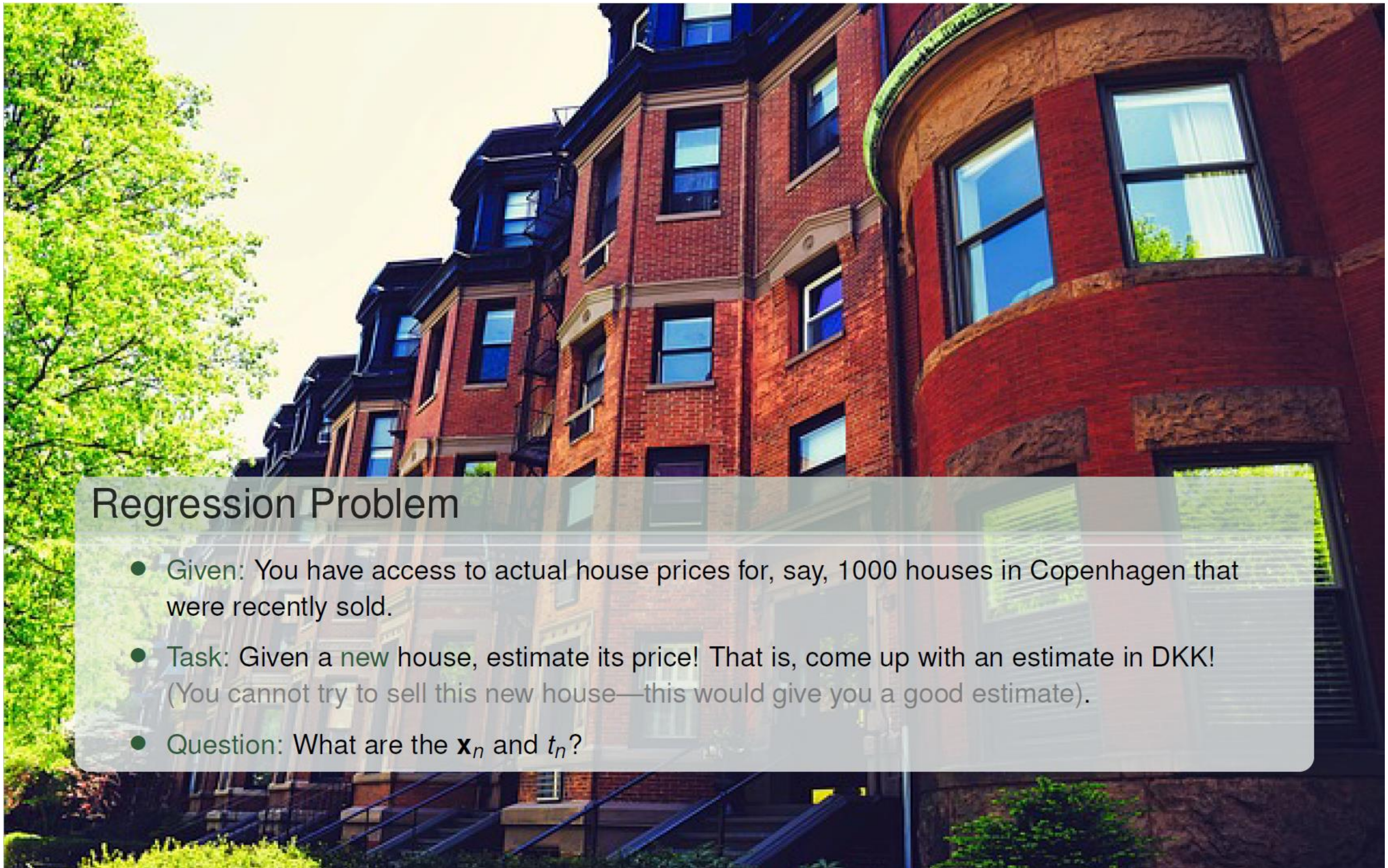


Figure: Murder rates versus unemployment rates in an American city[1]

---

[1] Helmut Spaeth, Mathematical Algorithms for Linear Regression, Academic Press, 1991, ISBN 0-12-656460-4; D G Kleinbaum and L L Kupper, Applied Regression Analysis and Other Multivariable Methods, Duxbury Press, 1978, page 150; http://people.sc.fsu.edu/ jburkardt/datasets/regression

Example: house price



Regression Problem

- Given: You have access to actual house prices for, say, 1000 houses in Copenhagen that were recently sold.
- Task: Given a new house, estimate its price! That is, come up with an estimate in DKK! (You cannot try to sell this new house—this would give you a good estimate).
- Question: What are the $\mathbf{x}_n$ and t_n?

Notation: vectors

- Let's say that our data is defined with D features. So one data sample $\boldsymbol{x}$ will look like:

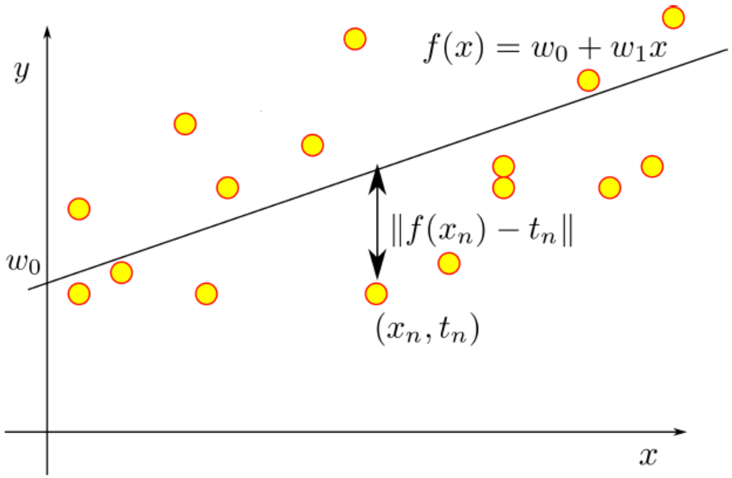
$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix}$$

- That's annoying to type, so we will write $\mathbf{x} = [\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_\mathrm{D}]^\mathbf{T}$

Linear regression: single feature data

- Let us start with $D = 1$, i.e., with input data of the form $x_n \in \mathbb{R}$.
- Let us consider models f of the form

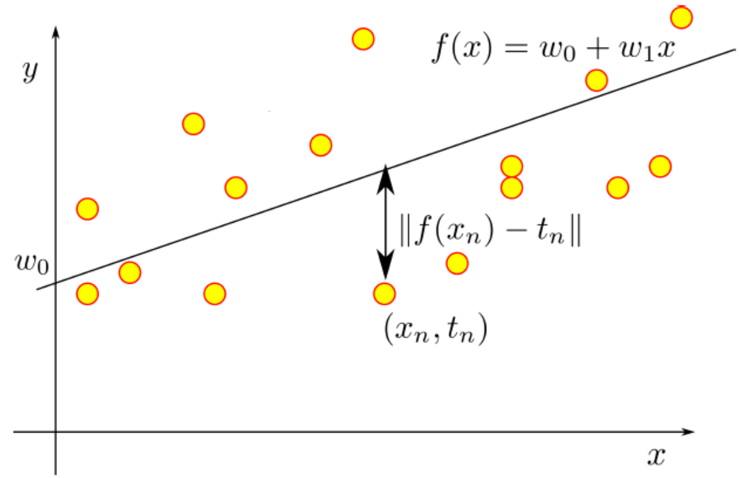
$$f(x) = f(x; w_0, w_1) = w_0 + w_1 x$$



Linear regression: single feature data

- Let us start with $D = 1$, i.e., with input data of the form $x_n \in \mathbb{R}$.
- Let us consider models f of the form

$$f(x) = f(x; w_0, w_1) = w_0 + w_1 x$$



- Comment: If we set $\boldsymbol{x} = [1, x]^T$ and $\boldsymbol{w} = [w_0, w_1]^T$, then we have:

$$f(\boldsymbol{x}) = f(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{x}^T \boldsymbol{w}$$

Example: murder rates

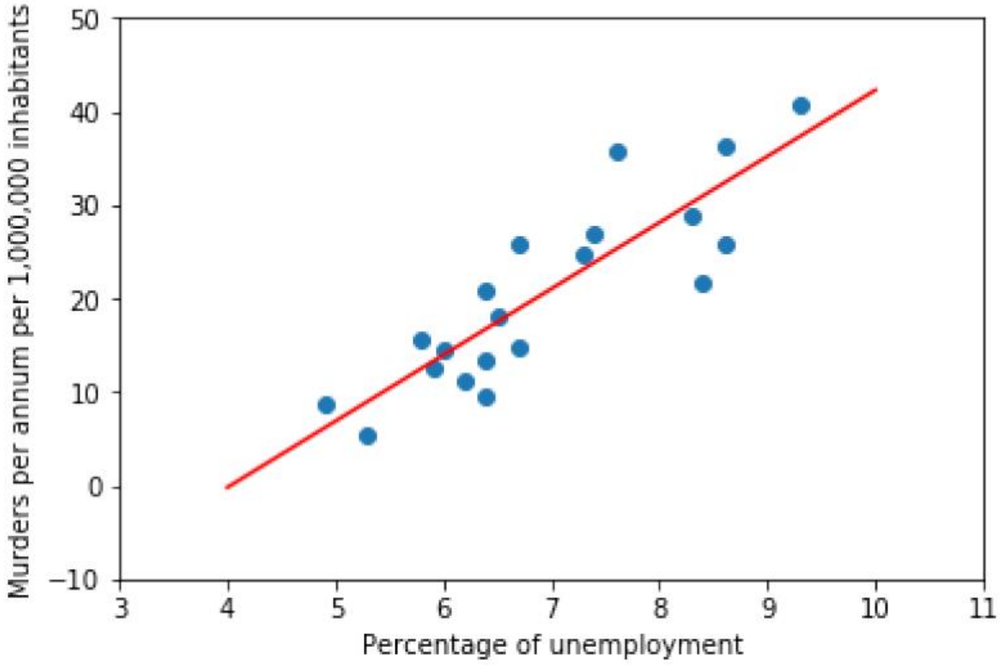


Figure: What is a “good” model? How can we measure its “quality”?

Performance evaluation

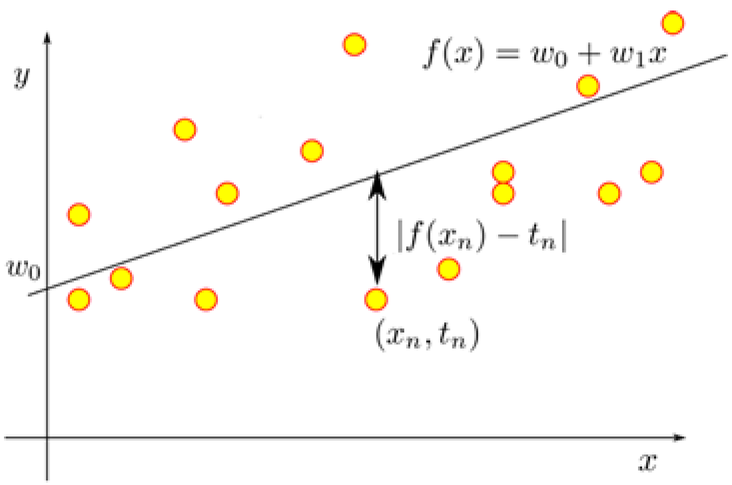
Regression: Labels are floating/integer numbers

- Mean absolute error

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

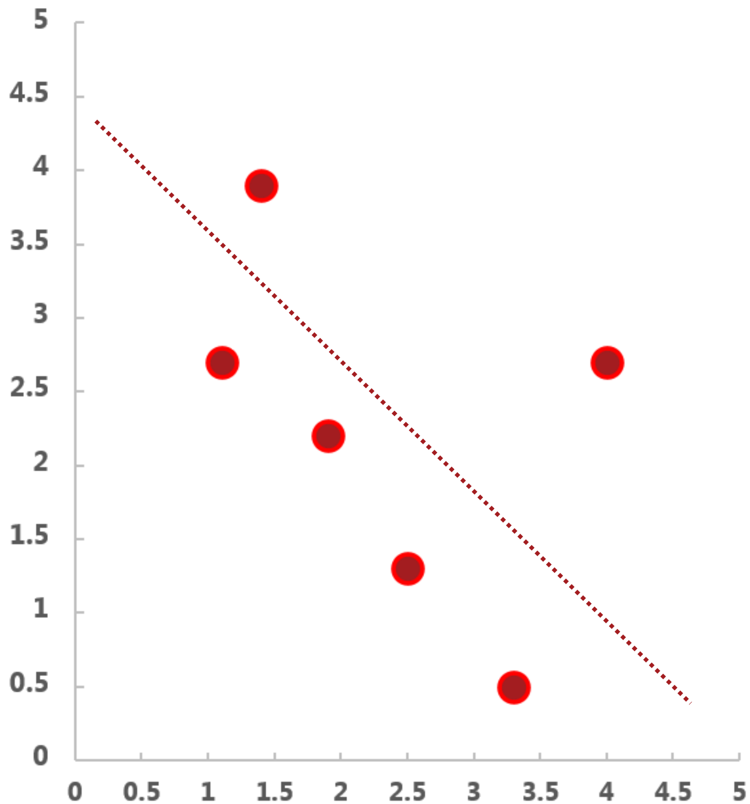
- Mean squared error

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

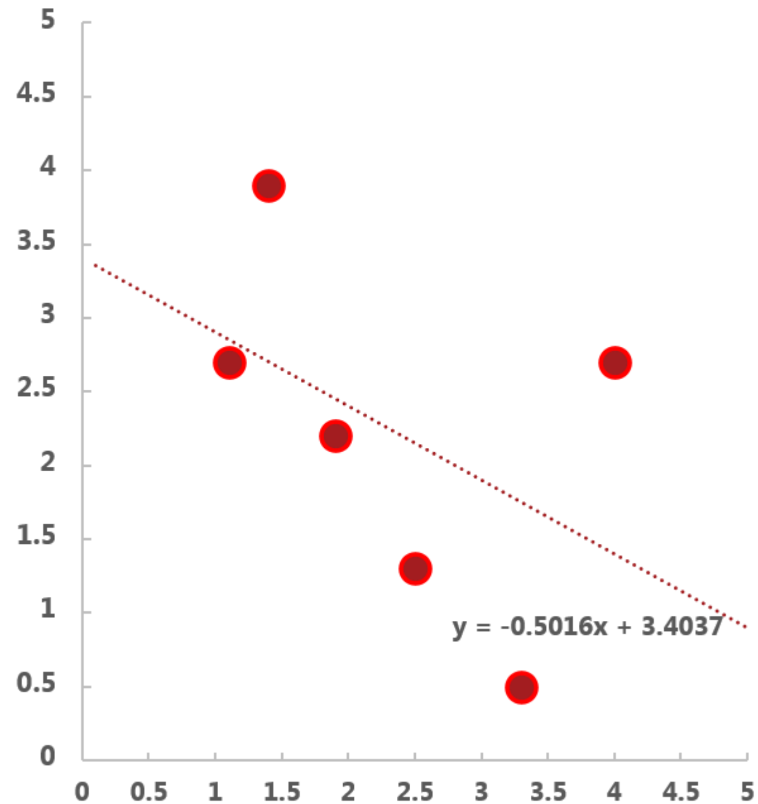


Performance evaluation

Which loss function to choose?



MAE, or L1 loss



MSE, or L2 loss

The square loss function

- We would like to minimize the "error" made when using f to predict values $f(x) = w_0 + w_1 x$ on the given data. One possible choice for such an error function is the square loss function

$$\left(f(x_n; w_0, w_1) - t_n\right)^2,$$

  which measures the discrepancy between a target t_n and the associated predicted value $f(x_n; w_0, w_1)$.

- We aim at a low loss for all the data points, i.e.:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \left(f(x_n; w_0, w_1) - t_n\right)^2$$

- Goal: Find optimal parameters $\hat{w}_0$ and $\hat{w}_1$ that minimize this overall loss:

$$(\hat{w}_0, \hat{w}_1) = \underset{w_0, w_1}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} \left(f(x_n; w_0, w_1) - t_n\right)^2$$

Computing the optimal parameters

$$\mathcal{L}(w_0, w_1) = \frac{1}{N}\sum_{n=1}^{N}(f(x_n; w_0, w_1) - t_n)^2 = \frac{1}{N}\sum_{n=1}^{N}((w_0 + x_n w_1) - t_n)^2$$

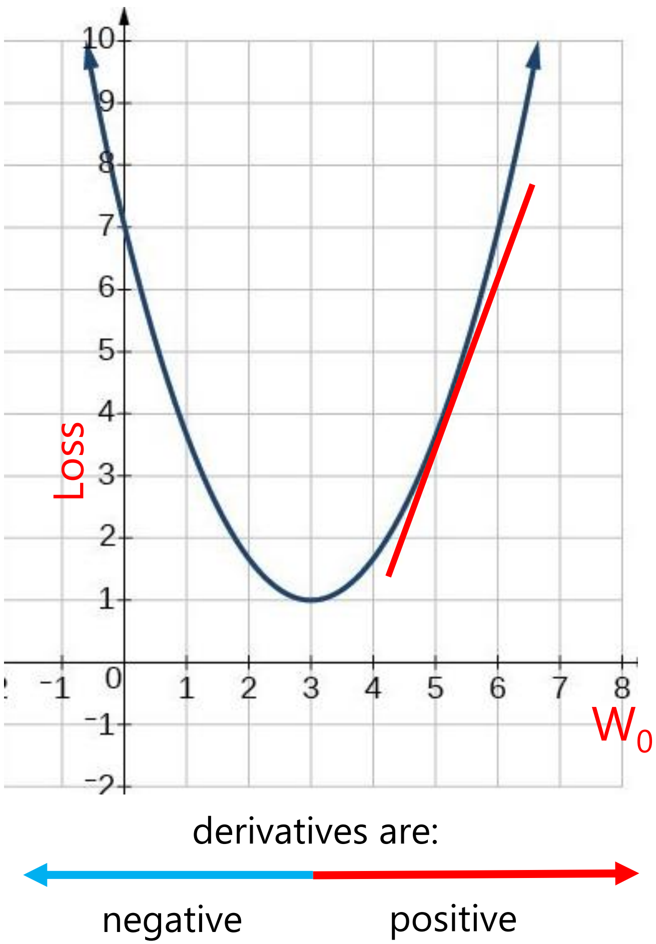
- We would like to find the two coefficients w_0 and w_1 that minimize the above objective! Question: How can we find these coefficients?

Iterative optimization: derivatives

We want the loss to be as small as possible, i.e. find its minimum.

We use derivatives to find minima/maxima of a function:

- How fast function changes
- Will it increase or decrease



Computing the optimal parameters

$$\mathcal{L}(w_0, w_1) = \frac{1}{N}\sum_{n=1}^{N}\left(f(x_n; w_0, w_1) - t_n\right)^2 = \frac{1}{N}\sum_{n=1}^{N}\left((w_0 + x_n w_1) - t_n\right)^2$$

- We have a function with two variables w_0 and w_1 and are searching for vector $\mathbf{w} = [w_0, w_1]^T$ corresponding to a minimum w.r.t. $\mathcal{L}$. Thus, the gradient of $\mathcal{L}$ must vanish at $\mathbf{w}$ (necessary condition!):

$$\nabla\mathcal{L}(w_0, w_1) = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial w_0} \\ \frac{\partial \mathcal{L}}{\partial w_1} \end{bmatrix} \overset{!}{=} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- Task: Compute both partial derivatives!

Computing the optimal parameters

- One can simplify the objective as follows:

$$\begin{aligned}
\mathcal{L}(w_0, w_1) &= \frac{1}{N}\sum_{n=1}^{N}((w_0 + x_n w_1) - t_n)^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}(w_0 + x_n w_1)^2 - 2(w_0 + x_n w_1)t_n + t_n^2 \\
&= \frac{1}{N}\sum_{n=1}^{N} w_0^2 + 2w_0 x_n w_1 + x_n^2 w_1^2 - 2w_0 t_n - 2x_n w_1 t_n + t_n^2
\end{aligned}$$

- Hence, one directly obtains the partial derivatives:

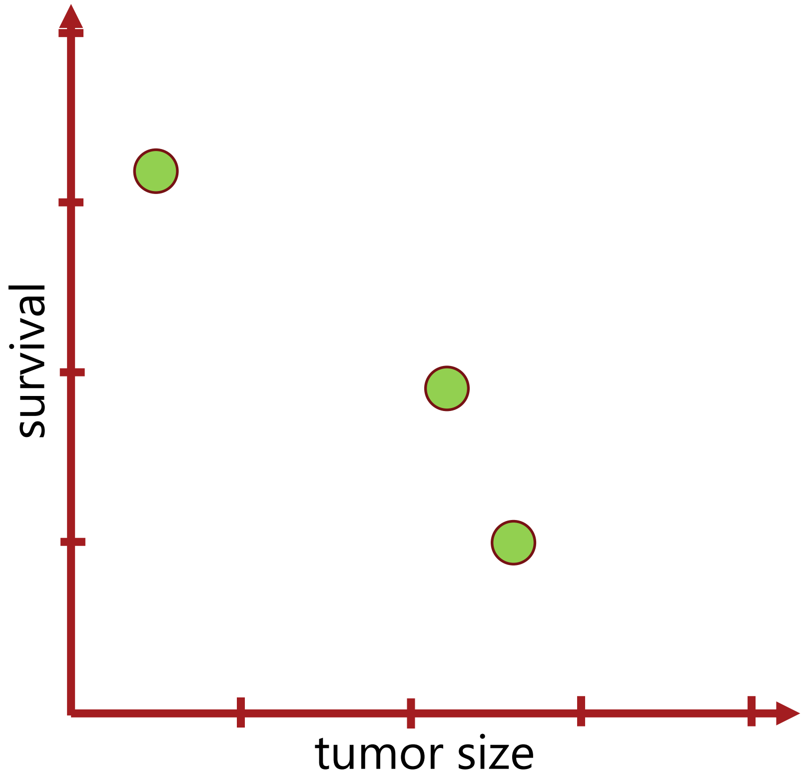
$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_0} &= 2w_0 + 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right) \\
\frac{\partial \mathcal{L}}{\partial w_1} &= 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} {x_n}^2\right) + \frac{2}{N}\left(\sum_{n=1}^{N} x_n(w_0 - t_n)\right)
\end{aligned}$$

Example

Patient survival depends on cancer size:

- Bigger tumors are worse
- Can we model this dependency?

	Tumor Size	Survival
Case1	0.5	3.2
Case2	2.3	1.9
Case3	2.9	1.0



Example

	Tumor Size	Survival
Case1	0.5	3.2
Case2	2.3	1.9
Case3	2.9	1.0

$$\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right)$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n{}^2\right) + \frac{2}{N}\left(\sum_{n=1}^{N} x_n(w_0 - t_n)\right)$$

The derivative against w_0:

$$\frac{\partial L}{\partial w_0} = 2w_0 + 2w_1 \frac{1}{3}(0.5 + 2.3 + 2.9) - \frac{2}{3}(3.2 + 1.9 + 1)$$
$$= 2w_0 + 3.8w_1 + 4.07 = 0$$

The derivative against w_1:

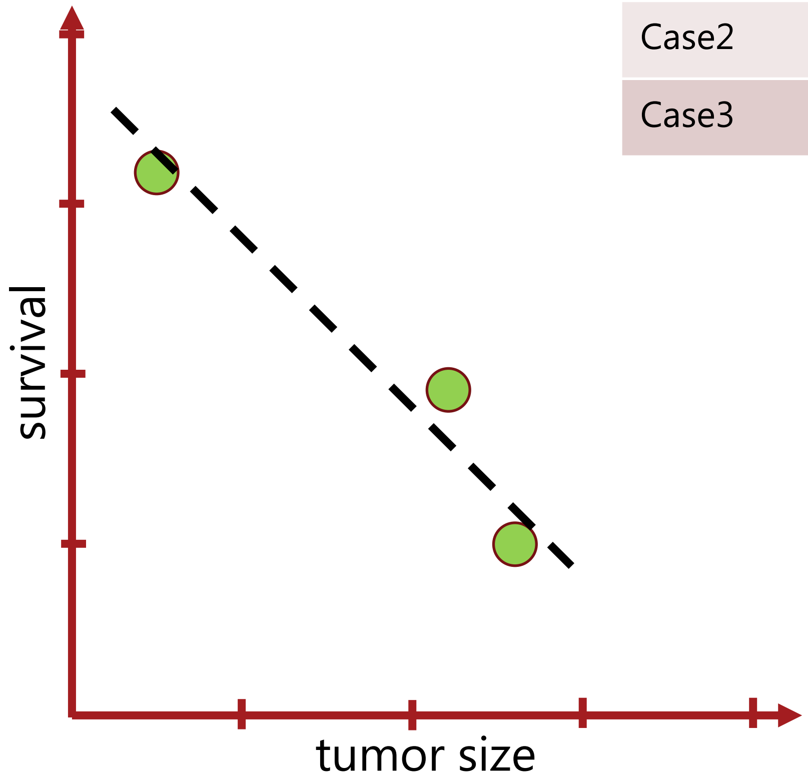
$$\frac{\partial L}{\partial w_1} = 2w_1 \frac{1}{3}(0.25 + 5.29 + 8.41) + \frac{2}{3}0.5(w_0 - 3.2) + \frac{2}{3}2.3(w_0 - 1.9) + \frac{2}{3}2.9(w_0 - 1) = 9.3w_1 + 3.8w_0 - 4.07 = 0$$

$$\boldsymbol{w_0 = 3.69; w_1 = -0.87}$$

Example

$$t = 3.69 - 0.87x$$

	Tumor Size	Survival	Predicted Survival
Case1	0.5	3.2	3.25
Case2	2.3	1.9	1.68
Case3	2.9	1.0	1.16



Coding

jupyter Linear regression in one variable Last Checkpoint: 3 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Help Trusted Python 3

Run Markdown

Import the usual libraries

In [1]:
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
```

We shall work with the dataset found in the file 'murderdata.txt', which is a 20 x 5 data matrix where the columns correspond to

Index (not for use in analysis)

Number of inhabitants

Percent with incomes below $5000

Percent unemployed

Murders per annum per 1,000,000 inhabitants

Reference:

Helmut Spaeth, Mathematical Algorithms for Linear Regression, Academic Press, 1991, ISBN 0-12-656460-4.

D G Kleinbaum and L L Kupper, Applied Regression Analysis and Other Multivariable Methods, Duxbury Press, 1978, page 150.

http://people.sc.fsu.edu/~jburkardt/datasets/regression

What to do?

We start by loading the data; today we will study how the number of murders relates to the percentage of unemployment.

In [2]:
```
data = np.loadtxt('murderdata.txt')
N, d = data.shape

unemployment = data[:,3]
murders = data[:,4]
```

Coding

jupyter Linear regression in one variable Last Checkpoint: 3 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Help

Trusted

Python 3

Run Markdown

Import the usual libraries

In [1]:
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
```

We shall work with the dataset found in the file 'murderdata.txt', which is a 20 x 5 data matrix where the columns correspond to

Index (not for use in analysis)

Coding Task

Compute the optimal coefficients:

- $\hat{w}_1 = \frac{\overline{xt} - \overline{x}\,\overline{t}}{\overline{x^2} - (\overline{x})^2}$
- $\hat{w}_0 = \overline{t} - \hat{w}_1 \overline{x}$

Make use of `np.dot` and `np.mean`. E.g., `np.dot(x,t) / N` computes $\overline{xt}$.

$\overline{t} = \frac{1}{N}\sum_{n=1}^{N} t_n$, $\overline{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$, $\overline{xt} = \frac{1}{N}\sum_{n=1}^{N} x_n t_n$, and $\overline{x^2} = \frac{1}{N}\sum_{n=1}^{N} {x_n}^2$

In [2]:
```
data = np.loadtxt('murderdata.txt')
N, d = data.shape

unemployment = data[:,3]
murders = data[:,4]
```

Questions?