Randomness as a type of warranty

Alexander Shen,
LIRMM,
on leave from IITP RAS,
sasha.shen@gmail.com

Randomness and philosophy

- Make 10^5 coin tosses and try to compress the resulting string with zip.
  Law: you never get more than 1% compression. (2^{-1000})

Randomness and philosophy

- Make 10^5 coin tosses and try to compress the resulting string with zip.
  Law: you never get more than 1% compression. (2^{-1000})
  Can this law be derived from laws of physics? How?

Randomness and philosophy

- Make 10^5 coin tosses and try to compress the resulting string with zip.
  Law: you never get more than 1% compression. (2^{-1000})
  Can this law be derived from laws of physics? How?
- Can an individual object be obviously non-random? Can the lottery with outcomes $0, 1, 0, 1, 0, 1, \ldots$ be contested?
  General: can statistic be used in the court?

Randomness and philosophy

- Make 10^5 coin tosses and try to compress the resulting string with zip.
  Law: you never get more than 1% compression. (2^{-1000})
  Can this law be derived from laws of physics? How?
- Can an individual object be obviously non-random? Can the lottery with outcomes $0, 1, 0, 1, 0, 1, \ldots$ be contested?
  General: can statistic be used in the court?
  Example: Sebastopol "referendum" (youtube, PiGz4U093XU)
  (registered,participation,'yes') =
  (306258,274101,262041).

Randomness and philosophy

- Make 10^5 coin tosses and try to compress the resulting string with zip.
  Law: you never get more than 1% compression. (2^{-1000})
  Can this law be derived from laws of physics? How?
- Can an individual object be obviously non-random? Can the lottery with outcomes $0, 1, 0, 1, 0, 1, \ldots$ be contested?
  General: can statistic be used in the court?
  Example: Sebastopol "referendum" (youtube, PiGz4U093XU)
  (registered,participation,'yes') =
  (306258,274101,262041).
  $P/R = 274101/306258 = 0.89500029387$;
  $0.895 \times 306258 = 274100.91$
  $[0.896 \times 306258 = 274407.168$, distance about 300]

Randomness and philosophy

- Make 10^5 coin tosses and try to compress the resulting string with zip.
  Law: you never get more than 1% compression. (2^{-1000})
  Can this law be derived from laws of physics? How?
- Can an individual object be obviously non-random? Can the lottery with outcomes $0, 1, 0, 1, 0, 1, \dots$ be contested?
  General: can statistic be used in the court?
  Example: Sebastopol "referendum" (youtube, PiGz4U093XU)
  (registered,participation,'yes') =
  (306258,274101,262041).
  $P/R = 274101/306258 = 0.89500029387$;
  $0.895 \times 306258 = 274100.91$
  [$0.896 \times 306258 = 274407.168$, distance about 300]
  $Y/P = 262041/274101 = 0.95600161984$;
  $0.956 \times 274101 = 262040.556$

Randomness and philosophy

- Make 10^5 coin tosses and try to compress the resulting string with zip.
  Law: you never get more than 1% compression. (2^{-1000})
  Can this law be derived from laws of physics? How?
- Can an individual object be obviously non-random? Can the lottery with outcomes $0, 1, 0, 1, 0, 1, \ldots$ be contested?
  General: can statistic be used in the court?
  Example: Sebastopol "referendum" (youtube, PiGz4U093XU)
  (registered,participation,'yes') =
  (306258,274101,262041).
  $P/R = 274101/306258 = 0.89500029387;$
  $0.895 \times 306258 = 274100.91$
  $[0.896 \times 306258 = 274407.168$, distance about 300]
  $Y/P = 262041/274101 = 0.95600161984;$
  $0.956 \times 274101 = 262040.556$
  coincidence $\approx (1/300)^2 < 2 \cdot 10^{-5}$

Randomness and copyright

Randomness and copyright

- Two sites that claim to provide strings of random bits

Randomness and copyright

- Two sites that claim to provide strings of random bits
- $A \oplus B = C$, where C is copyright protected (or state secret)

Randomness and copyright

- Two sites that claim to provide strings of random bits
- $A \oplus B = C$, where C is copyright protected (or state secret)
- Possibility: A truly random, $B = A \oplus C$

Randomness and copyright

- Two sites that claim to provide strings of random bits
- $A \oplus B = C$, where C is copyright protected (or state secret)
- Possibility: A truly random, $B = A \oplus C$
- Or vice versa: indistinguishable

Randomness and copyright

- Two sites that claim to provide strings of random bits
- $A \oplus B = C$, where C is copyright protected (or state secret)
- Possibility: A truly random, $B = A \oplus C$
- Or vice versa: indistinguishable
- Legal puzzles where two actions combined lead to some consequences. Difference: here each action could be completely innocent.

Randomness as incompressibility

Randomness as incompressibility

- Random tables. Preshuffled cards. Randomization of multiple-choice tests

- Random tables. Preshuffled cards. Randomization of multiple-choice tests
- Kolmogorov complexity as the bit length of minimal description

- Random tables. Preshuffled cards. Randomization of multiple-choice tests
- Kolmogorov complexity as the bit length of minimal description
- randomness: complexity close to length

Randomness as incompressibility

- Random tables. Preshuffled cards. Randomization of multiple-choice tests
- Kolmogorov complexity as the bit length of minimal description
- randomness: complexity close to length
- non-randomness can be proven, but randomness cannot

Randomness as incompressibility

- Random tables. Preshuffled cards. Randomization of multiple-choice tests
- Kolmogorov complexity as the bit length of minimal description
- randomness: complexity close to length
- non-randomness can be proven, but randomness cannot
- infinite case as "approximation from above": Martin-Löf random sequences avoid all effectively null sets, have maximal complexity of prefixes

Shafer – Vovk (inspired) approach

Shafer – Vovk (inspired) approach

- Probability and Finance: It's Only a Game! (G.Shafer, V.Vovk, 2001)

Shafer – Vovk (inspired) approach

- Probability and Finance: It's Only a Game! (G.Shafer, V.Vovk, 2001)
- shop: "random bits available here"

Shafer – Vovk (inspired) approach

- Probability and Finance: It's Only a Game! (G.Shafer, V.Vovk, 2001)
- shop: "random bits available here"
- customer pays \$1 and gets a sequence of N bits

Shafer – Vovk (inspired) approach

- Probability and Finance: It's Only a Game! (G.Shafer, V.Vovk, 2001)
- shop: "random bits available here"
- customer pays \$1 and gets a sequence of N bits
- why not sell zeros? in general, how customer may check that she gets a real product, not cheap imitation?

Randomness warranty

Randomness warranty

- customer brings a function $f\colon \{0,1\}^N \to \mathbb{R}^{\geq 0}$ = a list of 2^N non-negative reals, with average at most 1.

- customer brings a function $f\colon \{0,1\}^N \to \mathbb{R}^{\geq 0}$ = a list of 2^N non-negative reals, with average at most 1.
- she does not show it to the shop owner, but commits to it (puts its description in the closed envelope on the table).

Randomness warranty

- customer brings a function $f\colon \{0,1\}^N \to \mathbb{R}^{\geq 0}$ = a list of 2^N non-negative reals, with average at most 1.
- she does not show it to the shop owner, but commits to it (puts its description in the closed envelope on the table).
- later, when bit sequence x is provided, the customer is entitled to get back $f(x)$.

Randomness warranty

- customer brings a function $f\colon \{0,1\}^N \to \mathbb{R}^{\geq 0}$ = a list of 2^N non-negative reals, with average at most 1.
- she does not show it to the shop owner, but commits to it (puts its description in the closed envelope on the table).
- later, when bit sequence x is provided, the customer is entitled to get back $f(x)$.
- $f \equiv 1$: money back

Randomness warranty

- customer brings a function $f\colon \{0,1\}^N \to \mathbb{R}^{\geq 0}$ = a list of 2^N non-negative reals, with average at most 1.
- she does not show it to the shop owner, but commits to it (puts its description in the closed envelope on the table).
- later, when bit sequence x is provided, the customer is entitled to get back $f(x)$.
- $f \equiv 1$: money back
- f is 2^N at one point and 0 elsewhere: a lottery

Randomness warranty

- customer brings a function $f\colon \{0,1\}^N \to \mathbb{R}^{\ge 0}$ = a list of 2^N non-negative reals, with average at most 1.
- she does not show it to the shop owner, but commits to it (puts its description in the closed envelope on the table).
- later, when bit sequence x is provided, the customer is entitled to get back $f(x)$.
- $f \equiv 1$: money back
- f is 2^N at one point and 0 elsewhere: a lottery
- payments may be changed proportionally

Randomness warranty

- customer brings a function $f\colon \{0,1\}^N \to \mathbb{R}^{\geq 0}$ = a list of 2^N non-negative reals, with average at most 1.
- she does not show it to the shop owner, but commits to it (puts its description in the closed envelope on the table).
- later, when bit sequence x is provided, the customer is entitled to get back $f(x)$.
- $f \equiv 1$: money back
- f is 2^N at one point and 0 elsewhere: a lottery
- payments may be changed proportionally
- (and some profit for the owner and tax could be added)

Using randomness shop for hedging

Using randomness shop for hedging

- random string r used to decide whether (say) an input number is a prime.

Using randomness shop for hedging

- random string r used to decide whether (say) an input number is a prime.
- only small minority (say, 2^{-100}-fraction) of strings lead to wrong answer

Using randomness shop for hedging

- random string r used to decide whether (say) an input number is a prime.
- only small minority (say, 2^{-100}-fraction) of strings lead to wrong answer
- shopping for a string x, we bring a payment function that is 2^{100} on strings that give wrong answers, and 0 otherwise

Using randomness shop for hedging

- random string r used to decide whether (say) an input number is a prime.
- only small minority (say, 2^{-100}-fraction) of strings lead to wrong answer
- shopping for a string x, we bring a payment function that is 2^{100} on strings that give wrong answers, and 0 otherwise
- so if we are unlucky to get the wrong answer about primality (due to "bad" random string), at least we get monetary compensation

Using randomness shop for hedging

- random string r used to decide whether (say) an input number is a prime.
- only small minority (say, 2^{-100}-fraction) of strings lead to wrong answer
- shopping for a string x, we bring a payment function that is 2^{100} on strings that give wrong answers, and 0 otherwise
- so if we are unlucky to get the wrong answer about primality (due to "bad" random string), at least we get monetary compensation
- here the warranty function is given implicitly, we need to prove that the average is at most 1, and prove the value of the function on x before claiming the compensation.

Zero-sum games

Zero-sum games

- Two players A and B commit to their moves independently; each has finite number of options

Zero-sum games

- Two players A and B commit to their moves independently; each has finite number of options
- The winner is determined by a table $p(a, b)$ that says how much A gets from B (may be positive or negative)

Zero-sum games

- Two players A and B commit to their moves independently; each has finite number of options
- The winner is determined by a table $p(a, b)$ that says how much A gets from B (may be positive or negative)
- equilibruim cost c: there is a probabilistic strategy for A that for every possible move of B guarantees the expected return at least c [for A], and a probabilistic strategy for B that for every possible move of A guarantees the expected return at most c [for A].

Zero-sum games

- Two players A and B commit to their moves independently; each has finite number of options
- The winner is determined by a table $p(a, b)$ that says how much A gets from B (may be positive or negative)
- equilibruim cost c: there is a probabilistic strategy for A that for every possible move of B guarantees the expected return at least c [for A], and a probabilistic strategy for B that for every possible move of A guarantees the expected return at most c [for A].
- von Neumann: cost and corresponding probabilistic strategies always exist

Randomness game

Randomness game

- our game: A brings f with average at most 1 [strictly speaking, not a finite game];

Randomness game

- our game: A brings f with average at most 1 [strictly speaking, not a finite game];
- B brings x;

Randomness game

- our game: A brings f with average at most 1 [strictly speaking, not a finite game];
- B brings x;
- payment: A gets $f(x) - 1$.

Randomness game

- our game: A brings f with average at most 1 [strictly speaking, not a finite game];
- B brings x;
- payment: A gets $f(x) - 1$.
- cost of this game: 0

- our game: A brings f with average at most 1 [strictly speaking, not a finite game];
- B brings x;
- payment: A gets $f(x) - 1$.
- cost of this game: 0
- "universal lottery" where A chooses the rules.

Randomness game

- our game: A brings f with average at most 1 [strictly speaking, not a finite game];
- B brings x;
- payment: A gets $f(x) - 1$.
- cost of this game: 0
- "universal lottery" where A chooses the rules.
- special case: $N = 1$, $f(x)$ is equal to 2 on some value and 0 on the other one (guess-bit-game)

Randomness game

- our game: A brings f with average at most 1 [strictly speaking, not a finite game];
- B brings x;
- payment: A gets $f(x) - 1$.
- cost of this game: 0
- "universal lottery" where A chooses the rules.
- special case: $N = 1$, $f(x)$ is equal to 2 on some value and 0 on the other one (guess-bit-game)

Randomness existence postulate: there are physical sources (e.g., coin tossing) that allow B play this game more or less successfully

Relation to algorithmic randomness

Relation to algorithmic randomness

- randomness deficiency of n-bit x defined as $d(n) - KP(x)$

Relation to algorithmic randomness

- randomness deficiency of n-bit x defined as $d(n) - KP(x)$
- small for most strings, but almost n for highly compressible x

Relation to algorithmic randomness

- randomness deficiency of n-bit x defined as $d(n) - KP(x)$
- small for most strings, but almost n for highly compressible x
- $f(x) = 2^{d(x)}$

Relation to algorithmic randomness

- randomness deficiency of n-bit x defined as $d(n) - KP(x)$
- small for most strings, but almost n for highly compressible x
- $f(x) = 2^{d(x)}$
- average value is $\sum_x 2^{n-KP(x)}/2^n = \sum_x 2^{-KP(x)} \leq 1$

Relation to algorithmic randomness

- randomness deficiency of n-bit x defined as $d(n) - KP(x)$
- small for most strings, but almost n for highly compressible x
- $f(x) = 2^{d(x)}$
- average value is $\sum_x 2^{n-KP(x)}/2^n = \sum_x 2^{-KP(x)} \leq 1$
- so f is a valid warranty function: we can prove that average is at most 1, and prove high values of f

Relation to algorithmic randomness

- randomness deficiency of n-bit x defined as $d(n) - KP(x)$
- small for most strings, but almost n for highly compressible x
- $f(x) = 2^{d(x)}$
- average value is $\sum_x 2^{n-KP(x)}/2^n = \sum_x 2^{-KP(x)} \leq 1$
- so f is a valid warranty function: we can prove that average is at most 1, and prove high values of f
- f wins on every compressible string, so the randomness shop should avoid them to be safe against f

Relation to algorithmic randomness

- randomness deficiency of n-bit x defined as $d(n) - KP(x)$
- small for most strings, but almost n for highly compressible x
- $f(x) = 2^{d(x)}$
- average value is $\sum_x 2^{n-KP(x)}/2^n = \sum_x 2^{-KP(x)} \leq 1$
- so f is a valid warranty function: we can prove that average is at most 1, and prove high values of f
- f wins on every compressible string, so the randomness shop should avoid them to be safe against f
- but game requirement is not limited to computable f (example: experimental sample)

Relation to algorithmic randomness

- randomness deficiency of n-bit x defined as $d(n) - KP(x)$
- small for most strings, but almost n for highly compressible x
- $f(x) = 2^{d(x)}$
- average value is $\sum_x 2^{n-KP(x)}/2^n = \sum_x 2^{-KP(x)} \le 1$
- so f is a valid warranty function: we can prove that average is at most 1, and prove high values of f
- f wins on every compressible string, so the randomness shop should avoid them to be safe against f
- but game requirement is not limited to computable f (example: experimental sample)
- Caveat: f is not computable, and lower bounds for f take a long time to establish

Relation to pseudorandomness

Relation to pseudorandomness

- (Blum–Micali–Yao PRNG) $G\colon \{0,1\}^n \to \{0,1\}^N$

Relation to pseudorandomness

- (Blum–Micali–Yao PRNG) $G\colon \{0,1\}^n \to \{0,1\}^N$
- $n \ll N$

Relation to pseudorandomness

- (Blum–Micali–Yao PRNG) $G\colon \{0,1\}^n \to \{0,1\}^N$
- $n \ll N$
- G easy to compute (small circuit)

Relation to pseudorandomness

- (Blum–Micali–Yao PRNG) $G\colon \{0,1\}^n \to \{0,1\}^N$
- $n \ll N$
- G easy to compute (small circuit)
- for every simple test function f the average value of f on $G(x)$ for $x \in \{0,1\}^n$ is close to the average value of f on $\{0,1\}^N$.

Relation to pseudorandomness

- (Blum–Micali–Yao PRNG) $G\colon \{0,1\}^n \to \{0,1\}^N$
- $n \ll N$
- G easy to compute (small circuit)
- for every simple test function f the average value of f on $G(x)$ for $x \in \{0,1\}^n$ is close to the average value of f on $\{0,1\}^N$.
- "if customer is computationally limited, then seller can save some truly random bits using PRNG"