Computational Economics

- This is a 14-week long Ph.D. class on computational economics, held at the Department of economics at the University of Pennsylvania in the spring term 2023.

Class enrollment on the Nuvolos Cloud

- All lecture materials (slides, codes, and further readings) will be distributed via the Nuvolos Cloud.
- To enroll in this class, please click on this enrollment key, and follow the steps.

Purpose of the lectures

- This course is intended to confront Ph.D. students in economics, finance, and related fields with recent tools developed in applied mathematics, machine learning, computational science, and the computational economics literature to solve and estimate (dynamic stochastic) economic models.

- This course consists of three large topical blocks:

    - Weeks 1-3: Dynamic Optimization, the underlying numerical operations one needs to master (numerical integration, differentiation, constrained optimization, solving nonlinear sets of equations), and the classical solution algorithms (value function iteration, time iteration).
    - Weeks 4-9: A comprehensive overview of state-of-the-art machine-learning based methods to solve and estimate dynamic stochastic models.
    - Weeks 10-13: Parallel and high-performance computing.

- The methods presented in the lectures will be showcased in the context of applications in macroeconomics, finance, and climate-change economics.

- The lectures will be interactive, in a workshop-like style, that is, a mix of theory and actively playing with code examples (delivered in Python and deployed on a cloud computing infrastructure).

- The students are encouraged to bring their laptopts to the lectures.

Prerequisites

- Basic econometrics
- Basic programming in Python (see this link for a thorough introduction)
- Basic calculus and probability (Mathematics for Machine learning provides a good overview of skills participants are required to be fluent in).

Tentative schedule

Week 1 - Tuesday, Jan 17th, 2023

- A general intro to computational economics, and the organization of the semester.
- A primer to Python and its basic functionality.
- Introduction to Nuvolos, and github.

Week 2 - Tuesday, Jan 24th, 2023

- The basics for solving dynamic (stochastic) models numerically.
- Dynamic programming.
- Value function iteration algorithm.
- Time iteration algorithm.
- The basic operations one has to perform to solve models recursively

    - numerical approximation
    - (automatic) differentiation).
    - interpolation
    - constrained optimization

Week 3 - Tuesday, Jan 31th, 2023

- The curse of dimensionality
- Sparse grids (SG)
- Adaptive sparse grids (ASG)
- High-dimensional dynamic model representation (HDMR)
- Analytical examples
- Solving international real business cycle models with ASG and HDMR

Week 4 - Tuesday, Feb 7th, 2023

- A general introduction to machine learning
    - supervised, unsupervised, reinforcement learning
    - Cost functions, likelihood,
    - hyper-parameters,
    - numerical optimization (e.g., gradient descent, stochastic gradient descent,…)
- Deep learning basics
- Multi-layer perceptron
- Feed-forward networks
- Network training - SGD
- Error back-propagation
- Some notes on overfitting
- Throughout lectures - hands-on: Perceptron, gradient descent, Artificial neural networks: a simple MLP implementation and examples of applications
- An introduction to Tensorflow and Tensorboard (and Pytorch of time permits)

Week 5 - Tuesday, Feb 14th, 2023

- Deep learning methods for solving dynamic models (I)
    - Deep Equilibrium Nets
    - Deep Structural Estimation
    - Exploiting Symmetry
    - KrSm with DL.

Week 6 - Tuesday, Feb 21th, 2023

- Deep learning for solving models (II)
    - Solving continuous-time models/PDEs with Deep learning
- Deep learning for time-series data (Recurrent Neural Networks, LSTMs, Autoencoders)

Week 7 - Tuesday, Feb 28th, 2023

- Integrated assessment models (a.k.a merging economics with climate physics)
- Uncertainty quantification (showcased in the context of integrated assessment models)
- A brief midterm presentation (30% of final grade)

Spring break - March 4th - March 12th

Week 8 - Tuesday, March 14th, 2023

- Basics of Gaussian Process (GP) Regression
- Noise-free kernels
- Kernels with noise
- GP classification
- Throughout lectures - hands-on: Basics on GPs, Option-pricing examples

Week 9 - Tuesday, March 21th, 2023

- The curse of dimensionality and how to deal with it in the contect of GPs (e.g., active subspaces)
- Gaussian mixture models (unsupervised ML)
- Bayesian active learning
- Dynamic programming/optimal control with GPs
- An outlook to frontier topics of GPs (Limitations of GPs and “big data”/scalable GPs
- Throughout lectures - hands-on: A growth model solved with GPs and dynamic programming

Week 10 - Tuesday, March 28th, 2023

- A primer on C++ (a compiled promgramming language)
    - Basics on C++ (compilation, variable declarations, data types, header files, branching (if/else if/else), loops, pointers, arrays, references, static memory allocation, dynamic memory allocation, functions, pass by value, pass by reference, namespaces)
    - Preprocessing/compiling/linking
    - Static libraries
    - Productivity: Introduction to automated builds: Make and Cmake
    - pybind11: Mixing Python and C++ for performance and easiness

Week 11 - Tuesday, April 4th, 2023

- Introduction to high-performance computing & parallel programming (basic concepts, hardware, and terminology)
- OpenMP (shared memory parallelization; examples in C++)
- basics on (serial) code optimization

- (AVX) vectorization
- directives
- runtime library
- compiling OpenMP
- data scoping
- Slurm
- racing conditions
- loops parallelized
- sections
- reductions
- nested loops

Week 12 - Tuesday, April 11th, 2023

- MPI (distributed memory parallelization; examples in C++)
- What is MPI ("hello world" in MPI)
- point-to-point communication (basics on how to send and receive messages)
- Collective communication (Max., Sum.,…)

Week 13 - Tuesday, April 18th, 2023

- Parallel and High-performance computing with Python
    - Numba (Numba is an open source JIT compiler that translates a subset of Python and NumPy code into fast machine code)
    - Threading in Python
    - MPI4PY
    - JAX

Week 14 - Tuesday, April 25th, 2023

- Final presentation, hand-in of project.

Teaching philosophy

Lectures will be interactive, in a workshop-like style, using Python, scikit learn, Tensorflow, and TFP on Nuvolos, a browser-based cloud infrastructure in which files, datasets, code and applications work together, in order to directly implement and experiment with the introduced methods and algorithms.

Lecturer

- Simon Scheidegger (HEC, University of Lausanne)

Contacts

- Simon Scheidegger: simon.scheidegger@unil.ch
- Nuvolos Support: support@nuvolos.cloud

Assessment

Grading will be based on 20% on active class participation, 20% on a short, individual presentation in week 7 that demonstrates an application of one of the methods learned so far in the context of an own application, and 60% on a graded take home project, plus a 15' presentation. The final take home project will be proposed and developed by small groups consisting of a maximum of 3 persons, and will consist of applying the lecture content to interesting applications (e.g., by replicating part of a paper). The deliverable of the final project is a short write-up (6-10 pages maximum without references), the data set (if any), and the code on which the presented results in the report were based. The participants can work alone, or in teams of two.