Introducció

Curs Primavera 2019

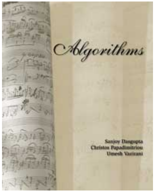
Curs d'algorísmia:

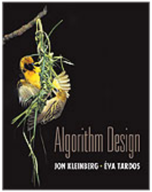
Què hauríeu de conèixer? (nivell EDA)

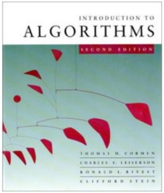
- Eines matemàtiques
  - Notació asimptòtica, recurrències, reduccions entre problemes
- Metodologia per a dissenyar algorismes:
  - Backtraking
  - Dividir i vèncer: Selecció lineal
  - Ordenació lineal
  - Voraços: algorismes d'aproximació, compressió de dades
  - Programació dinàmica
  - Distancies en grafs
  - Algorismes per a fluxos en xarxes: Aplicacions
  - Programació Lineal
  - Introducció a la criptoseguretat
  - Estructura de dades avançades: Blockchains,
- Problemes concrets i problemes reals

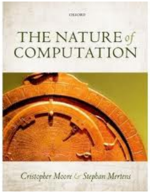
Bibliografia:

Referències principals:





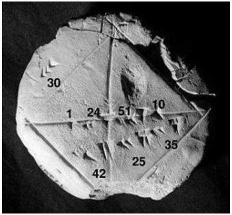




Algorismes.

Algorisme: recepta per a resoldre un problema.

Arrel (n)
$x_0 = 1$ $y_0 = n$
for $i = 1$ to 6 **do**
  $y_i = (x_{i-1} + y_{i-1})/2$
  $x_i = n/y_i$
end for



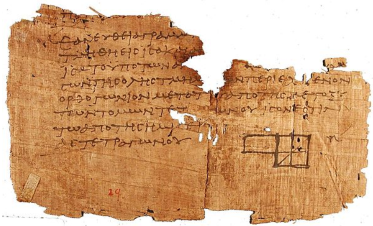
Babilònia (XVI BC)
Avui coneixem que: $\lim_{k\to\infty} x_k \to \sqrt{n}$ i ho fa ràpidament!

Un algorisme és correcte si per a qualsevol entrada s'atura i dóna una resposta correcta
Heurística: mètode per a resoldre un problema, que pot no aturar-se i pot no donar respostes exactes.

El primer algorisme no trivial

Donats $a, b \in \mathbb{Z}$:

mcd(a, b)
if $b = 0$ **then**
  return a
else if $b = 1$ **then**
  return 1
else
  mcd$(b, a \mod b)$
end if



Euclides 300 BC Proposició VII-2 als *Elements*

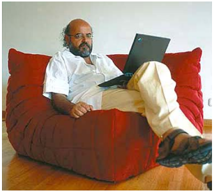
Conseqüència que si $b|a \Rightarrow \text{mcd}(a, b) = b$,
altrament $a = bt + r \Rightarrow \text{mcd}(a, b) = \text{mcd}(b, r)$

"The algorithmic lenses"

L'algorísmia representa una nova manera de mirar els problemes en diferents àmbits de la ciència i la tecnologia. L'algorísmia estudia quines són les particularitats de l'estructura interna del problemes que ajudin a dissenyar algorismes més eficients . A més de la informàtica, l'algorísmia ha canviat la manera d'abordar i resoldre els problemes en camps com:

- Matemàtiques
- Físiques
- Biologia i epidemiologia
- Economia
- Sociologia

⋮



Donat un algorisme: Què volem estudiar?

- Si és correcte,
- Si és eficient (el seu cost)

Què vol dir cost:

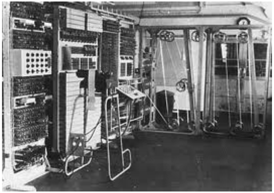
- Temps
- Memòria
- Capacitat de comunicació

$\vdots$

En aquest curs:
Cost d'un algorisme = temps de computació $T(n)$, parametrizat per grandària n.

Important

Les mesures de complexitat han de ser independents de les tecnologies existents

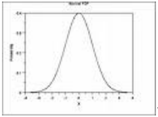
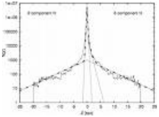


Anàlisis d'algorismes

:

Anàlisis cas pitjor: El màxim temps per a resoldre un problema amb grandària n, assumint l'entrada ve donada per un adversari.

Anàlisi mitjà: el nombre esperat de passos que un algorisme utilitza per a resoldre un problema, sobre totes les possibles entrades amb grandària n, agafades d'una distribució de probabilitat donada.

 $\longrightarrow$ $\boxed{\text{Deter. Algo.}}$ $\longrightarrow$ 

Escollir un estudiant per ajudar a la recerca

Tenim n estudiants, $\{1, \dots, n\}$ i volem entrevistar-los per agafar el més adient (cada estudiant ve identificat per un enter entre 1 i n)
Després d'entrevistar cada estudiant hem de decidir si el pre-seleccionem o no.
A la fi del procés escollim un sol estudiant, però indemnitzem a cadascun dels pre-seleccionats no escollits, amb $S > 0$ €. Volem minimitzar el cost de les indemnitzacions. Sigui $PS(n)$ el nombre de pre-seleccionats.

Hiring (n)
best:$=0$
$PS(n) = 0$
for $i = 1$ to n **do**
  interview i
  if i is better than best **then**
    best:$=i$ and pre-select i
    $PS(n) = PS(n) + 1$
  end if
end for

Complexitat

A la llista de l'adversari els estudiants estan escollits de manera que ordenadament van de menys apte a més apte. A cada pas l'algorisme es veu forçat a pre-seleccionar cada estudiant

Complexitat cas pitjor: $PS(n) \leq c_1 n$ (a on c_1 és una constant)

Notació asimptòtica.

Estudiem el comportament de $T(n)$ quan n pot prendre valors molt grans ($n \to \infty$)

if $n = 10$, $n^2 = 100$ i 2^n: 1024;

if $n = 100$, $n^2 = 10000$ i
$2^n = 1267650600228224014696703205376$;

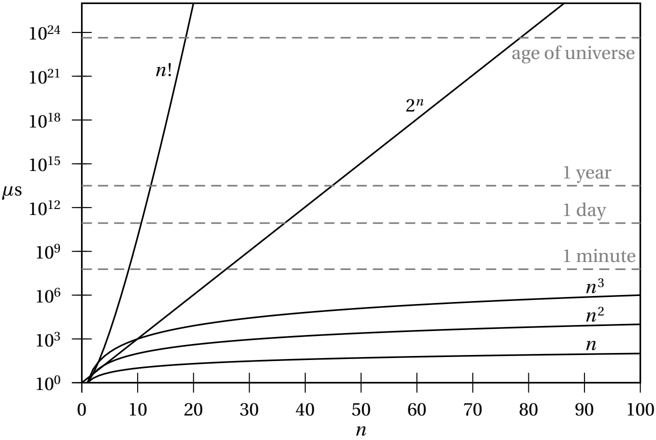
if $n = 10^3$ $n^2 = 10^6$ 2^n és un numero amb 302 dígits

10^{64} = nombre d'àtoms en la terra ($< 2^{213}$)

Notatció:
$\lg \equiv \log_2$; $\ln \equiv \log_e$; $\log \equiv \log_{10}$.

Comparació dels temps de computació amb els processadors actuals

Assumint una entrada amb grandària $n = 1$ es pot resoldre en 1 μsegon:



Temps de computació en funció de la grandària d'entrada n

	n	$n \lg n$	n^2	1.5^n	2^n
10	< 1s	< 1s	< 1s	< 1s	< 1s
50	< 1s	< 1s	< 1s	11m	36y
100	< 1s	< 1s	< 1s	12000y	10^{17}y
1000	< 1s	< 1s	< 1s	$> 10^{25}y$	$> 10^{25}y$
10^4	< 1s	< 1s	< 1s	$> 10^{25}y$	$> 10^{25}y$
10^5	< 1s	< 1s	< 1s	$> 10^{25}y$	$> 10^{25}y$
10^6	< 1s	20s	12d	$> 10^{25}y$	$> 10^{25}y$

Algorismes eficients i algorismes pràctics

Temps de computació: $n^{10^{10}}$ és un polinomi, però el temps de computació pot ser elevat.
De la mateixa manera, si tenim cn^2 per una constant $c = 10^{64}$, la constant té rellevància fins a entrades amb grandària $n = 10^{64}$.

A l'algorísmia, *factible (feasible)= temps polinòmic*, altrament *no factible (unfeasible)*.

A la pràctica és difícil implementar computacions amb valors més grans que n^4, per a valors "reals" de n.

A la pràctica, les constants grans tenen rellevància, però quan considerem asimptòtiques ($n \to \infty$) sempre podem agafar valors de n més grans que qualsevol constant.

Exemple O

Definition

Donades $f, g : \mathbb{Z} \to \mathbb{R}$, definim $f(n) \in O(g(n))$:

$$\limsup_{n\to\infty} \frac{f(n)}{g(n)} \leq c.$$

Utilitzant un abús de notació $f(n) = O(g(n))$
$n^{10} = O(e^n)$:
Escollir $c = 1$ and $n_0 = e^5$. Per tant, $\forall n \geq \lceil e^5 \rceil, 10 \ln n \leq n$,
$\Rightarrow e^{10 \ln n} \leq \lceil e^n \rceil$.

Notem que una constant $k > 0$, $k = O(1)$,
Però, la funció $f(x) = x$ no és $O(1)$

Omega: Ω. Fita inferior

Definition

Donades $f, g : \mathbb{Z} \to \mathbb{R}$, definim $f(n) \in \Omega(g(n))$:

$$\liminf_{n\to\infty} \frac{f(n)}{g(n)} \geq c > 0.$$

$f(n) = \Omega(g(n))$

$e^n = \Omega(n^{10})$

Theta: Θ

Definition

Donades $f, g : \mathbb{Z} \to \mathbb{R}$, definim $f(n) = \Theta(g(n))$:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c, \text{ on } c = \max\{c_1, c_2\}$$

$f(n) = \Theta(g(n))$ sii $f(n) = \Omega(g(n))$ i $f(n) = O(g(n))$

Donat $f(x) = a_k x^k + a_{k-1}x^{k-1} + \cdots + a_1 x + a_0$, llavors $f(x) = \Theta(x^k)$ (si $a_k \neq 0$).

La notació asimptòtica es pot utilitzar dintre d'una equació

$$2n^2 + 3n + 1 = 2n^2 + \Theta(n) = \Theta(n^2).$$

Little-oh: o

Definition
Donades $f, g : \mathbb{Z} \to \mathbb{R}$, definim $f(n) \in o(g(n))$

$$\limsup_{n \to \infty} \frac{f(n)}{g(n)} = 0.$$

$f(n) = o(g(n))$

$\forall \epsilon > 0, n^{1-\epsilon} = o(n)$ però $\forall \epsilon > 0, n^{1+\epsilon} \notin o(n)$

Little omega: ω

Definition

Donades $f, g : \mathbb{Z} \to \mathbb{R}$, definim $f(n) \in \omega(g(n))$

$$\liminf_{n\to\infty} \frac{f(n)}{g(n)} = \infty.$$

$f(n) = \omega(g(n))$

Notem: $n^2/2 = \omega(n)$ però $n^2/2 \notin \omega(n^2)$

Quadre resum

Símbol	$L = \lim_{n\to\infty} \frac{f(n)}{g(n)}$	intuïció ..
$f(n) = O(g(n))$	$L < \infty$	$f \leq g$
$f(n) = \Omega(g(n))$	$L > 0$	$f \geq g$
$f(n) = \Theta(g(n))$	$0 < L < \infty$	$f = g$
$f(n) = o(g(n))$	$L = 0$	$f < g$
$f(n) = \omega(g(n))$	$L = \infty$	$f > g$

Noms utilitzats per classes de funcions

classe	definition
polylogarirthmic	$f = O(\log^c n)$ for cte. c
polynomial	$f = O(n^c)$ for cte. c or $n^{O(1)}$
subexponential	$f = o(2^{n^\epsilon})$ $\forall 1 > \epsilon > 0$
exponential	$f = 2^{\mathrm{poly}(n)}$

Igualtat asimptòtica

Definition

Given functions $f, g : \mathbb{Z} \to \mathbb{Z}$, definim: $f(n) \sim g(n)$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$

Per tant, $f(n) \sim g(n)$ vol dir que quan n creix, les dues funcions esdevenen molt similars.

Recordant els grafs

En cas que no recordeu, mireu per ex. el Cap. 3 del Dasgupta, Papadimitriou, Vazirani.

Graf: $G = (V, E)$ a on V és el conjunt de vèrtexs, $|V| = n$. $E \subset V \times V$ és el conjunt d'arestes, $|E| = m$,

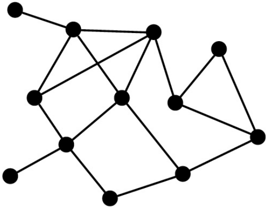
- Grafs: *dirigits* (digrafs) or *no-dirigits*.
- G no-dirigit es diu connex si hi ha un camí entre qualsevol 2 vèrtexs.
- Si G és connex, aleshores $\frac{n(n-1)}{2} \geq m \geq n - 1$.
- El grau d'un vèrtex v, $d(v)$ és el nombre d'arestes incidents a v.
- Una clica (clique) K_n amb n vèrtexs és un graf complert.

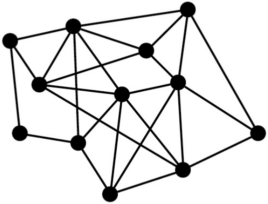
Digrafs

- Grafs amb arestes dirigides.
- El concepte de connectivitat en digrafs és el de connectivitat forta (strongly connected): hi ha un camí entre qualsevol 2 vèrtexs.
- Un digraf té com a màxim $n(n-1)$ arestes.

Densitat

Un graf G amb $|V| = n$ vèrtex es diu que és dens si $|E| = \Theta(n^2)$;
Es diu que és espars si $|E| = o(n^2)$.





Estructura de dades per a grafs.

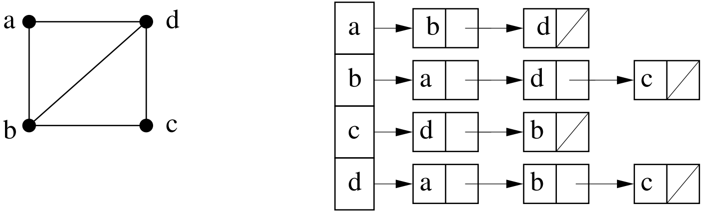
Els grafs són una de les eines més importats per a simular moltes situacions de la vida real.
Es necessiten maneres per a enmagatzemar i manipular de manera eficient els grafos.

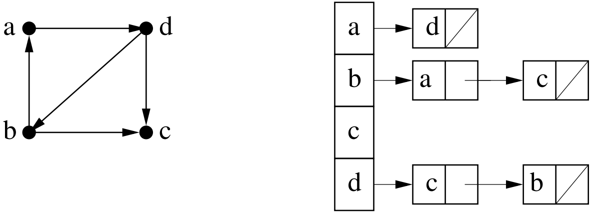
Sigui G un graf amb $V = \{1, 2, \ldots, n\}$. Les dues maneres més importants de representar grafs:

Llista d'adjacència

Matriu d'adjacència

Llista d'adjacència

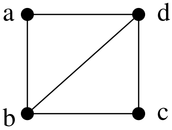




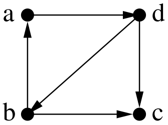
Matriu d'adjacència

Donat un graf, definim la seva matriu d'adjacència $n \times n$:

$$A[i,j] = \begin{cases} 1 & \text{if } (i,j) \in E, \\ 0 & \text{if } (i,j) \notin E. \end{cases}$$



$$\begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$



$$\begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Matriu d'adjacència

- Si G no és dirigit, la matriu d'adjacència és simètrica
- Si A és la matriu d'adjacència de G, aleshores A^2 dóna el nombre de camins amb longitud $= 2$ a G.
- Per a graphs amb pesos, $w_{i,j}$ a cada $(i,j) \in E$, la matriu d'adjacència té w_{ij} a la posició i,j.
- L'us de la matriu d'adjacència per representar un graf permet utilitzar L'àlgebra de matrius.

Comparació entre la representació amb llistes i amb matrius

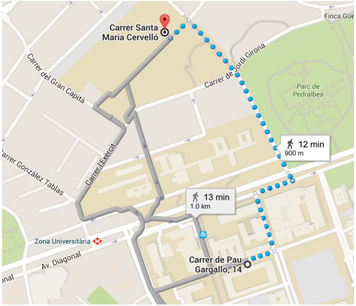
- Llistes utilitzen únicament un registre per vèrtex i un registre per aresta . Cada registre necessita 2×64 bits, per tant $128m$ bits $= \Theta(m)$.
- Matriu necessita $n \times n$ entrades, cadascuna pot ser de 1 bit: $\Theta(n^2)$ bits. Si el grafs te pesos, aleshores es necessita $64n^2$ bits.
- En grafs sense pesos, la representació amb matriu es millor per grafs densos, la representació per llistes és millor per grafs esparsos.

Representació llistes i matrius: Complexitat

- Afegir una aresta a G: Tant l'estructura de dades de matrius com les llistes necessiten $\Theta(1)$.
- "Query" si G te una aresta d'u a v:
  Matriu: $\Theta(1)$.
  Llista: $O(n)$
- Explorar tots els veïns del vèrtex v:
  Matriu: $\Theta(n)$
  Llista: $\Theta(|d(v)|)$
- Eliminar una aresta: Les dues estructures requereixen fer un "Query"

Searching a graph: Breadth First Search

1. Start with vertex v, visit and list all their neighbors at distance=1.
2. Then all their neighbors at distance 2 from v.
3. Repeat until all vertices visited.



BFS use a QUEUE, (FIFO) cua

Problem: If unknowingly the BFS revisits a vertex, the algorithm could yield the wrong notion of distance.
The solution to avoid that loop is to label each vertex we visit for first time, and ignore it when we revisit it.

Searching a graph: Depth First Search

explore

1. From current vertex, move to another.
2. Until you get stuck.
3. Then backtrack till new place to explore.

DFS use a STACK, (LIFO) pila

Time Complexity of DFS and BFS

- DFS:

For undirected and directed graphs: $O(|V| + |E|)$
In the case of sparse graphs $T(n) = O(n)$
For the case of a dense graph $T(n) = O(n^2)$

- BFS:

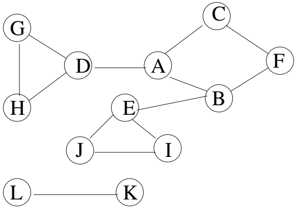
For undirected and directed graphs: $O(|V| + |E|)$

Therefore, the complexity of both procedures is linear in the size of the graph.

Connected components un undirected graphs.

Undirected Connected Components
INPUT: undirected graph G
QUESTION: Find if G is connected (if there is a path between any pair of vertices in $V(G)$).

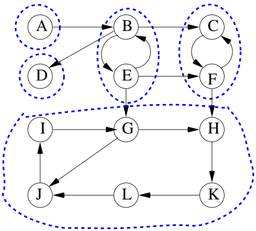
To find connected components in G apply DFS and count how many times **explore** is called. each time DFS calls **explore** on a vertex, it yields exactly the connected component to which the vertex belongs.

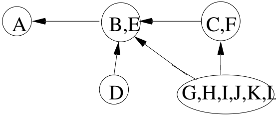


The problem can be solved in $O(|V| + |E|)$.

Strongly connected components in a digraph

Every digraph is a *directed acyclic graph (dag)* of its strongly connected components.





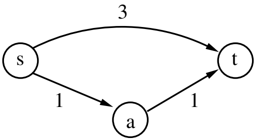
Kosharaju-Sharir's algorithm: Uses BFS (twice). Complexity $T(n) = O(|V| + |E|)$

Tarjan's algorithm: Based in using DFS. Complexity $T(n) = O(|V| + |E|)$

Both algorithms are optimal, i.e. are lineal, but in practice Tarjan's algorithm is easier to implement.

Caveat: BFS, DFS and weighted graphs

- BFS and DFS are algorithms to traverse a graph, if it is a weighted graph, it does not matter.
- However, using BFS or DFS to solve **optimization problems** (for example min or max distances) and doing it in the most efficient way, you need to use clever variations, to use the basic BFS (DFS) traverse.
- For instance applying BFS to the graph below, starting at vertex s, gives 3 as the min. distance $s \to t$:



- Implementing directly DFS or BFS on weighted graphs, **it does not yield necessarily an optimal solution**.

Ethics and Algorithms

- Ethics: the set of beliefs about right and wrong behavior.
- An algorithm is not a moral agent so it seems difficult to talk about the ethics of an algorithm.
- As algorithms become more powerful (learning-algorithms), their potential for causing great harm increases.
- Of the many facets of the topic, we could point to:
  - Ethics of algorithm designers and programmers, companies and institutions
  - Ethics associated with design or training of learning algorithms and their applications
  - The Singularity

Trollers, mobbing and hacking

- Trollers: Fake news, mobbing and bulling
  Some humans like to engage in diffusion of news with false and inflammatory content.
  It is known that in Twitter, fake news are 70% more likely to be retweeted than real news.
  Cambridge Analytics used Facebook to change the predictions in Brexit and in the 2016 US presidential elections.
- Cyberbulling: Using cyber-technology to threaten and intimidating people.
  In the US, during 2017, 34% of ten-agers (15-19 years) have been bullied, and 12% of tens admitted to have participated in a bullying. In the same period the suicides among the tens raised to 0.014%, most of the suicides are know due to cyberbullying
  http://www.pewinternet.org/2018/09/27/a-majority-of-teens-have-experienced-some-form-of-cyberbullying/

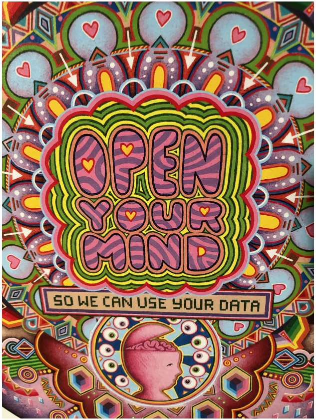
Trollers, mobbing and hacking

- Hacking: Activities to compromise digital devices. Sometimes, hacking might not be for malicious purposes, but most of the times it is. Phreaking, phishing and keyloggers are ways to stole information from individual people. In 2012-13, hackers stole data from 2×10^9 clients from Yahoo, Adobe and Dropbox.
  Some countries have created a division of their army to develop hackers, for a possible future cyber-war.
  In 2015 the Israel army basically destroyed the development of Iran's atomic program. In January 2018 Russia kept Ukraine without electricity for 3 days

Legally Collecting and selling of large amount of personal data

Brin & Page in *The Anatomy of a Large-Scale Hypertextual Web Search Engine (1996)*, stated:

"Advertising is wrong and bad and the search engine is corrupted if you sold advertising."



Self-learning algorithms (Deep-learning and so on)

- Machine learning algorithms learn from the training data they are given, regardless of any assumptions in the data.
- These algorithms can reflect, and magnify, the biases that are feed in the data.
- If an algorithm is trained on data that is racist or sexist, the result will also reflect this.
- Example: A chatbot is a program which can maintain a conversation. Microsoft put on-line on Twitter a chatbot Tay, which was designed to mimic the language level of a 19 year old, and to learn from interacting with human users of Twitter. Some users began tweeting inflammatory racist and sexists phrases, and after 20 hours Microsoft had to shut down Tay.

Self-learning algorithms

- Would you trust an algorithm trained to correct your exams?
- If we start trusting algorithms to make decisions, who will have the final word on important decisions? Will it be humans, or algorithms?
- Soon Self-driving cars are going to be widely deployed, who should be liable when accidents happen?
- Classical conundrum question: if a self-driving car has to choose between crashing into people or falling of a cliff, what should the car do?.
  Really, the conundrum is not well posed!
- When cars are safer than the average human drivers, should governments make human driving illegal?

The Power of Self-learning algorithms

The Power of Self-learning algorithms

THIS PERSON DOES NOT EXISTS!

It has been created artificially using an algorithm by a "generative adversarial networks" (GANs).

After adequate exposure to training data (provided by the trainer), GAN learn to infer the missing information by means of a competition

The Singularity

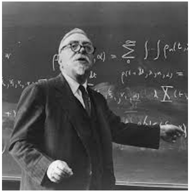
The singularity: the point in time where an artificial super-*intelligence* would surpass that of the brightest and most gifted human minds.
According to Ray Kuzwell (see references at the end) Singularity should happen in 2045.

- What is intelligence? Two attributes, clearly related to the concept are creativity and intuition.
- The physiological details of how those attributes develop in humans, are not well understood.
- I don't think we will see the Singularity in the XXIc., and I'm happy for the near-future of humanity.

Will AI achieve consciousness?

N. Wiener: The Human Use of Human Beings, 1950.

Wiener recognized that AI would not just imitate and replace human beings in many intelligent activities but would change human beings in the process.



The real danger is that such machines, though helpless by themselves, may be used by a human being or a block of human beings to increase their control over the rest of the race or that political leaders may attempt to control their populations by means not of machines themselves but through political techniques as narrow and indifferent to human possibility as if they had, in fact, been conceived mechanically. We're making tools, not colleagues, and the great danger is not appreciating the difference.

A possible solution: Licensing and bonding the makers and operators of these systems, whose errors and misjudgments can have dire consequences.

Bibliography: Courses

Different courses and papers:

- Course at Stony Brook U.:
  https://www3.cs.stonybrook.edu/~mueller/teaching/cse302/
- Paper by Mittelstadt et all:
  https://journals.sagepub.com/doi/abs/10.1177/2053951716679679
- Wikipedia on ethics and computers:
  https://en.wikipedia.org/wiki/Machine_ethics
- Wikipedia on ethics and AI:
  https://en.wikipedia.org/wiki/Ethics_of_artificial_intelligence

Bibliography: Books

Many books and papers:

- *The human use of human beings*. Norbert Wiener, 1950. https://monoskop.org/images/5/51/Wiener_Norbert_The_Human_Use_of_Human_Beings.pdf
- *The Singularity is near*. Ray Kurzweil. Viking 2005. http://www.singularity.com
- *Superintelligence: Paths, dangers, strategies*. Nick Bostrom. Oxford U.P 2014. https://www.goodreads.com/book/show/20527133-superintelligence
- Roman Yampolsky: *Pathway to Dangerous IA* ArXive (2015) https://arxiv.org/pdf/1511.03246.pdf