Kolmogorov complexity

Kolmogorov Andrey (1903 – 1987)

Probability theory

A lot of other topics

Link between randomness – learnability →compression

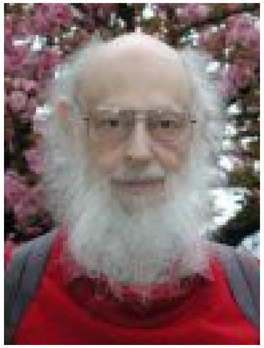
→ The Kolmogorov model (1962 – 1964 - 1975) (Solomonoff-Chaitin)

probabilistic view
a sequence of data s1, s2, …, sn
aim: to guess what is the next sn+1?
solution: choose the most probable (MAP principle)
BUT… what is this probability ?

A short history

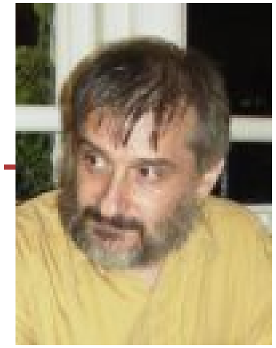
Andrei Kolmogorov →

Ray Solomonov →



Gregory Chaitin →

Leonid Levin →

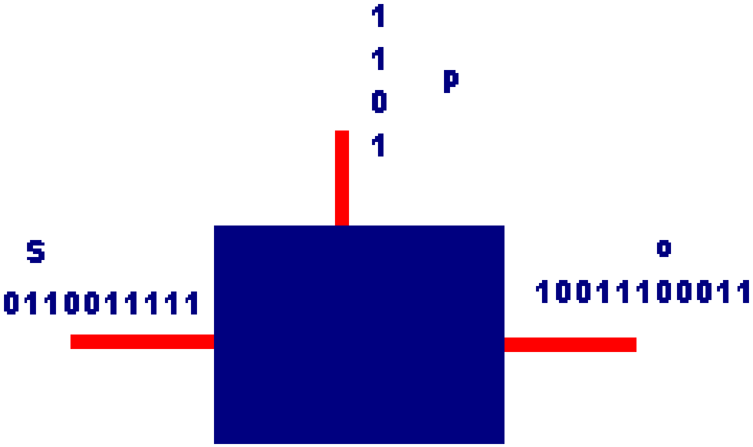


Turing machine

What is this ?

- Simple model of PC
- 3 tapes only = calculator C
  - Input s
  - Program p
  - Output o = C(p,s)

A simple picture



Kolmogorov complexity

Given a Turing calculator C

Given a finite input string y (=1000011001)

Given a program p (=11100001010110) can be infinite

2 options only:

- either C does not stop
- or C does stop and output a string x = C(p,y)

$$K(x/y) = \min \{|p| \mid C(p,y) = x\}$$

Taille du plus petit programme capable de produire X avec l'entrée Y

$$K(x) = \min \{|p| \mid C(p,Ø) = x\}$$

Taille du plus petit programme capable de produire X sans aucune entrées

Examples to understand

010101010101010101010101010101010101010101010
1010101010101010101…

Very simple: for i=1 to 10000 {write 0; write 1}

K(x) < 25*8=200

1100100000110000111011110111011001111101001000 01
00101011110010110

K(x) = 10000 ???

10^9 decimals of Л (Pi) : **simple C program to do it**

Properties of K

- *K(s)* **Turing machine independent (explain)**
- ***K(s) ≤ |s| + c (use print(s) for instance)***
- **There are s such that *K(s)* ≥ |*s*| *(to be proved)***
- **Compressible chains are quite rare *(to be proved)***
- **K(s) not computable ! *(to be explained)***
- **Relationship with Shannon entropy:**

  |E(s) – K(s) | < c

  So what ?

So what?

- K(x) = ultimate limit

Quantité d'info qu'elle contient

- K(x)= meaning of x – “informative content” of x
- K(x) = lower bound of any compression of x

K estimation using compression!

K(x) < bzip2(x) < gzip(x)< etc…..

OK … we can estimate ... then ???

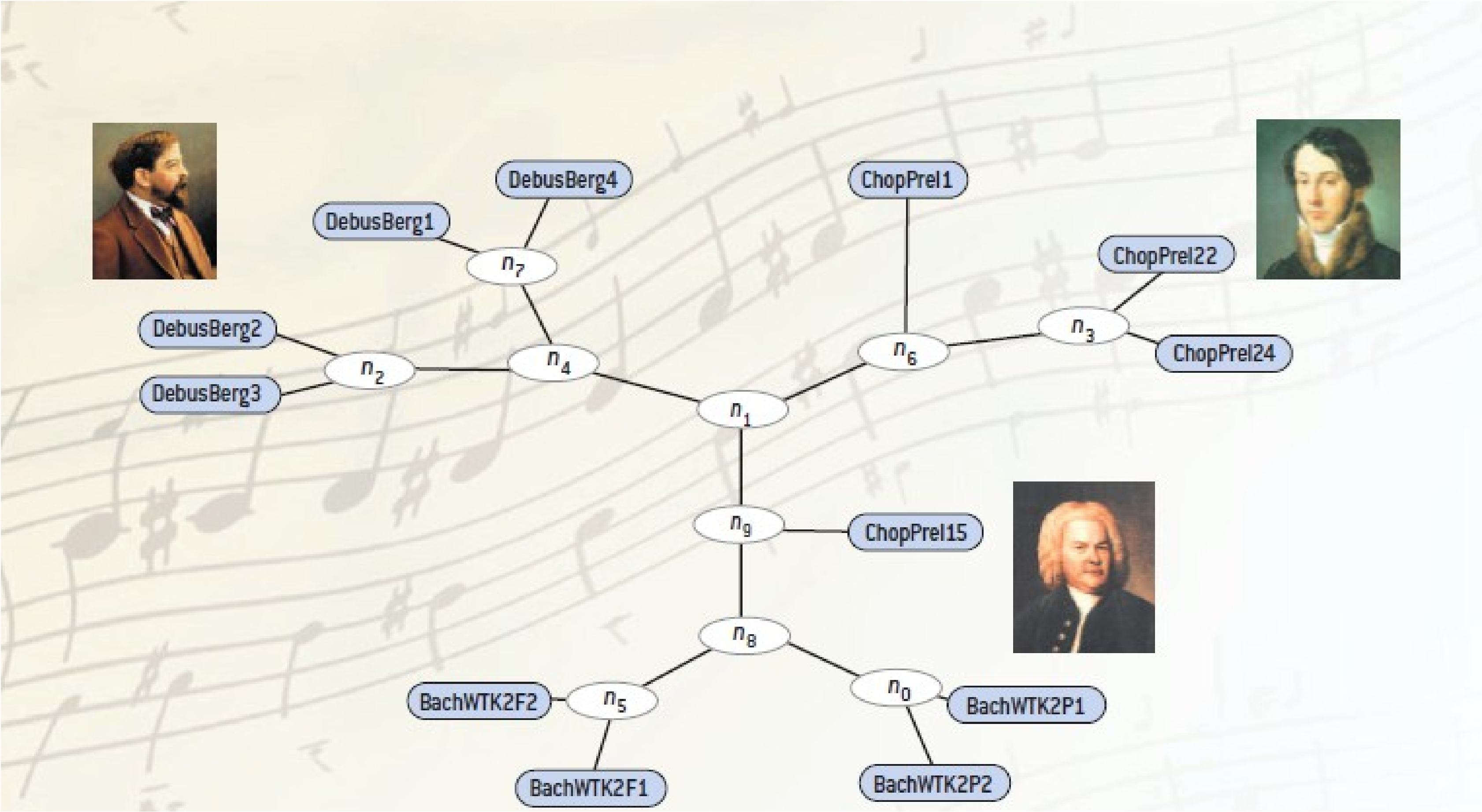
Information distance (Bennett)

Similarity using K:

- a -> K(a) (compress a and compute the size of the result)
  (draw a diagram on the board)
- b -> K(b)
- ab -> K(ab)
- m(ab)=K(a) + K(b) – K(ab) = measure of the common content
- d(a,b) = if K(a) > K(b) then 1 – m(a,b)/K(a)
  else 1 – m(a,b)/K(b)

And you know what ?

It works with music...



Can we do more ???? YES

Complearn.org
findfraud.exe

It works with everything...

- Pictures (to understand)
- Texts (Corneille wrote Moliere !)
- Student plagiarism/fraud: findFraud(www.complearn.org)
- Genome
- Spam
- Security (IDS) using K only (no distance needed)

But

Can we do more ???? YES

Solomonov probability measure

Main idea (back to initial problem):

$$p(x) = 2^{-K(x)}$$

- a priori probability (Bayes formula)
- p: the universal distribution
- s more complex, s less probable
- p(x) = probability for s to appear

Main problem:

p is not a proba distribution over $\{0,1\}^N$

We have to work a little bit more….

Reduced programs

Reduced programs set

- Choose a calculator C
- For each string x, program p stopping, reduced prog pr(p,x)

Pr(x) = {pr (p,x) ; p program and p stops with x as output}

Prefix free set

- Pr(x) is prefix-free (to explain)
- Idem for U_x Pr(x) (qui sont disjoints en fait)

Riemann measure on [0,1] (probability)

- pr(p,x) = 10001110… → 0.10001110… = real number in [0,1]
- Prob(pr) = mes({pr.q ; q in {0,1}   }) = $2^{-|pr|}$
- Prob(Pr(x)) = $\Sigma_{pr\ in\ Pr(x)}$Prob(pr) <= 1 for sure ! (prefix free)
- So Σ_x p(x) < 1 … OK

Help from Google

From K to p: Solomonov

From p to K : using log inverse function!

$$K(x) = -\log (p(x))$$

Mass probability generator = K estimator

Why not Google? (Cilibrasi-Vitanyi)

Keyword a → Google frequency = prob

NGD=approximation information distance

Experiments with Google

- learning with Google (prime number, etc…)
- emergency (with SVM)

Numbers versus colors via Google

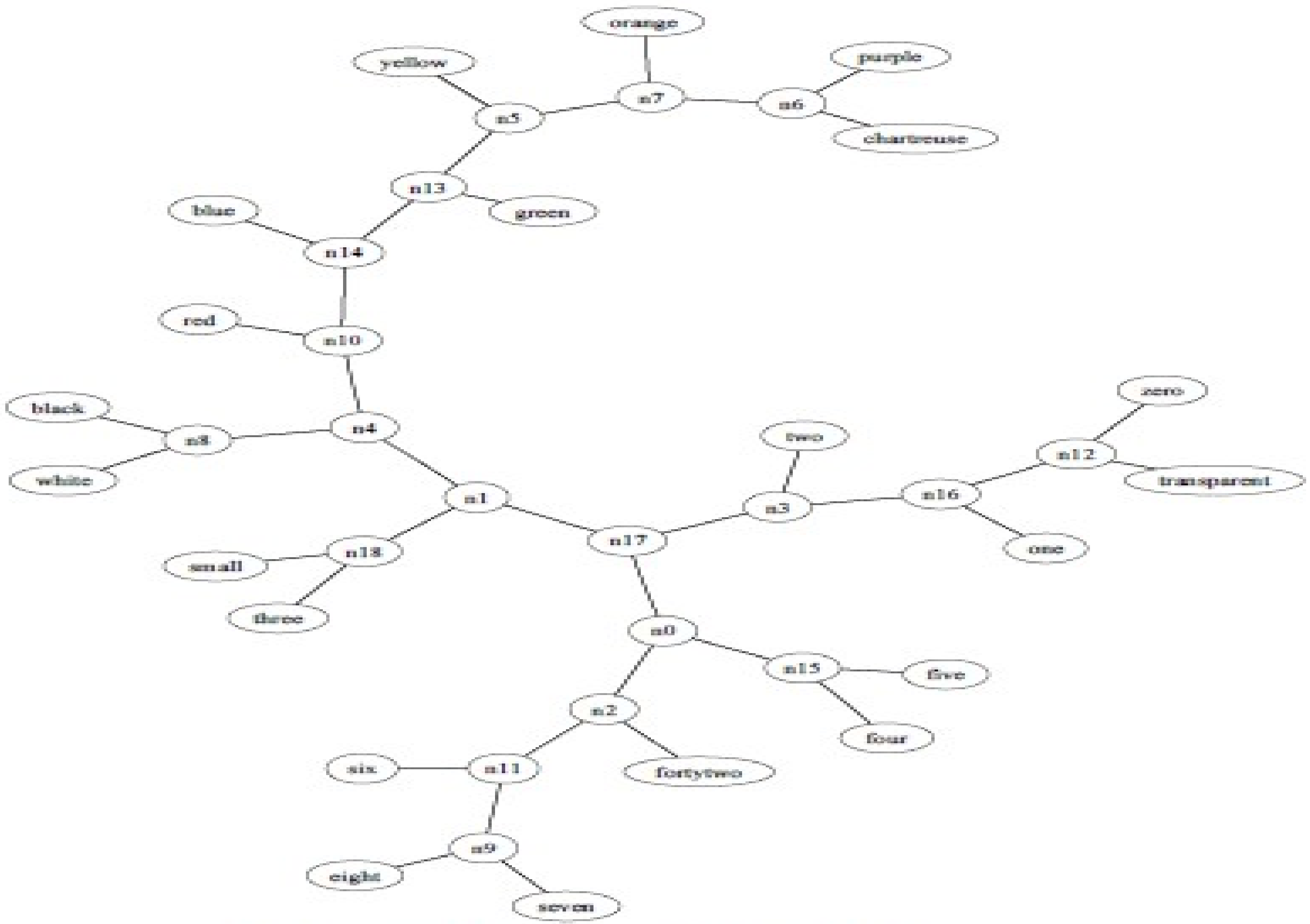


Figure 2: Colors and numbers arranged into a tree using NGD .

Learning emergency

Training Data

Positive Training	(22 cases)			
avalanche	bomb threat	broken leg	burglary	car collision
death threat	fire	flood	gas leak	heart attack
hurricane	landslide	murder	overdose	pneumonia
rape	roof collapse	sinking ship	stroke	tornado
train wreck	trapped miners			
Negative Training	(25 cases)			
arthritis	broken dishwasher	broken toe	cat in tree	contempt of cou
dandruff	delayed train	dizziness	drunkenness	enumeration
flat tire	frog	headache	leaky faucet	littering
missing dog	paper cut	practical joke	rain	roof leak
sore throat	sunset	truancy	vagrancy	vulgarity
Anchors	(6 dimensions)			
crime	happy	help	safe	urgent
wash				

Testing Results

	Positive tests	Negative tests
Positive Predictions	assault, coma, electrocution, heat stroke, homicide, looting, meningitis, robbery, suicide	menopause, prank call, pregnancy, traffic jam
Negative Predictions	sprained ankle	acne, annoying sister, campfire, desk, mayday, meal

Accuracy 15/20 = 75.00%

Figure 4: Google- SVM learning of "emergencies."

Chaitin number

- Prob. for a random program to stop: halting probability

$$\Omega = \text{Prob}(\text{Pr}(\varnothing)) = \Sigma p \text{ in } \text{pr}(\varnothing)\ 2^{-|p|} \leq 1$$

- Ω = 0.000000100000010000011000100001101000111111001011101110100001000 0... (84bits with only 64 reliable !)
  - = 0.0078749969978123844..
- Properties
  - Transcendental (to be explained)
  - Normal (to explain) (PI we do not know)
  - Uncomputable …. Why

Conclusion

- Kolmogorov:
  - very powerful
  - work still in progress

- ML = mix between theory and practice
  - to try before to prove;-)
  - not always clean explanation