Languages and Complexity

Matilde Marcolli

CS101: Mathematical and Computational Linguistics

Winter 2015

Shift dynamical systems associated to formal languages

- Regular languages $\mathcal{L} \Leftrightarrow$ Finite State Automaton
- finite directed graph: vertices = states Q, edges $e_{q,a,q'}$ = transitions τ with edge labels $a \in \mathfrak{A}$; source = initial state q_0; sinks = final states
- Markov shift: consider the set of all the possible infinite paths in the oriented graph starting at any state (not ending at sink)
- these are infinite words defining points of a Cantor set $\Lambda_{\mathcal{L}}$, with a shift map $\sigma : \Lambda_{\mathcal{L}} \to \Lambda_{\mathcal{L}}$ moving one step in the path
- probabilities $\mathbb{P}(\tau)$ assigned to transitions (production rules) in the language, together with probabilities π_q on set of states Q with $\sum_q \pi_q \mathbb{P}(e_{q,a,q'}) = \pi_{q'}$ determines Markov measure on $\Lambda_{\mathcal{L}}$

Random walks and Probabilistic Context Free Grammars

- Example of a Context Free Grammar:
$\mathcal{G} = (V_N = \{V, W\}, V_T = \{+, -\}, P, S = W)$
with production rules

$$W \to +V, \quad V \to +VV, \quad V \to -$$

- after three steps from start the list of all possible productions

$$+++VV+VV, \quad ++-+VV, \quad +++VV-, \quad ++--, \quad +-$$

- Language $\mathcal{L}_{\mathcal{G}}$ produced by this grammar = paths with barrier
all paths on a 1-dim lattice that eventually return to the origin but never enter the negative half axis
- not regular: contains $\{+^n -^n\}$

- make the example probabilistic:

$$\mathbb{P}(V \to +VV) + \mathbb{P}(V \to -) = 1$$

probability distribution $(p_+, p_-) = (p, 1-p)$

- if equally distributed $(p_+, p_-) = (1/2, 1/2)$ then diffusive random motion in which average distance from barrier grows line $\sqrt{n}$ for length n large: sufficiently far from the barrier like a Bernoulli random process

- if $p_- > 1/2$ then average distance from the barrier remains bounded: get a power spectrum for the sequence of increments with a peak corresponding to successions of consecutive bounces at the barrier

Context Sensitive Grammars and dynamical systems

- Self-avoiding random walk: in the plane 4 moves *u*,*d*,*l*,*r*
- to be self avoiding: exclude sequences *lr*, *ud*, *uldr*, ...
- generate the admissible symbols via a Turing machine (or linear bounded automaton):

(1) record on the tape coordinates of current position
(2) new alphabet symbol emitted
(3) head moves back on the tape to check if any previous position agree with last one, if so reject, if not record new position...

- number of different paths of length n conjectured

$$N(n) \sim A\, n^{\gamma-1} \mu^n$$

μ depends on dimension and lattice geometry, $\gamma \geq 1$ universal constant (independent of lattice, but depend on dimension) average displacement like $n^{2\nu}$: similar universal power law

- Entropy: $S \sim \log \mu$

• Previous examples show that one can think of (formal) languages as physical systems (dynamical systems, random walks)

• How does one measure complexity of a physical system?

• Kolmogorov complexity: measures length of a minimal algorithmic description

... but ... gives very high complexity to completely random things

• Shannon entropy: measures average number of bits, for objects drawn from a statistical ensemble

• Gell-Mann complexity: complexity is high in an intermediate region between total order and complete randomness

Kolmogorov complexity

• Let $T_{\mathcal{U}}$ be a universal Turing machine (a Turing machine that can simulate any other arbitrary Turing machine: reads on tape both the input and the description of the Turing machine it should simulate)

• Given a string w in an alphabet $\mathfrak{A}$, the Kolmogorov complexity

$$\mathcal{K}_{T_{\mathcal{U}}}(w) = \min_{P: T_{\mathcal{U}}(P)=w} \ell(P),$$

minimal length of a program that outputs w

• universality: given any other Turing machine T

$$\mathcal{K}_T(w) = \mathcal{K}_{T_{\mathcal{U}}}(w) + c_T$$

shift by a bounded constant, independent of w; c_T is the Kolmogorov complexity of the program needed to describe T for $T_{\mathcal{U}}$ to simulate it

- conditional Kolmogorov complexity

$$\mathcal{K}_{T_{\mathcal{U}}}(w \,|\, \ell(w)) = \min_{P: T_{\mathcal{U}}(P, \ell(w)) = w} \ell(P),$$

where the length $\ell(w)$ is given and made available to machine $T_{\mathcal{U}}$

$$\mathcal{K}(w \,|\, \ell(w)) \leq \ell(w) + c,$$

because if know $\ell(w)$ then a possible program is just to write out w: then $\ell(w) + c$ is just number of bits needed for transmission of w plus print instructions

- upper bound

$$\mathcal{K}_{T_{\mathcal{U}}}(w) \leq \mathcal{K}_{T_{\mathcal{U}}}(w \,|\, \ell(w)) + 2 \log \ell(w) + c$$

if don't know a priori $\ell(w)$ need to signal end of description of w (can show for this suffices a "punctuation method" that adds the term $2 \log \ell(w)$)

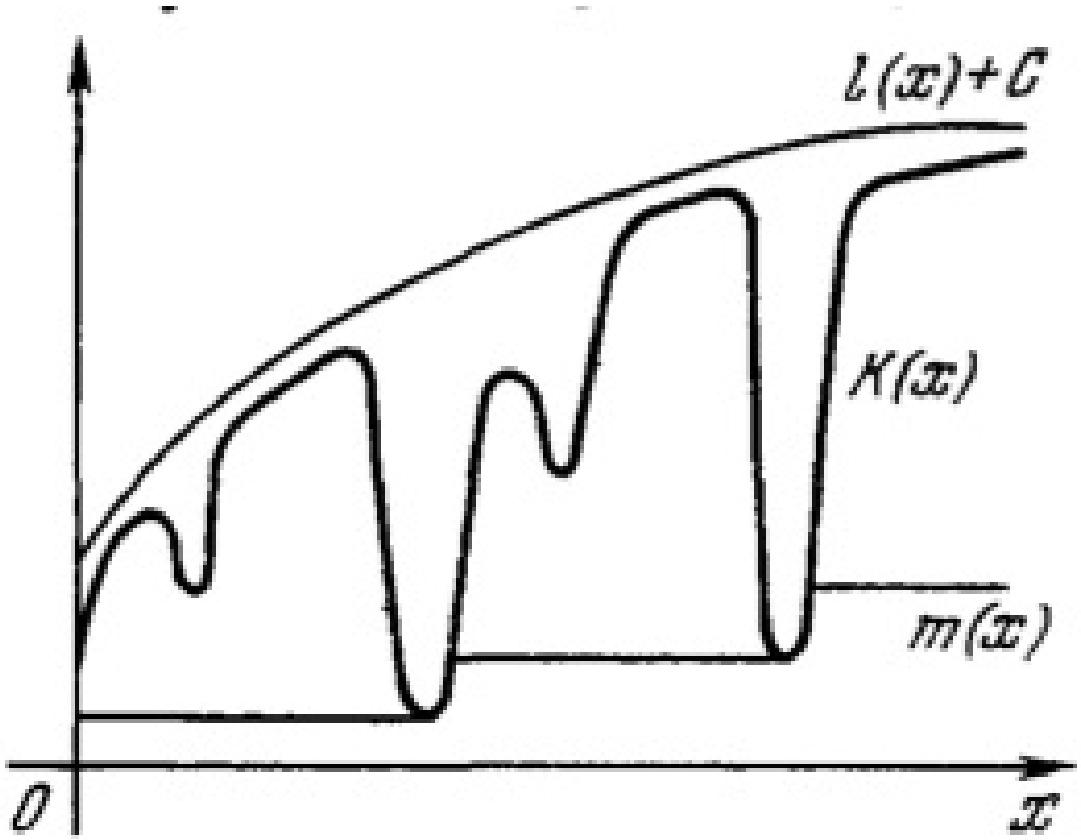
- any program that produces a description of w is an upper bound on Kolmogorov complexity $\mathcal{K}_{T_{\mathcal{U}}}(w)$

- think of Kolmogorov complexity in terms of data compression
- shortest description of w is also its most compressed form
- can obtain upper bounds on Kolmogorov complexity using data compression algorithms
- Example

Language	*Bits*	*Language*	*Bits*
Danish	7,159,576	Dutch	8,182,264
English	6,895,608	French	8,240,232
German	8,039,792	Haitian Creole	7,298,360
Hungarian	8,163,704	Icelandic	7,953,120
Italian	9,049,912	Latin	7,887,288
Maori	7,064,968	Spanish	7,412,232
Swedish	7,597,400		

upper bounds on Kolmogorov complexities of the same text (bible) in 13 different languages, based on compression algorithm `bzip2`

- finding upper bounds is easy... but NOT lower bounds



with $m(x) = \min_{y \geq x} \mathcal{K}(y)$

Morphological Complexity of languages

• `Linguistica` construct lists of stems and various possible affixes that occur in corpora of text

• selects "simplest model" as shortest description length ℓ over all stems, suffixes, and signatures

• Kolmogorov complexity of the language morphology is estimated from above by

$$\frac{\ell(\text{affixes}) + \ell(\text{signatures})}{\ell(\text{affixes}) + \ell(\text{signatures}) + \ell(\text{stems})}$$

Morphological complexity

... *are all languages equally complex*?

Example: Morphological complexity estimates for 20 languages

Language	*Metric*	*Language*	*Metric*
Latin	35.51%	English	16.88%
Hungarian	33.98%	Maori	13.62%
Italian	28.34%	Papiementu	10.16%
Spanish	27.50%	Nigerian Pidgin	9.80%
Icelandic	26.54%	Tok Pisin	8.93%
French	23.05%	Bislama	5.38%
Danish	22.86%	Kituba	3.40%
Swedish	21.85%	Solomon Pijin	2.91%
German	20.40%	Haitian Creole	2.58%
Dutch	19.58%	Vietnamese	0.05%

... adversus solem ne loquitor

- Reference: Max Bane, *Quantifying and Measuring Morphological complexity*, Proceedings of the 26th West Coast Conference on Formal Linguistics, 67–76, 2008.
- `http://linguistica.uchicago.edu`

Main problem

Kolmogorov complexity is NOT a computable function

- suppose list programs P_k (increasing lengths) and run through $T_{\mathcal{U}}$: if machine halts on P_k with output w then $\ell(P_k)$ is an upper bound on $\mathcal{K}_{T_{\mathcal{U}}}(w)$
- but... there can be an earlier P_j in the list such that $T_{\mathcal{U}}$ has not yet halted on P_j
- if eventually halts and outputs w then $\ell(P_j)$ is a better approximation to $\mathcal{K}_{T_{\mathcal{U}}}(w)$
- would be able to compute $\mathcal{K}_{T_{\mathcal{U}}}(w)$ if can tell exactly on which programs P_k the machine $T_{\mathcal{U}}$ halts
- but... halting problem is unsolvable

Kolmogorov Complexity and Entropy

- Independent random variables X_k distributed according to Bernoulli measure $\mathbb{P} = \{p_a\}_{a \in \mathfrak{A}}$ with $p_a = \mathbb{P}(X = a)$
- Shannon entropy $S(X) = -\sum_{a \in \mathfrak{A}} \mathbb{P}(X = a) \log \mathbb{P}(X = a)$
- $\exists c > 0$ such that for all $n \in \mathbb{N}$

$$S(X) \leq \frac{1}{n} \sum_{w \in \mathcal{W}^n} \mathbb{P}(w) \mathcal{K}(w \,|\, \ell(w) = n) \leq S(X) + \frac{\#\mathfrak{A} \log n}{n} + \frac{c}{n}$$

- expectaction value

$$\lim_{n \to \infty} \mathbb{E}(\frac{1}{n} \mathcal{K}(X_1 \cdots X_n \,|\, n)) = S(X)$$

average expected Kolmogorov complexity for length n descriptions approaches Shannon entropy

Kraft inequality for prefix-free codes

- prefix-free codes (prefix codes): code where no code word is a prefix of another code word (self-punctuating codes)
- Kraft inequality for prefix-free codes:

prefix code in an alphabet $\mathfrak{A}$ of size $N = \#\mathfrak{A}$; lengths of code words $\ell(w_1), \ldots \ell(w_m)$

$$\sum_{i=1}^{m} D^{-\ell(w_i)} \leq 1$$

and any such inequality is realized by lengths of code words of some prefix-free code

- Relation between optimal encoding and Shannon entropy

$$S_D(X) \leq \sum_{i=1}^{m} \mathbb{P}(w_i)\ell(w_i) \leq S_D(X) + 1$$

for $D = \#\mathfrak{A}$ and $S_D =$ Shannon entropy with $\log_D$ with $w_1, \ldots, w_m$ code words of optimal lengths for a source X randomly distributed according to Bernoulli $\mathbb{P} = \{p_a\}$

Why Kraft inequality?

- Main observation: a set of prefix-free binary code words corresponds to a binary tree and oriented paths from the root to one of the leaves (0 = turn right, 1 = turn left at the next node)
- for simplest tree with only one step equality $\frac{1}{2} + \frac{1}{2} = 1$
- for other binary trees, Kraft inequality proved inductively over subtrees: isolating root and first subsequent nodes
- Shannon entropy estimate from Kraft inequality

$$S(X) - \sum_{i=1}^{m} \mathbb{P}(w_i)\ell(w_i) \leq \sum_i \mathbb{P}(w_i) \log_2(\frac{2^{-\ell(w_i)}}{\mathbb{P}(w_i)})$$

$$= \log_2(e) \sum_i \mathbb{P}(w_i) \log(\frac{2^{-\ell(w_i)}}{\mathbb{P}(w_i)}) \leq \log_2(e) \sum_i \mathbb{P}(w_i)(\frac{2^{-\ell(w_i)}}{\mathbb{P}(w_i)} - 1) \leq 0$$

using $\log(x) \leq x - 1$ and Kraft inequality

Kraft inequality for Turing machines

- prefix-free Turing machine: programs on which it halts are prefix-free codes (unidirectional input/output tapes, bidirectional work tapes...)
- universal prefix-free Turing machine $T_{\mathcal{U}}$
- encode programs P using a prefix-free (binary) code
- Kraft inequality

$$\sum_{P\,:\,T_{\mathcal{U}}(P)\text{ halts}} 2^{-\ell(P)} \leq 1$$

- Universal (Sub)Probability

$$\mathbb{P}_{T_{\mathcal{U}}}(w) = \sum_{P\,:\,T_{\mathcal{U}}(P)=w} 2^{-\ell(P)} = \mathbb{P}(T_{\mathcal{U}}(P) = w)$$

over an ensemble of randomly drawn programs (expressed by binary codes) most don't halt (or crash) but some halt and output w

Levin's Probability Distribution

- prefix-free Kolmogorov complexity

$$\mathcal{KP}_{T_\mathcal{U}}(x) = \min_{P\,:\,T_\mathcal{U}(P)=x} \ell(P)$$

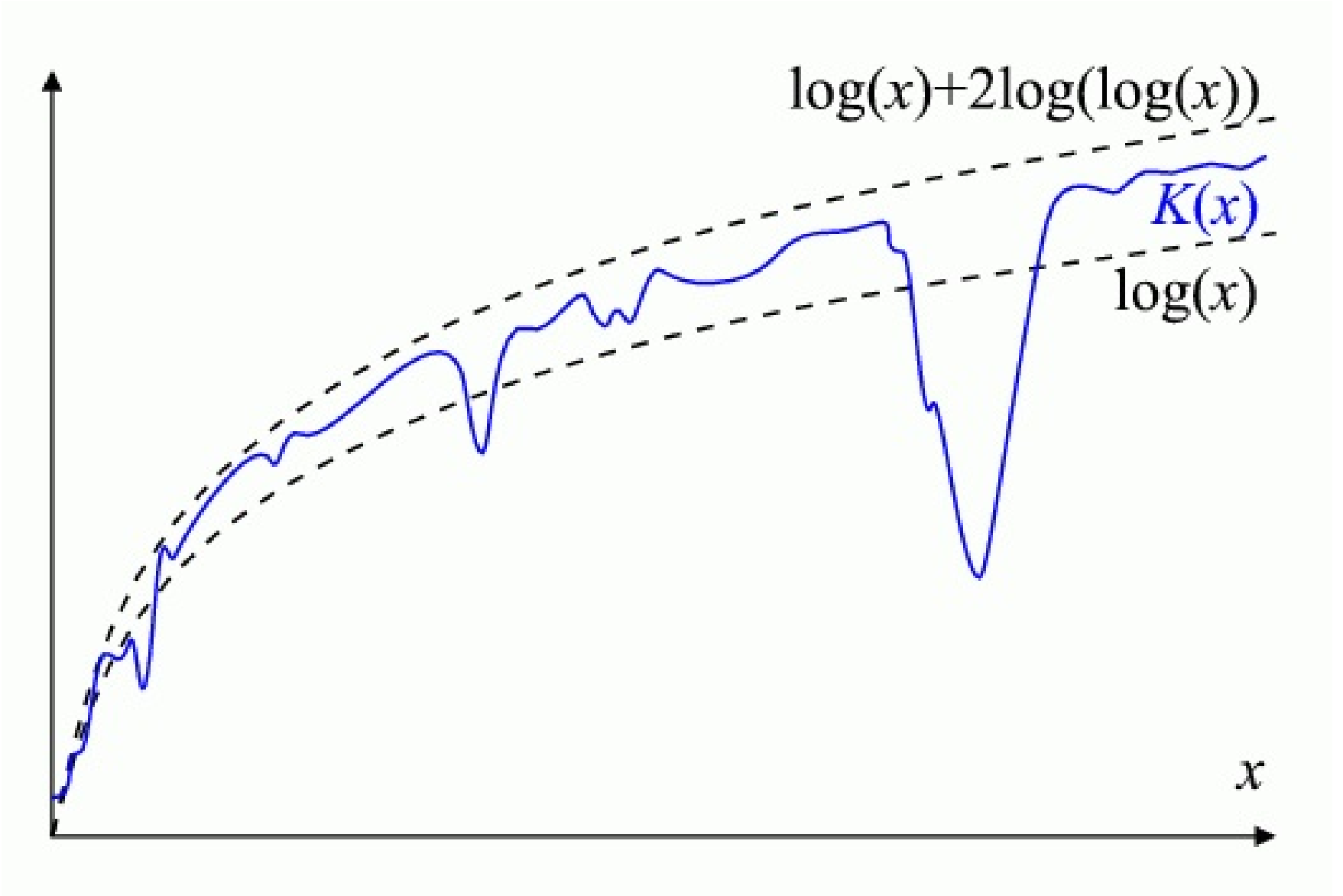
$T_\mathcal{U}$ = universal prefix-free Turing machine

- Relation of universal measure to Kolmogorov complexity:

$$\mathbb{P}_{T_\mathcal{U}}(w) \sim 2^{-\mathcal{KP}_{T_\mathcal{U}}(w)}$$

- dominance of shortest program
  - L.A. Levin, *Various measures of complexity for finite objects (axiomatic description)*, Soviet Math. Dokl., Vol.17 (1976) N.2, 522–526.
  - A.K. Zvonkin, L.A. Levin, *Complexity of finite objects and the development of the concept of information and randomness by means of the theory of algorithms*, Uspehi Mat. Nauk, Vol.25 (1970) no. 6(156), 85–127.

behavior of prefix-free Kolmogorov complexity



Syntactic Parameters and Complexity

- Languages encoded by the binary string of $\pm$ values (or 0/1 values) of their syntactic parameters
- For N parameters, the space of possible languages would contain 2^N points: number of actual languages much smaller
- Language acquisition, from this perspective should include a mechanism that identifies this subset
- *parameter expression*: a sentence that requires a certain syntactic parameter to be set to one or the other value in order to be able to assigned a well formed representation
- expect that a notion of complexity can be assigned to syntactic parameters based on the complexity of the syntactic structure of phrases that express that parameter

Complexity of Parse Trees

- Binary coding of all the nonterminal symbols in V_N
- Binary coding of the topological structure of the planar binary rooted tree
- Combine to get binary coding of all parse trees of a grammar
- Kolmogorov Complexity of Parse Trees: $\mathcal{K}_{T_{\mathcal{U}}}(\tau)$
- Estimated by compression algorithms on binary codings

Parameter Complexity

- Syntactic parameter Π with possible values $v \in \{\pm 1\}$
- The Kolmogorov Complexity of a parameter Π set to value v is

$$\mathcal{K}(\Pi = v) = \min_{\tau \text{ expressing } \Pi} \mathcal{K}_{T_{\mathcal{U}}}(\tau)$$

minimum of complexities of all the trees that express the syntactic parameter Π and require $\Pi = v$ to be grammatical in the language

- Reference: Robin Clark, *Kolmogorov complexity and the information content of parameters*, Technical Report. Philadelphia, Institute for Research in Cognitive Science, University of Pennsylvania, 1994.

... more subtle question: *are all languages equally complex in terms of parameter complexity?*
(languages very low in morphological complexity can be very high in syntactic complexity)

Minimum Description Length for Languages

- Within Principles and Paramaters model: assume there are N parameters
- Model space $\mathcal{A}$ cardinality 2^N
- Complexity of describing some data set X given $\mathcal{A}$: complexity of describing $\mathcal{A}$ plus the N bits required to select the model parameters for X
- Prefix complexity

$$\mathcal{K}(X \mid Y) = \min_{P: T_{\mathcal{U}}(P,Y)=X} \ell(P)$$

- consider all sets $\mathcal{A}$ with $\mathcal{K}(\mathcal{A}) + \log_2(\#\mathcal{A}) = \mathcal{K}(X)$
- minimal sufficient statistics for X is $\mathcal{A}_{\min}$ with minimal prefix complexity $\mathcal{K}(X \mid \mathcal{A})$

Some Difficulties in applying this principle to languages

- not a uniform encoding system for grammars
- Example: Pāṇini and the *anuvṛtti* encoding of "generative rules"
- Example: attribute value matrices (AVM) in Head-driven Phrase Structure Grammars (HPSG)
- also not clear generative rules are always a "shorter description"
- Fact: for every $n \in \mathbb{N}$ there is a language $\mathcal{L}_n$ containing $n^2 - n$ strings, which requires at least $O(n^2/\log(n))$ production rules in a grammar to generate
- Reference: Z. Tuza, *On the context-free production complexity of finite languages*, Discrete Applied Mathematics, 18 (1987) 293–304.

Gell-Mann Effective Complexity

- unlike Kolmogorov complexity does not measure description length of whole object
- based on description length of "regularities" (structured patterns) contained in the object
- a completely random sequence has maximal Kolmogorov complexity but zero effective complexity (it contains no structured patterns)
- for languages distinguish system complexity from structural complexity: grammatical complexity is complexity of the grammar versus complexity of the expressions in the language
- use of notions of effective complexity applied to phonetics, morphology and syntax of world languages currently provides a good challenge to the "equal complexity" idea

Zipf's Law $\mathcal{C}$ = a corpus of texts

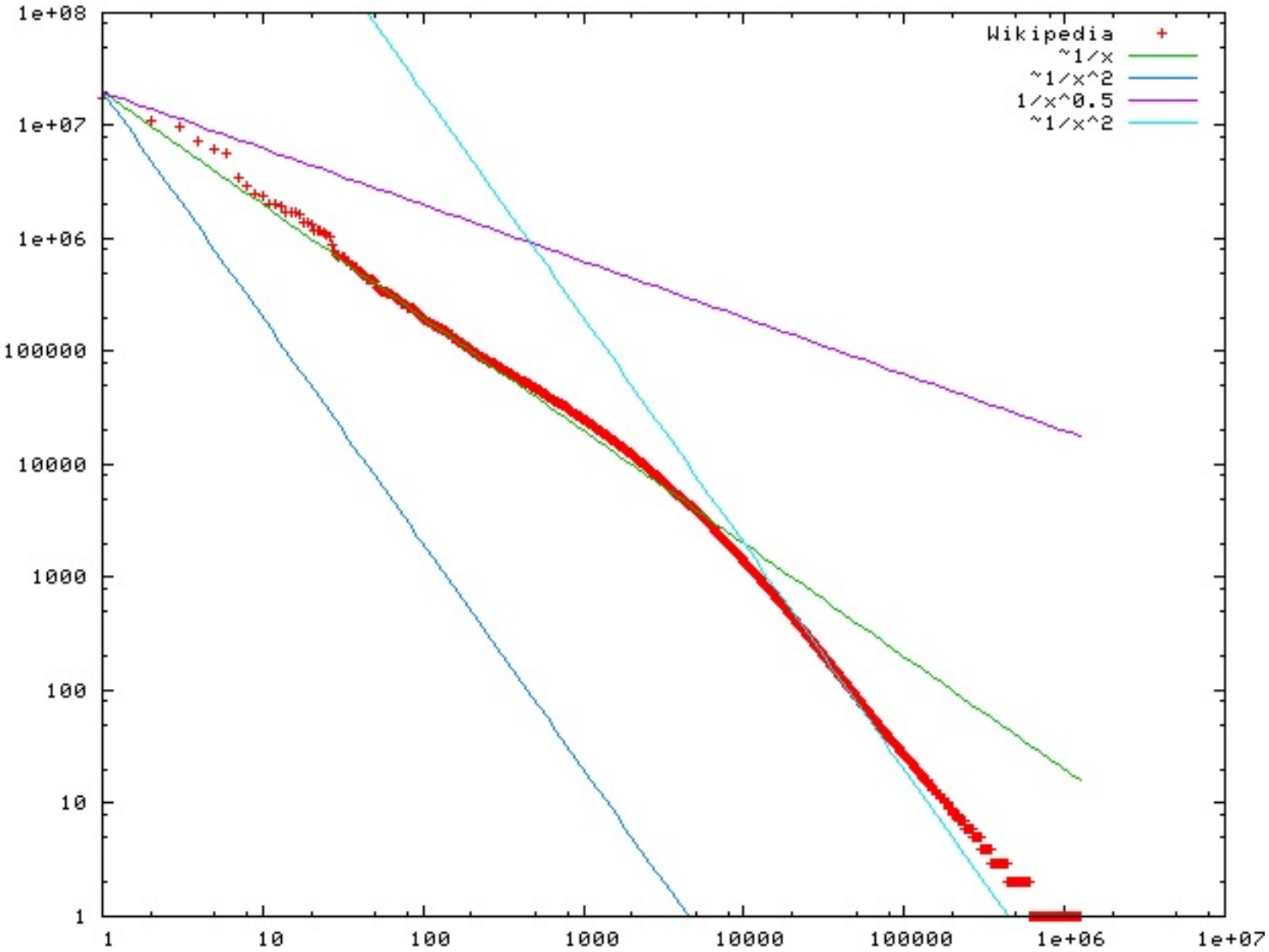
- N = size of corpus $\mathcal{C}$ (total number of words)
- word token (include capital letters, punctuation, special characters...); word type (ignoring all those extra features)
- absolute frequence $F_{\mathcal{C}}(w)$ = number of tokens for a given word type over corpus; relative frequence $F_{\mathcal{C}}(w)/N$
- list word types w_k by decreasing frequencies
- Zipf's law (experimental observation)

$$\log(F_{\mathcal{C}}(w_k)) = \kappa(N) - B\log(k)$$

the exponent B of the power law is close to $B \sim 1$

- What is a theoretical explanation?

Zipf's Law based on Wikipedia text



significant deviation from the $1/x$ law at both ends

Proposed explanations

• uniform generation process: new words are introduced by a small constant probability and old words keep being used with the same probability as before
- H.A. Simon, *On a class of skew distribution functions*, Biometrika 42 (1955) 425–440.

• random generation: (monkey at typewriter) choose symbol (blank space) and define word as string on all symbols that does not contain fixed one; generate next symbols by Bernoulli process (or Markov process, works too): obtain a Zipf's law with $B > 1$
- Benoit Mandelbrot, *An information theory of the structure of language based upon the theory of the statistical matching of messages and coding*, Second Symposium on Information Theory, pp. 486–500, 1952

• there followed a heated Mandelbrot–Simon debate that dragged on for many years

Consequences of Zipf's law

- for corpora with Zipf's law with exponent B vocabulary size $V(N) \sim cN^{1/B}$
- for corpora with Zipf's law with B: second Zipf's law for number $V(\ell, N)$ of word types that occur ℓ times

$$\log(\ell) = K(N) - D \log V(\ell, N), \qquad \text{with} \qquad D = \frac{B}{1+B}$$

- cumulative probability $\mathbb{P}_k$ of the k most frequent words

$$1 - \mathbb{P}_k = C_k \sum_{r=k+1}^{N^{1/B}} r^{-B} \sim \frac{C_k}{1-B}(N^{\frac{1-B}{B}} - k^{1-B})$$

so that $C_k \sim (1 - P_k)(B-1)k^{B-1}$ for large N

Zipf's law and Entropy

$$S = -\sum_{r=1}^{k} p_r \log_2(p_r) - \sum_{r=k+1}^{N^{1/B}} p_r \log_2(p_r)$$

approximate second sum as before

$$S = -S_k + \frac{1-\mathbb{P}_k}{\log(2)}\left(\frac{B}{B-1} - \log(B-1) + \log(k) - \log(1-\mathbb{P}_k)\right)$$

- several of these consequences are problematic for the critical value $B = 1$
- Mandelbrot argued for a deviation from $B = 1$ towards $B > 1$ once first few most frequent words are removed (clearly visible effect in corpora)

Zipf's law and Kolmogorov Complexity

• Assume that rank ordering is Kolmogorov ordering (by increasing complexity)

• Assume that the probability distribution producing Zipf's law is an approximant of the universal Levin probability distribution (weight by $2^{-\mathcal{KP}(x)}$)

• Then Zipf's law with critical exponent $B = 1$ follows from properties of Kolmogorov complexity and of the Levin measure

- Yuri I. Manin, *Zipf's law and L. Levin's probability distributions*, arXiv:1301.0427v2

References

- R. Badii, A. Politi, *Complexity. Hierarchical structures and scaling in physics*, Cambridge, 1997.
- M. Li, P. Vitányi, *An introduction to Kolmogorov complexity and its applications*, Springer, 2008.
- A. Kornai, *Mathematical Linguistics*, Springer, 2010.
- G. Sampson, D. Gil, P. Trudgill (Eds.) *Language Complexity as an evolving variable*, Oxford University Press, 2009.
- G.E. Barton, R.C. Berwick, E.S. Ristad, *Computational complexity and natural language*, MIT Press, 1987.