No localidad cuántica basada en secuencias

Ariel Bendersky

ICC, UBA-CONICET. DC, FCEyN, UBA

Agosto 2019, Buenos Aires

Overview

- Extremely brief introduction to Kolmogorov complexity and algorithmic randomness.
- Bell inequalities with pseudorandom inputs.
- Can Nature be non-local, computable and non-signaling?
- A simple principle ruling out many non-physical situations.

Overview

- Extremely brief introduction to Kolmogorov complexity and algorithmic randomness.
- Bell inequalities with pseudorandom inputs.
- Can Nature be non-local, computable and non-signaling?
- A simple principle ruling out many non-physical situations.

Overview

- Extremely brief introduction to Kolmogorov complexity and algorithmic randomness.
- Bell inequalities with pseudorandom inputs.
- Can Nature be non-local, computable and non-signaling?
- A simple principle ruling out many non-physical situations.

Overview

- Extremely brief introduction to Kolmogorov complexity and algorithmic randomness.
- Bell inequalities with pseudorandom inputs.
- Can Nature be non-local, computable and non-signaling?
- A simple principle ruling out many non-physical situations.

Randomness and Kolmogorov complexity

Which of these is more likely to come out from a fair coin tossing?

- 01010101010101010101010101010101010101010101010
  10101010101010101010101010101010101010101010101
  0101010101010101010101010101010101...
- 11001001000011111101101010100010001000010110100
  01100001000110100110001001100011001100010100010
  1110000000110111000001110011010001...
- 10000110110111111111000111100110000111101001100
  00000000000110001001100011100111110100111100011
  0101111010000110111000101011110001...

The third seems more random. What does that mean?

Why (almost) no one picks the first or second ones?

The first two are comprehensible, one can give them a meaning. Comprehensible implies, roughly, compressible.

The third seems more random. What does that mean?

Why (almost) no one picks the first or second ones?

The first two are comprehensible, one can give them a meaning. Comprehensible implies, roughly, compressible.

Algorithmic randomness

Algorithmic randomness codifies such intuition about randomness: How much information does one need to give to a machine to produce a given string? If one has to provide as much information as the length of the string, the string is *algorithmically random*.

$$\text{Randomness} \Longleftrightarrow \text{Incompressibility}$$

For infinite sequences

A sequence is algorithmically random if almost all of its prefixes are algorithmically random.

Algorithmic randomness

Algorithmic randomness codifies such intuition about randomness: How much information does one need to give to a machine to produce a given string? If one has to provide as much information as the length of the string, the string is *algorithmically random*.

$$\text{Randomness} \Longleftrightarrow \text{Incompressibility}$$

For infinite sequences

A sequence is algorithmically random if almost all of its prefixes are algorithmically random.

Algorithmic randomness

Algorithmic randomness codifies such intuition about randomness: How much information does one need to give to a machine to produce a given string? If one has to provide as much information as the length of the string, the string is *algorithmically random*.

$$\text{Randomness} \iff \text{Incompressibility}$$

For infinite sequences

A sequence is algorithmically random if almost all of its prefixes are algorithmically random.

What about Kolmogorov complexity?

Definition

The Kolmogorov complexity of a string X, denoted by $K(X)$ is the length of the shortest program that outputs X. It is defined up to an additive constant (this makes it independent of the computing model).

Explanation

It tells us how much we can compress a string (i.e., the amount of actual information it contains.)

Algorithmic randomness, formally

A sequence X is random if for almost all n it holds that $K(X \restriction n) \approx n$.

What about Kolmogorov complexity?

Definition

The Kolmogorov complexity of a string X, denoted by $K(X)$ is the length of the shortest program that outputs X. It is defined up to an additive constant (this makes it independent of the computing model).

Explanation

It tells us how much we can compress a string (i.e., the amount of actual information it contains.)

Algorithmic randomness, formally

A sequence X is random if for almost all n it holds that $K(X \upharpoonright n) \approx n$.

What about Kolmogorov complexity?

Definition

The Kolmogorov complexity of a string X, denoted by $K(X)$ is the length of the shortest program that outputs X. It is defined up to an additive constant (this makes it independent of the computing model).

Explanation

It tells us how much we can compress a string (i.e., the amount of actual information it contains.)

Algorithmic randomness, formally

A sequence X is random if for almost all n it holds that $K(X \upharpoonright n) \approx n$.

Two nice results

- A system equivalent to a Turing machine (any classical system according to the Church-Turing thesis) cannot generate a random sequence.
- It is impossible to prove, in any correct axiomatic system, that a given string is random (but for finitely many strings).

Two nice results

- A system equivalent to a Turing machine (any classical system according to the Church-Turing thesis) cannot generate a random sequence.
- It is impossible to prove, in any correct axiomatic system, that a given string is random (but for finitely many strings).

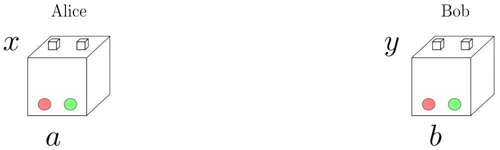
Some notation

- X, Y, Z, denote infinite sequences.
- $X \restriction n$ refers to the first n symbols from sequence X. That is, the prefix of length n of X.
- If $X = x_1 x_2 x_3 ...$ and $Y = y_1 y_2 y_3 ...$, then $X \oplus Y = x_1 y_1 x_2 y_2 x_3 y_3 ...$

Bell loophole

Bendersky, A., De La Torre, G., Senno, G., Figueira, S., Acín, A. (2016). Algorithmic pseudorandomness in quantum setups. Physical review letters, 116(23), 230402.

Bell scenario and non-locality



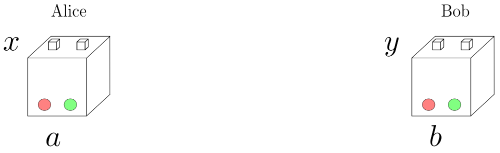
Locality

The system is local iff

$$P\left(a, b|x, y\right) = \sum_{\lambda} c_{\lambda} p_{\lambda}^{A}\left(a|x\right) p_{\lambda}^{B}\left(b|y\right).$$

In any other case, the distribution is called non-local.

Bell scenario and non-locality



Locality

The system is local iff

$$P(a,b|x,y)=\sum_{\lambda}c_{\lambda}p_{\lambda}^{A}(a|x)\,p_{\lambda}^{B}(b|y)\,.$$

In any other case, the distribution is called non-local.

Bell inequality

Bell inequality

$$2 \leq E(0,0) - E(0,1) + E(1,0) + E(1,1) \leq 2$$

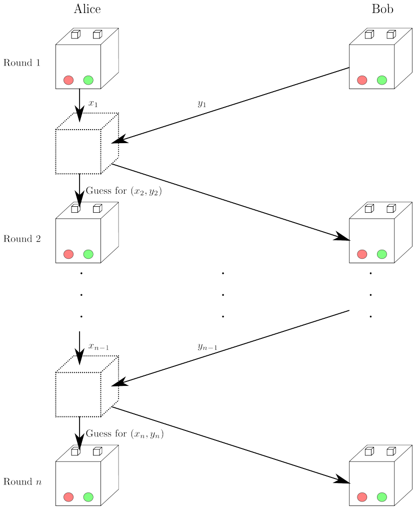
with
$E\left(x,y\right) = P\left(0,0|x,y\right) + P\left(1,1|x,y\right) - P\left(0,1|x,y\right) - P\left(1,0|x,y\right).$

is satisfied for every local distribution but can be violated by quantum mechanics.

The memory scenario

We will introduce an eavesdropper that prepares the boxes locally on every round with information on every input from previous rounds. This memory scenario still allows to see non-locality (Barret et al PRA 66:042111, Pironio et al Nature 464(7291):1021-1024, Pironio et al PRA 87:012336).

The memory scenario



The key idea

If the eavesdropper can predict the forthcoming inputs by Alice and Bob, she could prepare the boxes to give whatever probability distribution she wants.

Therefore

The problem is reduced to that of learnability of infinite sequences from prefixes. This problem has already been studied (Solomonoff, Gold, Zeugmann).

The key idea

If the eavesdropper can predict the forthcoming inputs by Alice and Bob, she could prepare the boxes to give whatever probability distribution she wants.

Therefore

The problem is reduced to that of learnability of infinite sequences from prefixes. This problem has already been studied (Solomonoff, Gold, Zeugmann).

Learnability

Main result

Any time (or space) bounded complexity class can be learned from prefixes. It means that after seeing a long enough prefix, every bit is predicted correctly. This classes include P, NP, PR, etc.

The idea

These classes are computably enumerable. The algorithm works as follows:

Seen bits:		1	0	1						
	s_1	0	0	0	0	0	0	0	...	
	s_2	0	0	1	1	0	1	1	...	
	s_3	1	0	0	0	1	0	1	...	
	s_4	1	1	1	1	0	0	0	...	
	s_5	0	1	0	1	0	1	0	...	
First match:	s_6	1	0	1	0	1	1	1	...	Next guess: 0
	s_7	1	1	0	1	1	0	1	...	
	$\vdots$									

Learnability

Main result

Any time (or space) bounded complexity class can be learned from prefixes. It means that after seeing a long enough prefix, every bit is predicted correctly. This classes include P, NP, PR, etc.

The idea

These classes are computably enumerable. The algorithm works as follows:

Seen bits:		1	0	1						
	s_1	0	0	0	0	0	0	0	...	
	s_2	0	0	1	1	0	1	1	...	
	s_3	1	0	0	0	1	0	1	...	
	s_4	1	1	1	1	0	0	0	...	
	s_5	0	1	0	1	0	1	0	...	
First match:	s_6	1	0	1	0	1	1	1	...	Next guess: 0
	s_7	1	1	0	1	1	0	1	...	
	$\vdots$									

Results

- If Alice and Bob use computable functions (they belong to some time complexity class), they can't rule out an eavesdropper learning a class bigger than the one they are using, therefore they can't conclude nonlocality.
- The eavesdropper, by choosing the class he learns, forces Alice and Bob to have to resort to harder functions. For instance, if the eavesdropper picks NP, then Alice and Bob will have to go beyond NP to have a proper violations of a Bell inequality.

Properties and random thoughts

- Once the sequence is learned, the overhead is small (the eavesdropper keeps simulating the same machine). The eavesdropper does not need more computational power than Alice and Bob.
- The first bits, before the sequences are learned, can give a proper violation. However, is this violation valid if in the long run there is a local model?

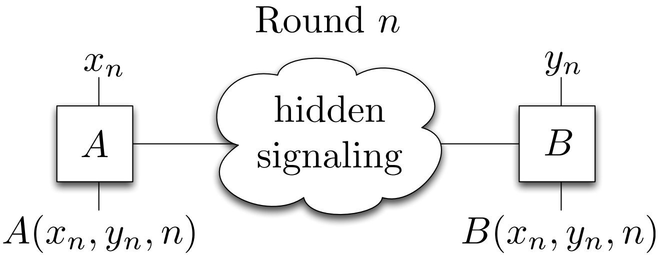
Properties and random thoughts

- Once the sequence is learned, the overhead is small (the eavesdropper keeps simulating the same machine). The eavesdropper does not need more computational power than Alice and Bob.
- The first bits, before the sequences are learned, can give a proper violation. However, is this violation valid if in the long run there is a local model?

Can Nature be non-local, computable and non-signaling?

Bendersky, A., Senno, G., De La Torre, G., Figueira, S., Acin, A. (2017). Nonsignaling deterministic models for nonlocal correlations have to be uncomputable. Physical review letters, 118(13), 130401.

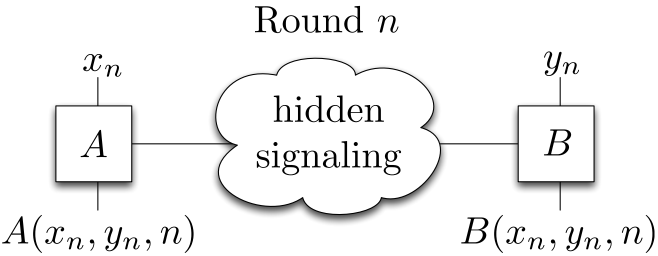
The model



Observations

- The functions A and B depend on the other party's input as this is the only relevant information to be signaled by the hidden signaling.
- We will assume functions A and B to be computable as a reasonable feature of a classic-like model.

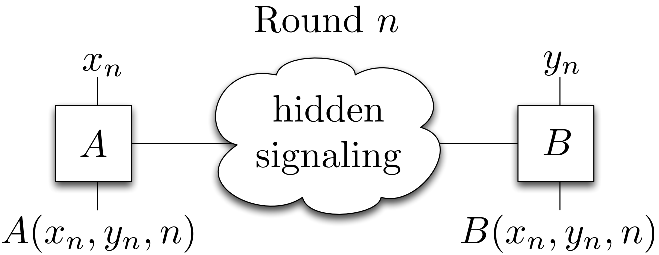
The model



Observations

- The functions A and B depend on the other party's input as this is the only relevant information to be signaled by the hidden signaling.
- We will assume functions A and B to be computable as a reasonable feature of a classic-like model.

The model



Observations

- The functions A and B depend on the other party's input as this is the only relevant information to be signaled by the hidden signaling.
- We will assume functions A and B to be computable as a reasonable feature of a classic-like model.

The model

As we assume that the boxes show non-local correlations when used with random inputs, this means that, without loss of generality, there exist infinitely many n and y_n such that:

$$B(0, y_n, n) \neq B(1, y_n, n)$$

Therefore...

If Bob knew the function his box (or Nature) computes, he'd be able to reconstruct Alice's input for infinitely many rounds. But he doesn't, B is Nature's secret.
Can it be kept this way?

Idea

Use learnability!

The model

As we assume that the boxes show non-local correlations when used with random inputs, this means that, without loss of generality, there exist infinitely many n and y_n such that:

$$B(0, y_n, n) \neq B(1, y_n, n)$$

Therefore...

If Bob knew the function his box (or Nature) computes, he'd be able to reconstruct Alice's input for infinitely many rounds. But he doesn't, B is Nature's secret.
Can it be kept this way?

Idea

Use learnability!

The protocol to learn B and signal

- Alice and Bob pre-share a random sequence (we will remove randomness later on) $S \in \{(0,0),(0,1),(1,0),(1,1),1,\ldots,m\}^{\omega}$.
- **Learning round:** if $S(n) = (x,y)$, Alice inputs x and Bob inputs y. Furthermore, Bob sets his current guess for B to the first program that reproduces every seen bit from the learning rounds.
- **Signaling round**: if $S(n) = i \in \{1,\ldots,m\}$, Alice inputs the ith bit of her message and Bob uses his current guess $\widetilde{B}$ for B to see if there is a y such that $\widetilde{B}(0,y,n) \neq \widetilde{B}(1,y,n)$. If there is such y, he inputs it. If not, he inputs 0. He then uses the output for statistics on the i-th bit from Alice's message.

The protocol to learn B and signal

Observations

- If Bob's guess for B is wrong, the randomness of S ensures that in some learning round the predicted bit will be different from the actual obtained bit. Therefore, sooner or later, B will be learned.
- Is it fair to allow for Alice and Bob to use randomness for S if we assume that even quantum mechanics is deterministic? We will see how to remove such requirement.

The protocol to learn B and signal

Observations

- If Bob's guess for B is wrong, the randomness of S ensures that in some learning round the predicted bit will be different from the actual obtained bit. Therefore, sooner or later, B will be learned.
- Is it fair to allow for Alice and Bob to use randomness for S if we assume that even quantum mechanics is deterministic? We will see how to remove such requirement.

Using a non-random S

- If sequence S was too simple, then the program used by Bob's box could be computing S to have a different behaviour for learning and for signaling rounds.
- How complex should it be if we assumed Bob's box to belong to a complexity class C?

Using a non-random S

- If sequence S was too simple, then the program used by Bob's box could be computing S to have a different behaviour for learning and for signaling rounds.
- How complex should it be if we assumed Bob's box to belong to a complexity class C?

Using a non-random S

C-randomness

A sequence S is called C random if no adversary with a computing power belonging to class C can bet on the next symbol from S and win an unbounded amount of money.

The result

There exist computable C-random sequences (See S. Figueira and A. Nies, Theory of Computing Systems 56, 439 (2015))

For our problem

If sequence S was C-random and Bob's box computes a function belonging to class C, our protocol is sound. Therefore, Alice can signal to Bob in the limit.

Using a non-random S

C-randomness

A sequence S is called C random if no adversary with a computing power belonging to class C can bet on the next symbol from S and win an unbounded amount of money.

The result

There exist computable C-random sequences (See S. Figueira and A. Nies, Theory of Computing Systems 56, 439 (2015))

For our problem

If sequence S was C-random and Bob's box computes a function belonging to class C, our protocol is sound. Therefore, Alice can signal to Bob in the limit.

Using a non-random S

C-randomness

A sequence S is called C random if no adversary with a computing power belonging to class C can bet on the next symbol from S and win an unbounded amount of money.

The result

There exist computable C-random sequences (See S. Figueira and A. Nies, Theory of Computing Systems 56, 439 (2015))

For our problem

If sequence S was C-random and Bob's box computes a function belonging to class C, our protocol is sound. Therefore, Alice can signal to Bob in the limit.

A signaling protocol if boxes used functions belonging to a class C known to Alice and Bob

.

- Alice and Bob pre-share a C–random sequence $S \in \{(0,0),(0,1),(1,0),(1,1),1,\dots,m\}^{\omega}$.
  - This is now easy, since we know there exist computable C–random sequences, they need to share a program computing one such sequence.
- **Learning round:** if $S(n)=(x,y)$, Alice inputs x and Bob inputs y. Furthermore, Bob sets his current guess for B to the first program that reproduces every seen bit from the learning rounds.
- **Signaling round**: if $S(n)=i\in\{1,\dots,m\}$, Alice inputs the ith bit of her message and Bob uses his current guess $\widetilde{B}$ for B to see if there is a y such that $\widetilde{B}(0,y,n)\neq\widetilde{B}(1,y,n)$. If there is such y, he inputs it. If not, he inputs 0. He then uses the output for statistics on the i-th bit from Alice's message.

Therefore...

Non-signaling deterministic models for non-local correlations have to be uncomputable.

Summary

- We saw a loophole that occurs when, on a Bell experiment, Alice and Bob use pseudorandom inputs. The only assumption for the local model is knowledge on the time (or space) complexity of Alice or Bob's program.
- We then analysed what would happen if Nature behaved in a computable manner generating non-locality via a hidden signaling mechanism. Such model for Nature would allow the parties to instantly communicate.

Thank you!