Open source spatial analysis: lessons for research and education from PySAL (Extended Abstract)

Author

Sergio J. Rey, GeoDa Center for Geospatial Analysis and Computation. School of Geographical Sciences and Urban Planning, Arizona State University, *USA*

KEYWORDS : Open source, spatial analysis, education, research

Abstract: This paper explores the intersection of open source with the areas of spatial analysis research and education. Drawing on lessons learned in the development of PySAL: Python Library for Spatial Analysis, it touches on the opportunities and challenges related to the adoption of open source practices and culture. While open source has had major impacts on pedagogy and research in spatial analysis, these are somewhat under-appreciated and at times seen as separate spheres. The paper reconsiders open source spatial analysis teaching and research from an integrated perspective and suggests some possible future developments.

1. Introduction

The open source revolution continues to have major impacts on science and education and the field of spatial analysis is no exception. A number of recent overviews of open source spatial analysis and geographic information science have recently appeared [3, 4, 6, 2, 7, 1] and my intent here is not to provide a similar comprehensive coverage of this area but rather to build on a particular set of themes originally raised by Rey [4]. I do so by drawing on the lessons learned in the development and evolution of the PySAL project [5] as it has intersected with my teaching and research activities.

2. Lessons for Education

It goes against the grain of modern education to teach students to program. What fun is there to making plans, acquiring discipline, organizing thoughts, devoting attention to detail, and learning to be self critical. Alan Perlis

Open source software and practices can have major empowering impacts on pedagogy. The free availability of the software offers a number of advantages in lab based courses. No longer are the students constrained to working in the school laboratory as they can now use the software installed on their own personal laptops, or home desktops, to complete exercises. This also allows for a greater degree of exploration and discovery by the student working by themselves and at their own pace.

These represent potential pedagogical wins for open source in geospatial education. My personal recent experience is that we still have a ways to go before these benefits are fully realized. During the fall of 2011 in my introductory course in GIScience, I decided to use QGIS as the software for the lab component in place of our traditional package of ArcGIS. This was something I had contemplated doing for quite sometime, but I always held back as the feature set and polish of QGIS were not yet at the stage where I felt comfortable doing so. By fall 2011, this had changed as the development of QGIS had reached an impressive state.

To my surprise, this switch was less than well received by the students. Emblematic of the main complaint was the following comment I received on an anonymous teaching evaluation: *I took this course as I heard we would be taught ArcGIS. I don't care about the science and the algorithms underneath the software, I want a job when I leave this class.*

While there were a minority of students who told me they appreciated the introduction to an open source alternative, the vast majority of the students were not happy about the switch. In addition, I received push-back from some of my colleagues who were concerned that not covering ArcGIS threatened relationships with community internship partners that had been carefully cultivated over the years.

I was completely blindsided by these responses and felt a mixture of disappointment and puzzlement. In hindsight, I admit that these potential negative impacts never entered my decision making calculus. At the same time, while I now see that these are pressing concerns, they also raise some important questions regarding the role of geospatial education. On the one hand, the current demands in the labor market for students trained in ArcGIS reflects the reality that previous generations of students we have trained in this software are now in key positions in these agencies and companies. Additionally, many of these agencies have invested much time and resources in their GIS infrastructures and are understandably conservative regarding any changes. But, what about the future? Is our task to train students for today's labor market or to equip them with the skill sets and knowledge so that they are ready for, and can create, the future geospatial labor market?

A second general lesson for geospatial education concerns graduate education and the seemingly the ironic situation of an embarrassment of riches in terms of freely
available high quality programming tools for geospatial research on the one hand, and on the other, a general lack of desire to do any programming. I believe this stems from the challenges facing geography graduate students in that they not only need to acquire knowledge of substantive and methodological areas of the discipline but also somehow become proficient in programming. We have done a fairly poor job on the latter with solutions running from recommending introductory courses in computer science departments to learning on the job as part of a research project. The former is rather inefficient as my experience is geography students taking most introductory computer science classes come away without any idea of how to apply core concepts to geographical problems. The mentoring approach scores higher on this point, but does not really scale well.

Taken these together I think that the reality of the situation of open source and geographic education is currently rather mixed. At the undergraduate level the impact has been much more limited than I would have originally believed, due mainly to the institutional factors raised above. The situation is more evolved at the graduate level. Here I've seen several instances where access to the source code in PySAL has enabled a motivated graduate student to gain a deeper understanding of a particular spatial analytical method. In hindsight, this mixed success may also suggest that a certain level of training and education may be required before the benefits of open source software can be experienced by students.

3. Lessons for Research

Open source has also had major impacts on research in GIScience. In the US, this is
clearly seen in research proposals to federal agencies as increasingly there are requirements that publicly funded projects include data and results management components so that subsequent research projects can replicate and extend funded projects. Having served on review panels for some of these agencies, a clear trend is that open source has been relied upon by many scientists to respond to these requirements. It should be emphasised that open source software offers clear advantages when it comes to replication as there are no longer any "black boxes" that conceal the implementation of a particular method or algorithm [8].

In an open source world it is also more likely that the scientific questions lead the way forward and the software itself is enhanced or modified to address these ques tions. This is in contrast to the proprietary world where the core programs themselves are not malleable. In the past this has led to the choice of research question being constrained by the functionality provided by the software. While the addition of scripting languages and plug-in or toolbox architectures do offer mechanisms for adding new functionality to address new types of questions, the issue of the social returns must also be considered.

For the academic engaged in open source software development, there are a number of challenges that are encountered. A chief one regards the academic evaluation and promotion system which places heavy emphasis on scientific publications. Development, maintenance and documentation of an open source spatial analysis package requires a significant investment in one's time and this cuts into time that could go towards writing and submitting journal articles and books. For a package that becomes widely adopted there is the possibility that scientists who use the package in their own research take care to cite the package, but my hunch is this is done less often than one would like. There have been positive developments in this regard with journals such as Journal of Statistical Software that provide an outlet dedicated to developments in statistical software. It also reflects a shift in attitudes towards scientific software in that it is seen as scholarly work that should come under peer-review.

4. Conclusion

You think you know when you can learn, are more sure when you can write, even more when you can teach, but certain when you can program. Alan Perlis

Although the lessons outlined above treated education and research separately, this was for purposes of exposition only. There are clearly strong synergies between education and research as the quote from Perlis suggests. At the same time, there are some challenges that can hinder these synergies. One of our overriding goals in the development of PySAL has been to keep the level of code readability as high as possible, and here we have relied on the clear syntax of the Python language. We have always felt that the code can serve as a powerful source of information for students interested to learn the exact manner in which a spatial analytical method was implemented. While we have by and large kept to this goal, we have encountered tensions along the way. Keeping the code readable has required that we limit the number of third party libraries that PySAL requires. These libraries are often written in lower level languages such as C, C++, or Fortran and can offer substantial speed gains over pure Python implementations. At the same time the lower level code can be more difficult for the newly initiated spatial analyst to decipher. Faced with this trade-off, we have chosen pedagogy over speed.

4.0.1 Acknowledgments

This research was partially supported by NSF Award OCI-1047917, SI2-SSI: CyberGIS Software Integration for Sustained Geospatial Innovation.

[1] BROVELLI, M., MITASOVA, H., NETELER, M., AND RAGHAVAN, V. Free and open source desktop andWeb GIS solutions. *Applied Geomatics* (2012), 1–2.

[2] HU, Q., CHEN, Y., ZHOU, Y., AND XIONG, C. An overview on open source GIS software with its typical applications. *Geomatics World 1* (2009).

[3] RAMSEY, P. The state of open source GIS. In *Free and Open Source Software for Geospatial (FOSS4G) conference* (2007).

[4] REY, S. J. Show me the code: Spatial analysis and open source. *Journal of Geographical Systems 11* (2009), 191–207.

[5] REY, S. J., AND ANSELIN, L. PySAL: A Python library of spatial analytical methods. In *Handbook of Applied Spatial Analysis*, M. M. Fischer and A. Getis, Eds. Springer, 2010, pp. 175–193.

[6] STEINIGER, S., AND BOCHER, E. An overview on current free and open source desktop gis developments. *International Journal of Geographical Information Science 23*, 10 (2009), 1345–1370.

[7] STEINIGER, S., AND HUNTER, A. Free and open source GIS software for building a spatial data infrastructure. In *Geospatial Free and Open Source Software in the 21st Century: Proceedings of the first Open Source Geospatial Research Symposium, 2009* (2012), E. Bocher and M. Neteler, Eds., Springer, pp. 247–261.

[8] YALTA, A. T., AND YALTA, A. Y. Should economists use open source software for doing research? *Computational Economics 35* (2010), 371–394.