Machine Learning and Random Number Generation Testing

Michael Mascagni
Department of Computer Science, Florida State University, Tallahassee, FL **USA**
Division of Applied and Computational Mathematics, National Institute of Standards and Technology, Gaithersburg, MD **USA**
mascagni@fsu.edu

Coauthor(s): John Thrasher, Jarret Kizer, Christoph Hagenauer, Department of Computer Science, University of South Carolina, Beaufort, SC **USA**

Special session: Testing and analysis of pseudo-random number generators

The standard for testing pseudorandom numbers is the testing software called `TestU01`, [1], through its `Crush` family of test suites. It has become customary to call a pseudorandom number generator "Crushproof" if it passes all of the tests in the largest of the `TestU01` test suites, `BigCrush`, [2]. While a `Crushproof` generator is desirable, even when a generator fails some tests, important information from `TestU01` is available. In fact, the presenter has had considerable experience using the output of `TestU01` to verify the identity of a generator when only its software implementation was available.

We are interested in using information from `TestU01` to train a machine-learning-based classifier to identify peudorandom number generators from their behavior with `TestU01`. To this end, we have taken the 6 generators that are available in the `S`calable `P`arallel `R`andom `N`umber `G`enerators, `SPRNG`, package to investigate the feasibility of creating such a classifier. `SPRNG` has 6 generator families each with multiple parameter choices to give one many different sub-families of generators. We will present how the classifier was constructed, and the results on the `SPRNG`. We will also comment on future prospects for extending the classifier. We hope to be able to construct a classifier that will provide insight even to generators that are `Crushproof`.

[1] Pierre L'Ecuyer and Richard Simard, (2007), "TestU01: A C library for empirical testing of random number generators," *ACM Transactions on Mathematical Software*, **33(4)**, 40 pages, https://doi.org/10.1145/1268776.1268777.

[2] John K. Salmon, Mark A. Moraes, Ron O. Dror and David E. Shaw, (2011), "Parallel random numbers: as easy as 1, 2, 3," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, article 16, 12 pages, https://doi.org/10.1145/2063384.2063405.

[3] M. Mascagni and A. Srinivasan (2000), "Algorithm 806: SPRNG: A Scalable Library for Pseudorandom Number Generation," *ACM Transactions on Mathematical Software*, **26**: 436-461, https://doi.org/10.1145/358407.358427.