Lecture 6. Prefix Complexity K, Randomness, and Induction

- The plain Kolmogorov complexity C(x) has a lot of "minor" but bothersome problems
  - Not subadditive: $C(x,y) \le C(x)+C(y)$ only modulo a log n term. There exists x,y s.t. $C(x,y)>C(x)+C(y)+\log n - c$. (This is because there are $(n+1)2^n$ pairs of x,y s.t. $|x|+|y|=n$. Some pair in this set has complexity $n+\log n$.)
  - Nonmonotonicity over prefixes
  - Problems when defining random infinite sequences in connection with Martin-Lof theory where we wish to identify infinite random sequences with those whose finite initial segments are all incompressible, Lecture 2
  - Problem with Solomonoff's initial universal distribution

$$P(x) = 2^{-C(x)}$$

but $\Sigma\ P(x)=\infty$.

In order to fix the problems …

- Let $x = x_0 x_1 \ldots x_n$, then

  $\bar{x} = \underline{x_0} 0 x_1 0 x_2 0 \ldots x_n 1$ and

  $x' = |x|\, x$
- Thus, x' is a prefix code such that $|x'| \le |x| + 2 \log|x|$
- x' is a self-delimiting version of x.
- Let reference TM's have only binary alphabet {0,1}, no blank B. The programs p should form an effective prefix code:

  $\forall p,p'$ [p is not prefix of p']
- Resulting self-delimiting Kolmogorov complexity (Levin, 1974, Chaitin 1975). We use K for prefix Kolmogorov complexity to distinguish from C, the plain Kolmogorov complexity.

Properties

- By Kraft's Inequality (proof – look at the binary tree):

  $$\Sigma_{x \in \Sigma^*} 2^{-K(x)} \leq 1$$

- Naturally subadditive
- Not monotonic over prefixes (then we need another version like monotonic Kolmogorov complexity)
- $C(x) \leq K(x) \leq C(x)+2 \log C(x)$
- $K(x) \leq K(x|n)+K(n)+O(1)$
- $K(x|n) \leq C(x) + O(1)$

  $$\leq C(x|n) + K(n)+O(1)$$

  $$\leq C(x|n)+\log^* n+\log n+\log\log n+\ldots+O(1)$$

Alice's revenge

- Remember Bob at a cheating casino flipped 100 heads in a row.
- Now Alice can have a winning strategy. She proposes the following:
  - She pays \$1 to Bob for every time she looses on 0-flip, gets \$1 for every time she wins on 1-flip.
  - She pays \$1 extra at start of the game.
  - She receives $2^{100-K(x)}$ in return, for flip sequence x of length 100.
- Note that this is a fair proposal as expectancy for 100 flips of fair coin is

  $\Sigma_{|x|=100} 2^{-100} 2^{100-K(x)} < \$1$

  But if Bob cheats with 1^{100}, then Alice gets $2^{100-\log 100}$

Chaitin's mystery number Ω

Define $\Omega = \sum_{p \text{ halts}} 2^{-|p|}$ (<1 by Kraft's inequality and there is a nonhalting program p). Now Ω is a nonrational number.

Theorem 1. Let $X_i=1$ iff the ith program halts. Then $\Omega_{1:n}$ encodes $X_{1:2^n}$. I.e., from $\Omega_{1:n}$ we can compute $X_{1:2^n}$

Proof. (1) $\Omega_{1:n} < \Omega < \Omega_{1:n}+2^{-n}$. (2) Dovetailing simulate all programs till $\Omega' > \Omega_{1:n}$. Then if p, $|p|\le n$, has not halted yet, it will not (since otherwise $\Omega > \Omega'+2^{-n} > \Omega$). QED

- Bennett: $\Omega_{1:10,000}$ yields all interesting mathematics.

Theorem 2. For some c and all n: $K(\Omega_{1:n}) \ge n - c$.

- Remark. Ω is a particular random sequence!

Proof. By Theorem 1, given $\Omega_{1:n}$ we can obtain all halting programs of length ≤ n. For any x that is not an output of these programs, we have $K(x)>n$. Since from $\Omega_{1:n}$ we can obtain such x, it must be the case that $K(\Omega_{1:n}) \ge n - c$. QED

Universal distribution

- A (discrete) semi-measure is a function P that satisfies $\Sigma_{x\in N}P(x)\leq 1$.
- An enumerable (=lower semicomputable) semi-measure P_0 is universal (maximal) if for every enumerable semi-measure P, there is a constant c_p, s.t. for all $x\in N$, $c_P P_0(x)\geq P(x)$. We say that P_0 dominates each P. We can set c_P = 2^{K(P)}. Next 2 theorems are due to L.A. Levin.

Theorem. There is a universal enumerable semi-measure ***m***.

We can set $\boldsymbol{m}(x)=\sum P(x)/c_P$ the sum taken over all enumerable probability mass functions P (countably many)

Coding Theorem. $\log 1/\boldsymbol{m}(x) = K(x) + O(1)$-Proofs omitted.

- Remark. This universal distribution ***m*** is one of the foremost notions in KC theory. As prior probability in a Bayes rule, it maximizes ignorance by assigning maximal probability to all objects (as it dominates other distributions up to a multiplicative constant).

Randomness Test for Finite Strings

- **Lemma.** If P is computable, then
- $\delta_0(x) = \log \boldsymbol{m}(x)/P(x)$
- is a universal P-test. Note $-K(P) \le \log \boldsymbol{m}(x)/P(x)$ by dominating property of ***m***.

- **Proof.** (i) δ_0 is lower semicomputable.
- (ii) $\sum_x P(x)2^{\delta_0(x)} = \sum_x \boldsymbol{m}(x) \le 1.$
- (iii) δ is a test $\rightarrow f(x) = P(x)2^{\delta(x)}$ is lower semicomputable & $\sum f(x) \le 1$.
- Hence, by universality of ***m***, $f(x) = O(\boldsymbol{m}(x))$.
- Therefore, $\delta(x) \le \delta_0(x) + O(1)$.
- QED

Individual randomness (finite |x|)

- Theorem. X is P-random iff log $\boldsymbol{m}(x)/P(x) \leq 0$ (or a small value).
- Recall: log $1/\boldsymbol{m}(x)=K(x)$ (ignore O(1) terms).
- Example. Let P be the uniform distribution. Then,
- log $1/P(x) =|x|$ and x is random iff $K(x) \geq |x|$.
- 1. Let x=00...0 ($|x|=n$). Then, $K(x) \leq \log n + 2 \log \log n$.
- So $K(x) << |x|$ and x is not random.
- 2. Let y = 011...01 ($|y|=n$ and typical fair coin flips).
- Then, $K(y) \geq n$. So $K(y) \geq |y|$ and y is random.

Occam' Razor

- ***m***(x) = 2^{-K(x)} embodies `Occam's Razor'.
- Simple objects (with low prefix complexity)
- have high probability and complex objects
- (with high prefix complexity) have low
- Probability.
- x=00...0 (n 0's) has K(x) ≤ log n + 2 log log n
- and ***m***(x) ≥ 1/n (log n)^2
- y=01...1 (length n random string) has K(y) ≥ n
- *a*nd ***m***(y) ≤ 1/2^n

Randomness Test for Infinite Sequences: Schnorr's Theorem

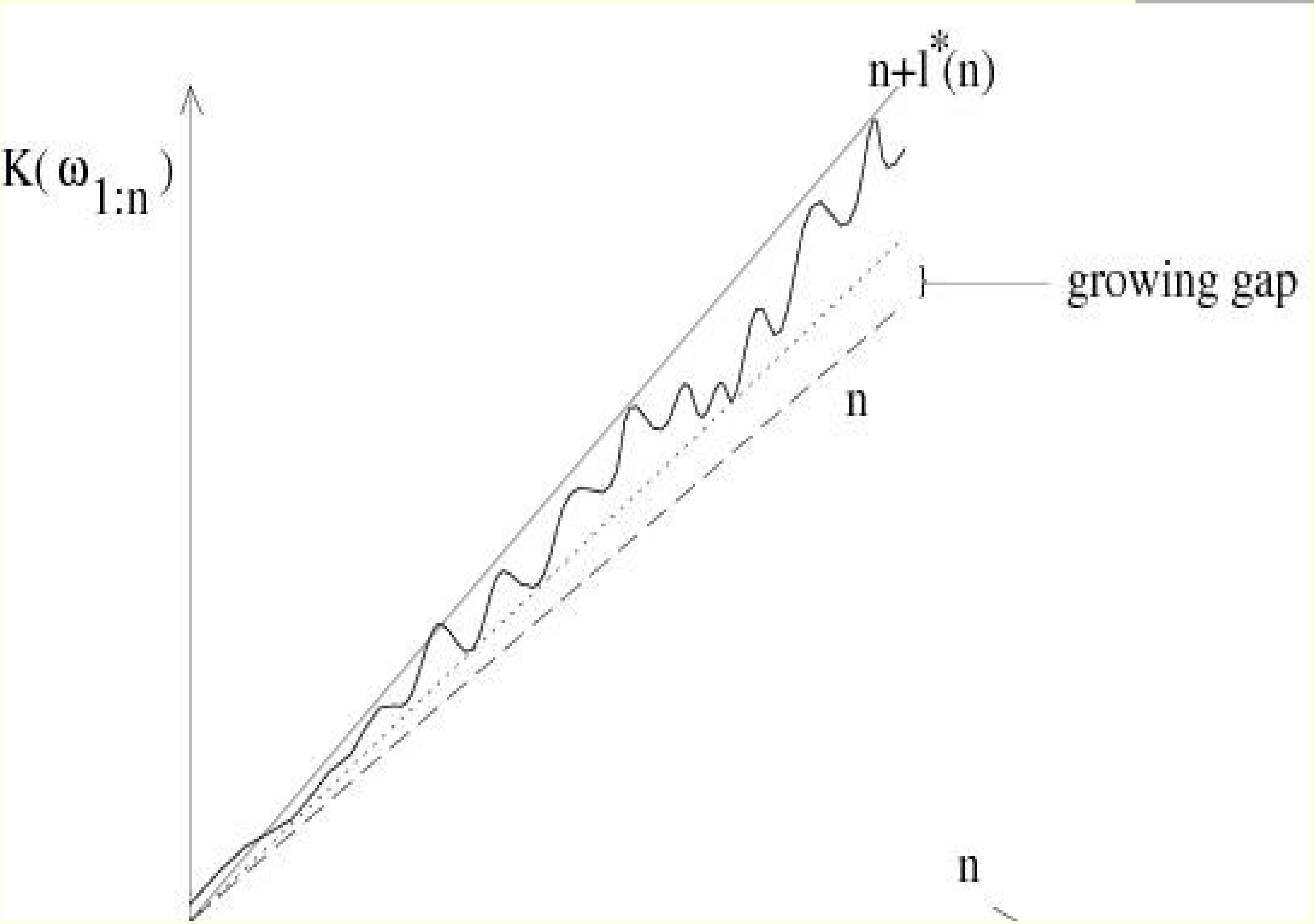
- Theorem. An infinite binary sequence ω is (Martin-Lof) random (random with respect to the uniform measure λ) iff there is a constant c such that for all n,

$$K(\omega_{1:n}) \geq n-c.$$

- Proof omitted---see textbook.

- (Note, please compare with Lecture 2, C-measure)

Complexity oscillations of initial segments of infinite high-complexity sequences



Entropy

- Theorem. If P is a computable probability mass function with finite entropy H(P), then

$$H(P) \leq \sum P(x)K(x) \leq H(P)+K(P)+O(1).$$

Proof.

Lower bound: by Noiseless Coding Theorem since {K(x)} is length set prefix-free code.

Upper bound: $\boldsymbol{m}(x) \geq 2^{-K(P)} P(x)$ for all x. Hence, $K(x) = \log 1/\boldsymbol{m}(x)+O(1) \leq K(P)+ \log 1/P(x)+O(1)$.

QED

Symmetry of Information.

- Theorem. Let x* denote shortest program for x (1st in standard enumeration). Then, up to an additive constant

$$K(x,y)=K(x)+K(y|x^*)=K(y)+K(x|y^*)=K(y,x).$$

Proof. Omitted---see textbook. QED

Remark 1. Let $I(x:y)=K(x)-K(x|y^*)$ (information in x about y). Then: $I(x:y)=I(y:x)$ up to a constant. So we call $I(x:y)$ the algorithmic mutual information which is symmetric up to a constant.

Remark 2. $K(x|y^*)=K(x|y,K(y))$.

Complexity of Complexity

- Theorem. For every n there are strings x of
- length n such that (up to a constant term):
- $\log n - \log\log n \le K(K(x)|x) \le \log n$.
- Proof. Upper bound is obvious since $K(x) \le n+2\log n$.
- Hence we have $K(K(x)|x) \le K(K(x)|n)+O(1) \le \log n + O(1)$.
- Lower bound is complex and omitted, see textbook. QED
- Corollary. Let length x be n. Then,
- $K(K(x),x) = K(x)+K(K(x)|x,K(x))=K(x)$, but
- $K(x)+K(K(x)|x)$ can be $K(x)+\log n - \log\log n$. Hence the
- Symmetry of Information is sharp.

Average-case complexity under *m*

Theorem [Li-Vitanyi]. If the input to an algorithm A is distributed according to ***m***, then the average-case time complexity of A is order-of-magnitude of A's worst-case time complexity.

Proof. Let T(n) be the worst-case time complexity. Define P(x) as follows:

- $a_n = \Sigma_{|x|=n} \boldsymbol{m}(x)$
- If |x|=n, and x is the first s.t. t(x)=T(n), then $P(x) := a_n$ else P(x):=0.

Thus, P(x) is enumerable, hence $c_P \boldsymbol{m}(x) \geq P(x)$. Then the average time complexity of A under ***m***(x) is:

$$T(n|\boldsymbol{m}) = \Sigma_{|x|=n} \boldsymbol{m}(x) t(x) / \Sigma_{|x|=n} \boldsymbol{m}(x)$$
$$\geq 1/c_P \, \Sigma_{|x|=n} P(x) T(n) / \Sigma_{|x|=n} \boldsymbol{m}(x)$$
$$= 1/c_P \, \Sigma_{|x|=n} [P(x)/\Sigma_{|x|=n} P(x)] \, T(n) = 1/c_P T(n). \qquad \text{QED}$$

Intuition: The x with worst time has low KC, hence large ***m***(x)

Example: Quicksort. With easy inputs, more likely incur worst case.

General Prediction

- Hypothesis formation, experiment, outcomes, hypothesis adjustment, prediction, experiment, outcomes, ...
- Encode this (infinite) sequence as 0's and 1's
- The investigated phenomenon can be viewed as a measure μ over the $\{0,1\}^\infty$ with probability $\mu(y|x)=\mu(xy)/\mu(x)$ of predicting y after having seen x.
- If we know μ then we can predict as good as is possible.

Solomonoff's Approach

- Solomonoff (1960, 1964): given a sequence of observations: S=010011100010101110 ..
- Question: predict next bit of S.
- Using Bayesian rule:

  P(S1|S)=P(S1)P(S|S1) / P(S)

  =P(S1) / P(S)

  here P(S1) is the prior probability, and we know P(S|S1)=1.
- Choose universal prior probability:

  P(S) = ***M***(S) = ∑ 2^-l(p) summed over all p which are shortest programs for which U(p…) = S....
- ***M*** is the continuous version of ***m*** (for infinite sequences in {0,1}^∞ .

Prediction a la Solomonoff

- Every predictive task is essentially extrapolation of a binary sequence:
- ...0101101░ 0 or 1 ?

- Universal semimeasure
- ***M***(x)= ***M***{x....: x ε {0,1}*} constant-multiplicatively dominates all (semi)computable semimeasures μ.

General Task

- Task of AI and prediction science: Determine for a phenomenon expresed by measure μ
- $$\mu(y|x) = \mu(xy)/\mu(x)$$
- The probability that after having observed data x the next observations show data y.

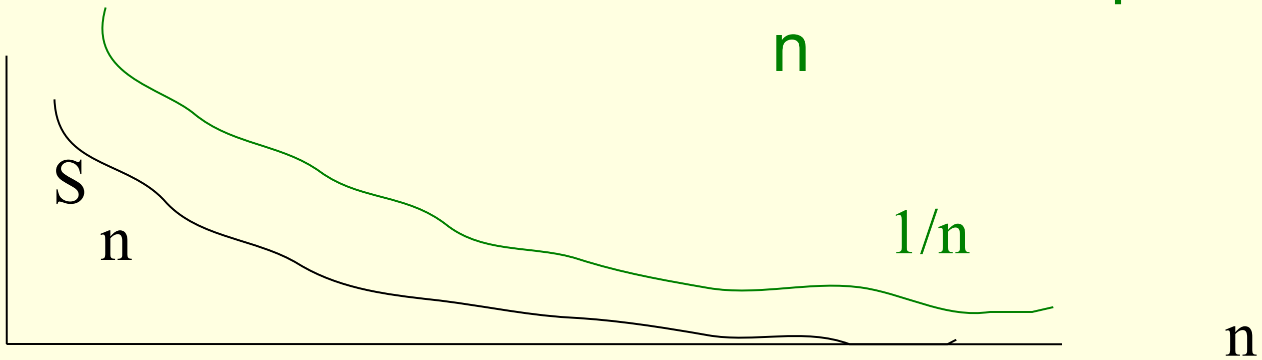
Solomonoff: *M*(x) is good predictor

- Expected error squared in the nth prediction:

$$S_n = \sum_{|x|=n-1} \mu(x) [\mu(0|x) - M(0|x)]^2$$

- Theorem. $\sum_n S_n \le$ constant (½K(μ) ln 2)

- Hence: Prediction error S_n in n-th prediction:



Predictor in ratio

- Theorem. For fixed length y and computable μ:
- $\boldsymbol{M}(y|x)/\mu(y|x) \rightarrow 1$ for $x \rightarrow \infty$
- with μ-measure 1.
- Hence we can estimate conditional μ-probability by ***M*** **with almost no error.**
- **Question: Does this imply Occam's razor:**
- **``shortest program predicts best''?**

M is universal predictor for all computable μ in expectation

- But M is a continuous measure over $\{0,1\}^\infty$ and weighs all programs for x, including shortest one:
- $$M(x) = \sum_{U(p\ldots)=x\ldots} 2^{-|p|} \quad \text{(p minimal)}$$
- Lemma (P. Gacs) For some x, log 1/ M(x) << shortest program for x. This is different from the Coding Theorem in the discrete case where always log 1/m(x) =K(x)+O(1).
- Corollary: Using shortest program for data is not always best predictor!

Theorem (Vitanyi-Li)

- For almost all x (i.e. with μ-measure 1):
- log 1/***M***(y|x) = Km(xy)-Km(x) +O(1) with Km the complexity (shortest program length |p|) with respect to U(p...)= x....
- Hence, it is a good heuristic to choose an extrapolation y that minimizes the length difference between the shortest program producing xy... and the one that produces x...
- I.e.; Occam's razor!