CSCI 5576 Fall-19
Community Analysis Report
Python package: Numpy
Name: Saurabh Mishra

Introduction and History:

Numpy is the fundamental package for scientific computing with python. It contains among other things:

- A powerful N-dimensional array object
- Sophisticated broadcasting function
- Tools for integrating C/C++ and Fortran code
- Other functionalities like useful linear algebra, Fourier transform, etc.

Numpy derives from an old library called Numeric, which was first array object build for python. It was fairly successful and was used in number of applications before being phased out. Numpy also incorporates features from a library called NumArray, which was written after numeric but before numpy.

Numeric was created by Jim Hugunin with contributions from several other developers. In early 2005, NumPy developer Travis Oliphant wanted to unify the community around a single array package and ported Numarray's features to Numeric, releasing the result as NumPy 1.0 in 2006. This new project was part of SciPy. To avoid installing the large SciPy package just to get an array object, this new package was separated and called NumPy. Support for Python 3 was added in 2011 with NumPy version 1.5.0. In 2011, PyPy started development on an implementation of the NumPy API for PyPy. It is not yet fully compatible with NumPy.

The Numpy Community:

The community of NumPy users and developers is large and decentralized. We try to direct certain conversations to certain channels.

- Use the #numpy tag on StackOverflow for *usage* questions (like “How do I do *X* in NumPy?”).
- Use the GitHub issue tracker for bug reports, documentation issues and feature requests.
- Use the NumPy mailing list for longer-form discussion items. This is for things that concern the broader NumPy community.

The Numpy Code of Conduct:

The code of conduct applies to all spaces managed by the NumPy project, including all public and private mailing lists, issue trackers, wikis, blogs, Twitter, and any other communication channel used by our community. More details on code of conduct and diversity statement can be found here.

Project Governance Model:

NumPy is a community-owned and community-run project. To the maximum extent possible, decisions about project direction are made by community consensus. Some members of the community additionally contribute by serving on the NumPy steering council, where they are responsible for facilitating the establishment of community consensus, for stewarding project resources, and in extreme cases for making project decisions if the normal community-based process breaks down. Normally, all project decisions are made by consensus of all interested contributors. The primary goal of this approach is to ensure that the people who are most affected by and involved in any given change can contribute their knowledge in the confidence that their voices will be heard, because thoughtful review from a broad community is the best mechanism we know of for creating high-quality software. Consensus means that everyone is trusted with the right to veto any change they feel it necessary. While this may sound like a recipe for obstruction and pain, this is not what happens. Instead, they found that most people take this responsibility seriously, and only invoke their veto when they judge that a serious problem is being ignored, and that their veto is necessary to protect the project. And in practice, it turns out that such vetoes are almost never formally invoked, because their mere possibility ensures that contributors are motivated from the start to find some solution that everyone can live with – thus accomplishing our goal of ensuring that all interested perspectives are taken into account.

Institutional Partner and Funding:

The Steering Council are the primary leadership for the project. No outside institution, individual or legal entity has the ability to own, control, usurp or influence the project other than by participating in the Project as Contributors and Council Members. However, because institutions can be an important funding mechanism for the project, it is important to formally acknowledge institutional participation in the project. These are Institutional Partners. For the first time ever, Numpy received grant funding. The proposal, “Improving NumPy for Better Data Science” received $645,020 from the Moore Foundation over 2 years, with the funding going to UC Berkeley Institute for Data Science. Donations to NumPy are managed by NumFOCUS. For donors in the United States, your gift is tax-deductible to the extent provided by law.

Contribution:

On inspecting the open issues in the numpy GitHub repository, I found an inconsistency in the implementation of dot function. The specific example mentioned the inconsistency is between dot, @ and matmul functions, the dot product between a numpy array of 0 and np.inf results in different solutions based on the shape of the arrays. On further inspection I found that the dot product doesn’t give warning flags for floating point inputs. E.g. np.dot([1,1], [np.inf, np.inf]) doesn’t raise any warning flags.