Computational Economics and Finance

Some Ideas for Term Papers

March 31, 2015

This document provides a list of suggestions for term papers covering a wide range of research areas. Please make sure that you receive an OK from the research assistants on your selected topic by May 26, 2015.

Macro-Finance Applications

Project MF1: Information Structures in Asset Pricing Models with Long-Run Risk

Asset pricing models including long-run risk (see Bansal and Yaron (2004)) have made great strides in solving many puzzling behaviors on financial markets like the high equity premium, volatile prices-dividend ratios, predictability patterns and bond price dynamics. One drawback of the original model is, that it requires an upward sloping equity term structure which is contrary to what we find in the data. Croce, Lettau and Ludvigson (2014) therefore present a bounded rationality model with limited information that produces a significant equity premium even with a downward sloping equity term structure. Reproduce the numerical results from the paper using the online appendix available at http://www.econ.nyu.edu/user/ludvigsons/Appendixlmi.pdf.

Project MF2: Long-Run Risk and Asset Prices

Asset pricing models including long-run risk (see Bansal and Yaron (2004)) have made great strides in solving many puzzling behaviors on financial markets like the high equity premium, volatile prices-dividend ratios, predictability patterns and bond price dynamics. Use the log-linearization as described in Bansal, Kiku and Yaron (2012) and Beeler and Campbell (2012) to reproduce the results from the original 2004 as well as the follow up 2012 paper.

Function Approximation

Project FA1: Function Approximation

The solution to many dynamic economic models does not come as a vector of real numbers, but rather as a function taking as arguments real numbers and mapping them to another vector of
reals; a very important example is the solution $V(x)$ to the Bellman equation

$$V_t(x) = \max_y \{ \pi(x, y) + \beta E[V_{t+1}(x')|x, y] \}$$

which can be solved for by backwards iteration (for finite horizon problems) or projection and collocation methods (for infinite horizon methods). Since even a one-dimensional function is generally an infinite-dimensional object, it generally cannot be represented exactly on a computer. (Imagine having to store its graph, so every pair $(x, y)$ for $x \in \mathbb{R} \ldots$) Therefore, functions are approximated using finitely many parameters. The task of this project is to apply different approximation schemes to sample functions, including polynomial, piecewise polynomial, and spline approximation; the sample of approximated functions should be heterogeneous, meaning that smooth functions as well as functions with singularities (kinks, or even discontinuities) should be tested. The approximation schemes used for the analysis should be well introduced, and efficient and stable methods for their construction should be used. Also, the methods should be compared in terms of accuracy, speed, and convergence (which means how fast does the approximation error become smaller as a function of the number of interpolation nodes). Most of the functions can be one-dimensional, but it would be nice to see at least one two-dimensional example and a brief discussion of what changes when moving from one to many dimensions.

**Project FA2: Error propagation in Recursive Likelihood Function Approximation**

The paper of Reich (2014) develops a method to efficiently estimate the parameters of dynamic discrete choice models with serially correlated unobserved state variables by maximum likelihood estimation. In this project, you are asked to replicate a small part of it, namely the recursion in Algorithm 1, without having to solve the full model. However, your task is to carry out numerical experiments to analyse how much the recursion algorithm is subject to numerical error, and, more important, its propagation throughout the recursion. For this purpose, you simulate random errors and analyse how the error propagates by measuring mean and variance of the resulting aggregated errors and comparing them to the individual error distribution you originally simulated. If insightful, the analysis can be extended to formally derive an error bound or a convergence rate. If you are interested in this project, please contact the author (Gregor Reich) in an early stage to get a recent version of the paper and additional help.

**Applications of Nonlinear Programming**

**Project NP1: Moral Hazard in Compensation Contracts**

Replicate the numerical results reported in Tables 1 to 9 in Armstrong et al. (2010) using Mathematica. This paper presents numerical solutions for two complementary generalized principal-agent models that incorporate features observed in real-world contracting environments (e.g.,
agents with power utility and limited liability, lognormal stock price distributions, and stock options) as mathematical programs with equilibrium constraints (MPEC).

**Project NP2: Index Fund Optimization Models**

Implement and solve the optimization models for international index funds of chapter 7.4 in Zenios (2007). In particular implement an integrated indexation model, a non-integrated model and an operational model. For the generation of the scenarios you can use the bootstrapping approach of chapter 9.4.1 in Zenios (2007).

**Mathematica Projects**

**Project M1: Dynamic Tool: Datasets & Regressions**

Use Mathematica to develop an interactive tool that allows to import any dataset (.xls/.xlsx/.csv) and automatically detects variables. Furthermore, the tool should allow to choose a combination of variables (or transformations of these variables) to dynamically perform linear regressions using `LinearModelFit` and display the regression equation, the parameter table, as well as the residual plot. This project involves extensively using `Dynamic`. The GUI does not need to be designed. A first prototype of the tool already exists.

**Project M2: UZH-Web (Webcrawler)**

Write a webcrawler using Mathematica (and import.io) that analyzes the www and fetches urls for all UZH webpages, and their connections to illustrate the UZH network. The webcrawler should yield (1) a list of all pairs (urls, page ID) in the network, (2) a list of all links (page ID to page ID) in the network, (3) the corresponding adjacency matrix, and (4) a graph showing the network. Mathematica has built-in functions to generate graphs, adjacency matrices, and a package to directly import data from import.io ([https://github.com/KristianHolsheimer/import.io_mathematica_package](https://github.com/KristianHolsheimer/import.io_mathematica_package)). The challenge of this project is to handle and store the data efficiently.

**Project M3: UZH-Web (PageRank and Search)**

Given a set of links (page ID to page ID) or an adjacency matrix and the corresponding urls, use Mathematica to calculate the PageRanks for all pages in the network. Build an application to dynamically display the top ten pages as well as a snippet of the network graph for given free form search (inputfield). Mathematica has built-in functions to calculate PageRanks and to visualize graphs and networks. The challenge of this project is to handle the data efficiently such that the applications run fluently.
Project M4: R vs. Mathematica

Define a set of basic statistical problems. For this set, implement the solutions in (1) R, (2) Mathematica, (3) the R-code in Mathematica (RLink) and compare the performance of the three implementations as well as the length of the code.

Project M5: UZH-Map

Using Mathematica and import.io, fetch the addresses for all buildings, faculties, departments, and chairs of the UZH. Visualize the geo positions of the addresses on a map of Zurich. Build an interactive tool that dynamically highlights all locations of a particular faculty/department and calculates the distance between any two locations and shows the directions on the map.

Project M6: Machine Learning: Credit Cards

Use the Mathematica built-in functions for data classifications to determine from a large set of customers and their credit card transaction, which customers are “good” debt and which ones are likely to default on their credit card payments. We have a large data set for you.

References


