ose-resources Documentation

Development Team

May 13, 2020

CONTENTS

Here we list resources for students or researchers who want to start to solve and estimate structural economic models. Due to our background we focus mainly on microeconomic applications but also try to include a few resources for macroeconomic models.

CHAPTER

ONE

OPENSOURCEECONOMICS

OpenSourceEconomics was founded in 2018 at the University of Bonn as a platform for discussion between students and academic researchers on issues regarding scientific programming. Apart from providing participants with an opportunity to present on and inform themselves about different software solutions during monthly meetings, the organization also offers a collection of open-source model and tool packages. All code is unit tested and can be installed with pip or conda. The model packages are mainly targeted at advanced users who want to estimate large structural models.

The following list provides an overview of the flagship projects:

1.1 Model Packages

1.1.1 respy.

respy is an open-source Python package for the simulation and estimation of a prototypical finite-horizon discrete choice dynamic programming model. It covers a range of models typically used in structural labor economics. Among the most notable examples are the models presented in Keane and Wolpin (1994, 1997).

If you are interested in using respy for your own work, check out the comprehensive online documentation.

1.1.2 grmpy

grmpy is an open-source Python package for the simulation and estimation of the generalized Roy model. It serves as a teaching tool to promote the conceptual framework of the generalized Roy model, illustrate a variety of issues in the econometrics of policy evaluation, and showcase basic software engineering practices. The developers build mainly on scientific contributions by Abbring and Heckman (2007) and Heckman and Vytlacil (2007a, 2007b).

If you are interested in using grmpy for your own work, check out the online documentation.

1.1.3 soepy.

`soepy` is an open-source Python package for the simulation and estimation of a dynamic model of human capital accumulation tailored to the German Socio-Economic Panel (SOEP). It is based on the seminal paper by Blundell, Dias, Meghir and Shaw. (2016).

If you are interested in using soepy for your own work, check out the online repository.

1.2 Tool Packages

1.2.1 estimagic

`estimagic` is a Python package that helps to build high-quality and user friendly implementations of (structural) econometric models. It is designed with large structural models in mind. However, it is also useful for any other estimator that numerically minimizes or maximizes a criterion function (Extremum Estimator). Examples are maximum likelihood estimation, generalized method of moments, method of simulated moments and indirect inference. It is in a very early stage and should not be used for major projects yet. However, we do encourage interested users to try it out, report bugs and provide feedback.

If you are interested in using estimagic for your own work, check out the online documentation.

CHAPTER

TWO

QUANTECON

QuantEcon is an initiative by Thomas J. Sargent and John Stachurski. It is very well suited for inexperienced programmers. Besides explaining some components of structural models in detailed and simple to follow tutorials, it also teaches the basics of Python and Julia.

The most notable links you should check out are:

2.1 Introduction to Python

This will guide you through the set-up of Python on your computer, explain the basic syntax and data structures. It is quite fast paced but self-contained.

2.2 Scientific Python Libraries

The core Python language is not designed for scientific computing. Instead, this is delegated to other packages. Here you will get an overview of the main Python packages you will need:

- `numpy` provides multi-dimensional arrays and functions to work on them. Check out the online documentation.
- `matplotlib` is for simple plotting in Python. While it is good to understand the basics of matplotlib, we would suggest that users use the much simpler seaborn library that builds on top of matplotlib and produces beautiful graphs out of the box. Check out the online documentation.
- `scipy` is a toolkit for scientific computing. Examples are optimization, numerical integration and random number generation. Check out the online documentation.
- `numba` is a just in time compiler for Python that can make Python about as fast as Fortran or C. Seriously, try it out. It's amazing! Check out the online documentation.

2.3 Dynamic Programming

A great introduction into the theory and implementation of simple dynamic programming models.

CHAPTER

THREE

ECON-ARK

The Econ-ARK project provides open-source toolkits for researchers trying to understand how economic and social outcomes result from the actions of heterogeneous individuals.

The primary goals of the project are:

- to make entry into the world of such modeling easy
- to accelerate the development of this kind of modeling for policy-making and academic research
- to increase the openness, replicability, and interoperability of modeling tools.

Econ-Ark is headed by Christopher D. Carroll, Professor of Economics at the Johns Hopkins University.

Originally, `econ-ark` was mainly targeted at macroeconomic applications, but is now extending their tools to support typical microeconomic models.

The main features of Econ-ARK are:

3.1 HARK

`econ-ark`, the Heterogeneous Agents Resources and Toolkit (HARK) is motivated by a sense that quantitative structural modeling of economic agents' behavior (consumers; firms), at present, is roughly like econometric modeling in the 1960s: Lots of theoretical results are available and a great deal can be done in principle, but actually using such tools for any specific research question requires an enormous investment of a scholar's time and attention to learn techniques that are fundamentally not related to economics but instead are algorithmic/computational (in the 1960s, e.g., inverting matrices; now, e.g., solving dynamic stochastic optimization problems). The toolkit is built using the suite of open source, transparent tools for collaborative software development that have become ubiquitous in other fields in the last few years: Github, object-oriented coding, and methods that make it much easier to produce plug-and-play software modules that can be (relatively) easily combined, enhanced and adapted to address new problems.

Check out the online documentation.

3.2 REMARK

This is the resting place for self-contained and complete projects written using the Econ-ARK. The content here should be executable by anyone with a suitably configured computer (see "Installation.md" in this directory).

Types of content include (see below for elaboration):

- Explorations: Use the `econ-ark`/HARK toolkit to demonstrate some set of modeling ideas
- Replications: Attempts to replicate the results of published papers written using other tools
- Reproductions: Code that reproduces the results of some paper that was originally written using the toolkit

CHAPTER

FOUR

MISCELLANEOUS

In this section we present some additional resources as well as a short list of relevant literature contributions.

4.1 Scientific Programming

4.1.1 Anaconda Distribution

The open-source Anaconda Individual Edition is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 19 million users worldwide, it is the industry standard for developing, testing, and training on a single machine.

4.1.2 Jupyter Lab

Jupyter Lab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab is extensible and modular: write plugins that add new components and integrate with existing ones.

Resources

- JupyterLab — A Next Gen Python Data Science IDE
- JupyterLab: The Next Generation Jupyter Web Interface
- Guide to R and Python in a Single Jupyter Notebook

4.1.3 Version Control

The Introduction to Git and GitHub course on coursera offers a well rounded intro into version control using Git and GitHub. It will teach you all the fundamentals as well as more advanced features.

The GitKraken video tutorial series provides you with the knowledge needed to start using Git. It covers the absolute basics as well as advanced procedures involving Git. The course consists of 15 concise videos split into three difficulty levels, all of which can be watched in under an hour.

4.1.4 SciPy

SciPy is a Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages: `numpy`, `scipy`, `matplotlib`, and `pandas`.

4.1.5 SciPy Lecture Notes

The Scipy lecture notes provide an excellent starting point for everyone interested in scientific programming in Python. They cover the main scientific packages of the Python ecosystem, namely `numpy`, `scipy` and `matplotlib`. Each chapter corresponds to a 1 to 2 hour course with the level of expertise increasing from beginner to expert.

Resources

- Pandas is for Everyone
- Introduction to Numerical Computing with NumPy
- Exploratory data analysis in python

4.1.6 Software Carpentry

Software Carpentry teaches researchers the computing skills they need to get more done in less time and with less pain. They have a lot of useful lessons on many different topics like Python and R programming or the Unix shell.

4.1.7 statsmodels

`statsmodels` is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. An extensive list of result statistics are available for each estimator. The results are tested against existing statistical packages to ensure that they are correct.

Check out the online documentation.

4.1.8 Textbooks

Ramalho, L. (2015). Fluent Python. Clear, concise, and effective programming. O'Reilly Media, Inc., Sebastopol, CA.

Rossant, C. (2018). IPython interactive computing and visualization cookbook. Packt Publishing, Birmingham, England.

VanderPlas, J. (2016). Python data science handbook. O'Reilly Media, Inc., Sebastopol, CA.

4.2 Software Engineering

4.2.1 Testing

Automated testing ensure the quality of our research software.

- Effective Python Testing With Pytest

4.2.2 Documentation

Proper documentation ensures that the research software is used as intended and helps in the recruitment of new contributors and users.

- Read the Docs: Documentation Simplified

4.2.3 Continuous Integration

Continuous Integration is a development practice where developers integrate code into a shared repository frequently, preferably several times a day. Each integration can then be verified by an automated build and automated tests.

- Continuous Integration Essentials
- GitHub Actions

- Setting up continuous integration using GitHub Actions

CHAPTER

FIVE

INFRASTRUCTURE

5.1 ose-server

Our group maintains a small virtual sever for teaching and prototyping purposes. It can utilize up to 12 Cores and 16GB of RAM, depending on resource allocation within the entire network at the time. The virtual server will either run on Xeon E7-8837 or Xeon E5-2699 based servers which the IT service group of the economcis departement maintains. For any related inquiries, please reach out to us on Zulip.

CHAPTER

SIX

BIBLIOGRAPHY

Abbring, J. H. and Heckman J. J. (2007). Econometric evaluation of social programs, part III: Distributional treatment effects, dynamic treatment effects, dynamic discrete choice, and general equilibrium policy evaluation. In J. J. Heckman & E. E. Leamer (Eds.), *Handbook of Econometrics* (Vol. 6B, p. 5145 - 5303). Elsevier Science.

Aguirregabiria, V. and Mira, P. (2010). Dynamic discrete choice structural models: A survey. *Journal of Econometrics*, 156(1): 38 - 67

Blundell R., Dias, M. C. , Meghir C. and Shaw, J. (2016). Female labor supply, human capital, and welfare reform. *Econometrica*, 84(5): 1705 - 1753.

DellaVigna, S. (2018). Chapter 7 - Structural behavioral economics. In B. D. Bernheim, S. DellaVigna and D. Laibson (Eds.), *Handbook of Behavioral Economics - Foundations and Applications 1* (Vol. 1, p. 613 - 723), Elsevier.

Heckman J. J. and Vytlacil E. J. (2007a). Econometric evaluation of social programs, part I: Causal models, structural models and econometric policy evaluation. In J. J. Heckman & E. E. Leamer (Eds.), *Handbook of Econometrics* (Vol. 6B, p. 4875 - 5143). Elsevier Science.

Heckman J. J. and Vytlacil E. J. (2007b). Econometric evaluation of social programs, part II: Using the marginal treatment effect to organize alternative econometric estimators to evaluate social programs, and to forecast their effects in new environments. In J. J. Heckman & E. E. Leamer (Eds.), *Handbook of Econometrics* (Vol. 6B, p. 4779 - 4874). Elsevier Science.

Keane, M. P. and Wolpin, K. I. (1994). The solution and estimation of discrete choice dynamic programming models by simulation and interpolation: Monte Carlo evidence. *The Review of Economics and Statistics*, 76(4): 648 - 672.

Keane, M. P. and Wolpin, K. I. (1997). The career decisions of young men. *Journal of Political Economy*, 105(3): 473 - 522.

Keane, M. P., Todd, P.E. and Wolpin, K. I. (2011). Chapter 4 - The structural estimation of behavioral models: Discrete choice dynamic programming methods and applications. In O. Ashenfelter and D. Card (Eds.), *Handbook of Labor Economics* (Vol. 4, p. 331 - 461). Elsevier Science.

If you know another good tutorial or python package for structural econometrics, please let us know, so we can add it. If you are the author of one of the packages or tutorials we list, and think our description is outdated or not accurate, please open an issue or submit a pull request on GitHub.