Python for Information Theoretic Analysis of Neural Data

Robin A. A. Ince★, Rasmus S. Petersen★, Daniel C. Swan+, Stefano Panzeri✿,★

★ Faculty of Life Sciences, University of Manchester, UK; + Institute of Cell and Molecular Biosciences, Newcastle University, UK
✿ Robotics, Brain and Cognitive Sciences Dept., Italian Institute of Technology, Genoa, Italy

MANCHESTER 1824 The University of Manchester

Python

What is Python?

"Python is a dynamic object-oriented programming language that can be used for many kinds of software development. It offers strong support for integration with other languages and tools, comes with extensive standard libraries, and can be learned in a few days. Many Python programmers report substantial productivity gains and feel the language encourages the development of higher quality, more maintainable code." [1]

Numerical libraries [2] provide Python with an efficient and powerful N-dimensional array object as well as a range of scientific functions, similar to the MATLAB package. The syntax is very similar to MATLAB, making any transition relatively straightforward. We have found using Python offers a number of advantages for our research (see Discussion).

Entropy and Information

Entropy is a measure of **uncertainty**, denoted by H(•). Mutual Information between a stimulus and a response is given by:

$$I(\mathbf{S};\mathbf{R}) = H(\mathbf{R}) - H(\mathbf{R}|\mathbf{S})$$

Information is the **reduction in uncertainty** about the response given knowledge that a particular stimulus was presented.

- Information is **symmetric**; applies to both encoding and decoding problem.
- Takes into account all correlations with **no underlying assumptions** about the system.
- Uses meaningful units (**bits**) that can be compared between experiments.

Sampling Bias

Origins of the Bias

Calculation of the entropy requires estimation of the underlying **probability distributions**. Since only limited quantities of data are available experimentally this leads to **limited sampling bias**. This causes **over-estimation** of the information.

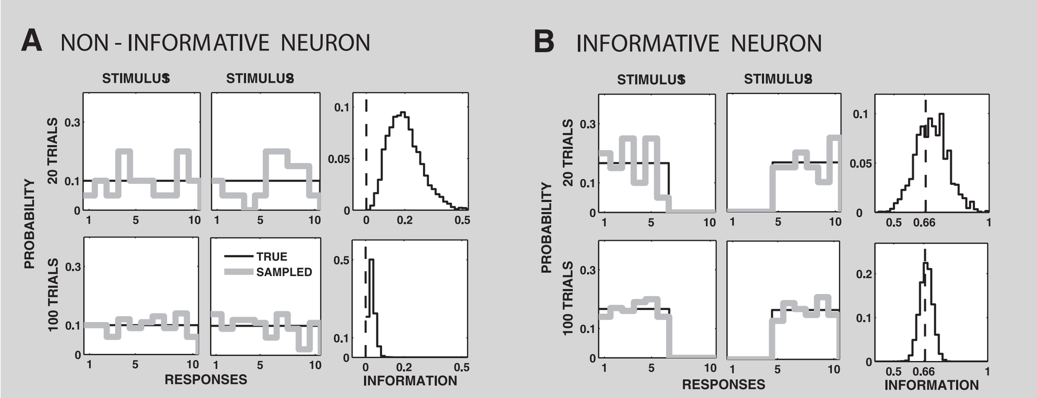

Figure 1: The effect of limited sampling. Reproduced from [3].

Bias Correction

Fortunately, a number of techniques have been developed to compensate for the bias effect.

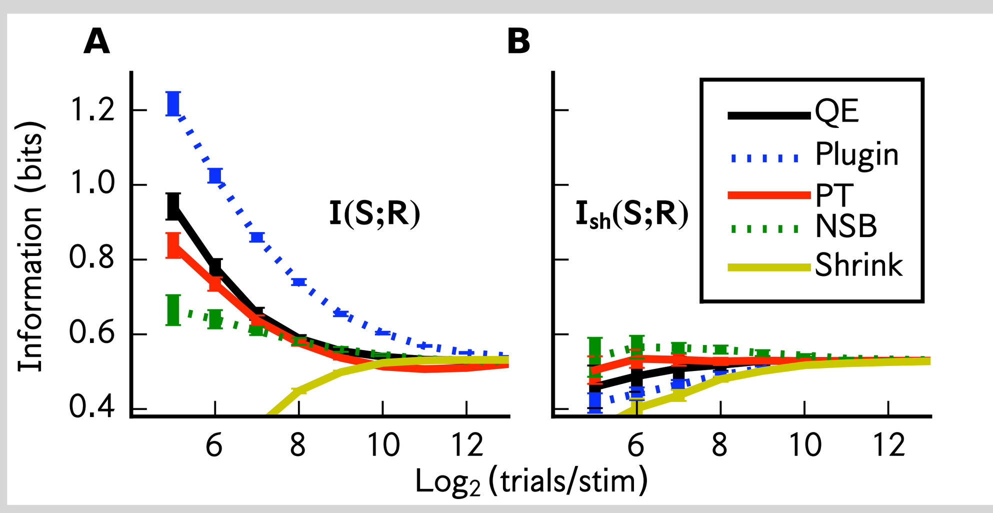

Figure 2: Comparison of the performance of different bias correction methods. Reproduced from [4].

These techniques can be difficult to implement. We have developed **PyEntropy**, a Python library to allow efficient calculation of bias-corrected entropy and information values [4,5].

```
# create DiscreteSystem object with response data
# X to stimuli in Y
s = DiscreteSystem(X,(8,2),Y,(1,13))
# compute entropies with Panzeri-Treves correction
s.calculate_entropies(method='pt',
        sampling='naive',
        calc=['HX','HXY','HiXY','HshXY'])
# compute the shuffled information estimator
s.Ish()
```

Maximum Entropy

Interactions in a system are **correlations** between outputs given a fixed input (or visa versa). The principle of **maximum entropy** provides a principled way to investigate such interactions.

The distribution with maximum entropy enforcing a set of constraints that preserve interactions of up to a certain order (that is between a certain number of variables) is the **least structured** distribution satisfying those constraints and by definition contains no higher order interactions. By comparing these maximum entropy solutions of different orders with those measured from experimental data we can investigate the effects of interactions.

We develop a general algorithm for finding maximum entropy distributions over finite spaces subject to marginal constraints up to a given order, using the information geometric approach of Amari [6]. By using different **co-ordinate systems** to describe probability distributions it is possible to obtain the maximum entropy solution as a projection.

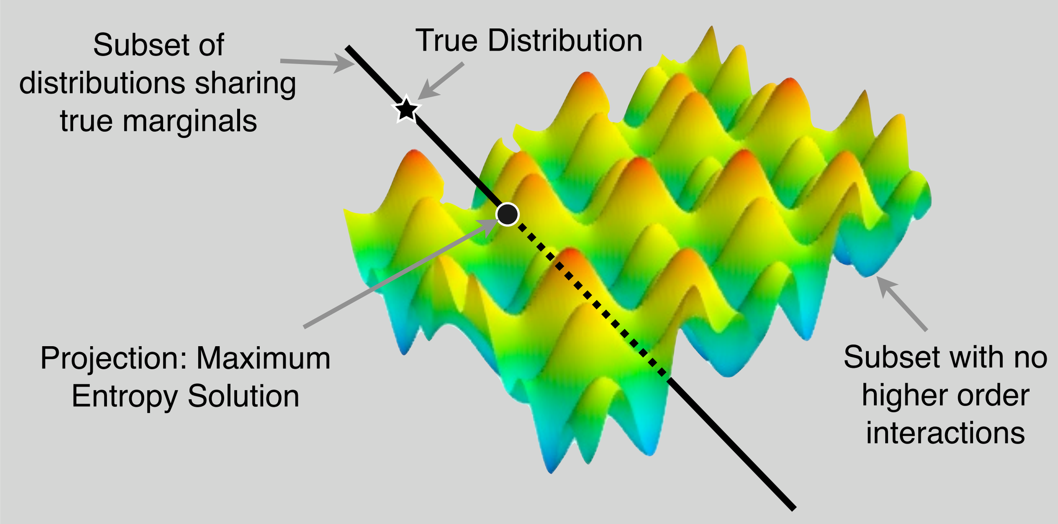

Figure 3: Geometrical approach to maximum entropy.

Using Python allowed us to implement this algorithm for a wider range of parameters than was possible with MATLAB, due to better sparse matrix support and lower level handling of memory issues.

The code is available online [5].

Discussion

The use of Python has several advantages for the work described here.

- Easier to develop reliable and maintainable code during the course of research.
- More powerful array manipulation (broadcasting, views, pass by reference).
- Easier to exploit multiprocessor machines and clusters.
- Easier to extend with C or FORTRAN for performance critical sections of code.
- Familiar to MATLAB users; similar interactive environment.
- Wide range of non-scientific libraries available; easier to create graphical interfaces, create web services, read/write different file formats etc.
- Open source - freely available allowing replication and verification of results without any commercial license.

References

[1] `http://www.python.org`
[2] NumPy and SciPy, `http://www.scipy.org`
[3] S Panzeri, R Senatore, MA Montemurro, RS Petersen (2007) "Correcting for the Sampling Bias in Spike Train Information Measures", *J. Neurophysiol.* **98** (3) 1064-1072
[4] RAA Ince, RS Petersen, DC Swan, S Panzeri (2009) "Python for Information Theoretic Analysis of Neural Data", *Front. Neuroinf.* **3** (4)
[5] `http://code.google.com/p/pyentropy`
[6] SI Amari (2001) "Information Geometry on Hierachy of Probability Distributions", *IEEE Trans. Info. Theory* **47** (5) 1701-1711

This work was supported by the CARMEN e-science project (`www.carmen.org.uk`) funded by the EPSRC (EP/002331/1)