'math+econ+code' masterclass on optimal transport and economic applications

Alfred Galichon (NYU+ScPo)

January 2022
Getting started

Preliminary stuff

- Schedule: Jan 10-14,2022. Classes meet 2pm-6pm Paris time / 8am-noon New York time.
- In addition, five advanced lectures will be taught online on a monthly basis, Jan-May 2022, 4-6pm Paris time / 10am-noon NY time: January 28 2022, February 18 2022, March 25 2022, April 15 2022, May 20 2022.
- Format: online.
- Course description: https://www.math-econ-code.org/jan-22
- Course material: https://github.com/math-econ-code/mec_optim

 Support from NSF DMS-1716489 and ERC CoG-866274 EQUIPRICE grants are acknowledged.

Introducting the team

- **Alfred Galichon**, instructor: professor of economics and of mathematics at NYU, and principal investigator, ERC grant EQUIPRICE, SciencesPo (ag133@nyu.edu)
- **Clement Montes** (SciencesPo), masterclass' graduate assistant (clement.montes@sciencespo.fr)
- **Antoine Chapel** (SciencesPo), research assistant
- **Giovanni Montanari** (NYU), research assistant
- Last but not least, **you** are now part of the team!

A group photo

- Put on your best smile anyone!

Launching docker while we chat

- See instructions in the github.

Remarks on this course

- Targeting Math and Econ students. Self-contained for both audiences
- Teaching format: 5 days; each day = four hours, 1/2 economics modeling, 1/2 Python coding. Plus 5 "special lectures" to deepen the participants' mathematical/technical understanding.
- Programming: our demos will be done Python and the support will be in these languages only, but you are welcome to use the language of your choice e.g. R, Matlab, C++, Julia...
- Parts of this course may be recorded, and the recording may be publicly broadcasted in part or in full. As a participant to the masterclass, you have consented that your voice and image should be broadcasted.
- To stay engaged with the course and interact optimally with other participants, we recommend you keep your camera on and turn your microphone off except when you wish to speak. You have the option to turn your camera and your microphone off at any time.
- Questions?

This course

- is focused on models of demand, matching models, and optimal transport methods, with various applications pertaining to labor markets, economics of marriage, industrial organization, matching platforms, networks, and international trade, from the crossed perspectives of theory, empirics and computation
- will introduce tools from economic theory, mathematics, econometrics and computing, on a needs basis, without any particular prerequisite other than the equivalent of a first year graduate sequence in Economics or in Applied Mathematics
- aims at providing a bridge between theory and practice, and has an unusual teaching format: each teaching "block" will be made of a mix of theory and coding (in Python), based on an empirical application related to the theory just seen.
- is closer to cooking lessons than to traditional lectures.
- is complementary to the m+e+c_equil masterclass (taught in June), but can be taken independently. Overlap is limited.

Readings

The closest reference is:

- Alfred Galichon, *Optimal Transport Methods in Economics*, Princeton University Press, 2016.

Additional references include:

- Pierre-André Chiappori, *Matching with Transfers: the Economics of Love and Marriage*, Princeton University Press, 2017.
- Gabriel Peyré and Marco Cuturi, *Computational Optimal Transport: With Applications to Data Science*, Foundations and Trends in Machine Learning: Vol. 11: No. 5-6, pp 355-607, 2019 (Available online for free)
- Cédric Villani, *Optimal Transport Old and New*, Springer, 2008.
- Filippo Santambrogio, *Optimal Transport for Applied Mathematicians – Calculus of Variations, PDEs and Modeling*, Birkhauser, 2015.

Outline: main week

- Day 1: Linear programming fundamentals
  - The diet problem
  - Optimal assignments and the Becker model
  - Intro to the toolchain: docker, github, jupyter
  - The Gurobi library
- Day 2: The optimal assignment problem
  - Entropic regularization, the IPFP algorithm
  - Semi-discrete case, Aurenhammer's algorithm
- Day 3: Discrete choice models
  - Discrete choice models and their inversion
  - Generalized linear models
  - Regularization and model selection: LASSO, elastic nets, etc.
  - The scikit-learn library
- Day 4: Demand estimation
  - The pure characteristics model
  - Berry-Levinsohn-Pakes' random coefficient logit model
  - The pyblp package
- Day 5: Matching estimation and the gravity equation
  - The Choo and Siow model and its offsprings
  - Matching estimation and the gravity equation

Outline: advanced lectures

The tentative plan for the advanced lectures in the semester is as follows. All run 4-6pm Paris time / 10am-noon NY time.

- Special Lecture 1 (January 28 2022): Networks 1
  - Introduction to networks: topology, Markov chains
  - The networkx and the osmnx libraries
- Special Lecture 2 (February 18 2022): Dynamic Programming 1
  - Finite-horizon dynamic programming
  - Backward and forward induction; linear programming formulation
- Special Lecture 3 (March 25 2022): Networks 2
  - Min-cost flow problems, shortest path problems
  - The Bellman-Ford and Dijkstra algorithms
- Special Lecture 4 (April 15 2022): Dynamic Programming 2
  - Infinite-horizon dynamic programming; stationarity; unit discount rate
  - Linear programming and LCP formulation
- Special Lecture 5 (May 20 2022): Quantile Methods
  - Quantiles and optimal transport; classical quantile regression
  - Vector quantile regression

Getting started

- From the command line, after installing docker, pull the image using: `docker pull alfredgalichon/mec_optim:2022-01`
- Create a local repository (here `/Users/alfre/Desktop/docker-tmp`)
- Place the Gurobi WLS license somewhere on your local machine (here `//c/Users/alfre/gurobi-wls-license/gurobi.lic`)
- Next, run the container using:
  `docker run -it --rm -p 8888:8888 -v //c/Users/alfre/Desktop/docker-tmp:/src/notebooks/my-work -v //c/Users/alfre/gurobi-wls-license/gurobi.lic:/opt/gurobi/gurobi.lic:ro mec_optim`
- Copy the URL and paste it on in your browser.

Other ways to run the notebooks

- Locally and without the container, using `anaconda` and `git`.
  - Simple, but some may run into version issues
- On Google Colab / mybinder: https://colab.research.google.com/
  - The simplest options, but with some limitations (limited lifetime and memory, some dependencies may not install)
- Run the container on GCP / AWS:
  - More powerful and flexible option, but also more involved. We will demonstrate this option later this week.