Random noise increases Kolmogorov complexity

alexander.shen@lirmm.fr, www.lirmm.fr/~ashen,
joint work with Gleb Posobin
based on the discussions with Peter Gacs

LIRMM CNRS & University of Montpellier, supported by RaCAF

July 3, 2018, GTC2018 & RaCAF meeting

Decreasing complexity by changing bits

Decreasing complexity by changing bits

- string $x \in \mathbb{B}^n$ has some complexity $\mathrm{C}(x) < n$

Decreasing complexity by changing bits

- string $x \in \mathbb{B}^n$ has some complexity $\mathrm{C}(x) < n$
- $\mathrm{C}(x) = \alpha n$

Decreasing complexity by changing bits

- string $x \in \mathbb{B}^n$ has some complexity $\mathrm{C}(x) < n$
- $\mathrm{C}(x) = \alpha n$
- change some small fraction of bits in x

Decreasing complexity by changing bits

- string $x \in \mathbb{B}^n$ has some complexity $\mathrm{C}(x) < n$
- $\mathrm{C}(x) = \alpha n$
- change some small fraction of bits in x
- what happens with $\mathrm{C}(x)$?

Decreasing complexity by changing bits

- string $x \in \mathbb{B}^n$ has some complexity $\mathrm{C}(x) < n$
- $\mathrm{C}(x) = \alpha n$
- change some small fraction of bits in x
- what happens with $\mathrm{C}(x)$?
- may increase or decrease: how much?

Decreasing complexity by changing bits

- string $x \in \mathbb{B}^n$ has some complexity $\mathrm{C}(x) < n$
- $\mathrm{C}(x) = \alpha n$
- change some small fraction of bits in x
- what happens with $\mathrm{C}(x)$?
- may increase or decrease: how much?
- decrease: $\min\{\mathrm{C}(y)\colon d(x,y) \leqslant \tau n\}$ as a function of τ

Decreasing complexity by changing bits

- string $x \in \mathbb{B}^n$ has some complexity $\mathrm{C}(x) < n$
- $\mathrm{C}(x) = \alpha n$
- change some small fraction of bits in x
- what happens with $\mathrm{C}(x)$?
- may increase or decrease: how much?
- decrease: $\min\{\mathrm{C}(y) \colon d(x,y) \leqslant \tau n\}$ as a function of τ
  $d(x,y)$: the Hamming distance (the number of changed bits)

Decreasing complexity by changing bits

- string $x \in \mathbb{B}^n$ has some complexity $\mathrm{C}(x) < n$
- $\mathrm{C}(x) = \alpha n$
- change some small fraction of bits in x
- what happens with $\mathrm{C}(x)$?
- may increase or decrease: how much?
- decrease: $\min\{\mathrm{C}(y) : d(x, y) \leqslant \tau n\}$ as a function of τ
  $d(x, y)$: the Hamming distance (the number of changed bits)
- τn-balls: what is the complexity of their simplest elements?

Decreasing complexity by changing bits

- string $x \in \mathbb{B}^n$ has some complexity $\mathrm{C}(x) < n$
- $\mathrm{C}(x) = \alpha n$
- change some small fraction of bits in x
- what happens with $\mathrm{C}(x)$?
- may increase or decrease: how much?
- decrease: $\min\{\mathrm{C}(y) \colon d(x, y) \leqslant \tau n\}$ as a function of τ
  $d(x, y)$: the Hamming distance (the number of changed bits)
- τn-balls: what is the complexity of their simplest elements?
- depends not only on $\mathrm{C}(x)$, but on the properties of x

Decreasing complexity by changing bits

- string $x \in \mathbb{B}^n$ has some complexity $\mathrm{C}(x) < n$
- $\mathrm{C}(x) = \alpha n$
- change some small fraction of bits in x
- what happens with $\mathrm{C}(x)$?
- may increase or decrease: how much?
- decrease: $\min\{\mathrm{C}(y) \colon d(x, y) \leqslant \tau n\}$ as a function of τ
  $d(x, y)$: the Hamming distance (the number of changed bits)
- τn-balls: what is the complexity of their simplest elements?
- depends not only on $\mathrm{C}(x)$, but on the properties of x
- algorithmic statistics for restricted families of models (Vereshchagin, Vitanyi) tells us what functions are possible

Decreasing complexity by changing bits

- string $x \in \mathbb{B}^n$ has some complexity $\mathrm{C}(x) < n$
- $\mathrm{C}(x) = \alpha n$
- change some small fraction of bits in x
- what happens with $\mathrm{C}(x)$?
- may increase or decrease: how much?
- decrease: $\min\{\mathrm{C}(y) \colon d(x, y) \leqslant \tau n\}$ as a function of τ
  $d(x, y)$: the Hamming distance (the number of changed bits)
- τn-balls: what is the complexity of their simplest elements?
- depends not only on $\mathrm{C}(x)$, but on the properties of x
- algorithmic statistics for restricted families of models (Vereshchagin, Vitanyi) tells us what functions are possible
- [random bits]$000\ldots000$

Decreasing complexity by changing bits

- string $x \in \mathbb{B}^n$ has some complexity $\mathrm{C}(x) < n$
- $\mathrm{C}(x) = \alpha n$
- change some small fraction of bits in x
- what happens with $\mathrm{C}(x)$?
- may increase or decrease: how much?
- decrease: $\min\{\mathrm{C}(y) \colon d(x,y) \leqslant \tau n\}$ as a function of τ
  $d(x,y)$: the Hamming distance (the number of changed bits)
- τn-balls: what is the complexity of their simplest elements?
- depends not only on $\mathrm{C}(x)$, but on the properties of x
- algorithmic statistics for restricted families of models (Vereshchagin, Vitanyi) tells us what functions are possible
- [random bits]$000\ldots000$
- random codeword: no decrease

Increasing complexity by changing bits

Increasing complexity by changing bits

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$

Increasing complexity by changing bits

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing τ-fraction of bits: $d(x, y) \leqslant \tau n$

Increasing complexity by changing bits

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing τ-fraction of bits: $d(x, y) \leqslant \tau n$
- is it always possible to increase complexity?

Increasing complexity by changing bits

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing τ-fraction of bits: $d(x, y) \leqslant \tau n$
- is it always possible to increase complexity?
- $\tau \mapsto \max\{\mathrm{C}(y) \colon d(x, y) \leqslant \tau n\}$

Increasing complexity by changing bits

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing τ-fraction of bits: $d(x,y) \leqslant \tau n$
- is it always possible to increase complexity?
- $\tau \mapsto \max\{\mathrm{C}(y)\colon d(x,y) \leqslant \tau n\}$
- Buhrman, Fortnow, Newman, Vereshchagin: $\Omega(n)$ increase is always possible

Increasing complexity by changing bits

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing τ-fraction of bits: $d(x, y) \leqslant \tau n$
- is it always possible to increase complexity?
- $\tau \mapsto \max\{\mathrm{C}(y) \colon d(x, y) \leqslant \tau n\}$
- Buhrman, Fortnow, Newman, Vereshchagin: $\Omega(n)$ increase is always possible
- the amount of increase depends on x

Increasing complexity by changing bits

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing τ-fraction of bits: $d(x, y) \leqslant \tau n$
- is it always possible to increase complexity?
- $\tau \mapsto \max\{\mathrm{C}(y) \colon d(x, y) \leqslant \tau n\}$
- Buhrman, Fortnow, Newman, Vereshchagin: $\Omega(n)$ increase is always possible
- the amount of increase depends on x
- open question: what functions can appear here?

Increasing complexity by changing bits

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing τ-fraction of bits: $d(x, y) \leqslant \tau n$
- is it always possible to increase complexity?
- $\tau \mapsto \max\{\mathrm{C}(y) \colon d(x, y) \leqslant \tau n\}$
- Buhrman, Fortnow, Newman, Vereshchagin: $\Omega(n)$ increase is always possible
- the amount of increase depends on x
- open question: what functions can appear here?
- maximal possible increase for random codewords

Increasing complexity by changing bits

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing τ-fraction of bits: $d(x, y) \leqslant \tau n$
- is it always possible to increase complexity?
- $\tau \mapsto \max\{\mathrm{C}(y) \colon d(x, y) \leqslant \tau n\}$
- Buhrman, Fortnow, Newman, Vereshchagin: $\Omega(n)$ increase is always possible
- the amount of increase depends on x
- open question: what functions can appear here?
- maximal possible increase for random codewords
- BFNV: minimal possible increase for Bernoulli random strings

Increasing complexity by changing bits

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing τ-fraction of bits: $d(x, y) \leqslant \tau n$
- is it always possible to increase complexity?
- $\tau \mapsto \max\{\mathrm{C}(y) \colon d(x, y) \leqslant \tau n\}$
- Buhrman, Fortnow, Newman, Vereshchagin: $\Omega(n)$ increase is always possible
- the amount of increase depends on x
- open question: what functions can appear here?
- maximal possible increase for random codewords
- BFNV: minimal possible increase for Bernoulli random strings
- combinatorial tool: Harper's theorem (Hamming balls have minimal neighborhoods)

Random change: what happens with complexity?

Random change: what happens with complexity?

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$

Random change: what happens with complexity?

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing a random τ-fraction of bits

Random change: what happens with complexity?

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing a random τ-fraction of bits
- better: each bit changed with probability τ independently

Random change: what happens with complexity?

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing a random τ-fraction of bits
- better: each bit changed with probability τ independently
- $N_\tau(x)$: noise of intensity τ added to x

Random change: what happens with complexity?

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing a random τ-fraction of bits
- better: each bit changed with probability τ independently
- $N_\tau(x)$: noise of intensity τ added to x
  $N_\tau(x) = x \oplus B_\tau$ where B_τ is a Bernoulli distribution with parameter τ

Random change: what happens with complexity?

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing a random τ-fraction of bits
- better: each bit changed with probability τ independently
- $N_\tau(x)$: noise of intensity τ added to x
  $N_\tau(x) = x \oplus B_\tau$ where B_τ is a Bernoulli distribution with parameter τ
- "random noise": probabilistic, not algorithmic randomness

Random change: what happens with complexity?

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing a random τ-fraction of bits
- better: each bit changed with probability τ independently
- $N_\tau(x)$: noise of intensity τ added to x
  $N_\tau(x) = x \oplus B_\tau$ where B_τ is a Bernoulli distribution with parameter τ
- "random noise": probabilistic, not algorithmic randomness
- $\mathrm{C}(N_\tau(x))$: a random variable

Random change: what happens with complexity?

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing a random τ-fraction of bits
- better: each bit changed with probability τ independently
- $N_\tau(x)$: noise of intensity τ added to x
  $N_\tau(x) = x \oplus B_\tau$ where B_τ is a Bernoulli distribution with parameter τ
- "random noise": probabilistic, not algorithmic randomness
- $\mathrm{C}(N_\tau(x))$: a random variable
- concentration inequalities: for every x this random variable has some typical value

Random change: what happens with complexity?

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing a random τ-fraction of bits
- better: each bit changed with probability τ independently
- $N_\tau(x)$: noise of intensity τ added to x
  $N_\tau(x) = x \oplus B_\tau$ where B_τ is a Bernoulli distribution with parameter τ
- "random noise": probabilistic, not algorithmic randomness
- $\mathrm{C}(N_\tau(x))$: a random variable
- concentration inequalities: for every x this random variable has some typical value
- some increase in complexity guaranteed with high probability

Random change: what happens with complexity?

- $x \in \mathbb{B}^n$, $\mathrm{C}(x) = \alpha n$
- changing a random τ-fraction of bits
- better: each bit changed with probability τ independently
- $N_\tau(x)$: noise of intensity τ added to x
  $N_\tau(x) = x \oplus B_\tau$ where B_τ is a Bernoulli distribution with parameter τ
- "random noise": probabilistic, not algorithmic randomness
- $\mathrm{C}(N_\tau(x))$: a random variable
- concentration inequalities: for every x this random variable has some typical value
- some increase in complexity guaranteed with high probability
- exact lower bound for this increase

Complexity increases with high probability

Theorem

Let $\alpha \in (0,1)$ and $\tau \in (0,1/2)$. There exists some $\beta > \alpha$ with the following property:

$$\mathrm{C}(x) \geqslant \alpha n \Rightarrow \Pr[\mathrm{C}(N_\tau(x)) \geqslant \beta n] \geqslant 1 - \frac{1}{n}$$

for sufficiently large n and for every x of length n

Complexity increases with high probability

Theorem

Let $\alpha \in (0,1)$ and $\tau \in (0,1/2)$. There exists some $\beta > \alpha$ with the following property:

$$\mathrm{C}(x) \geqslant \alpha n \Rightarrow \Pr[\mathrm{C}(N_\tau(x)) \geqslant \beta n] \geqslant 1 - \frac{1}{n}$$

for sufficiently large n and for every x of length n

regime: α, β and τ are fixed, $n \to \infty$

Complexity increases with high probability

Theorem

Let $\alpha \in (0,1)$ and $\tau \in (0,1/2)$. There exists some $\beta > \alpha$ with the following property:

$$\mathrm{C}(x) \geqslant \alpha n \Rightarrow \Pr[\mathrm{C}(N_\tau(x)) \geqslant \beta n] \geqslant 1 - \frac{1}{n}$$

for sufficiently large n and for every x of length n

regime: α, β and τ are fixed, $n \to \infty$
β is some function of α and τ

Complexity increases with high probability

Theorem

Let $\alpha \in (0,1)$ and $\tau \in (0,1/2)$. There exists some $\beta > \alpha$ with the following property:

$$\mathrm{C}(x) \geqslant \alpha n \Rightarrow \Pr[\mathrm{C}(N_\tau(x)) \geqslant \beta n] \geqslant 1 - \frac{1}{n}$$

for sufficiently large n and for every x of length n

regime: α, β and τ are fixed, $n \to \infty$

β is some function of α and τ

different combinatorial arguments possible

(Fourier analysis, hypercontractivity inequalities)

Complexity increases with high probability

Theorem

Let $\alpha \in (0,1)$ and $\tau \in (0,1/2)$. There exists some $\beta > \alpha$ with the following property:

$$\mathrm{C}(x) \geqslant \alpha n \Rightarrow \Pr[\mathrm{C}(N_\tau(x)) \geqslant \beta n] \geqslant 1 - \frac{1}{n}$$

for sufficiently large n and for every x of length n

regime: α, β and τ are fixed, $n \to \infty$
β is some function of α and τ
different combinatorial arguments possible
(Fourier analysis, hypercontractivity inequalities)
but they do not give an optimal bound for β

Complexity increases with high probability

Theorem

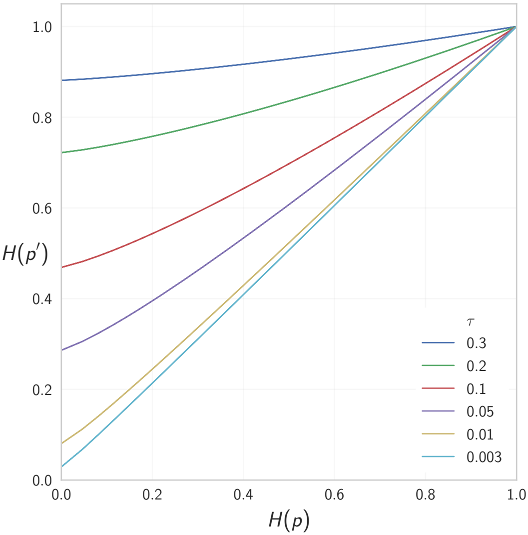
Let $\alpha \in (0,1)$ and $\tau \in (0,1/2)$. There exists some $\beta > \alpha$ with the following property:

$$\mathrm{C}(x) \geqslant \alpha n \Rightarrow \Pr[\mathrm{C}(N_\tau(x)) \geqslant \beta n] \geqslant 1 - \frac{1}{n}$$

for sufficiently large n and for every x of length n

regime: α, β and τ are fixed, $n \to \infty$
β is some function of α and τ
different combinatorial arguments possible
(Fourier analysis, hypercontractivity inequalities)
but they do not give an optimal bound for β
$1/n$ can be replaced by $1/n^d$ for arbitrary fixed d

Optimal lower bound for the complexity increase



The complexity of B_τ

The complexity of B_τ

- $N_\tau(0^n) = B_\tau$

The complexity of B_τ

- $N_\tau(0^n) = B_\tau$
- $\approx$ complexity of random string of length n with τn ones

The complexity of B_τ

- $N_\tau(0^n) = B_\tau$
- $\approx$ complexity of random string of length n with τn ones
- log(number of strings of length n with τn ones)

The complexity of B_τ

- $N_\tau(0^n) = B_\tau$
- $\approx$ complexity of random string of length n with τn ones
- log(number of strings of length n with τn ones)
- $\log\binom{n}{\tau n} = 2^{H(\tau)n}$, where

$$H(p) = p\log\frac{1}{p} + (1-p)\log\frac{1}{1-p}$$

is the Shannon entropy of for the $(p, 1-p)$ distribution

The complexity of B_τ

- $N_\tau(0^n) = B_\tau$
- $\approx$ complexity of random string of length n with τn ones
- log(number of strings of length n with τn ones)
- $\log \binom{n}{\tau n} = 2^{H(\tau)n}$, where

$$H(p) = p \log \frac{1}{p} + (1-p) \log \frac{1}{1-p}$$

  is the Shannon entropy of for the $(p, 1-p)$ distribution
- if B_p is a Bernoulli random string with probability p, then
  $N_\tau(B_p) = B_{N(p,\tau)}$
  $N(p,\tau) = p(1-\tau) + (1-p)\tau$

The complexity of B_τ

- $N_\tau(0^n) = B_\tau$
- $\approx$ complexity of random string of length n with τn ones
- log(number of strings of length n with τn ones)
- $\log \binom{n}{\tau n} = 2^{H(\tau)n}$, where

$$H(p) = p \log \frac{1}{p} + (1-p)\log \frac{1}{1-p}$$

  is the Shannon entropy of for the $(p, 1-p)$ distribution
- if B_p is a Bernoulli random string with probability p, then
  $N_\tau(B_p) = B_{N(p,\tau)}$
  $N(p,\tau) = p(1-\tau) + (1-p)\tau$
- complexity increase $H(p) \mapsto H(N(p,\tau))$ for Bernoulli random strings

Complexity increases with high probability: optimal bound

Theorem

Let $p \in (0, 1/2)$ *and* $\tau \in (0, 1/2)$*.*
Let $\alpha = H(p)$ *and* $\beta = H(N(\tau, p))$*. Then*

$$\mathrm{C}(x) \geqslant \alpha n \Rightarrow \Pr[\mathrm{C}(N_\tau(x)) \geqslant \beta n - o(n)] \geqslant 1 - \frac{1}{n}$$

for $n \to \infty$ *and for every* x *of length* n*. This* β *is the best possible bound.*

Complexity increases with high probability: optimal bound

Theorem

Let $p \in (0, 1/2)$ and $\tau \in (0, 1/2)$.
Let $\alpha = H(p)$ and $\beta = H(N(\tau, p))$. Then

$$\mathrm{C}(x) \geqslant \alpha n \Rightarrow \Pr[\mathrm{C}(N_\tau(x)) \geqslant \beta n - o(n)] \geqslant 1 - \frac{1}{n}$$

for $n \to \infty$ and for every x of length n. This β is the best possible bound.

Remark: for some strings (e.g., random codewords) we have better bounds, but the lower bound is optimal: one cannot improve β *for all strings*

Three approaches to measuring information

Three approaches to measuring information

- Kolmogorov (1965): combinatorial, algorithmic, probabilistic

Three approaches to measuring information

- Kolmogorov (1965): combinatorial, algorithmic, probabilistic
- combinatorial: an element of a set of size N has $\log N$ bits of information

Three approaches to measuring information

- Kolmogorov (1965): combinatorial, algorithmic, probabilistic
- combinatorial: an element of a set of size N has $\log N$ bits of information
- algorithmic: $\mathrm{C}(x)$, the minimal length of a program that produces x

Three approaches to measuring information

- Kolmogorov (1965): combinatorial, algorithmic, probabilistic
- combinatorial: an element of a set of size N has $\log N$ bits of information
- algorithmic: $\mathrm{C}(x)$, the minimal length of a program that produces x
- probabilistic: the Shannon entropy

Three approaches to measuring information

- Kolmogorov (1965): combinatorial, algorithmic, probabilistic
- combinatorial: an element of a set of size N has $\log N$ bits of information
- algorithmic: $\mathrm{C}(x)$, the minimal length of a program that produces x
- probabilistic: the Shannon entropy
- measures applied to different things (sets, strings, random variables) but they are deeply connected and this is our main tool

Three approaches to measuring information

- Kolmogorov (1965): combinatorial, algorithmic, probabilistic
- combinatorial: an element of a set of size N has $\log N$ bits of information
- algorithmic: $\mathrm{C}(x)$, the minimal length of a program that produces x
- probabilistic: the Shannon entropy
- measures applied to different things (sets, strings, random variables) but they are deeply connected and this is our main tool
- Buhrman et al. result uses the connection between combinatorial and algorithmic approaches

Three approaches to measuring information

- Kolmogorov (1965): combinatorial, algorithmic, probabilistic
- combinatorial: an element of a set of size N has $\log N$ bits of information
- algorithmic: $\mathrm{C}(x)$, the minimal length of a program that produces x
- probabilistic: the Shannon entropy
- measures applied to different things (sets, strings, random variables) but they are deeply connected and this is our main tool
- Buhrman et al. result uses the connection between combinatorial and algorithmic approaches
- we need all three

Buhrman et al. result revisited

Buhrman et al. result revisited

- (complexity version) for every string length n at complexity $\geqslant \alpha n$ one can change at most τn bits to get a string of complexity $\geqslant \beta n$

Buhrman et al. result revisited

- (complexity version) for every string length n at complexity $\geqslant \alpha n$ one can change at most τn bits to get a string of complexity $\geqslant \beta n$
- (combinatorial version) for every set of size at most $2^{\beta n}$ its τn-interior is of size at most $2^{\alpha n}$

Buhrman et al. result revisited

- (complexity version) for every string length n at complexity $\geqslant \alpha n$ one can change at most τn bits to get a string of complexity $\geqslant \beta n$
- (combinatorial version) for every set of size at most $2^{\beta n}$ its τn-interior is of size at most $2^{\alpha n}$
  (reformulation) for every set of size at least $2^{\alpha n}$ its τn-neighborhood is of size at least $2^{\beta n}$.

Buhrman et al. result revisited

- (complexity version) for every string length n at complexity $\geqslant \alpha n$ one can change at most τn bits to get a string of complexity $\geqslant \beta n$
- (combinatorial version) for every set of size at most $2^{\beta n}$ its τn-interior is of size at most $2^{\alpha n}$
  (reformulation) for every set of size at least $2^{\alpha n}$ its τn-neighborhood is of size at least $2^{\beta n}$.
- d-neigborhood of a set X: all strings at distance at most d from X (union of d-balls)

Buhrman et al. result revisited

- (complexity version) for every string length n at complexity $\geqslant \alpha n$ one can change at most τn bits to get a string of complexity $\geqslant \beta n$
- (combinatorial version) for every set of size at most $2^{\beta n}$ its τn-interior is of size at most $2^{\alpha n}$
  (reformulation) for every set of size at least $2^{\alpha n}$ its τn-neighborhood is of size at least $2^{\beta n}$.
- d-neigborhood of a set X: all strings at distance at most d from X (union of d-balls)
- d-interior of a set X: all strings y that are in X together with the entire d-ball centered at y

Buhrman et al. result revisited

- (complexity version) for every string length n at complexity $\geqslant \alpha n$ one can change at most τn bits to get a string of complexity $\geqslant \beta n$
- (combinatorial version) for every set of size at most $2^{\beta n}$ its τn-interior is of size at most $2^{\alpha n}$
  (reformulation) for every set of size at least $2^{\alpha n}$ its τn-neighborhood is of size at least $2^{\beta n}$.
- d-neigborhood of a set X: all strings at distance at most d from X (union of d-balls)
- d-interior of a set X: all strings y that are in X together with the entire d-ball centered at y
- Harper's theorem: minimal neighborhoods / maximal interiors happen for Hamming balls

combinatorics ⇒ complexity

combinatorics ⇒ complexity

- assume the combinatorial version: every set of size $\leqslant 2^{\beta n}$ has interior of size at most $2^{\alpha n}$

- assume the combinatorial version: every set of size $\leqslant 2^{\beta n}$ has interior of size at most $2^{\alpha n}$
- apply it to the set X of n-bit strings of complexity less than βn

combinatorics ⇒ complexity

- assume the combinatorial version: every set of size $\leqslant 2^{\beta n}$ has interior of size at most $2^{\alpha n}$
- apply it to the set X of n-bit strings of complexity less than βn
- $\#X \leqslant 2^{\beta n}$

combinatorics ⇒ complexity

- assume the combinatorial version: every set of size $\leqslant 2^{\beta n}$ has interior of size at most $2^{\alpha n}$
- apply it to the set X of n-bit strings of complexity less than βn
- $\#X \leqslant 2^{\beta n}$
- its τn-interior has size at most $2^{\alpha n}$

combinatorics ⇒ complexity

- assume the combinatorial version: every set of size $\leqslant 2^{\beta n}$ has interior of size at most $2^{\alpha n}$
- apply it to the set X of n-bit strings of complexity less than βn
- $\#X \leqslant 2^{\beta n}$
- its τn-interior has size at most $2^{\alpha n}$
- this interior is (computably) enumerable given n, βn, τn

combinatorics ⇒ complexity

- assume the combinatorial version: every set of size $\leqslant 2^{\beta n}$ has interior of size at most $2^{\alpha n}$
- apply it to the set X of n-bit strings of complexity less than βn
- $\#X \leqslant 2^{\beta n}$
- its τn-interior has size at most $2^{\alpha n}$
- this interior is (computably) enumerable given n, βn, τn
- its elements have complexity less than $\alpha n + O(\log n)$ ($\log n$ terms are ignored)

combinatorics $\Rightarrow$ complexity

- assume the combinatorial version: every set of size $\leqslant 2^{\beta n}$ has interior of size at most $2^{\alpha n}$
- apply it to the set X of n-bit strings of complexity less than βn
- $\#X \leqslant 2^{\beta n}$
- its τn-interior has size at most $2^{\alpha n}$
- this interior is (computably) enumerable given n, βn, τn
- its elements have complexity less than $\alpha n + O(\log n)$ ($\log n$ terms are ignored)
- so a string of complexity $\geqslant \alpha n + O(\log n)$ is *not* in this interior...

- assume the combinatorial version: every set of size $\leqslant 2^{\beta n}$ has interior of size at most $2^{\alpha n}$
- apply it to the set X of n-bit strings of complexity less than βn
- $\#X \leqslant 2^{\beta n}$
- its τn-interior has size at most $2^{\alpha n}$
- this interior is (computably) enumerable given n, βn, τn
- its elements have complexity less than $\alpha n + O(\log n)$ ($\log n$ terms are ignored)
- so a string of complexity $\geqslant \alpha n + O(\log n)$ is *not* in this interior...
- i.e., it can be changed in at most τn places to get outside X, i.e., to have complexity $\geqslant \beta n$

complexity ⇒ combinatorics

complexity ⇒ combinatorics

- assume the combinatorial statement: each string of complexity $\geqslant \alpha n$ can be changed in $\leqslant \tau n$ places to get a string of complexity $\geqslant \beta n$

complexity $\Rightarrow$ combinatorics

- assume the combinatorial statement: each string of complexity $\geqslant \alpha n$ can be changed in $\leqslant \tau n$ places to get a string of complexity $\geqslant \beta n$
- assume that combinatorial statement is false: there is a set X of size $2^{\beta n}$ whose τn-interior is (much) bigger that $2^{\alpha n}$

- assume the combinatorial statement: each string of complexity $\geqslant \alpha n$ can be changed in $\leqslant \tau n$ places to get a string of complexity $\geqslant \beta n$
- assume that combinatorial statement is false: there is a set X of size $2^{\beta n}$ whose τn-interior is (much) bigger that $2^{\alpha n}$
- let X be the *first* set with this property

complexity ⇒ combinatorics

- assume the combinatorial statement: each string of complexity $\geqslant \alpha n$ can be changed in $\leqslant \tau n$ places to get a string of complexity $\geqslant \beta n$
- assume that combinatorial statement is false: there is a set X of size $2^{\beta n}$ whose τn-interior is (much) bigger that $2^{\alpha n}$
- let X be the *first* set with this property
- then all elements of X have complexity at most βn (ignore $O(\log n)$ terms)

complexity ⇒ combinatorics

- assume the combinatorial statement: each string of complexity $\geqslant \alpha n$ can be changed in $\leqslant \tau n$ places to get a string of complexity $\geqslant \beta n$
- assume that combinatorial statement is false: there is a set X of size $2^{\beta n}$ whose τn-interior is (much) bigger that $2^{\alpha n}$
- let X be the *first* set with this property
- then all elements of X have complexity at most βn (ignore $O(\log n)$ terms)
- complexity statement implies that all the elements in the τn interior have complexity at most αn

complexity ⇒ combinatorics

- assume the combinatorial statement: each string of complexity $\geqslant \alpha n$ can be changed in $\leqslant \tau n$ places to get a string of complexity $\geqslant \beta n$
- assume that combinatorial statement is false: there is a set X of size $2^{\beta n}$ whose τn-interior is (much) bigger that $2^{\alpha n}$
- let X be the *first* set with this property
- then all elements of X have complexity at most βn (ignore $O(\log n)$ terms)
- complexity statement implies that all the elements in the τn interior have complexity at most αn
- but there are too many of them: contradiction

Random noise case

Random noise case

- (Shannon information) for a distribution P on n-bit strings: if $H(P) \geqslant \alpha n$, then $H(N_\tau(P)) \geqslant \beta n$.

Random noise case

- (Shannon information) for a distribution P on n-bit strings: if $H(P) \geqslant \alpha n$, then $H(N_\tau(P)) \geqslant \beta n$.
- (complexity) if $\mathrm{C}(x) \geqslant \alpha n$, then $\mathrm{C}(N_\tau(x)) \geqslant \beta n$ with probability at least $1 - \frac{1}{n}$

Random noise case

- (Shannon information) for a distribution P on n-bit strings: if $H(P) \geqslant \alpha n$, then $H(N_\tau(P)) \geqslant \beta n$.
- (complexity) if $\mathrm{C}(x) \geqslant \alpha n$, then $\mathrm{C}(N_\tau(x)) \geqslant \beta n$ with probability at least $1 - \frac{1}{n}$
- (combinatorial) if $\#B \leqslant 2^{\beta n}$, and every element of A get into B with probability at least $\frac{1}{n}$ after τ-noise, then $\#A \leqslant 2^{\alpha n}$.

Random noise case

- (Shannon information) for a distribution P on n-bit strings: if $H(P) \geqslant \alpha n$, then $H(N_\tau(P)) \geqslant \beta n$.
- (complexity) if $\mathrm{C}(x) \geqslant \alpha n$, then $\mathrm{C}(N_\tau(x)) \geqslant \beta n$ with probability at least $1 - \frac{1}{n}$
- (combinatorial) if $\#B \leqslant 2^{\beta n}$, and every element of A get into B with probability at least $\frac{1}{n}$ after τ-noise, then $\#A \leqslant 2^{\alpha n}$.
- (weak combinatorial) if $\#B \leqslant 2^{\beta n}$, and every element of A get into B with probability at least $1 - \frac{1}{n}$ after τ-noise, then $\#A \leqslant 2^{\alpha n}$.

All equivalent with precision $o(n)$ for complexity (log-cardinality)

Proof of equivalence

Proof of equivalence

- complexity $\Leftrightarrow$ combinatorial: as before

Proof of equivalence

- complexity $\Leftrightarrow$ combinatorial: as before
- complexity $\Rightarrow$ Shannon entropy: random i.i.d. copies have complexity close to entropy with high probability

Proof of equivalence

- complexity $\Leftrightarrow$ combinatorial: as before
- complexity $\Rightarrow$ Shannon entropy: random i.i.d. copies have complexity close to entropy with high probability
- entropy $\Rightarrow$ weak combinatorial: coding argument (apply the entropy inequality to the uniform distribution on A)

Proof of equivalence

- complexity $\Leftrightarrow$ combinatorial: as before
- complexity $\Rightarrow$ Shannon entropy: random i.i.d. copies have complexity close to entropy with high probability
- entropy $\Rightarrow$ weak combinatorial: coding argument (apply the entropy inequality to the uniform distribution on A)
- weak combinatorial $\Rightarrow$ combinatorial: concentration inequality (McDiarmid inequality, a version of Azuma–Hoeffding inequality)

How to prove the entropy inequality

How to prove the entropy inequality

- "one-letter case" P is a distribution on $\{0,1\}$ $(n=1)$

How to prove the entropy inequality

- "one-letter case" P is a distribution on $\{0, 1\}$ ($n = 1$)
- $P = B_p$ for some p

How to prove the entropy inequality

- "one-letter case" P is a distribution on $\{0,1\}$ $(n=1)$
- $P = B_p$ for some p
- $H(P) = H(p)$

How to prove the entropy inequality

- "one-letter case" P is a distribution on $\{0,1\}$ ($n = 1$)
- $P = B_p$ for some p
- $H(P) = H(p)$
- $H(N_\tau(P)) = H(N(p,\tau))$

How to prove the entropy inequality

- "one-letter case" P is a distribution on $\{0,1\}$ ($n=1$)
- $P = B_p$ for some p
- $H(P) = H(p)$
- $H(N_\tau(P)) = H(N(p,\tau))$
- exactly the curve mentioned in the lower bound

How to prove the entropy inequality

- "one-letter case" P is a distribution on $\{0,1\}$ ($n = 1$)
- $P = B_p$ for some p
- $H(P) = H(p)$
- $H(N_\tau(P)) = H(N(p,\tau))$
- exactly the curve mentioned in the lower bound
- "tensorization" + convexity argument

Tensorization lemma

Tensorization lemma

- P on n-bit strings

Tensorization lemma

- P on n-bit strings
- $(H(P), H(N_\tau(P)))$: which pairs are possible?

Tensorization lemma

- P on n-bit strings
- $(H(P), H(N_\tau(P))$: which pairs are possible?
- a set S_n in $[0, n] \times [0, n]$

Tensorization lemma

- P on n-bit strings
- $(H(P), H(N_\tau(P))$: which pairs are possible?
- a set S_n in $[0, n] \times [0, n]$

Lemma

$S_{n+m} \subset S_n + S_m$

Tensorization lemma

- P on n-bit strings
- $(H(P), H(N_\tau(P))$: which pairs are possible?
- a set S_n in $[0, n] \times [0, n]$

Lemma

$S_{n+m} \subset S_n + S_m$

Minkowski sum

Tensorization lemma

- P on n-bit strings
- $(H(P), H(N_\tau(P))$: which pairs are possible?
- a set S_n in $[0, n] \times [0, n]$

Lemma

$S_{n+m} \subset S_n + S_m$

Minkowski sum
correction: *above the convex closure* of $S_n + S_m$

Tensorization lemma

- P on n-bit strings
- $(H(P), H(N_\tau(P))$: which pairs are possible?
- a set S_n in $[0, n] \times [0, n]$

Lemma

$S_{n+m} \subset S_n + S_m$

Minkowski sum
correction: *above the convex closure* of $S_n + S_m$
lemma's proof: inequalities for Shannon entropies

Tensorization lemma

- P on n-bit strings
- $(H(P), H(N_\tau(P))$: which pairs are possible?
- a set S_n in $[0, n] \times [0, n]$

Lemma

$S_{n+m} \subset S_n + S_m$

Minkowski sum
correction: *above the convex closure* of $S_n + S_m$
lemma's proof: inequalities for Shannon entropies

It remains to check that the curves are convex (computation with power series)

Infinite consequences

Infinite consequences

- effective Hausdorff dimension of a binary sequence:

$$\dim(X) = \liminf_n \frac{\mathrm{C}(X_1 X_2 \ldots X_n)}{n}$$

Infinite consequences

- effective Hausdorff dimension of a binary sequence:

$$\dim(X) = \liminf_n \frac{\mathrm{C}(X_1 X_2 \ldots X_n)}{n}$$

- the effective dimension increases if random noise is applied to every bit (independently)

Infinite consequences

- effective Hausdorff dimension of a binary sequence:

$$\dim(X) = \liminf_n \frac{\mathrm{C}(X_1 X_2 \ldots X_n)}{n}$$

- the effective dimension increases if random noise is applied to every bit (independently)
- if $\dim(X) \geqslant \alpha = H(p)$, then $\dim(N_p(X)) \geqslant H(N(p,\tau))$ with probability 1

Infinite consequences

- effective Hausdorff dimension of a binary sequence:

$$\dim(X) = \liminf_n \frac{\mathrm{C}(X_1X_2\ldots X_n)}{n}$$

- the effective dimension increases if random noise is applied to every bit (independently)
- if $\dim(X) \geqslant \alpha = H(p)$, then $\dim(N_p(X)) \geqslant H(N(p,\tau))$ with probability 1
- the same lower bound curve for the increase

Infinite consequences

- effective Hausdorff dimension of a binary sequence:

$$\dim(X) = \liminf_n \frac{\mathrm{C}(X_1 X_2 \ldots X_n)}{n}$$

- the effective dimension increases if random noise is applied to every bit (independently)
- if $\dim(X) \geqslant \alpha = H(p)$, then $\dim(N_p(X)) \geqslant H(N(p,\tau))$ with probability 1
- the same lower bound curve for the increase
- one may use different noise levels for different positions

Infinite consequences

- effective Hausdorff dimension of a binary sequence:

$$\dim(X) = \liminf_n \frac{\mathrm{C}(X_1 X_2 \ldots X_n)}{n}$$

- the effective dimension increases if random noise is applied to every bit (independently)
- if $\dim(X) \geqslant \alpha = H(p)$, then $\dim(N_p(X)) \geqslant H(N(p,\tau))$ with probability 1
- the same lower bound curve for the increase
- one may use different noise levels for different positions
- every sequence of dimension α can be changed in a negligible fraction of positions (Besicovitch distance 0) to a strongly α-random sequence. [weakly random: Greenberg et al.]

Thanks!