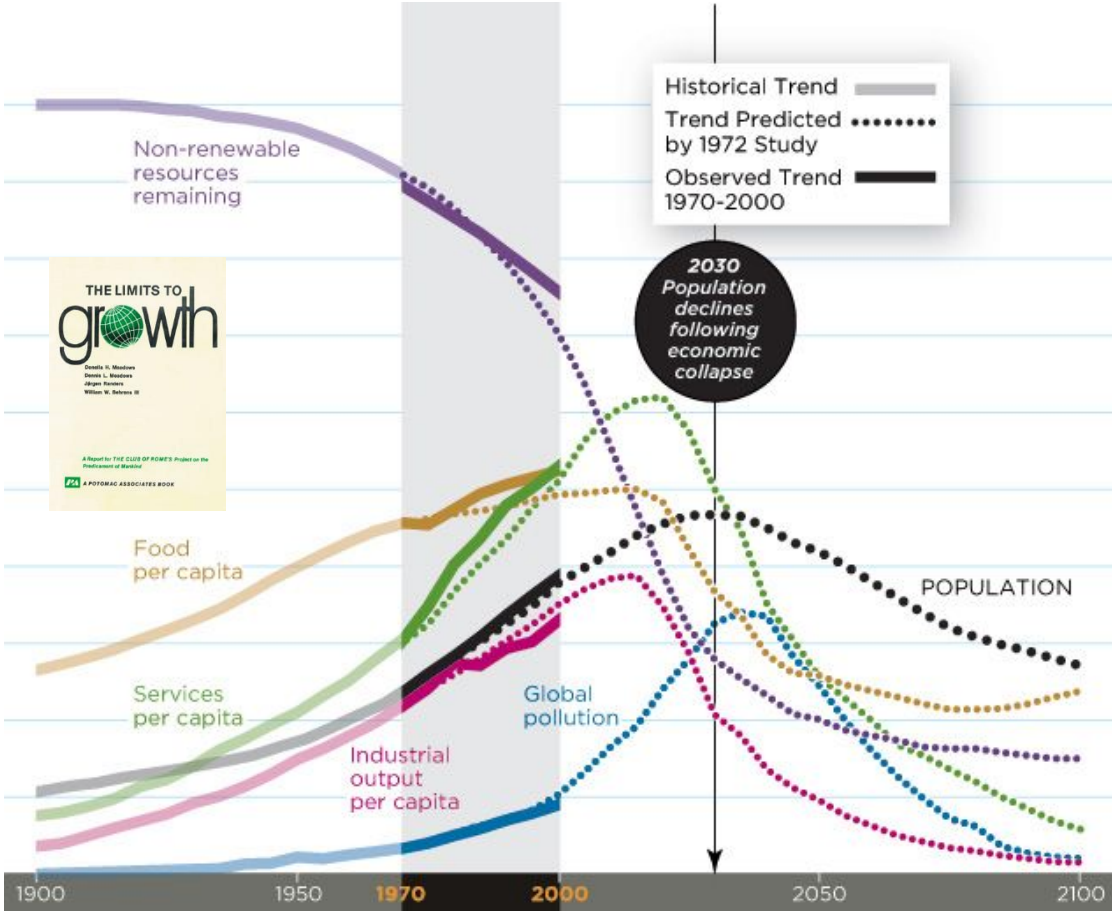
Applied Math
004 – Optimization

Linear and non-linear optimization

Christian Wallraven
Cognitive Systems Lab
Departments of Artificial Intelligence, Brain and Cognitive Engineering
christian.wallraven+AMF2023@gmail.com
http://cogsys.korea.ac.kr

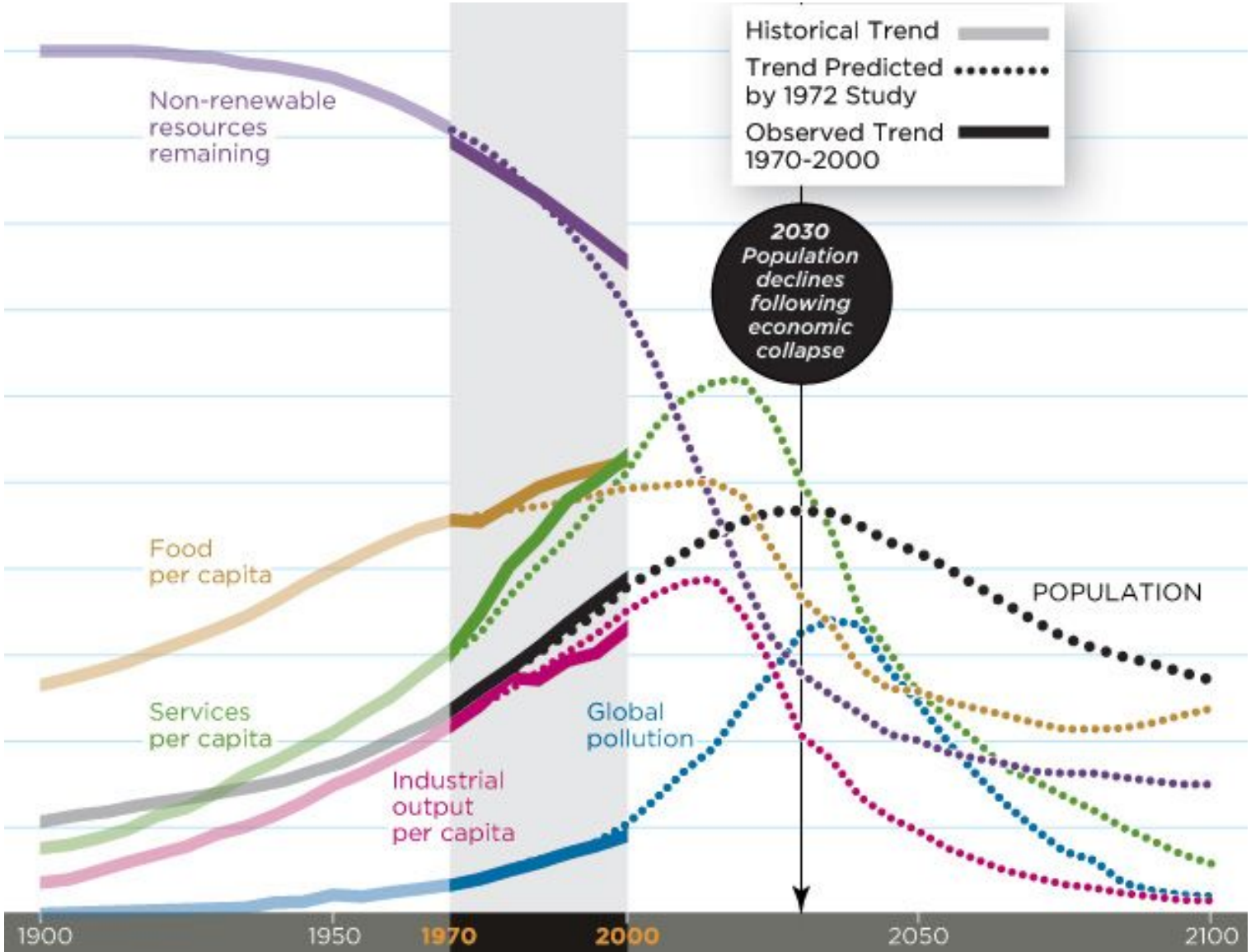
Motivation

- In the famous study ‘The Limits to Growth’ in 1972, scientists predicted a number of key variables of economic development
- Importantly, it predicted that as humanity uses up non-renewable resources, and as global pollution rises, an economic collapse may be imminent as early as 2030
- 40 years later we can look at how well the predictions do



http://www.smithsonianmag.com/science-nature/Looking-Back-on-the-Limits-of-Growth.html#

Motivation


Historical Trend
Trend Predicted by 1972 Study
Observed Trend 1970-2000
2030 Population declines following economic collapse
Non-renewable resources remaining
Food per capita
Services per capita
Industrial output per capita
Global pollution
POPULATION
1900
1950
1970
2000
2050
2100

Motivation

CognitiveSystems

- Fitting data is one of the most important methods in many scientific fields used for
  - model building and verification
  - making predictions
  - simplifying processes
  - assessing stability of systems (e.g., the financial market)
  - …

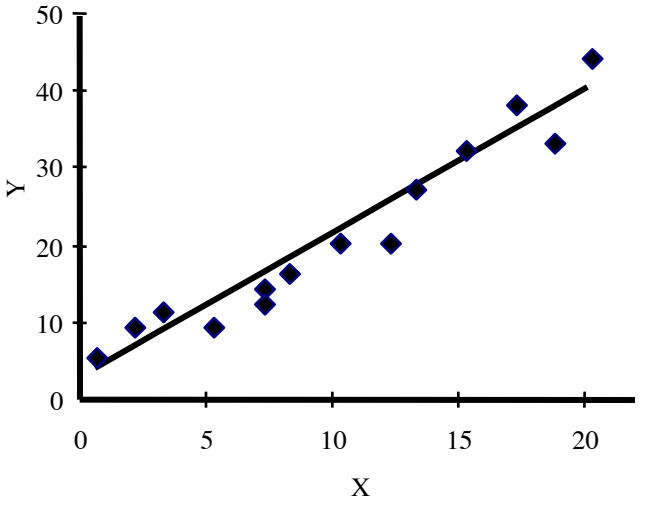
Outline

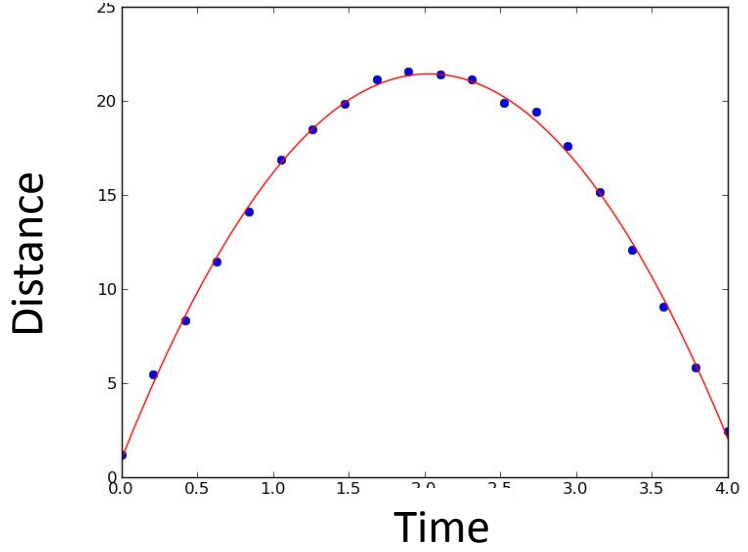
CognitiveSystems

- Problem formulation
  - the concept of an “error function” (“loss function”)
  - the concept of “least squares”
- Fitting a line to data
  - connection to correlation
  - robustness to outliers
- Linear least-squares with systems of equations
- Fitting polynomials to data
- Outlook

Problem formulation

- Given *n* observed data points (x_i, y_i) fit **known** function *f* with *k* parameters α_j such that the error becomes minimal
- We therefore need to define a suitable error-function *E*
  - sometimes also called loss-function, error, goodness-of-fit





Simple idea

- What do we need?
  - measure error from predicted $f(x_i)$ with measured values y_i
- At each point x_i, we could do: $error_i = y_i - f(x_i)$
- Then, in order to get the total error, we sum:

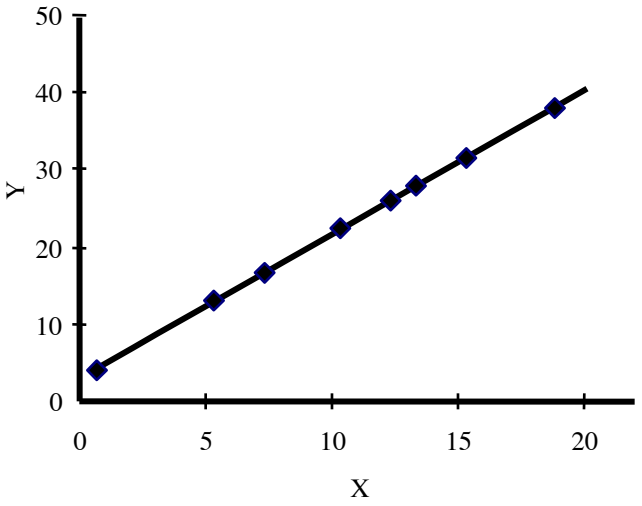
$$E(f) = \sum_{i=1}^{n} error_i = \sum_{i=1}^{n} y_i - f(x_i)$$

- But: this is not a good error measure, as the total error can become zero, when we make equal positive and negative prediction errors

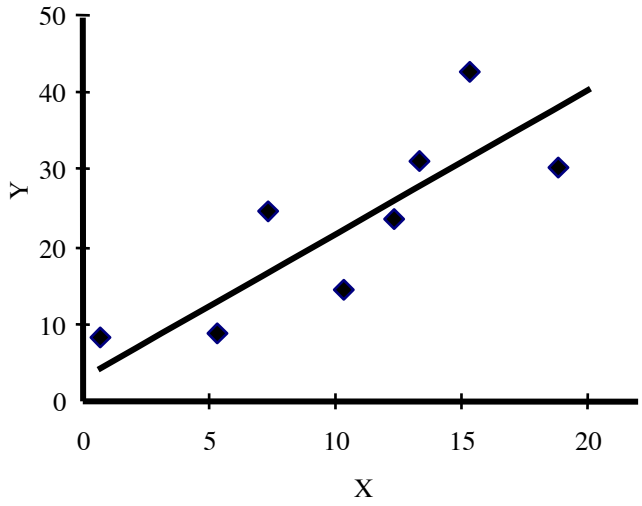
Simple idea

- But: this is not a good error measure, as the total error can become zero, when we make equal positive and negative prediction errors

$$E(f) = \sum_{i=1}^{n} error_i = \sum_{i=1}^{n} y_i - f(x_i)$$



$E(f) = 0$



$E(f) = 0$

Simple idea

- One step smarter:
- At each point x_i, we could do: $error_i = |y_i - f(x_i)|$
- Then, in order to get the total error, we sum:

$$E(f) = \sum_{i=1}^{n} error_i = \sum_{i=1}^{n} |y_i - f(x_i)|$$

- It turns out that using the absolute value results in a good error measure – just not one that is very easy to optimize
  - see Outlook!

Least-squares fitting

CognitiveSystems

- Instead of the absolute value, square each error!
- At each point x_i, we could do: $error_i = \left(y_i - f(x_i)\right)^2$
- Then, in order to get the total error, we sum:

$$E(f) = \sum_{i=1}^{n} error_i = \sum_{i=1}^{n} \left(y_i - f(x_i)\right)^2$$

- In order to determine our fit, we need to find the parameters α, for which *E(f)* is minimal

$$\underset{\alpha}{\operatorname{argmin}}\, E(f_\alpha) = \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - f_\alpha(x_i)\right)^2$$

- **This is called least-squares fitting**

Least-squares fitting

- Note that this formally is equivalent to:

$$E(f) = \sum_{i=1}^{n} \left(y_i - f(x_i)\right)^2 = \left\langle \vec{y} - \overrightarrow{f(x)}, \vec{y} - \overrightarrow{f(x)} \right\rangle = \left\| \vec{y} - \overrightarrow{f(x)} \right\|^2$$

  where the vectors have *n* dimensions and <,> denotes the standard scalar product.

- Hence, here we use the **L_2-norm** between two vectors to establish the error-function.

Fitting a line in a least-squares sense

- The most simple example of data fitting is that of fitting a line to some data: $f(x) = wx + b$
- With this, the error-function becomes:

$$E(w,b) = \sum_{i=1}^{n} \left(y_i - (wx_i + b) \right)^2$$

- As we will see, we can actually determine a closed-form solution for this this!

Fitting a line in a least-squares sense

- We are looking for:

$$\underset{w,b}{\operatorname{argmin}} E(w,b) = \underset{w,b}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - (wx_i + b) \right)^2$$

- Remembering calculus for the necessary condition for a minimum of a function:

$$\frac{\partial E(w,b)}{\partial w} = 0 \quad \wedge \quad \frac{\partial E(w,b)}{\partial b} = 0$$

Fitting a line in a least-squares sense

$$\frac{\partial E(w,b)}{\partial w} = \frac{\partial}{\partial \mathrm{w}} \sum \left(y_i - (wx_i + b) \right)^2 = \sum \frac{\partial}{\partial \mathrm{w}} \left(y_i - (wx_i + b) \right)^2$$

Fitting a line in a least-squares sense

$$\frac{\partial E(w,b)}{\partial b} = \frac{\partial}{\partial \mathrm{b}} \sum \left(y_i - (wx_i + b) \right)^2 = \sum \frac{\partial}{\partial \mathrm{b}} \left(y_i - (wx_i + b) \right)^2$$

Fitting a line in a least-squares sense

$$\frac{\partial E(w,b)}{\partial w} = 2\sum wx_i^2 + bx_i - x_i y_i$$

$$\frac{\partial E(w,b)}{\partial b} = 2\sum wx_i + b - y_i$$

Fitting a line in a least-squares sense

$$0 = 2(wn\mu_x + nb - n\mu_y)$$

$$\Leftrightarrow b = \mu_y - w\mu_x \therefore$$

Fitting a line in a least-squares sense

$$b = \mu_y - w\mu_x$$

$$w = \frac{\sigma_{xy}}{\sigma_x^2}$$

$$\Rightarrow f(x) = \frac{\sigma_{xy}}{\sigma_x^2} x + (\mu_y - w\mu_x)$$

- We have a closed-form solution for fitting a line in a least-squares sense to data!!
- The slope of the line depends on the variance and covariance of the data

Python code for fitting line

CognitiveSystems

```
[ ] # number of points (default =30)
    NUMP = 30
    # number of outliers (default =10)
    NUMO = 10

    # create points on noisy line y = 1.1x + 2
    X0 = 10*np.random.rand(1,NUMP)
    Y0 = (1.1*X0 +2 +.3*np.random.rand(1,NUMP))+.2*np.random.randn(1,NUMP)
```

```
[ ] # create outliers
    X1 = 10 * np.random.rand(1,NUMO)
    Y1 = 5 * np.random.rand(1,NUMO) + 5 *np.random.randn(1,NUMO)
```

```
[ ] # stack dataset together
    dataWithOutlierX=np.hstack((X0,X1))
    dataWithOutlierY=np.hstack((Y0,Y1))
```

```
[ ] # determine covariance for line data
    covXY = np.cov(X0,Y0)
    # determine covariance for all data
    covWithOutlierXY = np.cov(dataWithOutlierX,dataWithOutlierY)
```

```
[ ] # using covariances and variances, determine slopes
    w=covXY[0,1]/np.var(X0)
    wWithOutlier=covWithOutlierXY[0,1]/np.var(dataWithOutlierX)
```

```
[ ] # using slopes and mean, determine intercepts
    b=np.mean(Y0)-w*np.mean(X0)
    bWithOutlier=np.mean(dataWithOutlierY)-wWithOutlier*np.mean(dataWithOutlierX)
```

```
[ ] # plot result
    plt.plot(X0.T,w*X0.T+b,label='without outliers')
    plt.plot(dataWithOutlierX.T,wWithOutlier*dataWithOutlierX.T+bWithOutlier,label='with outliers')
    plt.grid()
    plt.scatter(X0,Y0,label='')
    plt.scatter(X1,Y1,label='')
    plt.legend()
    plt.show()
```

see notebook

Fitting a line in a least-squares sense

- Remembering the definition of correlation

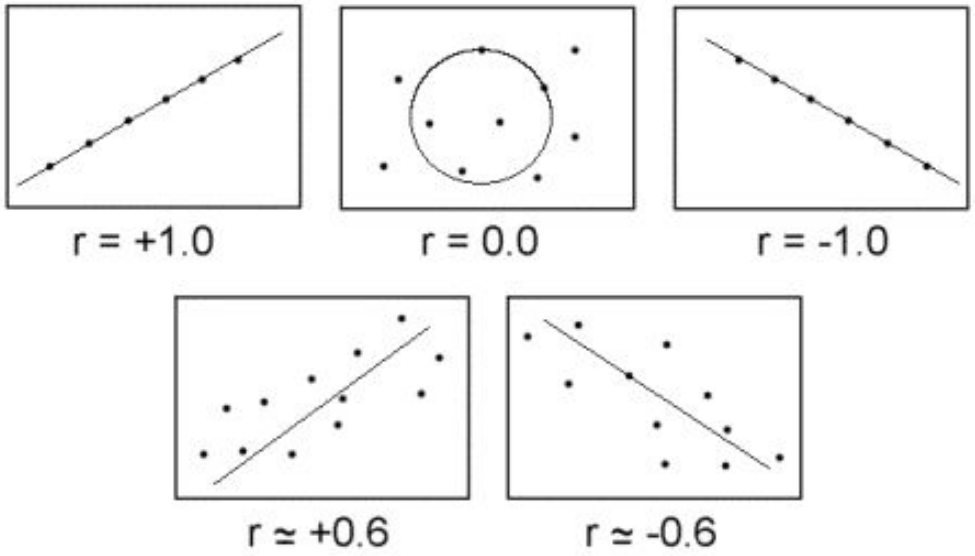
$$r = \frac{\sigma_{xy}}{\sqrt{\sigma_x^2}\sqrt{\sigma_y^2}}$$

$$\Rightarrow w = \frac{\sigma_{xy}}{\sigma_x^2} = r\frac{\sigma_y}{\sigma_x}$$

- Fitting a line in a least-squares sense to data means to calculate the correlation-coefficient of the data.
- Turning this around – every time you read about correlation, **you are reading about a linear fit!**

A few words about correlation

- Below are a few typical examples of correlation plots



- Note, that due to noise in your measurements, perfect correlations never will happen for real data!

http://allpsych.com/researchmethods/images/correlations.gif

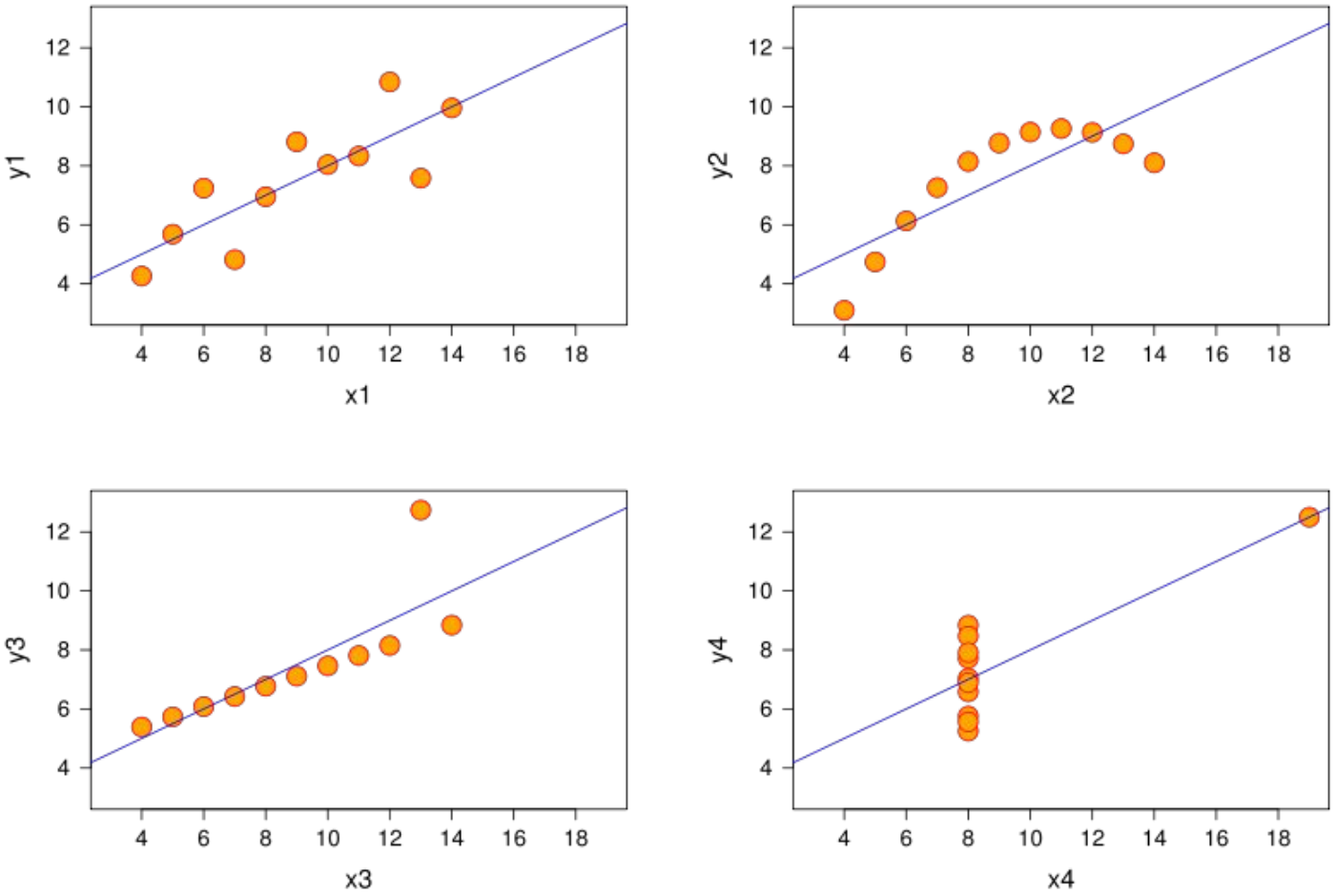
Determination coefficient r^2

- The correlation coefficient is given by r.
- r^2 amounts to the **proportion of variation** in one variable that is explained by the variation in the other variable
- A study of how visual thresholds correlate with acoustic thresholds finds an $r = 0.7$, amounting to an $r^2 = 0.49 = 49\%$
- Therefore, the correlation only explains 49% of the data variation, leaving the other half unexplained!

When you read a correlation value r, implicitly always square the value in order to get a feeling for the size of the effect!!

Dangers of correlation

- All of these datasets have r=0.816



http://upload.wikimedia.org/wikipedia/commons/b/b6/Anscombe.svg

Correlation: determining causation

- There is a strong correlation between people who had **oatmeal** for breakfast as a child and cancer, versus people who had **Frosted Flakes** for breakfast as a child
- Why?

http://fitnessbuff1.files.wordpress.com/2009/10/oatmeal.jpg

http://www.travelblog.org/Photos/2847481

Correlation: determining causation

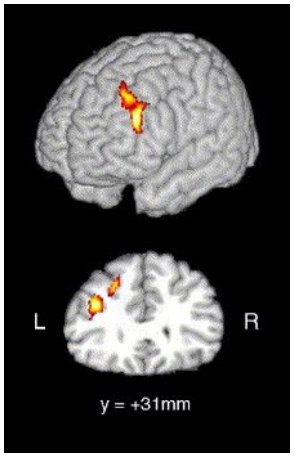
- Young children who sleep with the light on are much more likely to develop myopia in later life.
  - Quinn GE, Shin CH, Maguire MG, Stone RA (May 1999). "Myopia and ambient lighting at night". Nature 399 (6732): 113–4.
- Therefore, sleeping with the light on causes myopia.
- Really?

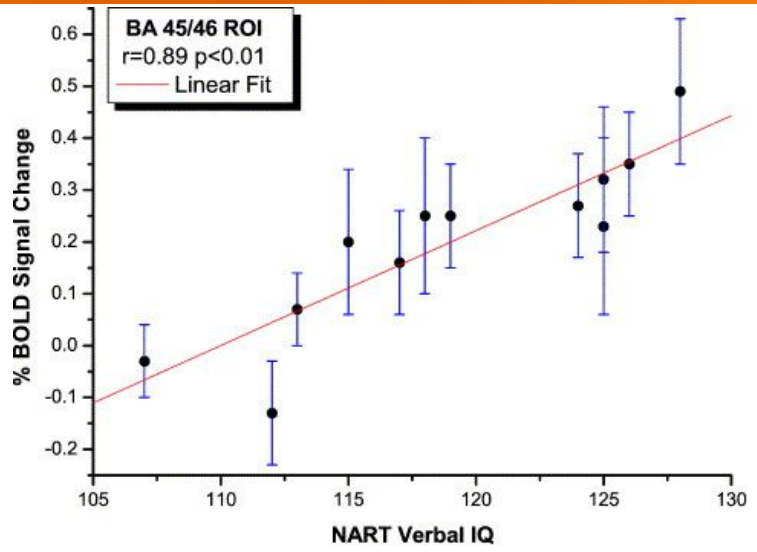
Correlation: determining causation

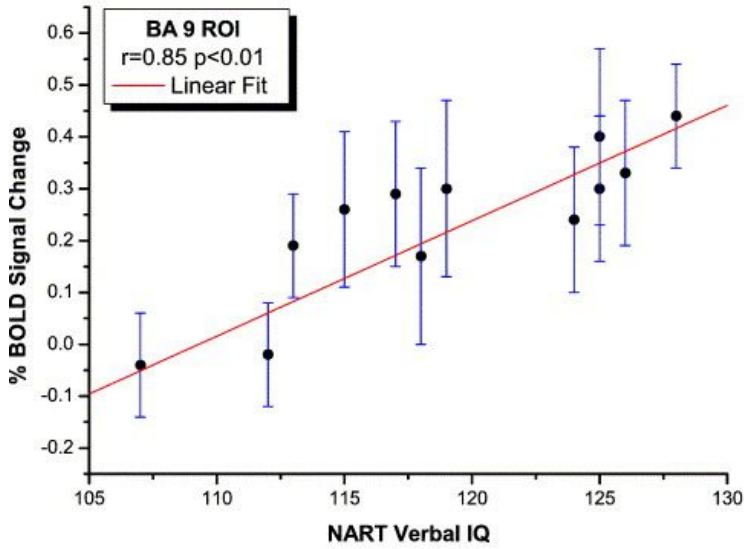
- Young children who sleep with the light on are much more likely to develop myopia in later life.
  - Quinn GE, Shin CH, Maguire MG, Stone RA (May 1999). "Myopia and ambient lighting at night". Nature 399 (6732): 113–4.
- Therefore, sleeping with the light on causes myopia.
- But:
  - parental myopia and the development of child myopia are linked,
  - myopic parents are more likely to leave a light on in their children's bedroom
  - Zadnik K, Jones LA, Irvin BC, et al. (March 2000). "Myopia and ambient night-time lighting". Nature 404 (6774): 143–4

Correlational Studies

- In a recent study, neural correlates of intelligence were investigated.
- The research correlated changes in bold activity in two ROIs with Verbal IQ levels and found high correlations in BA9 and BA45/46



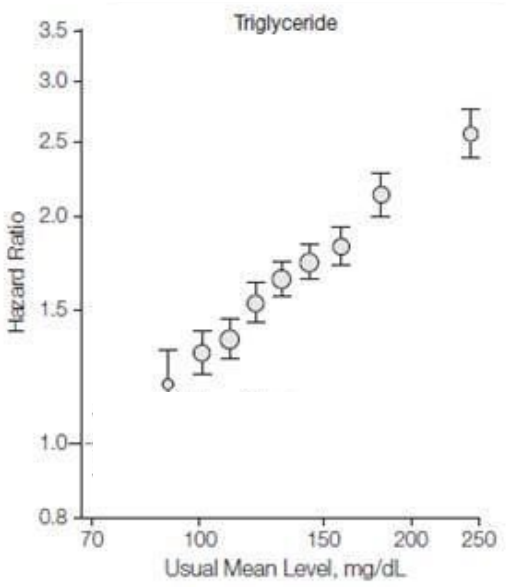




Geake and Hansen, Neuroimage, 2005

Correlation: determining causation

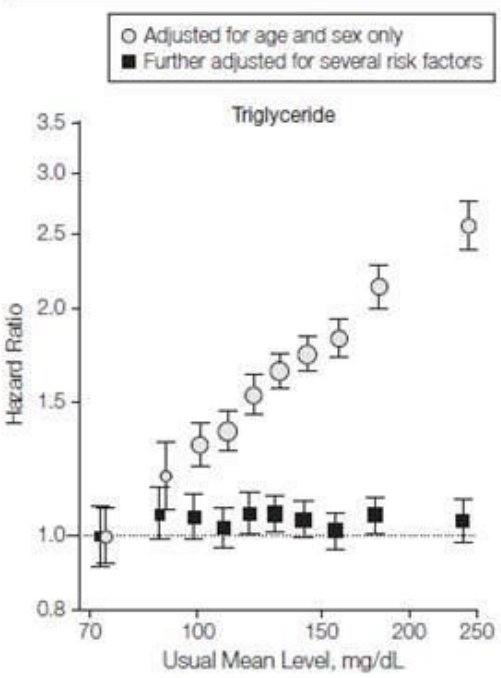
- Cardiovascular events are the leading cause of death in the US
- US people enjoy a very fatty diet – hence people started to look at analyzing markers in the blood and correlation it to prevalence of heart diseases



Getting the levels of triglycerides down would reduce your risk of heart diseases dramatically!

Correlation: determining causation

- Cardiovascular events are the leading cause of death in the US
- US people enjoy a very fatty diet – hence people started to look at analyzing markers in the blood and correlation it to prevalence of heart diseases

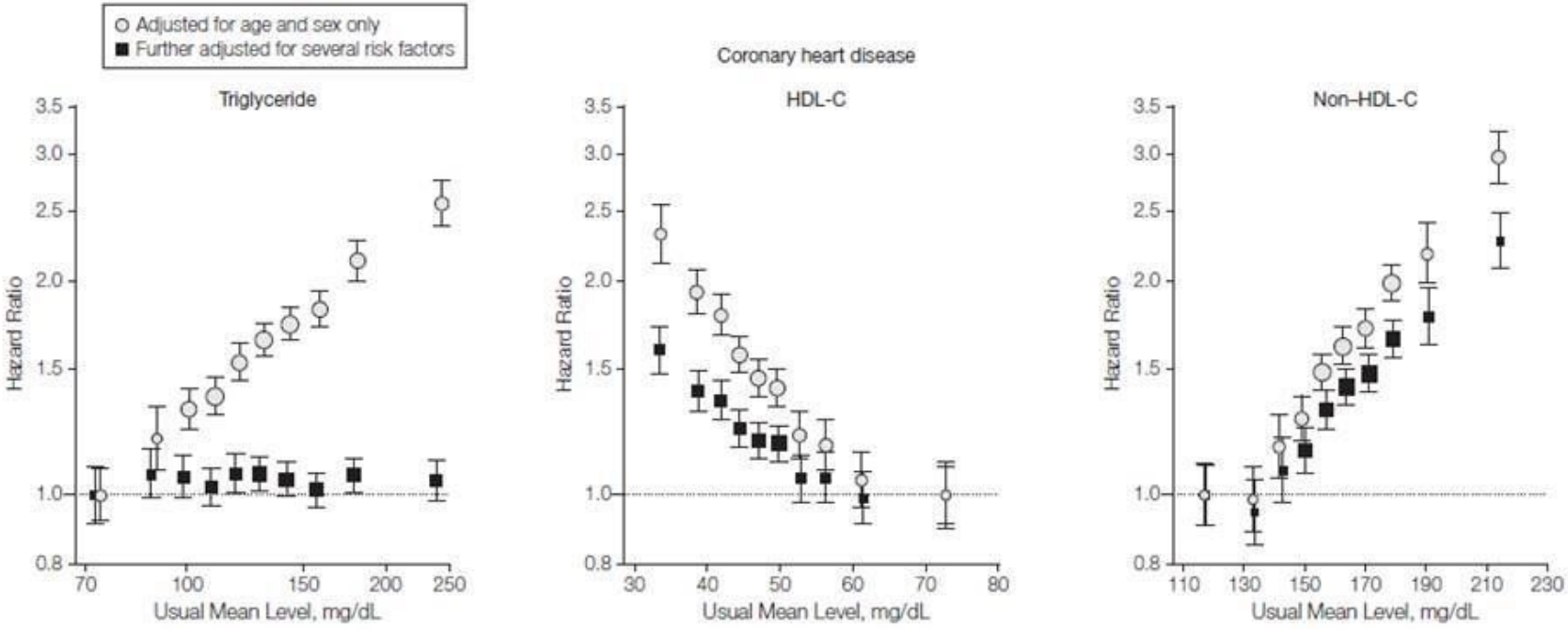


Or maybe not!

Correlation: determining causation

- Cardiovascular events are the leading cause of death in the US
- US people enjoy a very fatty diet – hence people started to look at analyzing markers in the blood and correlation it to prevalence of heart diseases

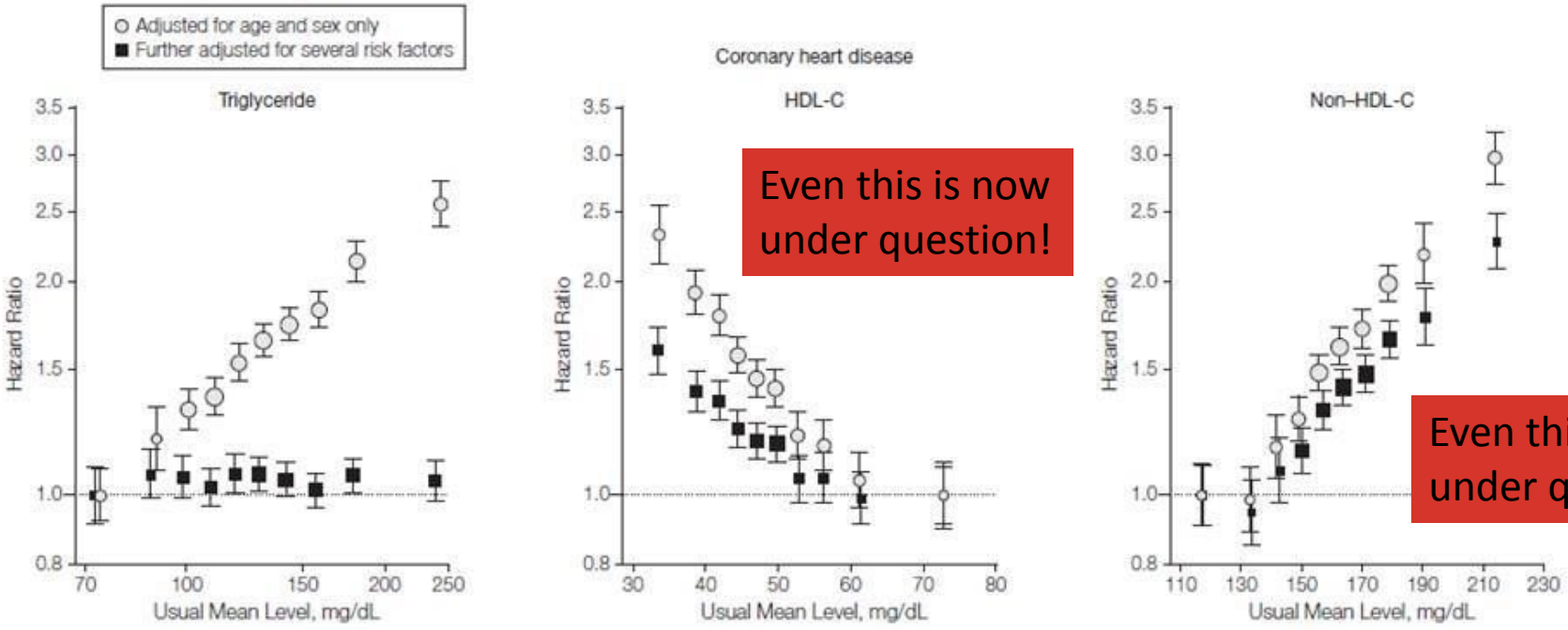
Figure 1. Hazard Ratios for Coronary Heart Disease or Ischemic Stroke Across Quantiles of Usual Triglyceride, HDL-C, and Non–HDL-C Levels



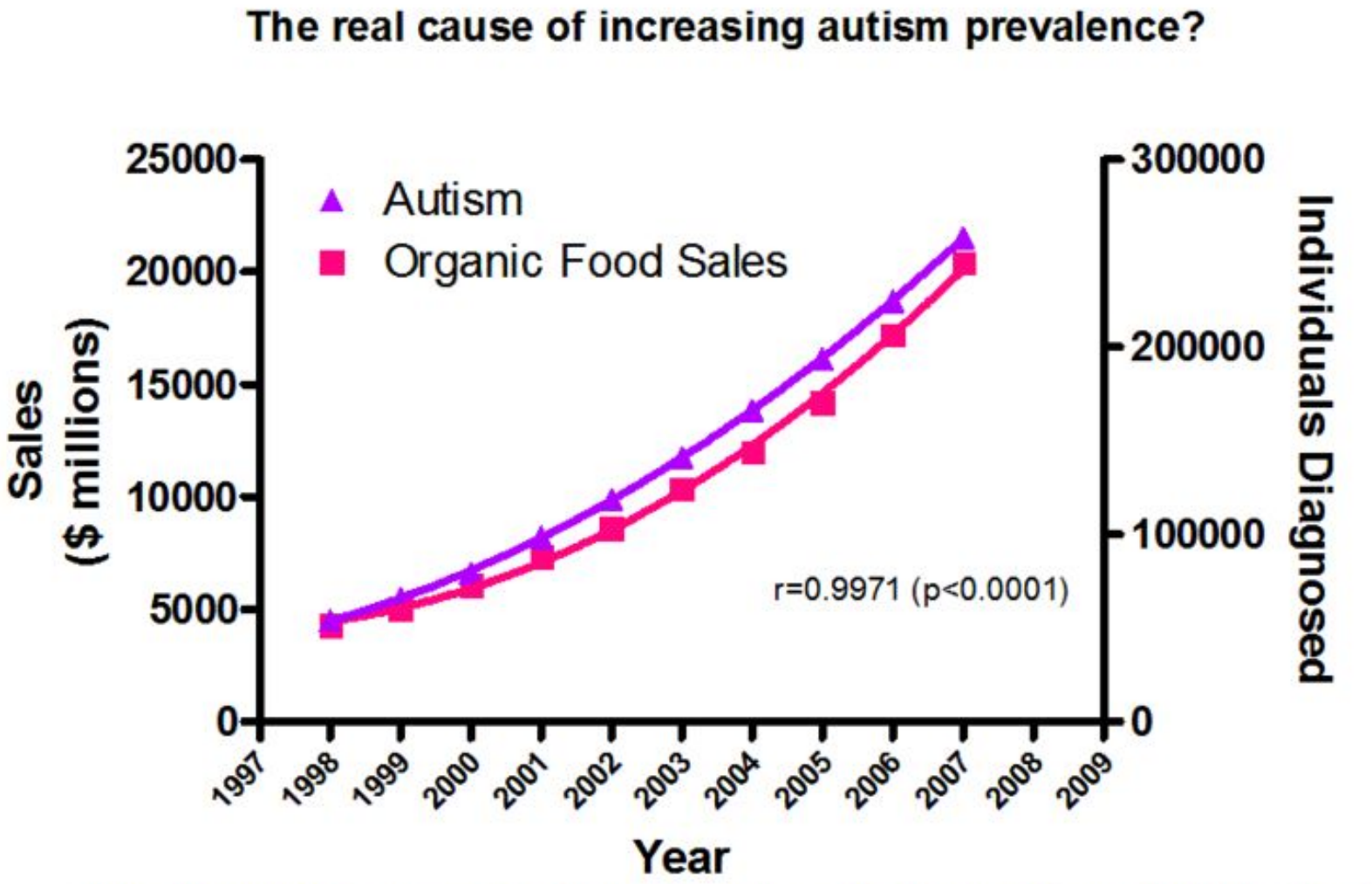
Correlation: determining causation

- Cardiovascular events are the leading cause of death in the US
- US people enjoy a very fatty diet – hence people started to look at analyzing markers in the blood and correlation it to prevalence of heart diseases

Figure 1. Hazard Ratios for Coronary Heart Disease or Ischemic Stroke Across Quantiles of Usual Triglyceride, HDL-C, and Non–HDL-C Levels

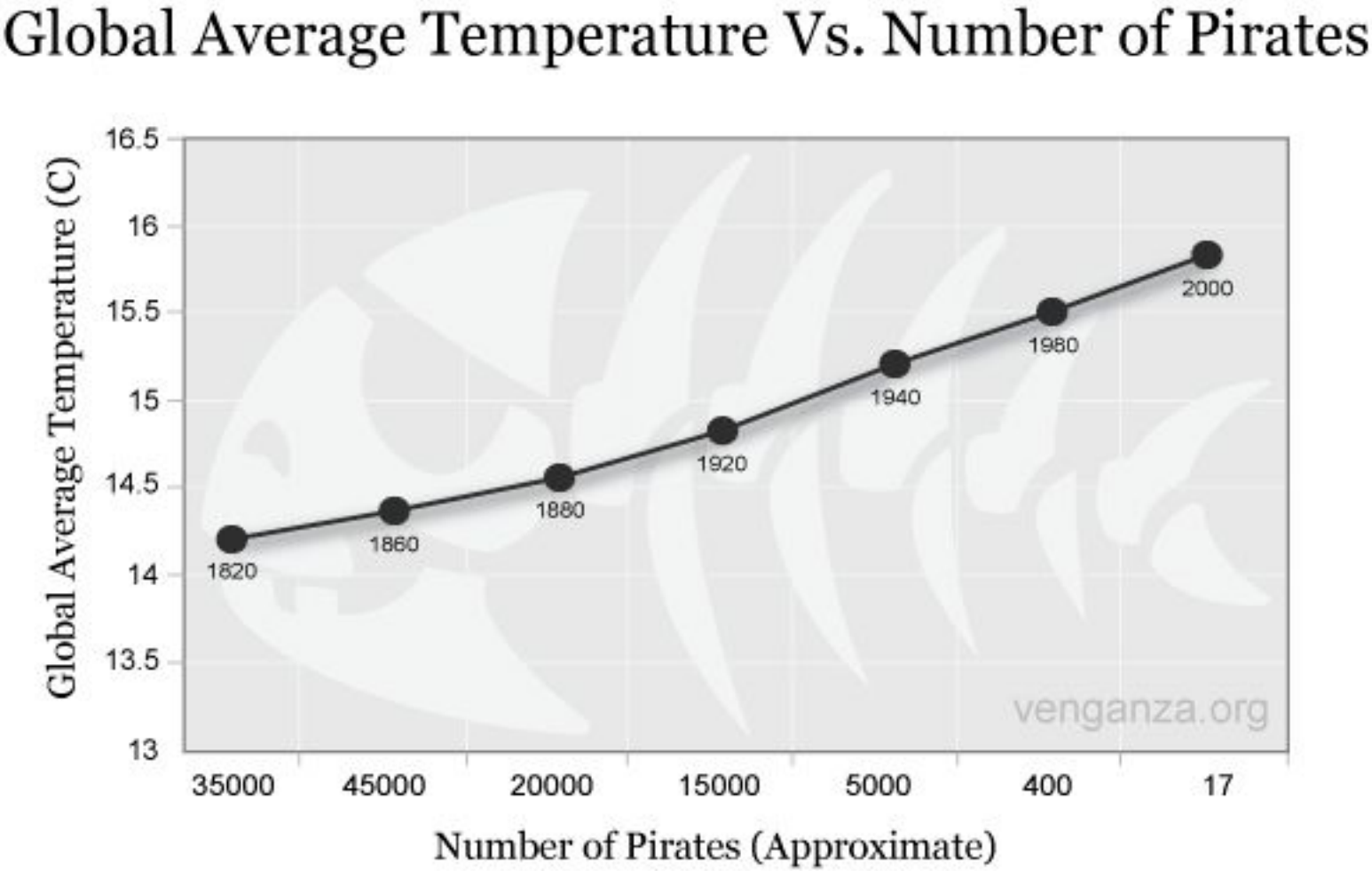


Humor



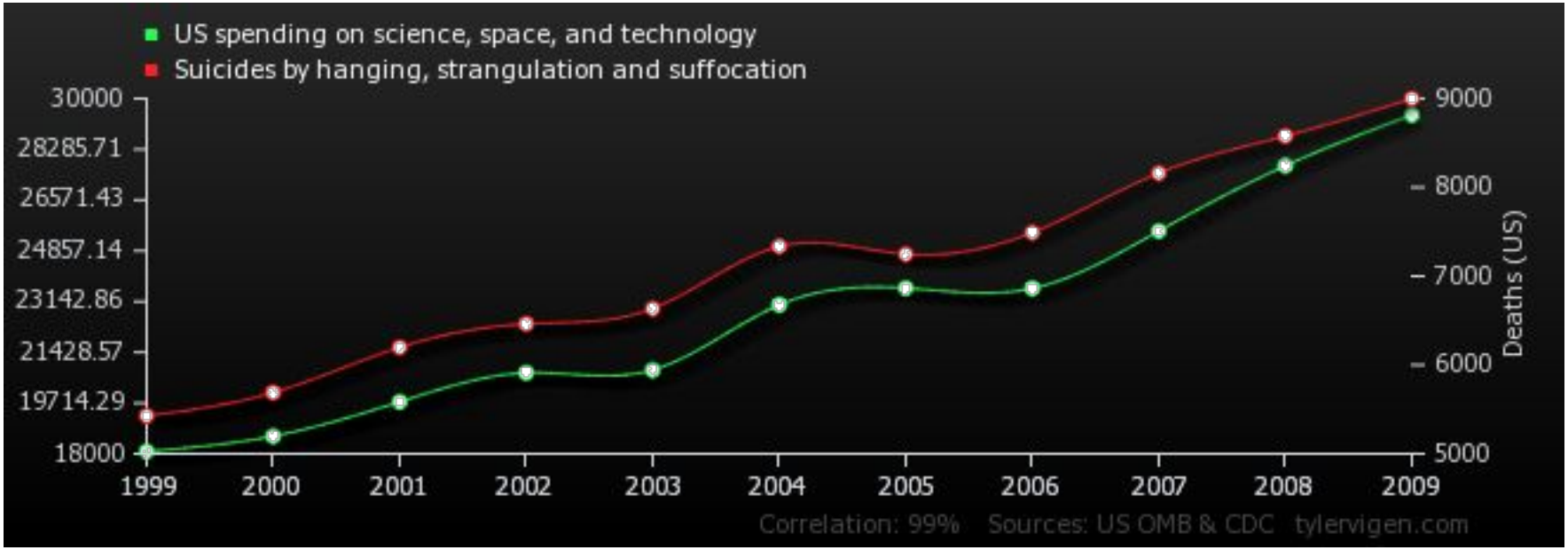
Sources: Organic Trade Association, 2011 Organic Industry Survey; U.S. Department of Education, Office of Special Education Programs, Data Analysis System (DANS), OMB# 1820-0043: "Children with Disabilities Receiving Special Education Under Part B of the Individuals with Disabilities Education Act

Humor


Global Average Temperature Vs. Number of Pirates
Global Average Temperature (C)
16.5
16
15.5
15
14.5
14
13.5
13
1820
1860
1880
1920
1940
1980
2000
venganza.org
35000
45000
20000
15000
5000
400
17
Number of Pirates (Approximate)

More spurious correlations

- http://tylervirgen.com



Illusory Correlation

CognitiveSystems

- A presumed relationship between two variables that, in fact, does not exist
- Often related to “confirmation bias”, that is, the tendency to notice only things that confirm our conceptions
- Redelmeier and Tversky (1996) probed 18 arthritis patients over 15 months, while taking comprehensive meteorological data.
  - Virtually all of the patients were certain that their condition was correlated with the weather. In fact the actual correlation was close to zero
  - confirmed in follow-up study in 2003 with n=154 patients!

Dangers of correlations in science

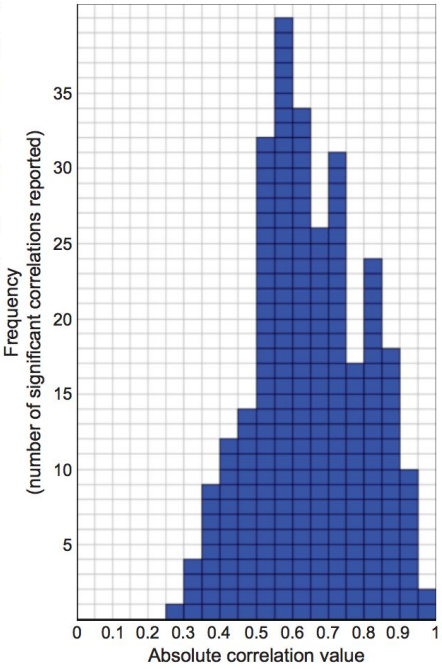
- A paper by Vul et al. describes a problem in fMRI analysis
- They found sometimes very high correlations in the studies th were reported
- Papers in social and cognitive neuroscience especially affected

Puzzlingly High Correlations in fMRI Studies of Emotion, Personality, and Social Cognition[1]

Edward Vul,[1] Christine Harris,[2] Piotr Winkielman,[2] & Harold Pashler[2]

[1]*Massachussetts Institute of Technology and* [2]*University of California, San Diego*

1. Eisenberger, Lieberman, and Williams (2003), writing in *Science*, described a game they created to expose individuals to social rejection in the laboratory. The authors measured the brain activity in 13 individuals while the actual rejection took place, and they later obtained a self-report measure of how much distress the subject had experienced. Distress was correlated at $r = .88$ with activity in the anterior cingulate cortex (ACC).



Dangers of correlations in science

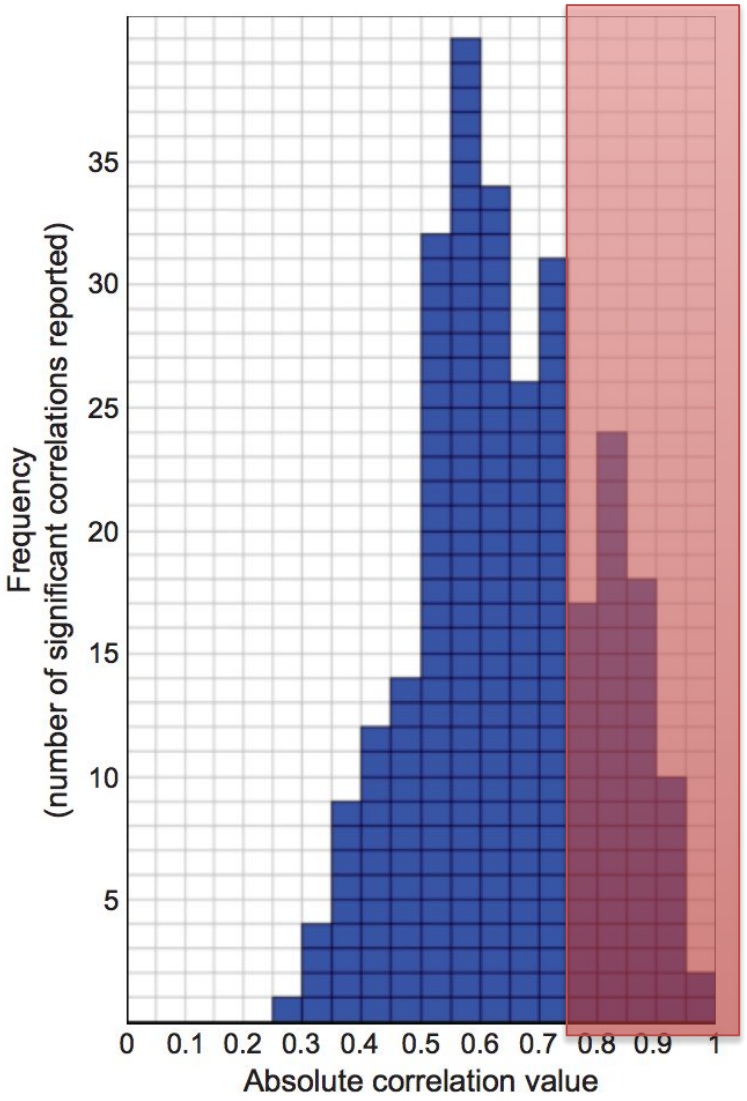
$$r_{X,Y} = \frac{\sigma_{XY}}{\sqrt{\sigma_X^2}\sqrt{\sigma_Y^2}}$$

$$\Rightarrow r_{ObsX,ObsY} = r_{X,Y}\sqrt{reliability(X)\cdot\ reliability(Y)}$$

- It turns out that the measured correlation of two variables is bounded by the reliability of each variable:
  - Real-world measurements will be corrupted by (independent) noise, thus the standard deviations of the measured distributions will be increased by the additional noise (with a magnitude assessed by the measure's reliability).
  - This will make the measured correlation lower than the true underlying correlation by a factor equal to the geometric mean of reliabilities.

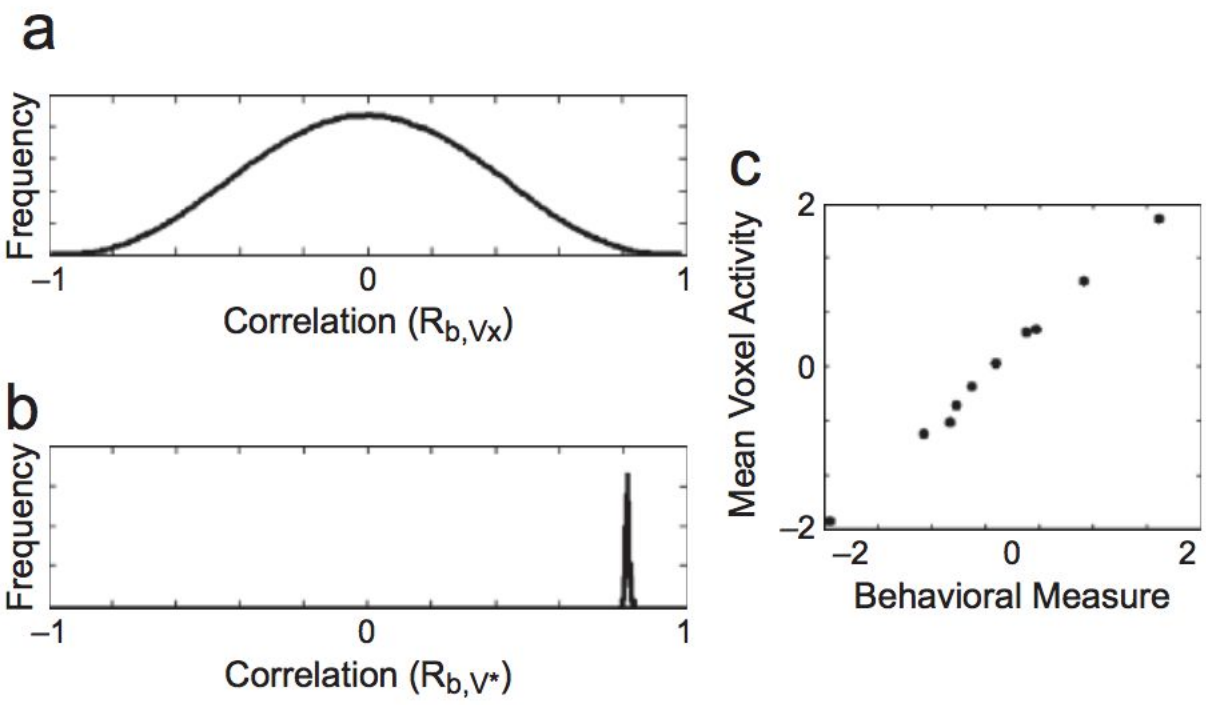
Dangers of correlations in science

- The puzzle:
  - reliabilities of personality scales are around r=.8 and the reliability of the BOLD signal is around r=.7
  - therefore the maximum observed correlations should be sqrt(.8 * .7) = .74!
- Why did the high reported correlations occur?



Dangers of correlations in science

- Double-dipping!
- You first simulate a no effect – model with many experiments (a)
- You then select only those voxels that pass a threshold p<.01, then correlations become very high! (b) and you get a beautiful scater plot (c)



Correlation Caveats

CognitiveSystems

- Correlation can be heavily affected by outliers – especially for small sample sizes
  - Always plot your data!

- r = 0 means that there is no **linear** association between your variables - they could still be otherwise associated
  - Always plot your data!

- Correlation does not imply causation
  - Be careful in interpreting your data!

THE SCIENCE NEWS CYCLE

Start Here

Your Research

Conclusion: **A is correlated with B (ρ=0.56)**, given C, assuming D and under E conditions.

...is translated by...

UNIVERSITY PR OFFICE
(YES, YOU HAVE ONE)

FOR IMMEDIATE RELEASE: SCIENTISTS FIND POTENTIAL LINK BETWEEN A AND B (UNDER CERTAIN CONDITIONS).

...which is then picked up by...

NEWS WIRE ORGANIZATIONS

A CAUSES B, SAY SCIENTISTS.

...who are read by ...

THE INTERNETS

Scientists out to kill us again.
POSTED BY RANDOM DUDE
Comments (377)
OMG! i kneeew it!!!
WTH???????

...then noticed by...

We saw it on a Blog!
A causes B all the time
What will this mean for Obama?
BREAKING NEWS BREAKING NEWS BREA

...and caught on ...

CNC Cable NEWS

...eventually making it to...

LOCAL EYEWITLESS NEWS

WHAT YOU DON'T KNOW ABOUT "A"... CAN KILL YOU! MORE AT 11...

A-KILLER AMONG US??
LOCAL EYEWITLESS NEWS

YOUR GRANDMA

I'M WEARING THIS TO WARD OFF "A"

WWW.PHDCOMICS.COM

Linear least-squares analysis
Matrix methods

Prof. Christian Wallraven

wallraven@korea.ac.kr

Introduction

CognitiveSystems

- Previously, we had fitted data with a straight line.
- In general, however, we would like to fit data with some kind of known model.
- If it is possible to write the model such that all **unknown parameters are linear**, we can find the solution using matrix-methods.
  - Note: the equations need to be linear in the parameters, but can, of course, be non-linear in the data!

Linear least-squares example

- There are 3 mountains *u,y,z* that from one site have been measured as 2474m, 3882m, and 4834m. But from *u*, *y* looks 1422m taller and the *z* looks 2354m taller, and from *y*, *z* looks 950m taller.
- This is equivalent to a linear system of equations
- Because there is more data than unknowns, we can only find an approximate solution

$$A\vec{x} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} u \\ y \\ z \end{pmatrix} \approx \begin{pmatrix} 2474 \\ 3882 \\ 4834 \\ 1422 \\ 2354 \\ 950 \end{pmatrix}$$

$$\min \left\| A\vec{x} - \vec{b} \right\|^2$$

http://commons.wikimedia.org/wiki/File:Eiger_M%C3%B6nch_Jungfrau.JPG

Normal equations

- We want to find: $\min \left\| A\vec{x} - \vec{b} \right\|^2$
- Again, from calculus:

$$\nabla f(x_1,\ldots,x_n) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right) = \vec{0}$$

$$\begin{aligned} \nabla \left\| A\vec{x} - \vec{b} \right\|^2 &= \nabla((A\vec{x} - \vec{b})^T (A\vec{x} - \vec{b})) \\ &= \nabla(\vec{x}^T A^T A\vec{x} - 2\vec{x}^T A^T \vec{b} + \vec{b}^T \vec{b}) \\ &= 2A^T A\vec{x} - 2A^T \vec{b} \\ &= \vec{0} \end{aligned}$$

$$\Leftrightarrow \vec{x}_{opt} = (A^T A)^{-1} A^T \vec{b}$$

- This assumes that the inverse $(A^TA)^{-1}$ exists

Normal equations

- Simpler: Multiply by A^T to get the **normal equations**:

  $\mathbf{A^T A x = A^T b}$

- The matrix $A^T A$ is symmetric .
- However, sometimes $A^T A$ can be *nearly singular*

$$A = \begin{pmatrix} 1 & 1 \\ \varepsilon & 0 \\ 0 & \varepsilon \end{pmatrix}, \quad \varepsilon = \text{smallest machine value}$$

$$\Rightarrow A^T A = \begin{pmatrix} 1+\varepsilon^2 & 1 \\ 1 & 1+\varepsilon^2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \text{ at machine precision}$$

Normal equations

- For the mountain example:

$$A^T A\vec{x} = \begin{pmatrix} 3 & -1 & 1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{pmatrix} \begin{pmatrix} u \\ y \\ z \end{pmatrix} = \begin{pmatrix} -1302 \\ 4354 \\ 8138 \end{pmatrix} = A^T \vec{b}$$

$$\Rightarrow u = 2472, y = 3886, z = 4832$$

- The additional information shifted the original measurements by -2m,4m,-2m

http://commons.wikimedia.org/wiki/File:Eiger_M%C3%B6nch_Jungfrau.JPG

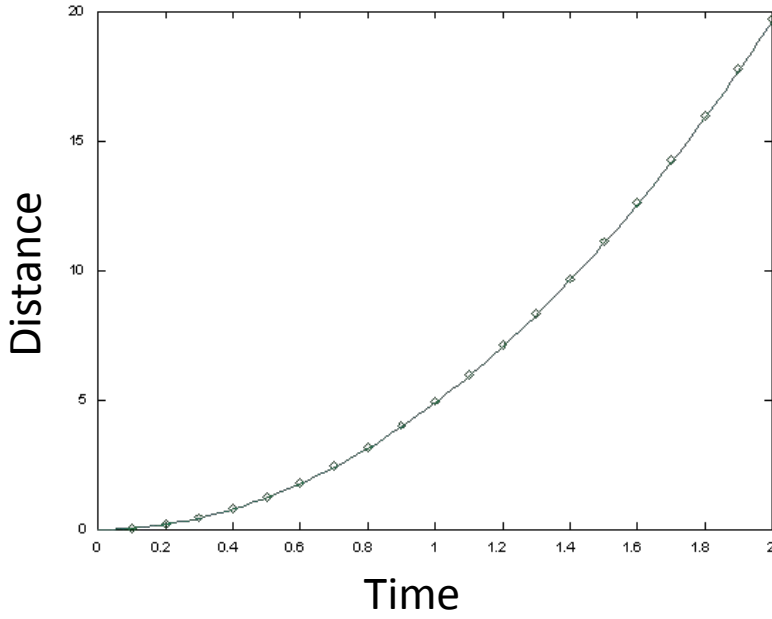
Fitting “gravitation”

- We want to determine the gravitational constant g from a dropped ball
- From physics, we know that:
- This is equivalent to:

$$d = d_0 + v_0 t + \frac{1}{2} g t^2$$

$$d = c_1 + c_2 t + c_3 t^2$$

- Goal:
  - given n measured distances d_i, derive c_1, c_2, c_3



http://people.rit.edu/andpph/photofile-c/a-stroboscopic-motion/drop-2balls-strobe-7994A.jpg

Derivation in least squares sense

$$Error = \epsilon(c_1, c_2, c_3) = \sum_{i=1}^{n} (c_1 + c_2 t_i + c_3 t_i^2 - d_i)^2$$

Error term in least squares sense

$$\frac{\partial \epsilon}{\partial c_1} = \sum_{i=1}^{n} 2 (c_1 + c_2 t_i + c_3 t_i^2 - d_i)$$

setting first derivative to zero

$$= 2 \left[n c_1 + c_2 \sum_{i=1}^{n} t_i + c_3 \sum_{i=1}^{n} t_i^2 - \sum_{i=1}^{n} d_i \right] = 0$$

$$n c_1 + \left[\sum_{i=1}^{n} t_i \right] c_2 + \left[\sum_{i=1}^{n} t_i^2 \right] c_3 = \sum_{i=1}^{n} d_i$$

$$\left[\sum_{i=1}^{n} t_i \right] c_1 + \left[\sum_{i=1}^{n} t_i^2 \right] c_2 + \left[\sum_{i=1}^{n} t_i^3 \right] c_3 = \sum_{i=1}^{n} d_i t_i$$

setting second derivative to zero

$$\left[\sum_{i=1}^{n} t_i^2 \right] c_1 + \left[\sum_{i=1}^{n} t_i^3 \right] c_2 + \left[\sum_{i=1}^{n} t_i^4 \right] c_3 = \sum_{i=1}^{n} d_i t_i^2$$

setting third derivative to zero

Full linear system

$$Error = \epsilon(c_1, c_2, c_3) = \sum_{i=1}^{n} (c_1 + c_2 t_i + c_3 t_i^2 - d_i)^2$$

Error term in least squares sense

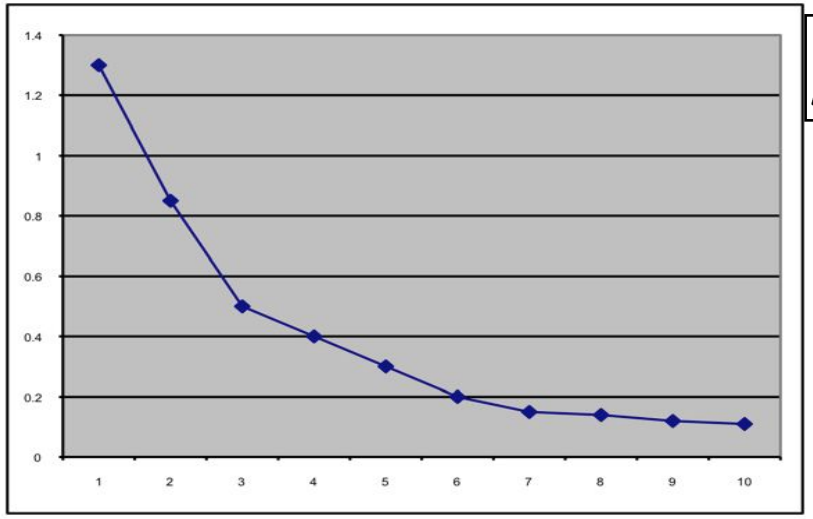
$$\begin{pmatrix} n & \sum_{i=1}^{n} t_i & \sum_{i=1}^{n} t_i^2 \\ \sum_{i=1}^{n} t_i & \sum_{i=1}^{n} t_i^2 & \sum_{i=1}^{n} t_i^3 \\ \sum_{i=1}^{n} t_i^2 & \sum_{i=1}^{n} t_i^3 & \sum_{i=1}^{n} t_i^4 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} d_i \\ \sum_{i=1}^{n} d_i t_i \\ \sum_{i=1}^{n} d_i t_i^2 \end{pmatrix}$$

Final linear system to solve

Note: this system has a unique solution!

Exponential example

- Assume you are given the following data:



t	1	2	3	4	5	6	7	8	9	10
b	1.3	0.85	0.5	0.4	0.3	0.2	0.15	0.14	0.12	0.11

- Which you think fits an exponential model with **known** exponent

$$b = u + we^{-0.5t} \quad \Rightarrow \quad A\begin{pmatrix} u \\ w \end{pmatrix} \approx b \text{ with } A = \begin{pmatrix} 1 & e^{-0.5} \\ 1 & e^{-1} \\ 1 & e^{-1.5} \\ \vdots & \vdots \\ 1 & e^{-5} \end{pmatrix}$$

Fitting polynomials

- The general form of a polynomial of degree *n* is:

$$f(x) = a_0 + a_1 x^1 + a_2 x^2 + \ldots + a_n x^n = \sum_{i=0}^{n} a_i x^i$$

- In order to fit a polynomial of degree *n* to *k* data-points *(x,y)* in a least-squares sense (*k*>>*n*), we create the so-called Vandermonde matrix and solve:

$$V = \begin{pmatrix} 1 & x_1 & \cdots & x_1^n \\ 1 & x_2 & & x_2^n \\ \vdots & \vdots & & \vdots \\ 1 & x_k & \cdots & x_k^n \end{pmatrix} \qquad V \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ \vdots \\ y_k \end{pmatrix}$$

Norm optimization

$L_1 - L_2$ norm

Prof. Christian Wallraven

wallraven@korea.ac.kr

Norm optimization

- As we have seen, fitting a line to points relies on defining a suitable error-function

$$E(w,b) = \sum_{i=1}^{n} \left\| y_i - (wx_i + b) \right\|$$

- where ||..|| is a suitable norm and we seek to find w, b, such that we minimize E(w,b)
- For ||..|| = L_2-norm, we are able to find a closed, algebraic solution
- What about other norms?

Recap – the Norm

Basic idea: Notion of length/distance between vector.

Definition (Norm): Let $\ell : X \mapsto \mathcal{R}^{\geq 0}$ be a mapping from a vector space to non-negative reals. The function ℓ is a norm iff

1. $\ell(\vec{x}) = 0$ iff 0,
2. $\ell(a\vec{x}) = |a|\ell(\vec{x})$
3. $\ell(\vec{x} + \vec{y}) \leq \ell(\vec{x}) + \ell(\vec{y})$ (**Triangle Inequality**).

"Length" of $\vec{x}$: $\ell(\vec{x})$.

"Distance" between $\vec{x}, \vec{y}$ is $\ell(\vec{y} - \vec{x})(= \ell(\vec{x} - \vec{y}))$.

L_2 – norm (Euclidean norm)

Distance between (x_1, y_1) to (x_2, y_2) is $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.
In general, $||\vec{x}||_2 = \sqrt{\vec{x}^T . \vec{x}}$.
Distance between $\vec{x}, \vec{y}$ is given by $||\vec{x} - \vec{y}||_2$.

L_p – norm

For $p \geq 1$, we define L_p norm as

$$||\vec{x}||_p = (|x_1|^p + |x_2|^p + \ldots + |x_n|^p)^{\frac{1}{p}}.$$

The L_p norm distance between two vectors is: $||\vec{x} - \vec{y}||_p$

L_1 **norm:** $||\vec{x}||_1 = |x_1| + |x_2| + \ldots + |x_n|$.

L_∞ – norm

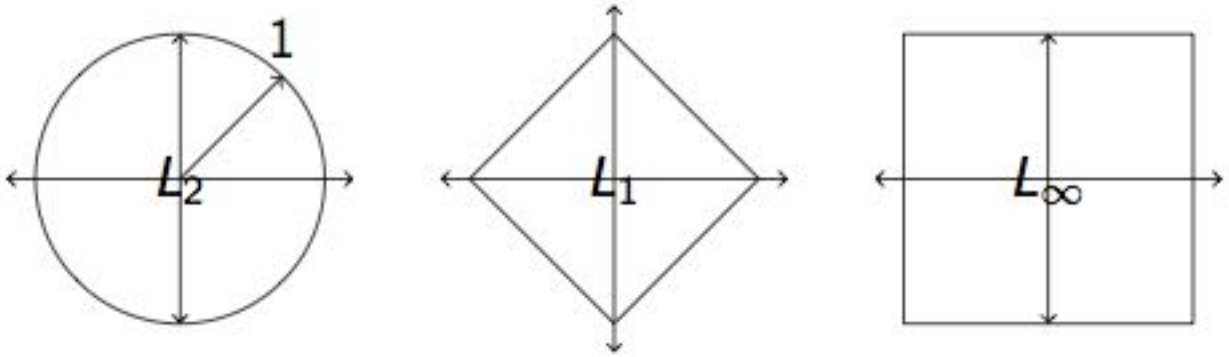
Obtained as the limit $lim_{p\to\infty}\ ||\vec{x}||_p$.

Definition: For vector $\vec{x}$, $||\vec{x}||_\infty$ is defined as

$$||\vec{x}||_\infty = \max(|x_1|, |x_2|, \ldots, |x_n|)\,.$$

Norm – Equal Distance Spheres

ϵ-Sphere: $\{\vec{x} \mid d(\vec{x}, 0) \leq \epsilon\}$ for norm d. The 1−spheres corresponding to the norms L_1, L_∞, L_2.



Note: Norm spheres are convex sets.

Norm minimization

General form:

$$\begin{aligned} \text{minimize} \quad & ||x||_p \\ Ax \quad & \leq \quad b \end{aligned}$$

Note-1: We always minimize the norm functions (in this course).

Note-2: Maximization of norm is generally a hard (non-convex) problem.

Unconstrained minimization

Lets first study the following problem:

$$\text{min. } ||Ax - b||_p \,.$$

1. For $p = 2$, this problem is called (unconstrained) least squares. We will solve this using calculus.
2. For $p = 1, \infty$, this problem is called $L_1 (L_\infty)$ least squares. We will reduce this problem to LP.
3. Applications: solving (noisy) linear systems, regularization, denoising, max. likelihood estimation, regression and so on.

Unconstrained least-squares

Decision Variables: $\vec{x} = (x_1, \ldots, x_n) \in \mathcal{R}^n$.
Objective function: $\min. ||A\vec{x} - \vec{b}||_2$.
Constraints: No explicit constraint.

Least-squares: Example

Example: $\min. ||(2x + 3y - 1, y)||_2$. f(x,y)!

Solution: We will equivalently minimize $(2x + 3y - 1)^2 + y^2$. Using fundamental theorem of calculus, we obtain the conditions:

$$\begin{aligned} \frac{\partial (2x+3y-1)^2+y^2}{\partial x} &= 0 \\ \frac{\partial (2x+3y-1)^2+y^2}{\partial y} &= 0 \end{aligned}$$

In other words, we obtain:

$$\begin{aligned} 4(2x + 3y - 1) &= 0 \\ 6(2x + 3y - 1) + 2y &= 0 \end{aligned}$$

The optima lies at is $y = 0, x = \frac{1}{2}$.
Verify minima by computing checking second order derivative (Hessian matrix).

Unconstrained Least-squares

CognitiveSystems

Problem: minimize $||A\vec{x} - \vec{b}||_2$.

Solution: We will use calculus minimum finding using partial derivatives.

Recall:

$$\nabla f(x_1, \ldots, x_n) = (\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f).$$

Criterion for optimal point for $f(\vec{x})$ is that $\nabla f = (0, \ldots, 0)$.

$$\begin{aligned}
\nabla(||A\vec{x} - \vec{b}||_2^2) &= \nabla((A\vec{x} - \vec{b})^{\mathrm{T}}(A\vec{x} - \vec{b})) \\
&= \nabla(\vec{x}^{\mathrm{T}} A^{\mathrm{T}} A\vec{x} - 2\vec{x}^{\mathrm{T}} A^{\mathrm{T}} \vec{b} + \vec{b}^{\mathrm{T}} \vec{b}) \\
&= 2A^{\mathrm{T}} A\vec{x} - 2A^{\mathrm{T}} \vec{b} \\
&= 0
\end{aligned}$$

Minima will occur at $\vec{x}^* = (A^{\mathrm{T}} A)^{-1} A^{\mathrm{T}} \vec{b}$ (assume A is "full rank").

Similar solution for L_p norm for even number p.

L_1 - minimization

Decision Variables: $\vec{x} = (x_1, \ldots, x_n) \in \mathcal{R}^n$.
Objective function: min.$||A\vec{x} - \vec{b}||_1$.
Constraints: No explicit constraint.
We will reduce this to a linear program.

L_1-minimization: Example

Example: $\min. ||(2x + 3y - 1, y)||_1 = |2x + 3y - 1| + |y|$.
Trick: Let $t_1 \geq 0, t_2 \geq 0$ be s.t.

$$\begin{aligned} |2x + 3y - 1| &\leq t_1 \\ |y| &\leq t_2 \end{aligned}$$

Consider the following problem:

$$\begin{aligned} \min. \quad & t_1 + t_2 \\ |2x + 3y - 1| &\leq t_1 \\ |y| &\leq t_2 \\ t_1, t_2 &\geq 0 \end{aligned}$$

Note: $|y| \leq t_2$ can be rewritten as $y \leq t_2 \ \wedge \ -y \leq t_2$.

L_1-minimization: Example as a Linear Program

CognitiveSystems

$$
\begin{array}{lrcl}
\text{min.} & t_1 + t_2 & & \\
& 2x + 3y - 1 & \leq & t_1 \\
& -2x - 3y + 1 & \leq & t_1 \\
& y & \leq & t_2 \\
& -y & \leq & t_2 \\
& t_1, t_2 & \geq & 0
\end{array}
$$

Solution: $x = \frac{1}{2}, y = 0.$

L_1-minimization

Problem: $\min. ||Ax - b||_1$.

Solution: Let A be $m \times n$ matrix. Add variables $t_1, \ldots, t_m$ correspoding to rows of m.

$$\left[\begin{array}{lrcl} \min. & \sum_{i=1}^{m} t_i & & \\ & |A_i\vec{x} - \vec{b}| & \leq & t_i \\ & t_1, \ldots, t_m & \geq & 0 \end{array}\right] \Rightarrow \left[\begin{array}{lrcl} \min. & \sum_{i=1}^{m} t_i & & \\ & A\vec{x} - \vec{b} & \leq & \vec{t} \\ & -A\vec{x} + \vec{b} & \leq & \vec{t} \\ & t_1, \ldots, t_m & \geq & 0 \end{array}\right]$$

We write it in the general form:

$$\begin{array}{lrcl} \max. & -\vec{1}^T\vec{t} & & \\ & A\vec{x} - \vec{t} & \leq & \vec{b} \\ & -A\vec{x} - \vec{t} & \leq & -\vec{b} \\ & t_1, \ldots, t_m & \geq & 0 \end{array}$$

Linear programming (LP)

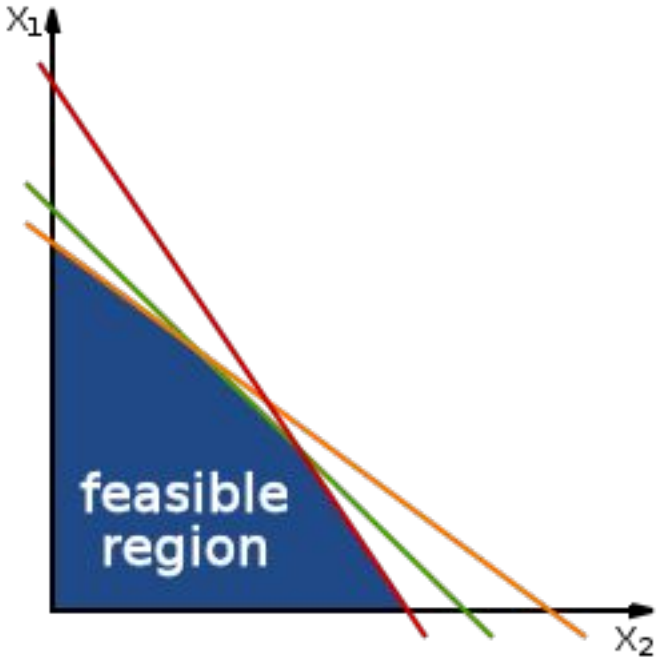
- A system of linear equations subject to equality and inequality constraints can be solved using LP
- The canonical (usual) form of a LP is as follows:

$$\begin{aligned} \text{maximize} \quad & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{b} \\ \text{and} \quad & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

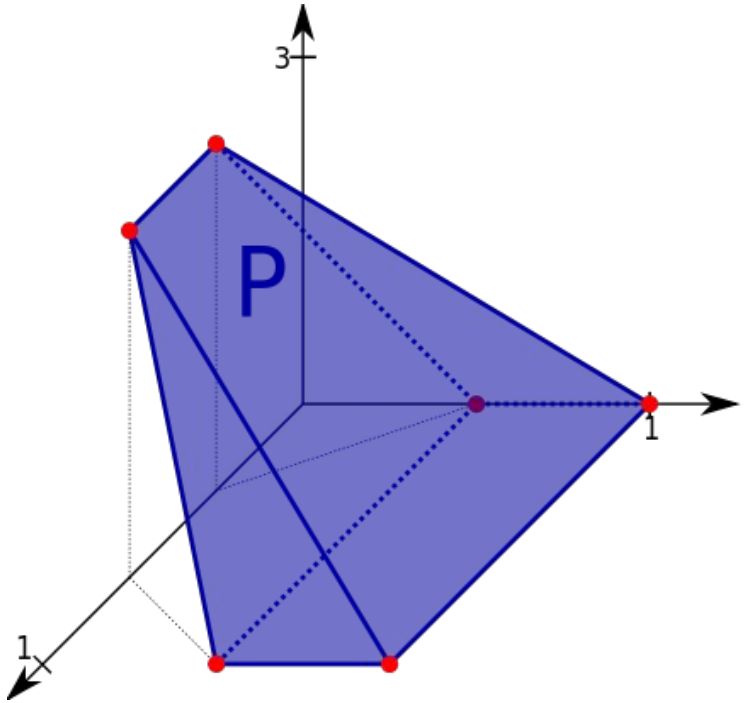
- with **x** the unknown vector of variables, **c**,**b** known vectors of coefficients, and A known matrix of coefficients
- $\mathbf{c}^{\mathrm{T}}\mathbf{x}$ is the objective function, and $A\mathbf{x} \leq \mathbf{b}$ are the constraints

Geometric interpretation

- Since each of the constraint equations is linear, the resulting search space (**the feasible region**) of optimal solutions consists of a polygon



feasible region for 2 variables



feasible region for 3 variables

L_1-minimization: Example as a Linear Program

- How do we solve this in general??

$$
\begin{array}{lrcl}
\text{min.} & t_1 + t_2 & & \\
& 2x + 3y - 1 & \leq & t_1 \\
& -2x - 3y + 1 & \leq & t_1 \\
& y & \leq & t_2 \\
& -y & \leq & t_2 \\
& t_1, t_2 & \geq & 0
\end{array}
\quad \square \quad
F = \begin{pmatrix} 2 & 3 \\ 0 & 1 \end{pmatrix}, \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}, \mathbf{s} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}
$$

$$
\begin{array}{ll}
\text{maximize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\
\text{subject to} & A\mathbf{x} \leq \mathbf{b} \\
\text{and} & \mathbf{x} \geq \mathbf{0}
\end{array}
\quad \square \quad
\max \mathbf{c}^T \mathbf{x} \Rightarrow \max -\mathbf{1}^T \mathbf{t}
$$

$$
\text{s.t. } A\mathbf{x} \leq \mathbf{b} \Rightarrow \text{s.t. } \begin{pmatrix} F & -I \\ -F & -I \\ 0 & -I \end{pmatrix} \begin{pmatrix} \mathbf{t} \\ \mathbf{t} \\ \mathbf{t} \end{pmatrix} \leq \begin{pmatrix} \mathbf{s} \\ -\mathbf{s} \\ \mathbf{0} \end{pmatrix}
$$

and $\mathbf{x} \geq \mathbf{0}$

Linear programming (LP)

- The problem for optimization consist in the inequalities – algorithmically, things become easier if equalities are used
- Trick: introduce "dummy" or "slack" variables
- Example: $A\mathbf{x} \leq \mathbf{b}$ with $x = (x_1, x_2)^T$

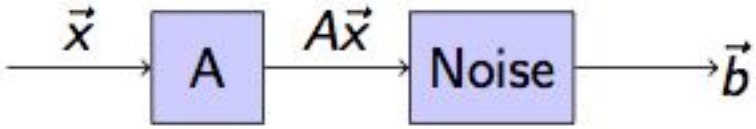
  $a_{11}x_1 + a_{12}x_2 \leq b_1 \quad \square \quad a_{11}x_1 + a_{12}x_2 + x_3 = b_1$
  $a_{21}x_1 + a_{22}x_2 \leq b_2 \quad \square \quad a_{21}x_1 + a_{22}x_2 + x_4 = b_2$
- The slack variables capture the "unused" quantity in the inequality!
- How to solve this?
  - Simplex algorithm – not covered here…

Estimation

Problem: We are trying to solve the linear equation

$$A\vec{x} = \vec{b}$$

1. The entries in $A, \vec{b}$ could have random errors (noisy measurements).
2. There are many more equations than variables.



Estimation

Example Problem: We are trying to measure a quantity x using noisy measurements:

$$
\begin{aligned}
2x &= 5.1 \\
3x &= 7.6 \\
4x &= 9.91 \\
5x &= 12.45 \\
6x &= 15.1
\end{aligned}
$$

Estimation

Example Problem: We are trying to measure a quantity x using noisy measurements:

$$\begin{aligned} 2x &= 5.1 \\ 3x &= 7.6 \\ 4x &= 9.91 \\ 5x &= 12.45 \\ 6x &= 15.1 \end{aligned}$$

Find x that nearly fits all the measurements., i,e,

$$\min. ||(2x - 5.1, 3x - 7.6, 4x - 9.91, 5x - 12.45, 6x - 15.1)||_p \, .$$

We can choose $p = 1, 2, \infty$ and see what answer we get.

Estimation

L_2 norm: Apply least squares for

$$A = \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix} \quad \vec{b} = \begin{pmatrix} -5.1 \\ -7.6 \\ -9.91 \\ -12.45 \\ -15.1 \end{pmatrix}$$

Solving for $x = \text{argmin}||Ax - \vec{b}||_2$, yields $x \sim 2.505$.
Residual error:

$$(0.09, 0.08, -.11, -0.08, -.06)\,.$$

Estimation

L_1 norm: Reduce to linear program using

$$A = \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}, \ \vec{b} = \begin{pmatrix} -5.1 \\ -7.6 \\ -9.91 \\ -12.45 \\ -15.1 \end{pmatrix}, .$$

Solution: $x = 2.49$.
Residual Error:

$$(-.12, -.13, -0.05, 0.0, .16).$$

L_∞ norm: Reduce to linear program.
Solution: $x = 2.501$
Residual Error:

$$(-.1, -.1, .1, .05, -.1)$$

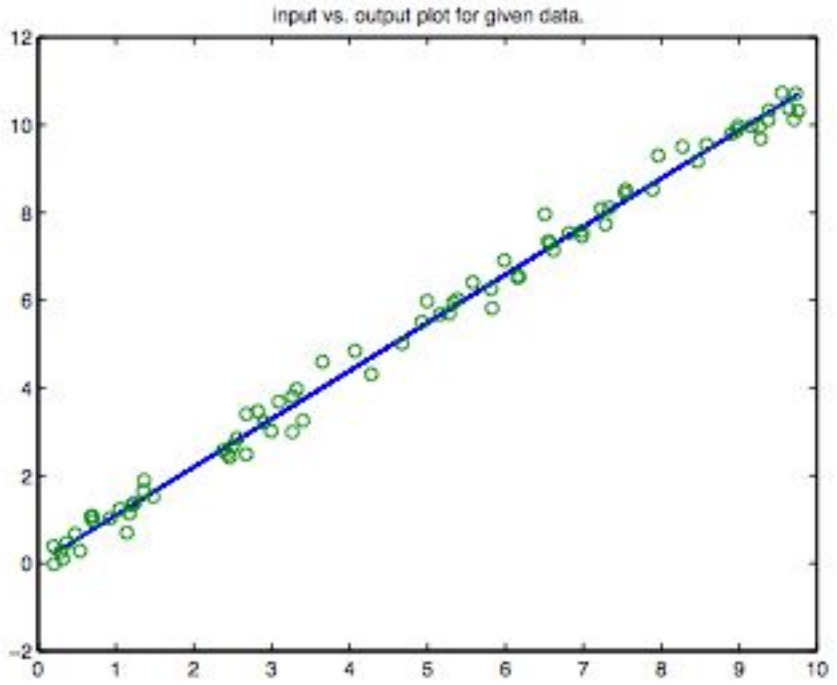
Linear Regression

Inputs: Data points $\vec{x}_i : (x_{i1}, \ldots, x_{in})$ and outcome y_i.

Goal: Find a function

$$f(\vec{x}) = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n + c_0$$

that best fits the data.



Linear Regression

Let X be the data matrix and $\vec{y}$ be corresponding outcome matrix.

Note: The i^{th} row X_i represents data point i
y_i is the corresponding outcome for X_i

If $\vec{c}^T \vec{x} + c_0$ is the best line fit, the error for i^{th} data point is:

$$\epsilon_i : (X_i \cdot \vec{c}) + c_0 - y_i .$$

We can write the overall error as:

$$\left\| [X \quad \vec{1}]\vec{c} - \vec{y} \right\|_p .$$

Linear Regression

Inputs: Data X, outcome $\vec{y}$.
Decision Vars: $c_0, c_1, \ldots, c_n$ the coefficients of the straight line.
Solution:minimize $\left\| [X \quad \vec{1}]\vec{c} - \vec{y} \right\|_p$.

Note: Instead of the norm, we could use any penalty function.

Ordinary Regression: min $\left\| [X \quad \vec{1}]\vec{c} - \vec{y} \right\|_p^p$.

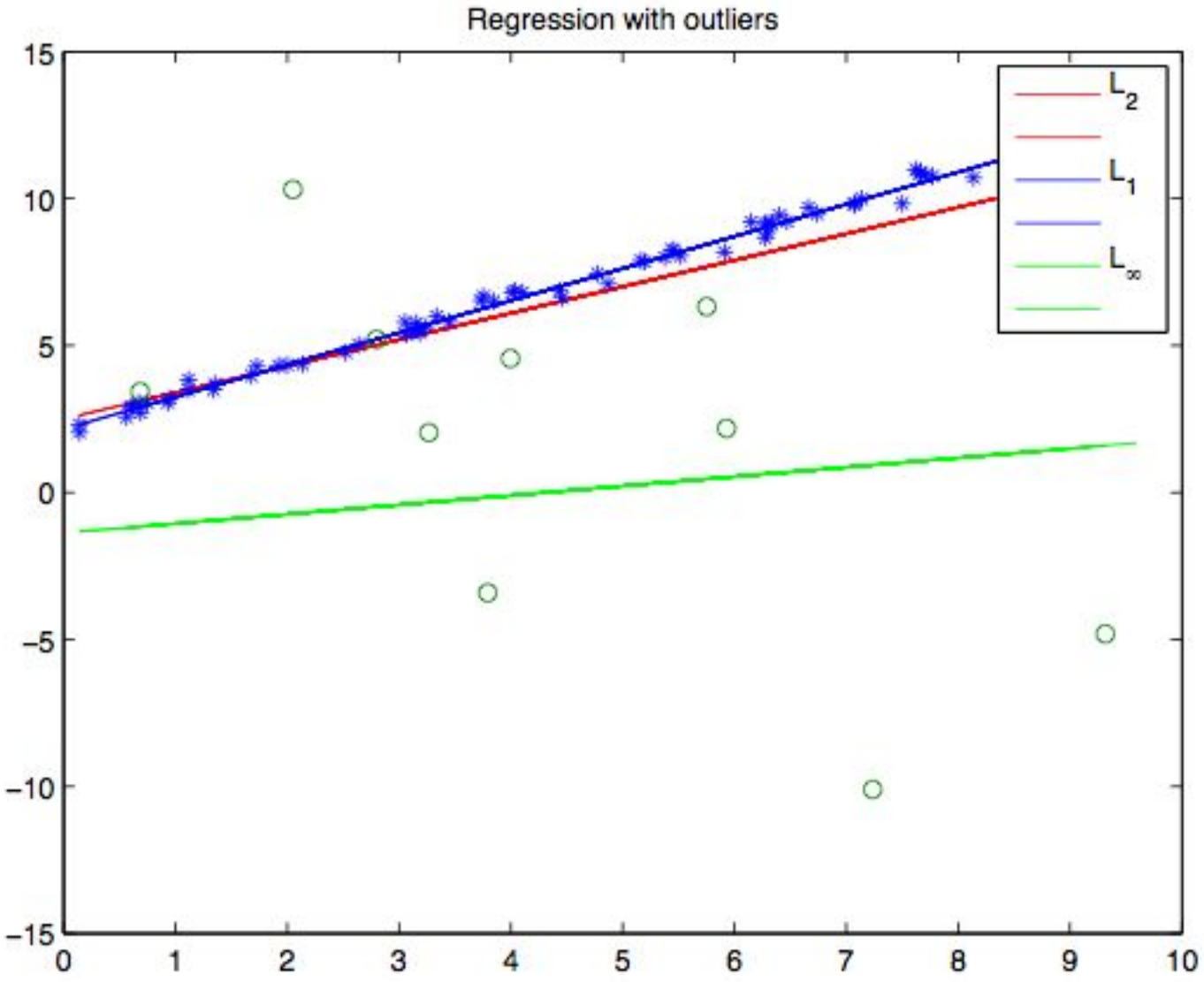
Linear Regression

Experiment: Create noisy data set with outliers.

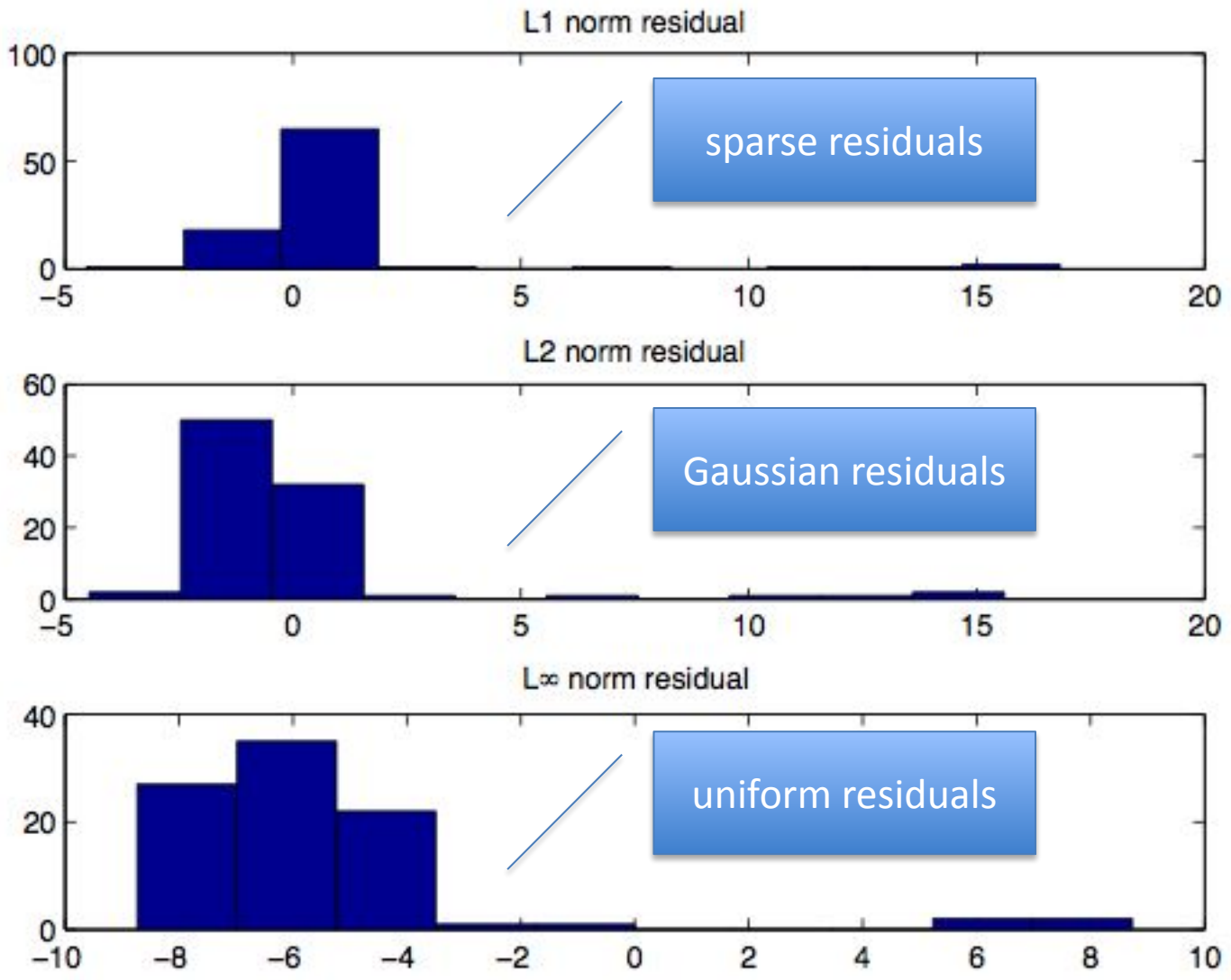
80 regular data points with low noise.
5 outlier datapoints with large noise.

Goal: Compare effect of L_1, L_2, L_∞ norm regressions.

Linear Regression



Linear Regression



Non-linear optimization

Prof. Christian Wallraven

wallraven@korea.ac.kr

Applications

- Non-linear least-squares optimization is one of the most important optimization methods

- Inverse kinematics
- Physically-based animation
- Data-driven motion synthesis
- Many other problems in graphics, vision, machine learning, robotics, etc.

Problem Definition

- Most optimization problem can be formulated as a nonlinear least squares problem

$$x^* = \arg\min_x \frac{1}{2}\sum_{i=1}^{m}(f_i(x))^2$$

$$x^* = \arg\min_x \frac{1}{2} f(x)^T f(x)$$

Where $f_i : R^n \to R$, i=1,…,m are given functions, and m>=n

Data Fitting

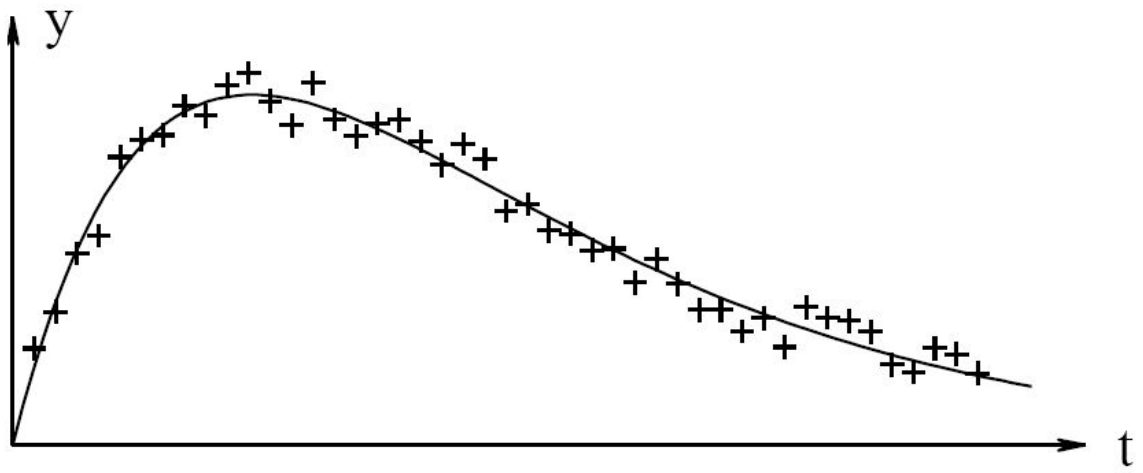


Figure 1.1. *Data points* $\{(t_i, y_i)\}$ *(marked by +) and model* $M(\mathbf{x}, t)$ *(marked by full line.)*

$$M(\mathbf{x}, t) = x_3 e^{x_1 t} + x_4 e^{x_2 t}$$

Data Fitting

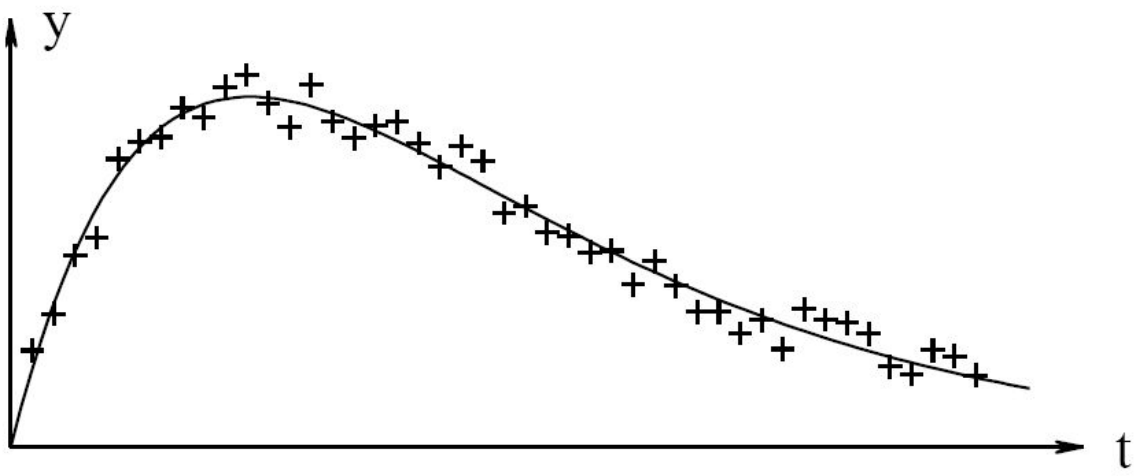


Figure 1.1. *Data points* $\{(t_i, y_i)\}$ *(marked by +) and model* $M(\mathbf{x}, t)$ *(marked by full line.)*

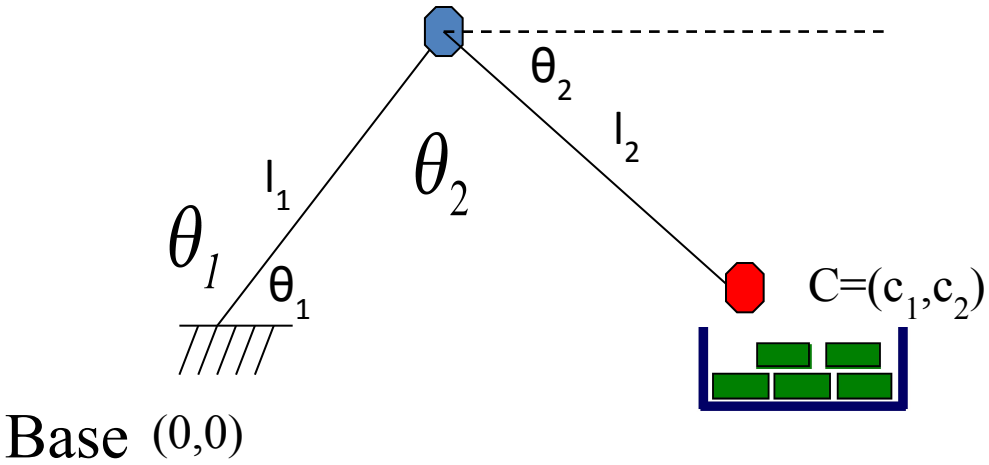
For any choice of $\mathbf{x}$ we can compute the *residuals*

$$\begin{aligned} f_i(\mathbf{x}) &= y_i - M(\mathbf{x}, t_i) \\ &= y_i - x_3 e^{x_1 t_i} - x_4 e^{x_2 t_i}, \quad i = 1, \ldots, m \end{aligned}$$

Inverse Kinematics

- Find the joint angles θ that minimizes the distance between the character position and user specified position

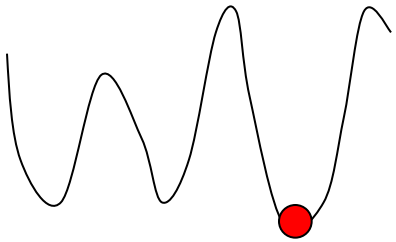
$$\arg\min_{\theta_1,\theta_2}(l_1\cos\theta_1 + l_2\cos(\theta_2) - c_1)^2 + (l_1\sin\theta_1 - l_2\sin(\theta_2) - c_2)^2$$



Global Minimum vs. Local Minimum

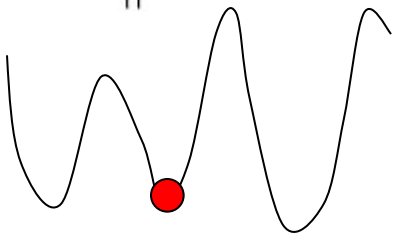
- Finding the global minimum for nonlinear functions is very hard

$$\text{Given } F: \mathbb{R}^n \mapsto \mathbb{R}. \text{ Find}$$
$$\mathbf{x}^+ = \text{argmin}_{\mathbf{x}}\{F(\mathbf{x})\}$$



- Finding the local minimum is much easier

$$\text{Given } F: \mathbb{R}^n \mapsto \mathbb{R}. \text{ Find } \mathbf{x}^* \text{ so that}$$
$$F(\mathbf{x}^*) \leq F(\mathbf{x}) \quad \text{for} \quad \|\mathbf{x} - \mathbf{x}^*\| < \delta$$



Assumptions

- The cost function *F* is differentiable and so smooth that the following Taylor expansion is valid,

$$F(\mathbf{x}+\mathbf{h}) = F(\mathbf{x}) + \mathbf{h}^\top \mathbf{g} + \tfrac{1}{2}\mathbf{h}^\top \mathbf{H}\,\mathbf{h} + O(\|\mathbf{h}\|^3)$$

where $\mathbf{g}$ is the *gradient*,

$$\mathbf{g} \equiv \mathbf{F}'(\mathbf{x}) = \begin{bmatrix} \frac{\partial F}{\partial x_1}(\mathbf{x}) \\ \vdots \\ \frac{\partial F}{\partial x_n}(\mathbf{x}) \end{bmatrix}$$

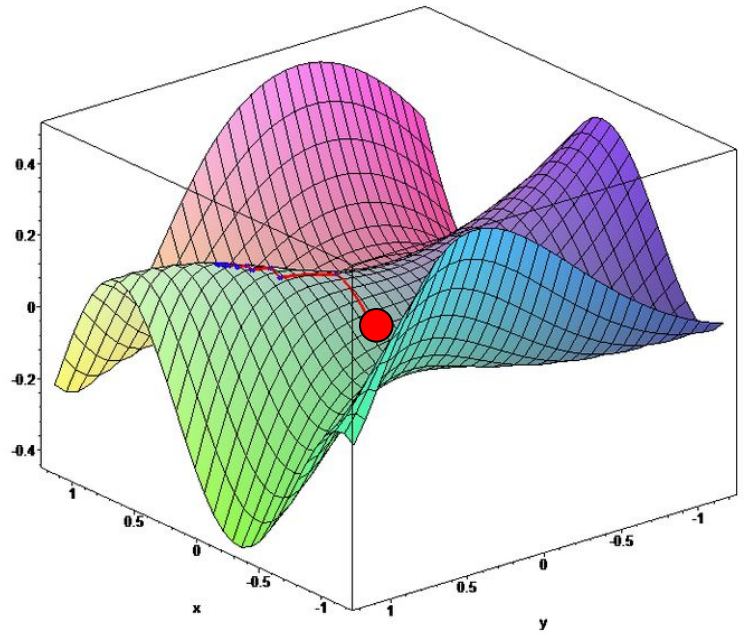
$\mathbf{H}$ is the *Hessian*,

$$\mathbf{H} \equiv \mathbf{F}''(\mathbf{x}) = \left[\frac{\partial^2 F}{\partial x_i \partial x_j}(\mathbf{x})\right]$$

Gradient Descent

Objective function:

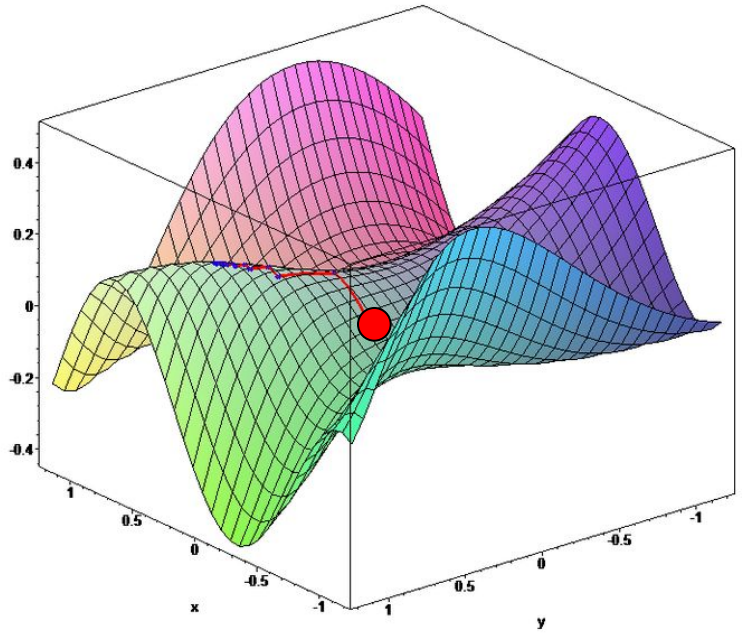
$$F(x,y) = \sin\left(\frac{1}{2}x^2 - \frac{1}{4}y^2 + 3\right)\cos(2x + 1 - e^y)$$



Which direction is optimal?

Gradient Descent

$$\lim_{\lambda \to 0} \frac{F(x) - F(x + \lambda h)}{\lambda \|h\|} = \frac{-h^T F'(x)}{\|h\|} = -F'(x) \cos\theta$$

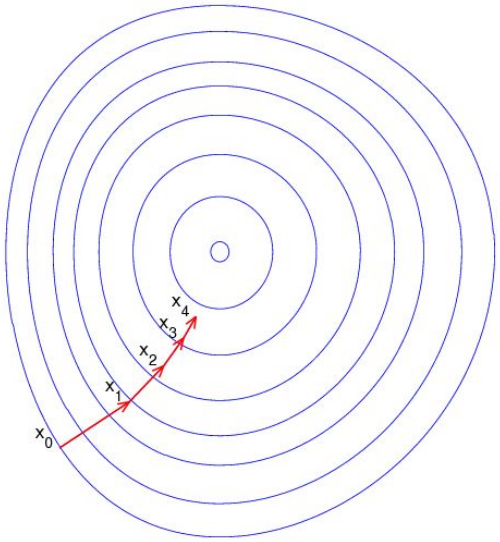


Which direction is optimal? $h = -F'(x)$

Gradient Descent

A first-order optimization algorithm.

To find a local minimum of a function using gradient descent, one takes steps proportional to the *negative* of the gradient of the function at the current point.



Gradient Descent

- Initialize k=0, choose x0

- While k<kmax

$$x_k = x_{k-1} - \lambda \nabla F(x_{k-1})$$

Gradient Descent

CognitiveSystems

- Usually slow method as it is first-order (depends on the first derivative only)
- As with any iterative method needs good starting estimate to converge
  - common solution: try many random starting points
- Needs knowledge of first derivative of error-function
  - numerical derivatives work as well, of course

Newton's Method

- Quadratic approximation

$$f(x+\Delta x) \approx f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2$$

- What's the minimum solution of the quadratic approximation

$$\Delta x = -\frac{f'(x)}{f''(x)}$$

Newton's Method

- High dimensional case:

$$F(x+\Delta x) \approx F(x) + \nabla F(x)\Delta x + \frac{1}{2}\Delta x^T H(x)\Delta x$$

- What's the optimal direction?

$$\Delta x \approx -H(x)^{-1}\nabla F(x)$$

Newton's Method

- Initialize k=0, choose x0

- While k<kmax

$$x_k = x_{k-1} - \lambda H(x)^{-1} \nabla F(x_{k-1})$$

Newton's Method

- Finding the inverse of the Hessian matrix is often expensive

- Approximation methods are often used
  - conjugate gradient method
  - quasi-newton method

Comparison

- Newton's method vs. Gradient descent
- Newton's method is much faster