Speaker: Pamphile T. Roy

Quasi-Monte Carlo for everyone: `scipy.stats.qmc` and `torch.quasirandom`

Pamphile T. Roy
iTranslate, roy.pamphile@gmail.com

Maximilian Balandat
Facebook, balandat@gmail.com

The Python programming language is becoming more and more popular, not just as a general purpose language, but as a scripting language for science and engineering. Besides its accessibility and flexibility, a key reason for this is its rich library ecosystem for scientific computing.

On one end of the spectrum, there are fundamental libraries such as SciPy [5], which provide basic and well established scientific tools for thousands of other packages. Anyone using Python for science is most certainly using SciPy either directly or as a dependency from another tool. On the other end, there are modern deep learning libraries such as PyTorch [4], which provide a modular and flexible high-level python interface to high-performance numerical backends with support for autodifferentiation and modern hardware such as GPUs, TPUs and custom accelerators. These libraries are evolving fast and used by millions of end-users; in fact, the overwhelming majority of modern machine learning development is done in such libraries. Initially focused on deep learning, they have been developing more and more towards supporting general purpose scientific computing workloads.

Our work introduced QMC capabitlities to both SciPy and PyTorch, making QMC available to millions of users through the libraries themselves and the many others that build on them . Bringing QMC to everyone will help advance the state-of-the-art in QMC methods in two ways:

1. With QMC methods now much easier to use and available "out-of-the-box", researchers and engineers will be able to utilize their benefits across a much broader range of problems, expanding their reach and impact far beyond their current usage.
2. As QMC methods are applied to more and a more diverse set of applications, this will raise a whole set of new practical (i.e., implementation-related) as well as fundamental methodological questions. For instance, recent applications of QMC in the context of sample average approximation in Bayesian optimization [1] led to the development of a strong law of large numbers for QMC [3], and our implementation work inspired additional work on highlighting the importance of including the first point in the *Sobol'* sequence [2].

Since SciPy 1.7, the following set of features is available:

- *Sobol'* and *Halton* sequences (scrambled and unscrambled),
- Multinominal and Multivariate sampler,
- LHS (optimized on C^2 coming soon),
- Discrepancy measures (C^2, wrap around, star-L_2, mixed),
- Scaling utility,
- Fast numerical inverse using Hermite spline (sample any distribution with QMC).

As for PyTorch, the same version of *Sobol'* as SciPy is provided, but implemented in PyTorch's native ATen Tensor library.

In this presentation, we propose to walk through the new features using examples. We hope to generate discussions, gather feedbacks to improve the libraries and call for new contributors.

Acknowledgements

The authors acknowledge Professors Art B. Owen, Fred Hickernell and Sergei Kucherenko for helpful discussions. The SciPy maintainer team also provided support and help regarding the design and integration. We thank especially Ralf Gommers and Matt Haberland for their thorough reviews.

[1] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21524–21538. Curran Associates, Inc., 2020.

[2] Art B. Owen. On dropping the first Sobol' point. *arXiv e-prints*, page arXiv:2008.08051, August 2020.

[3] Art B. Owen and Daniel Rudolf. A strong law of large numbers for scrambled net integration. *SIAM Review*, 63(2):360–372, 2021.

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala.

Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[5] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, Aditya Vijaykumar, Alessandro Pietro Bardelli, Alex Rothberg, Andreas Hilboll, Andreas Kloeckner, Anthony Scopatz, Antony Lee, Ariel Rokem, C. Nathan Woods, Chad Fulton, Charles Masson, Christian Häggström, Clark Fitzgerald, David A. Nicholson, David R. Hagen, Dmitrii V. Pasechnik, Emanuele Olivetti, Eric Martin, Eric Wieser, Fabrice Silva, Felix Lenders, Florian Wilhelm, G. Young, Gavin A. Price, Gert-Ludwig Ingold, Gregory E. Allen, Gregory R. Lee, Hervé Audren, Irvin Probst, Jörg P. Dietrich, Jacob Silterra, James T. Webber, Janko Slavič, Joel Nothman, Johannes Buchner, Johannes Kulick, Johannes L. Schönberger, JoséVinícius de Miranda Cardoso, Joscha Reimer, Joseph Harrington, Juan Luis Cano Rodríguez, Juan Nunez-Iglesias, Justin Kuczynski, Kevin Tritz, Martin Thoma, Matthew Newville, Matthias Kümmerer, Maximilian Bolingbroke, Michael Tartre, Mikhail Pak, Nathaniel J. Smith, Nikolai Nowaczyk, Nikolay Shebanov, Oleksandr Pavlyk, Per A. Brodtkorb, Perry Lee, Robert T. McGibbon, Roman Feldbauer, Sam Lewis, Sam Tygier, Scott Sievert, Sebastiano Vigna, Stefan Peterson, Surhud More, Tadeusz Pudlik, Takuya Oshima, Thomas J. Pingel, Thomas P. Robitaille, Thomas Spura, Thouis R. Jones, Tim Cera, Tim Leslie, Tiziano Zito, Tom Krauss, Utkarsh Upadhyay, Yaroslav O. Halchenko, Yoshiki Vázquez-Baeza, and SciPy 1. 0 Contributors. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272, 2020.