MATLAB files for Pareto Extrapolation

Émilien Gouin-Bonenfant*[1] and Alexis Akira Toda†[2]

[1]Department of Economics, Columbia University
[2]Department of Economics, University of California San Diego

1 Introduction

This note explains the functionalities in the MATLAB package `PE`, which implements the Pareto Extrapolation algorithm. The user should use these files at their own responsibility. Whenever you use these codes for your research, please cite Gouin-Bonenfant and Toda (2018).

2 Package content

There are three main functionalities for Pareto extrapolation:

- `getZeta.m`
- `getQ.m`
- `getTopShares.m`

In addition, `example.m` contains a simple example.

2.1 Computing Pareto exponent

`getZeta.m` computes the Pareto exponent using the Beare and Toda (2017) formula. The usage is

```
zeta = getZeta(PS,PJ,p,G,zetaBound)
```

where

- `PS` is the $S \times S$ transition probability matrix of exogenous states indexed by $s = 1, \dots, S$,

*Email: eg3041@columbia.edu.
†Email: atoda@ucsd.edu.

- `PJ` is the $S^2 \times J$ matrix of conditional probabilities of transitory states indexed by $j = 1, \ldots, J$,
- `p` is the birth/death probability $p \in [0, 1)$,
- `G` is the $S^2 \times J$ matrix of gross growth rates, and
- `zetaBound` is a vector $(\underline{\zeta}, \bar{\zeta})$ that specifies the lower and upper bounds to search for the Pareto exponent (optional).

The S^2 rows in `PJ` and `G` should be ordered such that

$$(s, s') = (1, 1), \ldots, (1, S); \ldots; (s, 1), \ldots, (s, S); \ldots; (S, 1), \ldots, (S, S).$$

If `PS` $= P = (p_{ss'})$, `PJ` $= (\pi_{ss'j})$, and `G` $= (G_{ss'j})$, then the Pareto exponent $z = \zeta$ is the solution to

$$(1 - p)\rho(P \odot M(z)) = 1,$$

where ρ is the spectral radius and $M(z) = (M_{ss'}(z))$,

$$M_{ss'}(z) = \sum_{j=1}^{J} \pi_{ss'j} G_{ss'j}^z,$$

and $\odot$ is the Hadamard (entry-wise) product.

`PJ` can be either $1 \times J$, $S \times J$, or $S^2 \times J$. If it is $1 \times J$, it assumes $\pi_{ss'j} = \pi_j$ depends only on j. If it is $S \times J$, it assumes $\pi_{ss'j} = \pi_{sj}$ depends only on (s, j).

`G` can be either $S \times J$ or $S^2 \times J$. If it is $S \times J$, it assumes $G_{ss'j} = G_{sj}$ depends only on (s, j).

2.2 Computing joint transition probability matrix

`getQ.m` computes the $SN \times SN$ joint transition probability matrix $Q = (q_{sn,s'n'})$ and the stationary distribution $\pi = (\pi_{sn})$ for the exogenous state s and wealth. The usage is

```
[Q,pi] = getQ(PS,PJ,p,x0,xGrid,gstjn,Gstj,zeta,h)
```

where

- `PS`, `PJ`, `p` are as above,
- `x0` is the initial wealth of newborn agents,
- `xGrid` is the $1 \times N$ grid of wealth (size variable) w_n,
- `gstjn` is the $S^2 \times JN$ matrix of law of motion for wealth $g_{ss'j}(w_n)$,

- `Gstj` is the $S^2 \times J$ matrix of asymptotic slopes of law of motion $G_{ss'j}$ (optional),
- `zeta` is the Pareto exponent (optional), and
- `h` is the grid spacing for hypothetical grid points (optional).

The JN columns of `gstjn` must be ordered such that the first N columns correspond to $j = 1$, the next N columns correspond to $j = 2$, and so on. `Gstj` is the same as `G` in `getZeta.m`. If unspecified, it uses the slope of the law of motion between the two largest grid points. If `zeta` is unspecified, it calls `getZeta.m` to compute. If `h` is unspecified, it uses the distance between the two largest grid points.

`gstjn` can be either $S \times JN$ or $S^2 \times JN$. If it is $S \times JN$, it assumes $g_{ss'j}(w_n) = g_{sj}(w_n)$ depends only on (s, j, n).

2.3 Computing top wealth shares

`getTopShares.m` computes the top wealth shares. The usage is

```
topShare = getTopShares(topProb,wGrid,wDist,zeta)
```

where

- `topProb` is the vector of top probabilities to evaluate top shares,
- `wGrid` is the $1 \times N$ vector of wealth grid,
- `wDist` is the $1 \times N$ vector of wealth distribution, and
- `zeta` is the Pareto exponent (optional).

Given the stationary distribution π computed using `getQ.m`, one can compute the wealth distribution as $\pi_n = \sum_{s=1}^{S} \pi_{sn}$. If `zeta` is unspecified, `getTopShares.m` uses spline interpolation to compute top wealth shares.

References

Brendan K. Beare and Alexis Akira Toda. Geometrically stopped Markovian random growth processes and Pareto tails. 2017. URL `https://arxiv.org/abs/1712.01431`.

Émilien Gouin-Bonenfant and Alexis Akira Toda. Pareto extrapolation: Bridging theoretical and quantitative models of wealth inequality. 2018. URL `https://ssrn.com/abstract=3260899`.