On time-adaptive statistical testing for random number generators

Boris Ryabko

September 15, 2020

Summary

The problem of constructing effective statistical tests for random sequences of binary digits is considered. The effectiveness of such statistical tests is mainly estimated on the basis of experiments with various random number generators. We consider this problem in the framework of mathematical statistics and find an asymptotic estimate for the p-value of the optimal test in the case when the alternative hypothesis is an unknown stationary ergodic source.

Introduction-1

Random number generators (RNG) and pseudo-random number generators (PRNG) are widely used in many applications. RNGs are based on physical sources, while pseudo-random numbers are generated by computers.

The goal of RNG and PRNG is to generate sequences of binary digits (that is, 0 and 1) , which are distributed as a result of throwing an "honest" coin, or, more precisely, to obey the Bernoulli distribution with parameters $(1/2, 1/2)$.

This property is verified experimentally with the help of statistical tests developed for this purpose.

Example of a test: sequence $x_1 ... x_l, \tau(x_1 ... x_l) = \#(0)/l$. *If* $l =$ *1000,* $\#$ *(0)* $=$ *900, then* $\tau = 0.9$.

Introduction-test batteries

Informally, an ideal RNG should generate sequences that pass all tests. In practice, especially in cryptographic applications, this requirement is formulated as follows:
an RNG must pass a so-called battery of statistical tests, that is, some fixed set of tests. When a battery is applied, each test in the test battery is applied separately to the RNG. Among these batteries, we mention the Marsaglia's Diehard battery, which contains 16 tests *(weak battery)*, **the National Institute of Standards and Technology (NIST) battery of 15 tests** *(weak battery)* several batteries proposed by L'Ecuyer and Simard , which contain from 10 to 106 tests and many others.
Note that practically used RNG should be tested from time to time like any physical equipment, and therefore these test batteries should be used continuously.

The problem

How to evaluate large batteries of tests? On the one hand, the larger the test battery, the more likely it is to find flaws in the tested RNG. On the other hand, the larger the battery, the more time is required for testing.
Another view is as follows: in reality, the time available to study any RNG is limited.

Example:
$p(0) = 0,51, p(1) = 0,49; \qquad n = 1000$
$p(0) = 0,5001, p(1) = 0,4999; \qquad n = 1000\,000$
Given a certain time budget, one can either use more tests and relatively short sequences generated by the RNG, or use fewer tests, but longer sequences and, in turn, this gives more chances to find deviations of the randomness of the considered RNG.

The main idea -1

In order to reduce this trade-off, we propose time-adaptive testing of RNGs, in which, informally speaking, first all the tests are executed on relatively short sequences generated by the RNG, and then a few "promising" tests are applied for the final testing.

The key question here is which tests are promising. It can be done based on a so-called pi-value.

$Definition\ \pi(x_1 ... x_l) = \#\{y_1 ... y_l\ such\ that \tau((x_1 ... x_l) \geq \tau(y_1 ... y_l)\}$

The main idea -2

For example, if a battery of two tests is applied to (relatively short) sequences of the same length, it can be assumed that the smaller the p-value, the more promising the test.

But a more complicated situation may arise when we have sequences different lengths (for example, the first test was applied to a sequence of length l_1, and the second to a sequence of the length of l_2, $l_1 \neq l_2$).

We show that a good strategy is to choose the test i for which the ratio $-\log(p_{value_i})/l_i$ is maximum.

This recommendation is based on the following

theorem: if an RNG can be modelled by a stationary ergodic source, the value $-log\, \pi(x_1 x_2 ... x_n)/n$ goes to $1-h$, if n grows, where $x_1 x_2 ...$ is a generated sequence, $\pi(\,)$ is the p-value of the most powerful test, h is the limit Shannon entropy of the stationary ergodic source.

The results

We describe a time-adaptive test and some experiments designed to study its capabilities. We first look at a case where the goal is to reduce testing time.

It turns out that testing time for the adaptive test can be done much less than on the original battery without losing test power.

Secondly, we consider the case when the testing time is fixed. It turns out that the power of the adaptive test may be greater than that of the original battery of tests.

Let us consider a situation where the randomness testing is performed by conducting a battery of statistical tests for randomness.
Suppose that the battery contains s tests and α_i is the significance level of $i-$th test, $i = 1, ..., s$. If the battery is applied in such a way that the hypothesis H_0 is rejected when at least one test in the battery rejects it, then the significance level α of this battery satisfies the following inequality:

$$\alpha \leq \sum_{i=1}^{s} \alpha_i \, .$$

We have considered a scenario in which a test is applied to a single sequence generated by an RNG, and then the researcher makes a decision on the RNG based on the test results.

The scheme of the time-adaptive testing.

Let there be an RNG which generates binary sequences, and a battery of s tests with statistics $\tau_1, \tau_2, ..., \tau_s$. In addition, suppose that the total available testing time is limited to a certain amount T and the level of significance is $\alpha \in (0, 1)$.

The calculations are separated into a preliminary stage and a final one. The result of the preliminary stage is the list of values

$$\gamma_1 = \frac{-\log \pi_{\tau_1}(x_1^1 x_2^1 ... x_{n_1}^1)}{n_1}, \gamma_2 = \frac{-\log \pi_{\tau_2}(x_1^2 x_2^2 ... x_{n_2}^2)}{n_2}$$

$$, ..., \ \gamma_s = \frac{-\log \pi_{\tau_s}(x_1^s x_2^s ... x_{n_s}^s)}{n_s}, \tag{1}$$

Then, taking into account the values γ_i, we choose some tests and apply them to the longer sequence, calculate new values γ, and so on. When the preliminary stage is carried out, several tests from the battery should be chosen for the next stage.

The final stage is as follows. First, we divide the significance level α into $\alpha_1, \alpha_2, ..., \alpha_k$ in such a way that $\sum_{i=1}^{k} \alpha_i = \alpha$. Then, we obtain new sequence(s) $y_1^1 y_2^1 ... y_{m_1}^1, ..., y_1^k y_2^k ... y_{m_k}^k$, which are **independent of** $x_1^1 x_2^1 ... x_{n_1}^1$**,** $..., x_1^s x_2^s ... x_{n_s}^s$ and calculate

$$\pi_{\tau_{i_1}}(y_1^1 y_2^1 ... y_{m_1}^1), ..., \pi_{\tau_{i_k}}(y_1^k y_2^k ... y_{m_k}^k)\,. \qquad (2)$$

The hypothesis H_0 will be accepted, if $\pi_{\tau_{i_j}}(y_1^j y_2^j ... y_{m_j}^j) > \alpha_j$ for all $j = 1, ..., k$. Otherwise, H_0 is rejected. The parameters of the test should be chosen in such a way that the total time of calculation is not greater than the given limit of T.

Experiments.

We carried out some experiments with the time-adaptive test basing on the battery Rabbit to compare the original Rabbit battery and its new adaptive test.

Let us first describe the choice of RNG for our experiments. Nowadays there are many “bad” PRNGs and “good” ones. In other words, the output sequences of some known PRNGs have some deviations from the randomness, which are quite easy to detect with many known tests, while other PRNGs do not have deviations that can be detected by known tests. So, we need to have some families of RNGs with such deviations from the randomness that they can be detected only for quite large output sequences.

We take a good generator MRG32k3a, and a bad one LCG from, generate sequences $g_1 g_2...$ and $b_1 b_2...$ by these two generators and then prepared a "mixed" sequence $m_1 m_2...$ in such a way that

$$m_i = \begin{cases} g_i & \text{if } i \mod D \ \neq 0 \\ b_i & \text{if } i \mod D \ = 0 \end{cases} \tag{3}$$

where D is a parameter. In different experiments, we used different "good" generators (MRG32k3a, Java 48, lfsr 113) and "bad" ones (LCG, Taus and Visual Basic).

The time-adaptive testing was organised as follows: during the preliminary stage we first generated a file $m_1 m_2 ... m_{l_1}$ with $l_1 = 2\,000\,000$ bytes, tested it by 25 tests from the Rabbit battery and calculated the values (1) with $\log \equiv \log_2$, see the left part of Table 1. (This battery contains 26 tests, but one of them cannot be applied to such a short sequence.) Then we chose 5 tests with the biggest value $-log\, \pi_{t_i}(m_1 ... m_{l_i})/l_1$ (let they be $t_{i_1}, ..., t_{i_5}$), generated a sequence $m_1 ... m_{l_2}$ with $l_2 = 6\,000\,000$ bytes and applied the tests $t_{i_1}, ..., t_{i_5}$ for testing this sequence (see the example in the right part of Table 1). After that we found a test t_f for which

$$-log\, \pi_{t_f}/l_f = \max_{r=1,...,25;\, j=i_1 ... i_5} \{-log \pi_r(m_1 ... m_{l_1})/l_1,$$

$$-log \pi_j(m_1 ... m_{l_2})/l_2\}\,.$$

(In other words, for t_f the value $-log \pi_r(m_1 ... m_{l_k})/l_k$ is maximal for $k = 1, 2$ and all r (see the Table 1). The preliminary stage was finished.

Then, during the second stage, we generated a 40 000 000 byte sequence, and applied the test t_f to it. If the obtained p-value was less than 0.001, the hypothesis H_0 was rejected. (Note that the sequence length $l_1 = 2\,000\,000$ and $l_2 = 6\,000\,000$ are 5% and 15% from the final length of 40 000 000 bytes. So, the total length of the sequences tested by all the tests during the preliminary stage is $25 \times 0.05 + 5 \times 0.15 = 2$ the final length, i.e. $2 \times 40\,000\,000$. On the other hand, if one applies the battery Rabbit to the sequence of the same length, the total length of investigated sequences is $25 \times 40\,000\,000$, i.e. 8,33 times more.

Table: Time-adaptive testing. Preliminary stage.

test	length (l) (bytes)	p-value (π)	$-\log\pi/l$	length (l) (bytes)	p-value	$-\log\pi/l$
t1	$2\,10^6$	0.42	$6.3\ 10^{-7}$			
t2	$2\,10^6$	0.37	$7.3\ 10^{-7}$			
t3	$2\,10^6$	0.028	$26\ 10^{-7}$	$6\,10^6$	0,23	$3.6\ 10^{-7}$
t4	$2\,10^6$	0.78	$1.8\,10^{-7}$			
t5	$2\,10^6$	0.4	$6.5\,10^{-7}$			
t6	$2\,10^6$	0.37	$7.2\,10^{-7}$			
t7	$2\,10^6$	0.059	$20\,10^{-7}$			
t8	$2\,10^6$	0.026	$26\,10^{-7}$	$6\,10^6$	0.0037	$26\ 10^{-7}$
t9	$2\,10^6$	0.72	$2.4\,10^{-7}$			
t10	$2\,10^6$	0.72	$2.4\,10^{-7}$			
t11	$2\,10^6$	0.63	$3.3\,10^{-7}$			
t12	$2\,10^6$	0.74	$2.2\,10^{-7}$			

Table 1 contains the results of all the calculations carried out during the preliminary stage. So, we can see that the value $-\log_2 \pi / l$ is maximal for the test $t13$. Hence, at the final stage, we applied the test $t13$ to the new 40 000 000-byte sequence. It turned out that $\pi_{t13} = 2.9\ 10^{-26}$ and, hence, H_0 is rejected. After that, we conducted an additional experiment to get the full picture. Namely, we calculated p-values for all tests and for the same 40 000 000-byte sequence and the estimated total time of calculations. It turned out that the p-values of the two tests were less than 0.001. Namely, $\pi_{t13} = 2.9\ 10^{-26}$, $\pi_{t22} = 1.1\ 10^{-6}$. Besides, we estimated the time of calculations for all experiments. So, the described time-adaptive testing revealed one of the two most powerful tests, **while the time used is 8 times.**

We carried out similar experiments 20 times for $D = 2, 3, 4$ with different good and bad generators.Besides, we investigated several modifications of the considered scheme. In particular, we considered a case where during the preliminary stage we, as before, first chose 5 the best tests and them two of the best tests for the finale stage (instead of one, as in the experiment above).

In all cases the battery Rabbit rejects H_0 and the time-adaptive testing rejected H_0, too. But the time was significantly reduced.

Experiments designed to improve the quality of testing.

We first applied the original Rabbit battery to a specific generator, and then time-adaptive testing so that the total size of the investigated files was the same for both tests. For both cases, we took a significance level of 0.001.

Then we generated 26 different files, every 120 megabytes (MB) in size, and applied one battery test to one file (note that the total size of all files is 3120 MB.).

The time adaptive testing was as follows. As in the previous section, we applied a two-step preliminary stage. First, we generated 26 files with a length of 50 MB each, then we selected 5 tests with a minimum p-value and applied these tests to five new files with a length of 150 MB. Then, based on the results obtained, we select one test with a maximum value of $-\log \pi()/length$) and use this test for a file of size 1000 MB (= 1 GB). If the p-value for this test was less than 0.001, H_0 was rejected, otherwise accepted. The main results are presented in Table 2.

Bad generator	Good generator	D in (3)	Result of adaptive testing	Result of original battery
LCG	Java 48	4	H_1	H_1
LCG	Java 48	16	H_1	H_1
LCG	Java 48	64	H_1	H_1
LCG	**Java 48**	256	H_1	H_0
LCG	Java 48	1024	H_0	H_0
LCG	lfsr 113	4	H_1	H_1
LCG	lfsr 113	16	H_1	H_1
LCG	lfsr 113	64	H_1	H_1
LCG	**lfsr 113**	256	H_1	H_0
LCG	lfsr 113	1024	H_0	H_0
LCG	MRG 32 k3a	4	H_1	H_1
LCG	**MRG 32 k3a**	16	H_1	H_0

This table shows that there are several cases where both batteries either accept or reject H_0 together, and there are cases where the adaptive battery rejects H_0, while the original battery accepts this hypothesis. Taking into account that in both cases the significance level (0.001) was the same, **we see that there are situations when the adaptive testing detects deviations from randomness, while the original battery does not find them.**

Conclusion

We showed that the proposed time-adaptive testing is promising for RNG testing. On the other hand, we note that the proposed time-adaptive testing does not offer exact values of numerous parameters (the number of steps at the preliminary stage, the number of tests compared in one step, the length of the tested sequences, the rule for choosing tests at different stages, etc. There are many methods available to solve such problems (for example, neural networks and other AI algorithms), and some of them can be used along with time-adaptive testing.

We believe that the **proposed approach, combined with multidimensional optimization, allows researchers to investigate and optimize time-adaptive testing.**

Thank you for attention!