Faculty of Science

L1 – Introduction & Linear Regression I

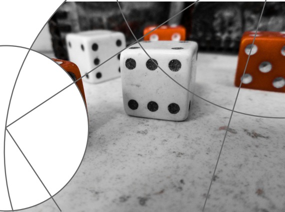
Modelling and Analysis of Data

Fabian Gieseke

Machine Learning Section
Department of Computer Science
University of Copenhagen

Universitetsparken 1, 1-1-N110
fabian.gieseke@di.ku.dk

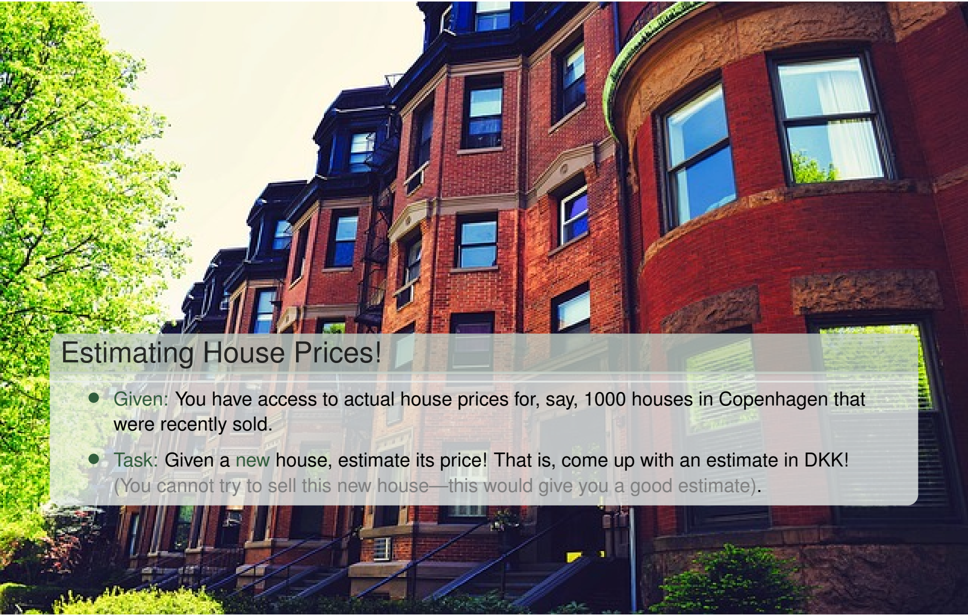
19 November 2019



Outline

Outline

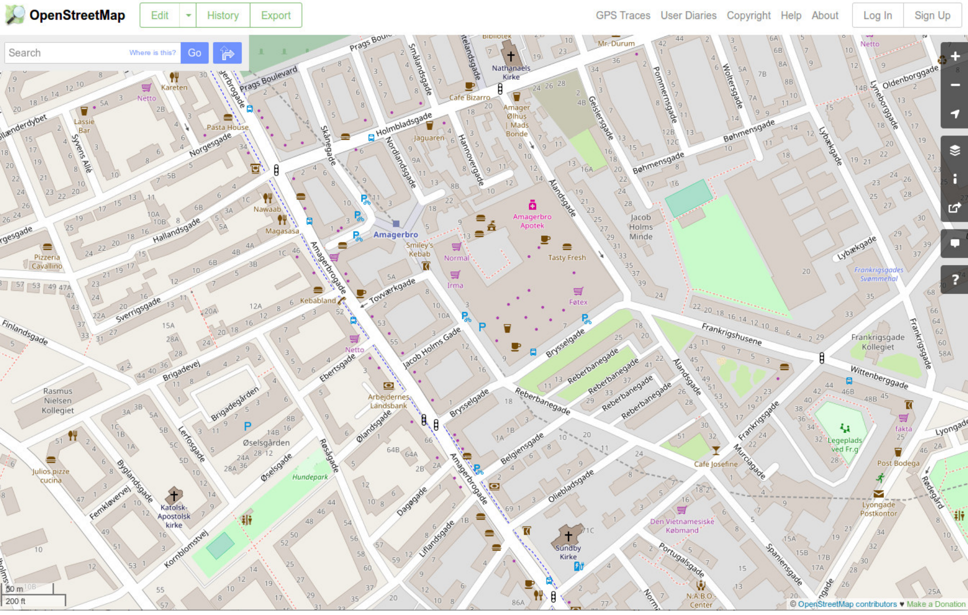
Motivation



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
Finlandsgade
Brigadevej
Amagerbrogade
Amagerbro
Tovværkgade
Jacob Holms Gade
Brysselgade
Reberbanegade
Frankrigshusene
Frankrigsgade
Wittenberggade
Bøhmensgade
Oldenborggade
Lybækgade
Ølandsgade
Øselsgade
Kornblomstvej
Byglandsgade
Lerfosgade
Rosagade
Dagøgade
Liflandsgade
Belgiensgade
Oliebladsgade
Portugalsgade
Spaniensgade
Murciagade
Ålandsgade
Geislersgade
Pommernsgade
Woltersgade
Nordlandsgade
Hannovergade
Skånegade
Smålandsgade
Syvens Allé
Ebertsgade
Femkløvervej
Netto
Kareten
Pasta House
Lassie Bar
Cafe Bizarro
Nathanaels Kirke
Mr. Durum
Amager Ølhus | Mads Bonde
Jaguaren
Nawaab
Magasasa
Pizzeria Cavallino
Amagerbro Apotek
Smiley's Kebab
Normal
Irma
Tasty Fresh
Føtex
Kebabland
Jacob Holms Minde
Frankrigsgades Svømmehal
Frankrigsgade Kollegiet
Rasmus Nielsen Kollegiet
Arbejdernes Landsbank
Øselsgården
Hundepark
Katolsk-Apostolsk kirke
Julios pizza cucina
Cafe Josefine
Legeplads ved Fr.g
Post Bodega
fakta
Lyongade Postkontor
Den Vietnamesiske Købmand
Sundby Kirke
N.A.B.O. Center
50 m
200 ft
© OpenStreetMap contributors ♥ Make a Donation

Motivation

Task: Regression

1. Given some data related to houses, estimate the price $y \in \mathbb{R}$ in DKK for each house!
2. Given some astronomical object, estimate its distance $y \in \mathbb{R}$ to Earth!
3. Given some stock, estimate the value $y \in \mathbb{R}$ it will have in ten days!
4. ...

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

Example

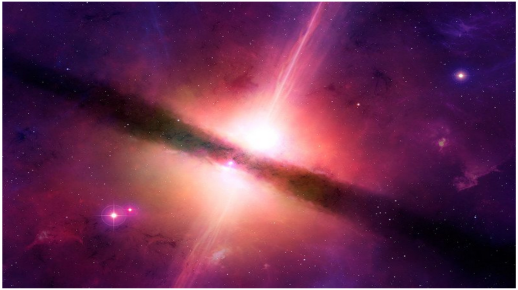
New Scientist

HOME NEWS TECHNOLOGY SPACE PHYSICS HEALTH EARTH HUMANS LIFE TOPICS EVENTS JOBS SUBSCRIBE SEARCH LOG IN

Home | News | Physics | Space

DAILY NEWS 6 December 2017

Most distant quasar ever seen is way too big for our universe



Quasars – discs of gas around supermassive black holes – are incredibly bright
Mark Garlick/Science Photo Library

By **Leah Crane**

A quasar has been spotted 13 billion years away from us. It's the farthest one we've ever seen, and it already existed 690 million years after the birth of the universe. Finding a quasar – a supermassive black hole with a bright disc of material circling it – from so long ago indicates that huge black holes must have formed quickly in

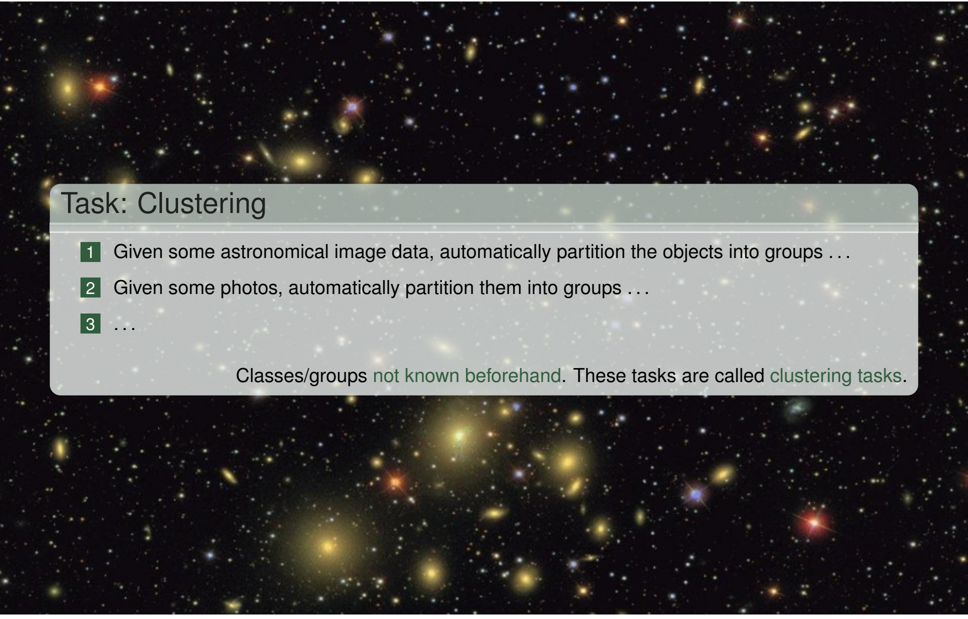
Motivation

Task: Classification

1. Given some astronomical image data, classify each object as star ($y = 0$) or galaxy ($y = 1$).
2. Given some photos, classify them into “cats” ($y = 0$), “dogs” ($y = 1$), or “other” ($y = 2$).
3. ...

These tasks are called classification tasks since we are interested in a class $y \in \mathcal{Y}$ with $|\mathcal{Y}| < \infty$.

Motivation

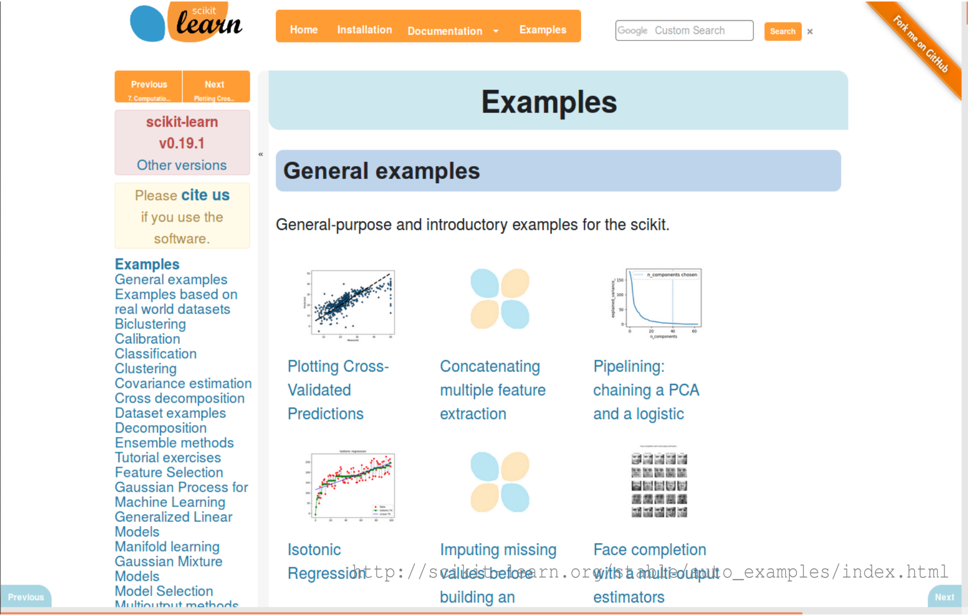


Task: Clustering

1. Given some astronomical image data, automatically partition the objects into groups . . .
2. Given some photos, automatically partition them into groups . . .
3. . . .

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

Demo: Machine Learning & Scikit-Learn



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

About Us

Lecturers

Kim Steenstrup Pedersen
(course responsible)
kimstp@di.ku.dk

Fabian Gieseke
fabian.gieseke@di.ku.dk

Bulat Ibragimov
bulat@di.ku.dk

Teaching Assistants

- Alessandro Falcione
- Camilla Kergel Petersen
- Nikolaj Overgaard Sørensen (A)
- Johan Pedersen
- Nichlas Langhoff Rasmussen
- Bjarke Wheatley Enkelund
- Rune Vium Søndergaard (A)

About You

Show of hands: How many of you . . .

1. did not attend MASD?

About You

Show of hands: How many of you . . .

1. did not attend MASD?
2. are not CS students?

About You

Show of hands: How many of you . . .

1. did not attend MASD?
2. are not CS students?
3. have never taken a statistics course?

About You

Show of hands: How many of you …

1. did not attend MASD?
2. are not CS students?
3. have never taken a statistics course?
4. have never programmed in Python?

About You

Show of hands: How many of you . . .

1. did not attend MASD?
2. are not CS students?
3. have never taken a statistics course?
4. have never programmed in Python?
5. have not worked with Jupyter notebooks yet?

Tentative Schedule

1. Introduction & Linear Regression I (FG)
2. Linear Regression II (FG)
3. Non-Linear Regression & Regularization (FG)
4. Statistics (BI)
5. Inequalities, Convergence of Random Variables, and Hypothesis Tests (KSP)
6. Linear Modelling: A MLE Approach + Coin Game (KSP)
7. Bayesian Perspective of Regression (KSP)
8. Principal Component Analysis (BI)
9. Classification I (BI)
10. Classification II (BI)
11. Sampling (KSP)
12. Clustering & Evaluation & Wrap-Up (BI)

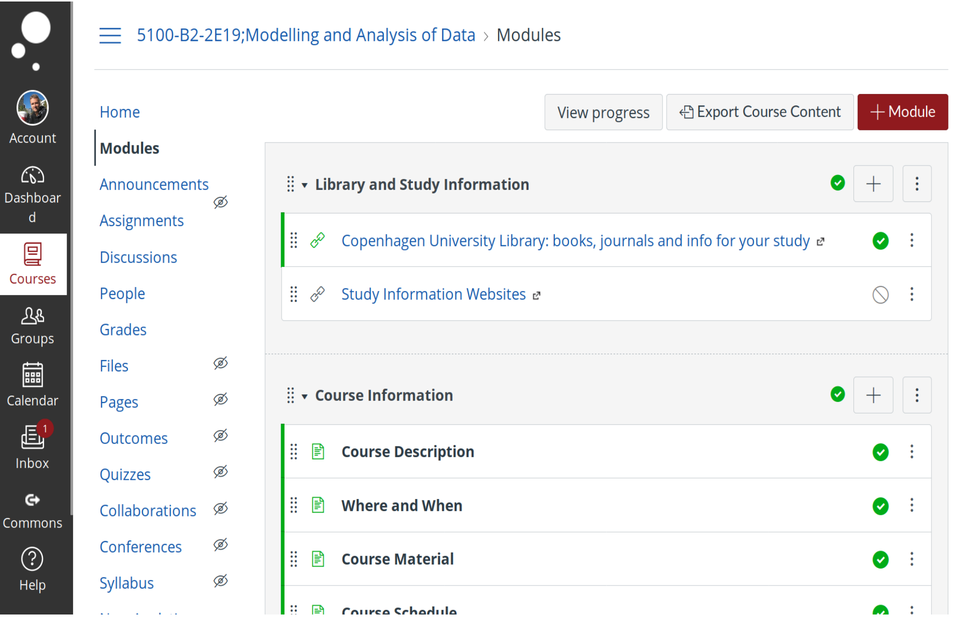
Qualifications

Recommended Academic Qualifications

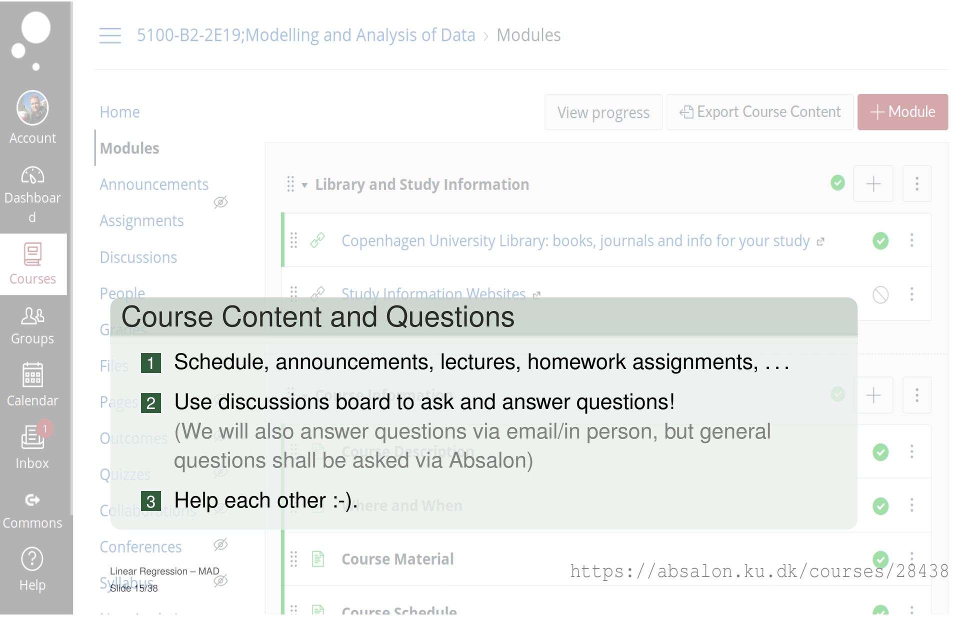
"Mathematical knowledge equivalent to those obtained in the courses LinAlgDat, DMA, and MASD or similar. Basic knowledge of programming."

- MAD is a partner course with MASD (Mathematical Analysis and Statistics for Computer Scientists) that took place in block 1.
  1. MASD focused on the statistical approach to data science (basics).
  2. MAD will turn towards more advanced statistics and machine learning.
- MAD builds on the statistics and calculus from MASD.
- MAD also relies heavily on linear algebra!

Course Organization: Absalon



Course Organization: Absalon



Course Content and Questions

1. Schedule, announcements, lectures, homework assignments, . . .
2. Use discussions board to ask and answer questions!
   (We will also answer questions via email/in person, but general questions shall be asked via Absalon)
3. Help each other :-).

https://absalon.ku.dk/courses/28438

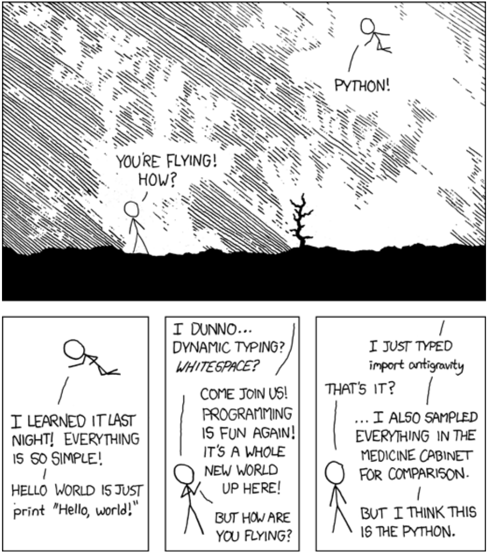
Where and When?

Course

1. Lectures
   - Tuesday: 09:15-11:00: Aud 01, Universitetsparken 5, HCO
   - Thursday: 10:15-12:00: Teilum A, Frederik Vs vej 1
2. Homework Café
   - Tuesday: 11:15-13:00: Aud 01, Universitetsparken 5, HCO
3. Practical Sessions
   1. TA session 1: Thursday: 13:15-15:00: A110 (Universitetsparken 5, HCO)
   2. TA session 2: Thursday: 13:15-15:00: A107 (Universitetsparken 5, HCO)
   3. TA session 3: Thursday: 13:15-15:00: A102 (Universitetsparken 5, HCO)
   4. TA session 4: Thursday: 13:15-15:00: A101 (Universitetsparken 5, HCO)
   5. TA session 5: Thursday: 13:15-15:00: A112 (Universitetsparken 5, HCO)
   6. TA session 6: Thursday: 13:15-15:00: A105 (Universitetsparken 5, HCO)
   7. TA session 7: Thursday: 13:15-15:00: C103 (Universitetsparken 5, HCO)

Whenever there is a lecture in the morning, there will be practical sessions in the afternoon.

We will make use of Python!!!



https://xkcd.com/353/

Need help with Python? Homework café today!

Assignments & Exam (Tentative)

Assignments

There will be five take-home assignments. The assignments will be handed out on Monday morning (around 10:00) and will have to be handed in **1-2 weeks later by Tuesday night, 23:59**.

1. A1 (18.11.2019 - 26.11.2019)
2. A2 (25.11.2019 - 03.12.2019)
3. A3 (02.12.2019 - 10.12.2019)
4. A4 (09.12.2019 - 17.12.2019)
5. A5 (16.12.2019 - 07.01.2020)

See the individual assignments for details and potential changes.

Assignments & Exam (Tentative)

Assignments

There will be five take-home assignments. The assignments will be handed out on Monday morning (around 10:00) and will have to be handed in **1-2 weeks later by Tuesday night, 23:59**.

1. A1 (18.11.2019 - 26.11.2019)
2. A2 (25.11.2019 - 03.12.2019)
3. A3 (02.12.2019 - 10.12.2019)
4. A4 (09.12.2019 - 17.12.2019)
5. A5 (16.12.2019 - 07.01.2020)

See the individual assignments for details and potential changes.

- **All but one of these must be passed in order to be eligible for the exam.** In general, passing means to get $\geq 40\%$ of the points per assignment.
- For the assignments, you are allowed and encouraged to discuss with each other. However, **the assignments are individual**; don't copy code or text from each other. This will be considered plagiarism.

Assignments & Exam (Tentative)

Exam

- The exam is a final take-home exam for 7 days (calendar week 3, 13.01.2020 – 19.01.2020)
- **For the exam, you are not allowed to work/discuss with each other.**

Next Steps

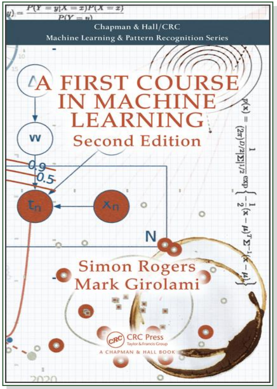
What to do next?

1. Optional: Join the homework café today (11:15–13:00, Aud01). Plan for today:
   - Get started with Assignment 1.
   - Get Python on your laptop up and running. Go through:
     - `https://docs.python.org/3/tutorial/`
     - `https://docs.scipy.org/doc/numpy/user/quickstart.html`
2. Work on Assignment 1 (deadline: 26.11.2019)
3. Next lecture: Thursday, 10:15-12:00 (Teilum A, Frederik Vs vej 1)

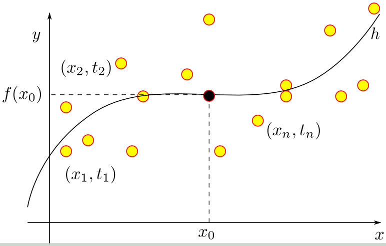
Outline

1. Motivation & Organization
2. Linear Regression I
3. Summary & Outlook

Course Material (Next Lectures)



Regression



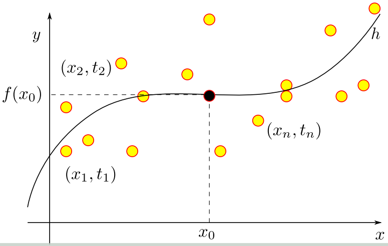
A Learning Problem

- **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



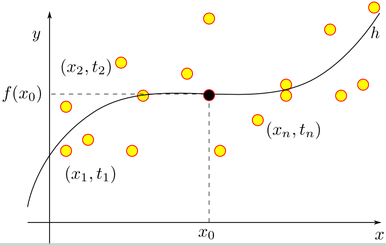
A Learning Problem

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

Regression



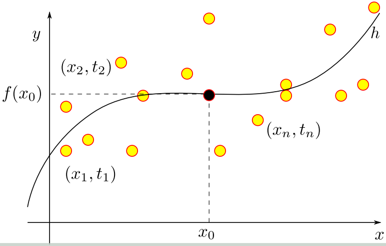
A Learning Problem

- **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 $f(\mathbf{x}_0)$!

Case: Murder Rates

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

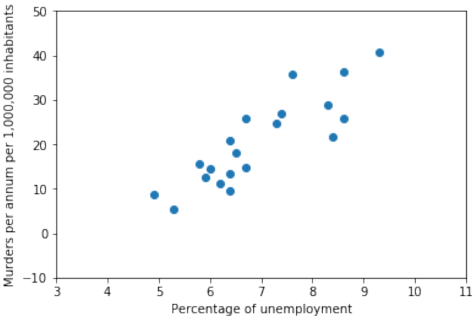
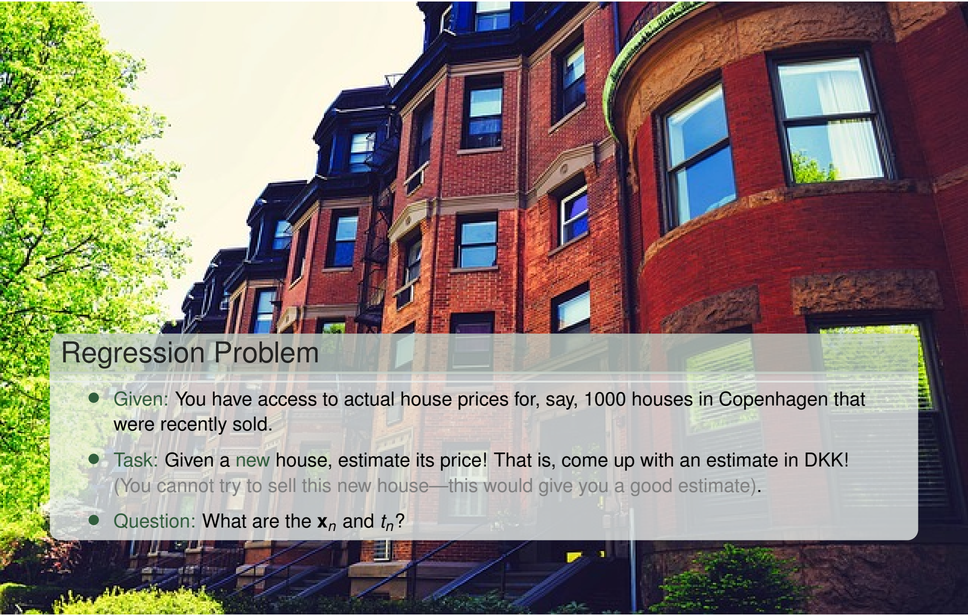


Figure: Murder rates versus unemployment rates in an American city[1]

[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

Case: House Prices



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 are Column Vectors

- In most of the ML literature, vectors are written as column vectors:

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

- That's annoying to type, so we will write $\boldsymbol{x} = [x_1, x_2, \ldots, x_D]^T$.

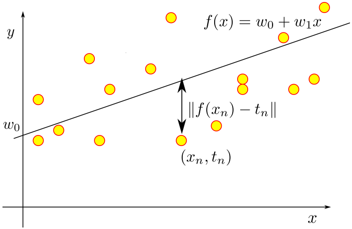
Linear Regression: Single Input Variable

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

Linear Regression: Single Input Variable

- 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

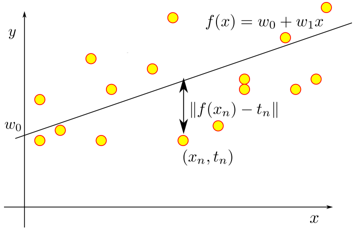
$$f(x) = f(x; w_0, w_1) = w_0 + w_1 x$$



Linear Regression: Single Input Variable

- 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:

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

Case: Murder Rates

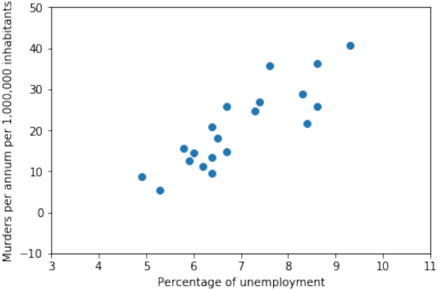


Figure: Murder rates versus unemployment rates in an American city

Case: Murder Rates

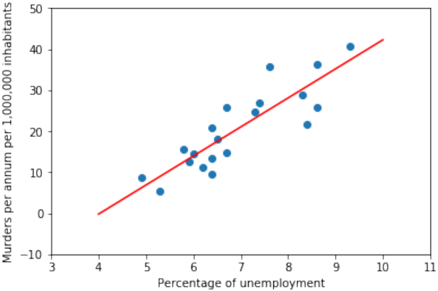


Figure: What is a “good” model? How can we measure its “quality”?

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 $f(x_n; w_0, w_1)$.

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

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

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.:

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

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

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

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.:

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

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

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

Computing the Optimal Model

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

- 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?

Computing the Optimal Model

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

- We would like to find the two coefficients w_0 and w_1 that minimize the above objective! Question: How can we find these coefficients?
- We have a function with two variables w_0 and w_1 and are searching for vector $\boldsymbol{w} = [w_0, w_1]^T$ corresponding to a minimum w.r.t. $\mathcal{L}$. Thus, the gradient of $\mathcal{L}$ must vanish at $\boldsymbol{w}$ (necessary condition!):

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

Computing the Optimal Model

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

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

- One can simplify the objective as follows:

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

Computing the Optimal Model

- One can simplify the objective as follows:

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

- Hence, one directly obtains the partial derivatives:

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

Proof I (Warm-Up)

- Let $f(x, y)$ be a function in two variables.

Proof I (Warm-Up)

- Let $f(x, y)$ be a function in two variables.
- Assume that we can find, for any fixed but arbitrary x, a global minimum y^* (either a constant or a term that depends on x). Let's write $y^*(x)$ to emphasize that it might still depend on x (e.g., $y^*(x) = 3$ or $y^*(x) = 1 - x$).

Proof I (Warm-Up)

- Let $f(x, y)$ be a function in two variables.
- Assume that we can find, for any fixed but arbitrary x, a global minimum y^* (either a constant or a term that depends on x). Let's write $y^*(x)$ to emphasize that it might still depend on x (e.g., $y^*(x) = 3$ or $y^*(x) = 1 - x$).
- Next, let us consider the following function: $g(x) = f(x, y^*(x))$. Further, assume that we can find a x^* that is a global minimum for g.

Proof I (Warm-Up)

- Let $f(x, y)$ be a function in two variables.
- Assume that we can find, for any fixed but arbitrary x, a global minimum y^* (either a constant or a term that depends on x). Let's write $y^*(x)$ to emphasize that it might still depend on x (e.g., $y^*(x) = 3$ or $y^*(x) = 1 - x$).
- Next, let us consider the following function: $g(x) = f(x, y^*(x))$. Further, assume that we can find a x^* that is a global minimum for g.
- Then, we have

$$f(x, y) \geq f(x, y^*(x)) = g(x) \geq g(x^*) = f(x^*, y^*(x^*))$$

for any x and y.

Proof I (Warm-Up)

- Let $f(x, y)$ be a function in two variables.
- Assume that we can find, for any fixed but arbitrary x, a global minimum y^* (either a constant or a term that depends on x). Let's write $y^*(x)$ to emphasize that it might still depend on x (e.g., $y^*(x) = 3$ or $y^*(x) = 1 - x$).
- Next, let us consider the following function: $g(x) = f(x, y^*(x))$. Further, assume that we can find a x^* that is a global minimum for g.
- Then, we have

$$f(x, y) \geq f(x, y^*(x)) = g(x) \geq g(x^*) = f(x^*, y^*(x^*))$$

  for any x and y.
- Hence, the point $(x^*, y^*(x^*))$ is a global minimum of f!

Proof I (Warm-Up)

- Let $f(x,y)$ be a function in two variables.
- Assume that we can find, for any fixed but arbitrary x, a global minimum y^* (either a constant or a term that depends on x). Let's write $y^*(x)$ to emphasize that it might still depend on x (e.g., $y^*(x) = 3$ or $y^*(x) = 1 - x$).
- Next, let us consider the following function: $g(x) = f(x, y^*(x))$. Further, assume that we can find a x^* that is a global minimum for g.
- Then, we have

$$f(x,y) \geq f(x, y^*(x)) = g(x) \geq g(x^*) = f(x^*, y^*(x^*))$$

  for any x and y.
- Hence, the point $(x^*, y^*(x^*))$ is a global minimum of f!
- Warning: Not always possible! For instance: $f(x,y) = x^2 + y^2 - 10xy$

Similarly: Global maximum.

Proof I

- $\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right) \stackrel{!}{=} 0$ leads to $\hat{w}_0 = \overline{t} - w_1 \overline{x}$.

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

Proof I

- $\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right) \stackrel{!}{=} 0$ leads to $\hat{w}_0 = \overline{t} - w_1 \overline{x}$.
- Since $\frac{\partial^2 \mathcal{L}}{\partial w_0^2} = 2 > 0$, we know that this is a global minimum (the second derivative is a positive constant; hence single global minimum!). Thus, for any w_1, we know the optimal $\hat{w}_0$!

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

Proof I

- $\frac{\partial \mathcal{L}}{\partial w_0} = 2w_0 + 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right) \stackrel{!}{=} 0$ leads to $\hat{w}_0 = \bar{t} - w_1\bar{x}$.
- Since $\frac{\partial^2 \mathcal{L}}{\partial w_0^2} = 2 > 0$, we know that this is a global minimum (the second derivative is a positive constant; hence single global minimum!). Thus, for any w_1, we know the optimal $\hat{w}_0$!
- Plugging in $\hat{w}_0$ leads to

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

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

Proof I

- Plugging in $\hat{w}_0$ leads to

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

- Now, enforcing $\frac{\partial \mathcal{L}}{\partial w_1} \stackrel{!}{=} 0$ leads to $\hat{w}_1 = \frac{\overline{xt} - \overline{x}\,\overline{t}}{\overline{x^2} - (\overline{x})^2}$

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

Proof I

- Plugging in $\hat{w}_0$ leads to

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

- Now, enforcing $\frac{\partial \mathcal{L}}{\partial w_1} \stackrel{!}{=} 0$ leads to $\hat{w}_1 = \frac{\overline{xt} - \overline{x}\,\overline{t}}{\overline{x^2} - (\overline{x})^2}$
- Since $\frac{\partial^2 \mathcal{L}}{\partial w_1^2} = 2\left(\frac{1}{N}\sum_{n=1}^{N} {x_n}^2\right) - 2\overline{x}\,\overline{x} = \frac{2}{N}\sum_{n=1}^{N}(x_n - \overline{x})^2 > 0$, we know that $\hat{w}_1$ is a global minimum as well (here, we assume that not all the x_n are the same).

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

Proof I

- Plugging in $\hat{w}_0$ leads to

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

- Now, enforcing $\frac{\partial \mathcal{L}}{\partial w_1} \stackrel{!}{=} 0$ leads to $\hat{w}_1 = \frac{\overline{xt} - \overline{x}\,\overline{t}}{\overline{x^2} - (\overline{x})^2}$
- Since $\frac{\partial^2 \mathcal{L}}{\partial w_1^2} = 2\left(\frac{1}{N}\sum_{n=1}^{N} {x_n}^2\right) - 2\overline{x}\,\overline{x} = \frac{2}{N}\sum_{n=1}^{N}(x_n - \overline{x})^2 > 0$, we know that $\hat{w}_1$ is a global minimum as well (here, we assume that not all the x_n are the same).
- Thus, we have $\mathcal{L}(w_0, w_1) \geq \mathcal{L}(\hat{w}_0, w_1) \geq \mathcal{L}(\hat{w}_0, \hat{w}_1)$ for any (w_0, w_1), i.e., $(\hat{w}_0, \hat{w}_1)$ is a global minimum of $\mathcal{L}$!

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

Coding Time!

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.

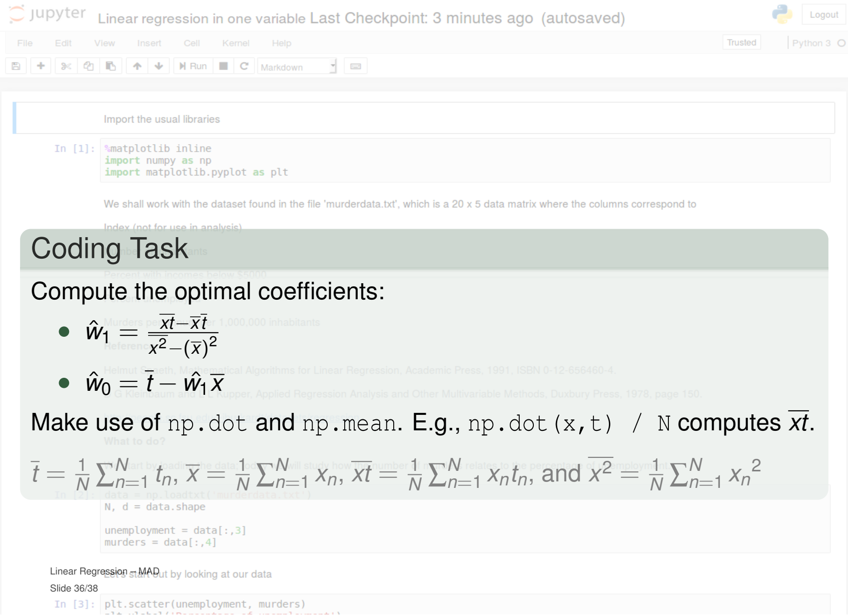
In [2]:
```
data = np.loadtxt('murderdata.txt')
N, d = data.shape

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

Let's start out by looking at our data

In [3]:
```
plt.scatter(unemployment, murders)
```

Coding Time!



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 $\overline{x^2} = \frac{1}{N}\sum_{n=1}^{N} {x_n}^2$

Outline

Summary & Outlook

Today

- Seen how to model a real-world problem.
- Seen how to derive and implement the linear regression model for $1D$.
- Recalled the linear algebra needed to phrase and solve these models.

Outlook

- We will consider the “multi-dimensional” case ...
- We will implement the multivariate case in Python ...
- We will prove the optimality of the solution ...