Evolutionary and RL models of exchange

N. Botta

- Evolutionary models
- RL models
- Learning
- Agents as types
- Modeling perspective

Evolutionary models: basic ideas

- **Agent-specific prices** evolve according to a rule that mimics natural selection / learning by imitation:
  - First, the fitness of the agents' prices w.r.t. trading with fixed prices-dependent rules is estimated in a *trading game*.
  - Then, agents with low trading fitness adopts the prices of *similar* agents with higher fitness and the prices undergo a random mutation.
- **No individual agents.** Individual economic entities are represented by sizeable *populations* of similar agents.
- **Population-specific utilities, trading rules.** All agents of a population share the same *utility* function and trade according to the same rules in order to maximize this utility.

Evolutionary models: basic ideas

- **No learning-by-doing.** Agents cannot improve their strategies while trading with other agents. They learn at the end of a game through imitation and mutation.
- **Unique optimal stocks.** For each system of prices $p \in G \to \mathbb{R}_{>0}$ and initial stock $x_0 \in G \to \mathbb{R}_{\geq 0}$ there exists a unique stock $x^*\, x_0\, p$ which maximizes the utility of a population under the constraint $(x^*\, x_0\, p) \cdot p \leq x_0 \cdot p$.
- **Utility maximizing agents.** The agents of a population knows how to compute their optimal stock $x^*\, x_0\, p$ for arbitrary agent-specific x_0, p.
- **Trading game.** A trading game consists of sequences of *elementary bilateral trades*.

Evolutionary models: trading

- **Elementary bilateral trade.** An EBT of $g, g' \in G$ between two agents consists of three steps:
  - The agents issue demands $d_{g'}, d'_g \in \mathbb{R}_{\geq 0}$ and offers $o_g, o'_{g'} \in \mathbb{R}_{\geq 0}$ on the basis of population-specific *offer/demand policies*. These depends on the agent's prices explicitly and implicitly through its optimal stocks.
  - The offers and the demands of are matched by a *trade resolving policy*. The policy depends on the prices of the interacting agents explicitly and implicitly through their offers and demands. It return the quantities $\delta_g, \delta_{g'}$ to be exchanged.
  - The agents exchange $\delta_g, \delta_{g'}$ and increment their fitness by the utilities of the new stocks.

RL models: basic ideas

- **No utility maximizing agents.** Agents do not have any a-priori notion of utility or optimal demand.
- **No foresight.** Agents cannot evaluate the (possibly non-deterministic) outcome of trades unless they actually perform them.
- **Learning-by-doing.** Agents do not learn by imitating the strategies of successful agents. They improve their policies at each trade on the basis of the outcome of that trade.
- **Individual agents.** Individual economic entities can be represented by individual agents.

RL models: trading

- **Elementary bilateral trade.** In an EBT of $g, g' \in G$ an agent does:
  - Select an offer and demand $z = \pi(x, g, g') \in Z = \mathbb{R} \times \mathbb{R}$ on the basis of its current stocks x, of g, g' and of its current *policy* π. This is a function from $Y = X \times G \times G$ into Z. In the RL parlance Y and Z are called the agent's *state* and *action* spaces.
  - Enter the trade with $y = (x, g, g')$, z and get a new stock x' and a *reward* $r \in R$.
  - Learn a new policy $\pi' \succcurlyeq \pi$ on the basis of (y, z, x', r).

RL models: implicit assumptions

- **Actions and feedbacks.** Agents act on an environment and receive feedbacks from their environment.
- **Goals.** Agents aim at maximizing rewards. An agent considers $\pi' \succcurlyeq \pi$ if selecting actions according to π' yields, in the long run, higher rewards than selecting actions according to π.
- **Learning capabilities.** Agent are able to improve their policies.

Learning: dynamic programming, perfect foresight

Assume (for a moment) that:

- Y, Z and R are finite.
- The agent knows the probability $\rho\, y\, z\, r$ of getting reward r when selecting action z in y and the probability $\sigma\, y\, z\, y'$ of getting a new state y' when selecting action z in y.

With ρ and σ, the agent could compute the value in terms of discounted rewards of entering 1, 2, ..., EBTs with policy π

Learning: dynamic programming, perfect foresight

$$V_1\,\pi\,y \;=\; \sum_{r\in R} r * (\rho\, y\, (\pi\, y)\, r)$$

$$V_2\,\pi\,y \;=\; \sum_{r\in R} r * (\rho\, y\, (\pi\, y)\, r) \;+\; \beta * \sum_{y'\in Y} (V_1\,\pi\,y') * (\sigma\, y\, (\pi\, y)\, y')$$

$$\ldots$$

$$V\,\pi\,y \;=\; \sum_{r\in R} r * (\rho\, y\, (\pi\, y)\, r) \;+\; \beta * \sum_{y'\in Y} (V\,\pi\,y') * (\sigma\, y\, (\pi\, y)\, y')$$

. . . and value policies according to the preference

$$\pi' \succcurlyeq \pi \;\equiv\; \forall\, y \in Y\;\; V\,\pi'\,y \geq V\,\pi\,y$$

Learning: dynamic programming, perfect foresight

For $\succcurlyeq$, the agent can find a π' which is at least as good as π. Let

$$\omega\, y \;=\; \underset{z \in Z}{\operatorname{argmax}} \left(\sum_{r \in R} r * (\rho\, y\, z\, r) + \beta * \sum_{y' \in Y} (V\, \pi\, y') * (\sigma\, y\, z\, y') \right)$$

And let π' consist of making one step with action $\omega\, y$ and then following π:

$$\begin{aligned} V\, \pi'\, y &= \sum_{r \in R} r * (\rho\, y\, (\omega\, y)\, r) \;+\; \beta * \sum_{y' \in Y} (V\, \pi\, y') * (\sigma\, y\, (\omega\, y)\, y') \\ &\geq \sum_{r \in R} r * (\rho\, y\, (\pi\, y)\, r) \;+\; \beta * \sum_{y' \in Y} (V\, \pi\, y') * (\sigma\, y\, (\pi\, y)\, y') \\ &= V\, \pi\, y \end{aligned}$$

Learning: believes, expectations

Assume the agent has *believes* ρ_k, σ_k about ρ, σ. It could then:

1. Select an action $z_k = \pi_k\, y_k$ according to its current state y_k and policy π_k, enter the trade and get a new stock x_{k+1} and a reward r_k.
2. Compute $V\,\pi_k$ on the basis of its current believes ρ_k, σ_k.
3. Compute a new policy $\pi_{k+1} \succcurlyeq \pi_k$ on the basis of $V\,\pi_k$, ρ_k, σ_k.
4. Update its believes ρ_k, σ_k on the basis of (y_k, z_k, x_{k+1}, r_k):

$$\rho_{k+1}\, y\, z\, r = \begin{cases} (1-\epsilon) * (\rho_k\, y\, z\, r) + \epsilon * (\delta\, r_k\, r) & \text{if } y = y_k \\ & \wedge\ z = z_k \\ \rho_k\, y\, z\, r & \text{otherwise} \end{cases}$$

RL agents as adaptive types

- In evolutionary models agents play the role of passive containers (of commodities and prices). Their offer and demand policies are constant. There are no obvious advantages from introducing agent *abstract data types*.
- In RL models it could be convenient to represent agents as *adaptive values* (Bauer+Erwin+Fern+Pinto DSL2011):
  - Agents can select actions on the basis of *states*.
  - States include agent-specific properties and properties of *environments* the agents interact with.
  - To select actions, agents first observe their environments and perceive states.
  - Agents adapt to *feedbacks* from their environments.

RL agents as adaptive types

We say that a data type represents an agent if it is an instance of

```
class Agent α where
  type Environment α
  type State α
  type Action α
  type Feedback α
  observe :: α -> Environment α -> State α
  select :: α -> State α -> Action α
  adapt :: α -> State α -> Action α -> Feedback α -> α
```

And a data type represent the environment of an agent if it is an instance of

```
class AgentEnvironment ε where
  type ActionOfAgent ε
  type FeedbackOnAgent ε
  react :: ε -> ActionOfAgent ε -> (ε,FeedbackOnAgent ε)
```

RL agents as adaptive types

With `Agent`, `AgentEnvironment` and

```
step :: (Agent a,
         AgentEnvironment e,
         Environment a ~ e,
         Action a ~ ActionOfAgent e,
         Feedback a ~ FeedbackOnAgent e) =>
         (a, e) -> (a, e)

step (a,e) = (a',e')
  where a' = adapt a state action feedback
        (e',feedback) = react e action
        action = select a state
        state = observe a e
```

one can

- Specify concrete agent types as independent components.
- Develop and test agents in artificial environments.

Evolutionary and RL models of exchange

- In RL models of exchange there are no explicit prices. Could one decode prices from actions, policies or rewards and compare these prices with those of evolutionary models ?
- How do allocations in evolutionary and RL models of exchange relate to Walrasian equilibrium allocations ?
- Is it possible to compute rewards for RL models of exchange to behave as evolutionary models ? In which sense ?