Von Neumann Normalisation and Symptoms of Randomness

A. A. Abbott C. S. Calude

University of Auckland

www.cs.auckland.ac.nz/~{aabb009,cristian}

UC 2011, Turku, Finland

June 2011

- Quantum Random Number Generators
  - What are they?
  - What do we mean by "random"?
- Normalisation Techniques
  - von Neumann normalisation
- Normalisation and Randomness
  - Probability space transformations
  - Effects on randomness
- Probability Spaces and QRNGs

Quantum Random Number Generators

Recently there has been a hotbed of activity in using quantum mechanics to generate "randomness" — quantum random number generators (QRNGs).

Many devices have surfaced, primarily based on the 'photon + beamsplitter' idea.

- Quantis — the first commercial such device

Quantum Random Number Generators

Recently there has been a hotbed of activity in using quantum mechanics to generate "randomness" — quantum random number generators (QRNGs).

Many devices have surfaced, primarily based on the 'photon + beamsplitter' idea.

- Quantis — the first commercial such device

Most claims of randomness are based on beliefs of quantum mechanics, not on solid mathematical groundings.

Randomness

Kolmogorov axiomatic probability theory assigns probabilities to sets of outcomes and shows how to calculate with such probabilities: **it assumes randomness, but does not distinguish between individual random and non-random elements.**

- QM theory formally predicts these probabilities, but in no way guarantees the randomness of these events.

Randomness

Kolmogorov axiomatic probability theory assigns probabilities to sets of outcomes and shows how to calculate with such probabilities: **it assumes randomness, but does not distinguish between individual random and non-random elements.**

- QM theory formally predicts these probabilities, but in no way guarantees the randomness of these events.

Imagine a QRNG left to run *ad infinitum* generating an infinite sequence of bits. The sequence constructed by concatenating the binary representation of the integers

$$\mathbf{x} = 011011100101110\ldots$$

is uniformly distributed and has high entropy. But, is it random?

True randomness?

AIT uses computability theory to model "infinite M-L/C random sequences", i.e. sequences which pass all computably enumerable tests of randomness.

It is impossible to prove a sequence is random. Such sequences are incomputable.

AIT proves that there are no infinite sequences that are truly random.

No infinite sequence of bits can pass all tests of randomness, but generators of truly random bits proliferate.

Nature's claim (doi:10.1038/news.2010.181, 14 April 2010):

Symptoms of Randomness

A good QRNG would ideally produce a sequence satisfying the following (independent) "symptoms of randomness":

1. **Unpredictability**: it should be impossible to predict exactly in advance the values of any bits of the sequence. Formally, this is a manifestation of strong incomputability of the sequence.

Symptoms of Randomness

A good QRNG would ideally produce a sequence satisfying the following (independent) "symptoms of randomness":

1. **Unpredictability**: it should be impossible to predict exactly in advance the values of any bits of the sequence. Formally, this is a manifestation of strong incomputability of the sequence.
2. **Uniformity**: All finite patterns of bits should occur equally often. Formally, this is a manifestation of Borel-normality of the sequence.

Unpredictability of QRNGs

The formulation of QM only allows us to make probabilistic claims, predicting the uniformity of all possible outcomes.

The apparent unpredictability of QM stems from empirical evidence, and various no-go theorems forbidding certain types of deterministic (predictable) theories.

Unpredictability of QRNGs

The formulation of QM only allows us to make probabilistic claims, predicting the uniformity of all possible outcomes.

The apparent unpredictability of QM stems from empirical evidence, and various no-go theorems forbidding certain types of deterministic (predictable) theories.

The Kochen-Specker Theorem provides a very strong argument for the strong incomputability of sequences of quantum random bits, giving some mathematical grounding to claims of unpredictability.

The Need for Normalisation

Experimental reality makes exact uniformity impossible to achieve by QRNGs.

Normalisation techniques, working under the assumption of independence and constancy of bias, are used by most QRNGs to mitigate this fact.

The Need for Normalisation

Experimental reality makes exact uniformity impossible to achieve by QRNGs.

Normalisation techniques, working under the assumption of independence and constancy of bias, are used by most QRNGs to mitigate this fact.

But, what effect do such normalisation procedures have on the unpredictability and uniformity?

Normalisation Techniques

The simplest technique is due to von Neumann, although more efficient methods exist and are often used.

To formally discuss normalisation techniques, we need to work with the probability space $(\Omega, \mathcal{F}, P_n)$ associated with a QRNG:

- Ω is the sample space. For finite strings $\Omega = \{0,1\}^n$,
- $\mathcal{F}$ is a σ-algebra on Ω. $\mathcal{F} = 2^{\{0,1\}^n}$ for finite strings,
- $P_n : \mathcal{F} \to [0,1]$ is the probability distribution of the source.

We model the source as an independent-bit-source, so that

$$P_n(X) = \sum_{x \in X} p_0^{\#_0(x)} p_1^{\#_1(x)}.$$

- Is this a good model?

von Neumann Normalisation

von Neumann's method groups bits into pairs and performs the mapping $01 \to 0$, $10 \to 1$, $00, 11 \to \lambda$.

For example:

01 01 10 01 11 00 11 11 01 01 00 11 00 10 1

$\downarrow$

0 0 1 0 0 0 1

von Neumann Normalisation

von Neumann's method groups bits into pairs and performs the mapping $01 \to 0$, $10 \to 1$, $00, 11 \to \lambda$.

For example:

$$01\ 01\ 10\ 01\ 11\ 00\ 11\ 11\ 01\ 01\ 00\ 11\ 00\ 10\ 1$$

$$\downarrow$$

$$0\ 0\ 1\ 0\ 0\ 0\ 1$$

For strings of length n, the normalised string has length between 0 and $\lfloor n/2 \rfloor$, with expected length np_0p_1.

von Neumann Normalisation

Formally, we introduce the functions VN_n which perform this normalisation on n-bit strings, along with the inverses $VN_{n,m}^{-1}$ defined for $x \in \{0,1\}^m$ as

$$VN_{n,m}^{-1}(x) = \{y \in \{0,1\}^n : VN_n(y) = x\}.$$

von Neumann Normalisation

Formally, we introduce the functions VN_n which perform this normalisation on n-bit strings, along with the inverses $VN_{n,m}^{-1}$ defined for $x \in \{0,1\}^m$ as

$$VN_{n,m}^{-1}(x) = \{y \in \{0,1\}^n : VN_n(y) = x\}.$$

The probability space induced by normalisation is

$$P_{n\to m}(X) = \frac{P_n(VN_{n,m}^{-1}(X))}{P_n(VN_{n,m}^{-1}(\{0,1\}^m))}.$$

Theorem (von Neumann): $P_{n\to m} = U_m$, the uniform distribution.

Infinite Sequences from QRNGs

Mathematical arguments show that to obtain results on the incomputability of QRNG sequences it is necessary to work with infinite sequences of bits. To study the effect of normalisation on such sequences, we must extend it to infinite sequences also.

For infinite sequences the sample space is the Cantor space $\{0,1\}^\omega$ and the probability measure is defined by the cylinders $x\{0,1\}^\omega$ with $x \in \{0,1\}^*$ as

$$\mu_P(x\{0,1\}^\omega) = P_{|x|}(x).$$

The normalisation function VN is extended as expected, and defined VN^{-1} as $VN^{-1}(x) = \{y \in \{0,1\}^* : VN_{|y|}(y) = x\}$.

VN for Infinite Sequences

The normalised probability measure is thus defined as

$$\mu_{P_{VN}}(x\{0,1\}^\omega) = \frac{\mu_P(VN^{-1}(x)\{0,1\}^\omega)}{\mu_P(VN^{-1}(\{0,1\}^{|x|})\{0,1\}^\omega)}.$$

VN for Infinite Sequences

The normalised probability measure is thus defined as

$$\mu_{P_{VN}}(x\{0,1\}^\omega) = \frac{\mu_P(VN^{-1}(x)\{0,1\}^\omega)}{\mu_P(VN^{-1}(\{0,1\}^{|x|})\{0,1\}^\omega)}.$$

Theorem: $\mu_{P_{VN}} = \mu_L$, the Lebesgue measure.

This shows that as far as the probability space of the source is concerned, normalisation ensures uniformity. But this is not the same as uniformity of bits!

So what about properties of individual sequences?

Defining randomness

Formally, fix a universal prefix-free TM U, and define the complexity of a string as

$$H_U(\sigma) = \min\{|p| : U(p) = \sigma\}.$$

The choice of U is not important.

A sequence $\mathbf{x} \in \{0,1\}^\omega$ is ε-random if there exists a constant c such that

$$H_U(\mathbf{x}(n)) \geq \varepsilon \cdot n - c$$

for all $n \geq 1$. Sequences which are 1-random are simply called random.

The Infinite Collapse

- Let y be any string and $\mathbf{x} = x_1 x_2 \ldots$, then

$$VN(y_1 \bar{y_1} \ldots y_{|y|} \bar{y}_{|y|} x_1 x_1 x_2 x_2 \ldots) = VN_{|y|}(y),$$

  a finite string.
- Since $\mathbf{x}$ can be ML-random, this "collapse" is not due to low complexity of sequences.

The Infinite Collapse

- Let y be any string and $\mathbf{x} = x_1 x_2 \ldots$, then

$$VN(y_1 \bar{y_1} \ldots y_{|y|} \bar{y}_{|y|} x_1 x_1 x_2 x_2 \ldots) = VN_{|y|}(y),$$

  a finite string.
- Since $\mathbf{x}$ can be ML-random, this "collapse" is not due to low complexity of sequences.

Theorem: $\{\mathbf{x} \in \{0,1\}^\omega : VN(\mathbf{x}) \in \{0,1\}^*\}$ has μ_P measure zero.

Increasing or Decreasing Randomness?

Focusing on sequences for which this infinite collapse does not occur, we see that normalisation can both increase and decrease the randomness of a sequence.

Proposition: There exist continuously many infinite ε-random sequences $\mathbf{x}$ such that $VN(\mathbf{x}) = 000\ldots00\ldots$, for any computable $0 < \varepsilon < 1$.

Proposition: There exist continuously many infinite ε-random sequences $\mathbf{x}$ such that $VN(\mathbf{x})$ is 1-random for any computable $0 < \varepsilon < 1$.

Effect on Normality and Randomness

Given that normalisation has the potential to decrease the randomness of a sequence, we would like to check that this doesn't happen for any "good random" sequence.

Theorem: If $\mathbf{x}$ is a Borel normal sequence w.r.t. μ_L, then $VN(\mathbf{x})$ is also Borel normal w.r.t μ_L.

Effect on Normality and Randomness

Given that normalisation has the potential to decrease the randomness of a sequence, we would like to check that this doesn't happen for any "good random" sequence.

Theorem: If $\mathbf{x}$ is a Borel normal sequence w.r.t. μ_L, then $VN(\mathbf{x})$ is also Borel normal w.r.t μ_L.

Theorem: If $\mathbf{x}$ is a random sequence w.r.t. μ_P, then $VN(\mathbf{x})$ is also random w.r.t $\mu_{P_{VN}} = \mu_L$.

Effect on Normality and Randomness

Given that normalisation has the potential to decrease the randomness of a sequence, we would like to check that this doesn't happen for any "good random" sequence.

Theorem: If $\mathbf{x}$ is a Borel normal sequence w.r.t. μ_L, then $VN(\mathbf{x})$ is also Borel normal w.r.t μ_L.

Theorem: If $\mathbf{x}$ is a random sequence w.r.t. μ_P, then $VN(\mathbf{x})$ is also random w.r.t $\mu_{P_{VN}} = \mu_L$.

Theorem: The set $\{\mathbf{x} \in \{0,1\}^\omega : VN(\mathbf{x}) \text{ is computable}\}$ has measure zero in μ_P.

Probability Distributions and Individual Sequences

Normalisation

- helps to realise the probability distribution which QM theory predicts,
- gives unbiasing of the distribution, but does not produce "true randomness",
- cannot guarantee incomputability or normality of the sequences produced.

Probability Distributions and Individual Sequences

Normalisation

- helps to realise the probability distribution which QM theory predicts,
- gives unbiasing of the distribution, but does not produce "true randomness",
- cannot guarantee incomputability or normality of the sequences produced.

From Kochen-Specker Theorem one can deduce that is impossible to obtain computable sequences from a QRNG.

The set of computable numbers has measure zero in the generated distribution, but this result is weaker than, although not in contradiction with, the impossibility result.

Implications for QRNGs

Algorithmic randomness and Borel normality are preserved under normalisation, but it is not known if QRNGs are guaranteed to produce such sequences.

Probabilistically, normalisation ensures this is the case with probability one.

It is not immediately clear from these results that the incomputability produced by QRNGs is preserved by normalisation. Indeed, we have seen that normalisation can both increase or decrease the degree of randomness of a sequence.

Normalisation should be applied with a little more caution and awareness.

Thank you!

A. A. Abbott, C. S. Calude. Von Neumann normalisation of a quantum random number generator, *CDMTCS Research Report* 392, 2010.

C. S. Calude, K. Svozil. Quantum randomness and value indefiniteness, *Advanced Science Letters*, 1:165–168, 2008.

A. A. Abbott, C. S. Calude, K. Svozil. A quantum random number generator certified by value indefiniteness, *CDMTCS Research Report* 396, 2010.

A. Stefanov, N. Gisin, O. Guinnard, L. Guinnard, H. Zbinden. Optical quantum random number generator. *Journal of Modern Optics*, 47(4):595–598, 2000.

S. Pironio, A. Acín, S. Massar, A. B. de la Giroday, D. N. Matsukevich, P. Maunz, S. Olmchenk, D. Hayes, L. Luo, T. A. Manning, C. Monroe. Random numbers certified by Bell's Theorem. *Nature*, 464(09008), 2010.

Borel-normality

A sequence $\mathbf{x}$ is Borel-normal if for every $m \geq 1$ and every $1 \leq i \leq 2^m$ one has

$$\lim_{n\to\infty} \frac{N_i^m(\mathbf{x}(n))}{\lfloor n/m \rfloor} = 2^{-m},$$

where N_i^m count the number of non-overlapping occurrences of the ith binary string of length m.