rgamer

A package to help students learn game theory using R

Yoshio Kamijo (Waseda U)

Yuki Yanai (Kochi U of Tech)

Motivation

- Free students from difficult calculations
  - Let them spend more time examining the validity and interpretation of results
  - Help them gain a deeper understanding of game theory through simulations

What you can do with rgamer

- Normal-form games
- Extensive-form games
- Simulating learning processes
- Matching: one-to-one or many-to-one matchings and more!

Normal-form games

Stag hunt

Tab 1. Payoff matrix created by `solve_nfg()`, where best responses are marked by ^

		Yanai		
	strategy	Stag	Hare	
Kamijo	Stag	10^, 10^	0, 8	p
	Hare	8, 0	7^, 7^	$1-p$
		q	$1-q$	

R code

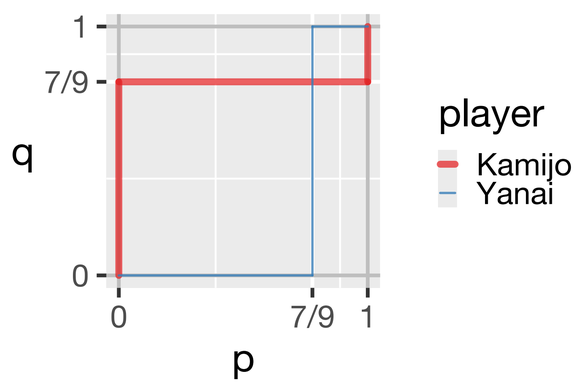

Fig 1. Best-response plot created by `solve_nfg()`

Cournot competition

Firm 1: $\pi_1(x_1, x_2) = (40 - x_1 - x_2)x_1 - 20x_1$

Firm 2: $\pi_2(x_1, x_2) = (40 - x_1 - x_2)x_2 - 24x_2$

R code

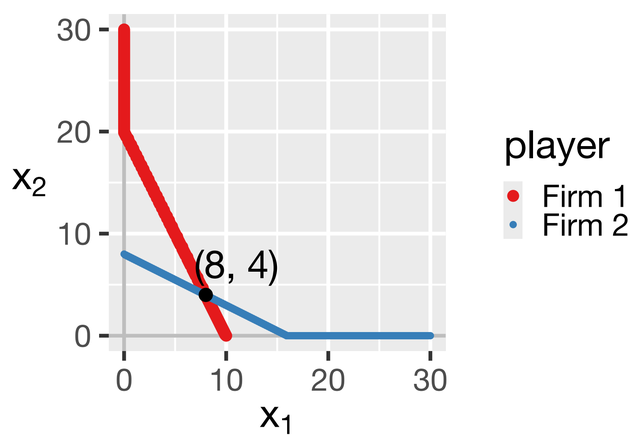

Fig 2. Best-response plot with a NE shown, created by `solve_nfg()`

Sequential games of perfect info

Price competition

R code

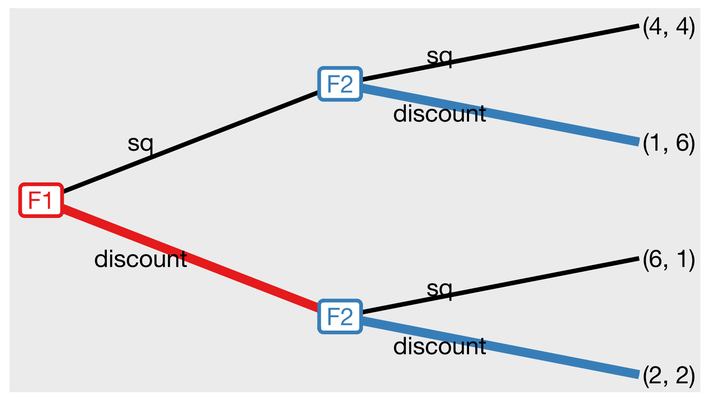

Fig 3. Game tree of a price competition with backward-induction solution marked, created by `solve_efg()`

Tab 2. Payoff matrix created by `to_matrix() |> solve_nfg()`

		F2			
	strategy	(sq, sq)	(sq, discount)	(discount, sq)	(discount, discount)
F1	(sq)	4, 4	4^, 4	1, 6^	1, 6^
	(discount)	6^, 1	2, 2^	6^, 1	2^, 2^

Sequential games of imperfect info

Ultimatum game

R code

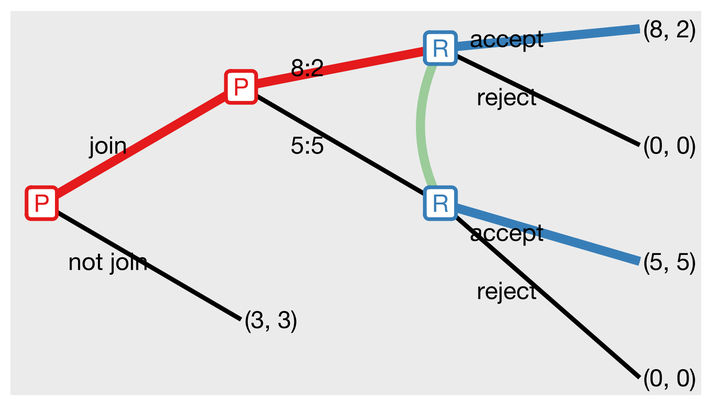

Fig 4. Game tree of an ultimatum game with a sub-game perfect equilibrium marked, created by `solve_efg()`

Simulating learning processes

Reinforcement learning in Cournot competition

R code

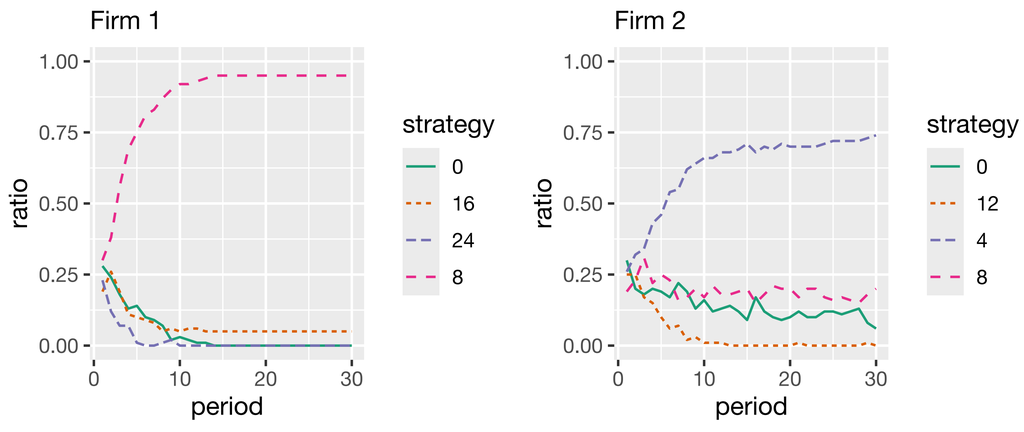

Fig 5. Simulation results of the reinforcement learning in Cournot competition, implemented by `sim_learning()`