Project Description: Reinforcement Learning for Markets

860-0011-00L Agent-Based Modeling and Social System Simulation - With Coding Project

2019
September

1 Agent Setting

In the proposed project **the goal is to design and implement agents that participate and play in a double auction game**. The environment is a market that consists of the so called agents, who may assume two discrete roles: (i) a seller $j \in \mathbb{J}$ and (ii) a buyer $i \in \mathbb{I}$. The sellers are selling a single unit of a product, with production cost $r_j^- \in \mathbb{R}^+$ at an ask price $a_j \in \mathbb{R}^+$. For a seller to be rational, it can be *assumed* that:

$$r_j^- \leq a_j \tag{1}$$

Selling at lower price than the cost would actually introduce losses to the seller. The buyers are buying a single unit of the same product with available budget $r_i^+ \in \mathbb{R}^+$ at an ask price $b_i \in \mathbb{R}^+$. For a buyer to be rational, it can be *assumed* that:

$$r_i^+ \geq b_i \tag{2}$$

Since it would not be possible for the buyer to buy at a higher price than the available budget.

The agents can submit their offers, i.e. bids and asks, at discrete uniform time steps t. A pair of a buyer and a seller is matched when at a given time step t, when the buyer bid is enough to cover the seller ask:

$$a_j \leq b_i \tag{3}$$

A deal price $d_{j,i} \in [a_j, b_i]$ is decided by a matching mechanism.

Agents stop participating in the game, i.e. submitting offers, once they are matched. Every T time steps or when no more matches are possible[1], the environment resets, meaning that all the agents are able to submit their offers again. This time interval is referred to as round. A collection of K rounds is a game. During a game, the buyer budgets r_i^+ and the seller costs r_j^- are *assumed* constant. Furthermore, in each game it is *assumed* that a single market setting is applied.

2 Market Setting

A market setting controls two aspects of a game:

1. the matching mechanism, i.e. the extra rules behind matching a pair of agents, e.g. regarding tie breaking, match frequency etc.

2. the information setting, which controls what information are available to the agent at each time step. These information are used by the agent to determine the next bid.

Some default market settings are provided, and the projects can be fully developed upon them. In the projects additional market settings may be proposed. Yet, it should be stressed out the choice of new market settings needs to be both well-explained and well-defined, e.g. closer to real-world scenarios, or more general mathematical properties. **In the end, an agent implementation is always required.**

2.1 Information Settings

Before taking a decision at time t, an agent $l \in \mathbb{J} \cup \mathbb{I}$ observes information $o_{l*,t}$. The following information settings are available:

2.1.1 Full Information Setting

Each agent l knows all the offers (bids and asks) that were submitted in the last time step $t-1$ in the market:

$$o_{l*,t} = \{b_{i,t-1}\}_{i \in \mathbb{I}} \cup \{a_{j,t-1}\}_{j \in \mathbb{J}} \tag{4}$$

[1]e.g. in a round, where there are more sellers than buyers, no matches are possible when all buyers are matched.

2.1.2 Same Side Information Setting

Each agent knows all the offers from the agents of same role (bids or asks) that happened in the last time step in the market:

$$o_{j*,t} = \{a_{j,t-1}\}_{j\in\mathbb{J}} \tag{5}$$

and:

$$o_{i*,t} = \{b_{i,t-1}\}_{i\in\mathbb{I}} \tag{6}$$

2.1.3 Opposite Side Information Setting

Each agent knows all the offers from the agents of different role (bids or asks) that happened in the last time step in the market:

$$o_{i*,t} = \{a_{j,t-1}\}_{j\in\mathbb{J}} \tag{7}$$

and:

$$o_{j*,t} = \{b_{i,t-1}\}_{i\in\mathbb{I}} \tag{8}$$

2.1.4 Black-box Information Setting

Each agent knows nothing about other agents or sellers. Therefore agents can only observe some information based on their own past offers and deals.

2.2 Matching Mechanisms

The proposed matching mechanism selects the deal price $d_{j,i}$ by sampling a random uniform distribution $d_{j,i} \sim \mathcal{U}(a_j, b_i)$ for each match. Other possible implementations of matching mechanisms are at seller price or buyer price, or at a price that the matching mechanisms keeps part of the bid.

3 Agent Design Considerations

As it will be pointed out in the lectures, many considerations can be taken into account during agent design. Some possible considerations are the following:

- *Memory*: Does the agent evaluate any information from previous time steps to reach a decision?
- *Generalized agents*: The same agents may be exposed on different settings and games (even asynchronously).

- *Type*: The agents can be deterministic, stochastic or a mix.
- *Learning*: Agents may learn to change their decision making process from their interactions with the market.
- *Group Dynamics*: Agents might be independent, co-operative, competitive. Combinations of agents with different group dynamics might also be possible.
- *Formalization*: A formalization of the agent mechanism and its environment needs to be demonstrated, and all relevant sources need to be cited.

4 Resources

4.1 Reading Material

For the implementation of the project, the description present in this document is expected to be sufficient. Still, for a deeper and clearer view, the following proposed literature might prove useful:

4.1.1 Game Theory

The following links contain relevant material:

- "Price Formation in Double Auctions" by Steven Gjerstad and John Dickhaut: `https://www.sciencedirect.com/science/article/pii/S0899825697905765`
- Lecture notes on "Cooperative games" by Mihai Manea: `https://ocw.mit.edu/courses/economics/14-126-game-theory-spring-2016/lecture-notes/MIT14_126S16_cooperative.pdf`
- "Introduction to the Theory of Cooperative Games" by Bezalel Peleg and Peter Sudhölter: `https://link.springer.com/book/10.1007%2F978-3-540-72945-7`
- "The theory of linear economic models" by David Gale: `https://www.press.uchicago.edu/ucp/books/book/chicago/T/bo3619410.html`

To be able to skim through search engines and literature efficiently, the following key-phrases might help: "market clearing price", "Walrasian double auction", "competitive equilibrium"

4.1.2 Reinforcement Learning

Although not mandatory, it is strongly recommended to use reinforcement learning as the underlying decision making mechanism for the behavior of agents. More specifically, the goal is to design agents that collect experience via processing observations $o_{l,t}$ and use it to determine optimal decisions for offers (b or a) that yield to the optimal deal prices. Optimality of deal price can be defined differently for each agent role. The recommended approach would be to consider profit maximization(seller) and cost minimization(buyer). Still, other definitions of optimality can be proposed, if well-positioned.

The following links contain relevant material:

- "Reinforcement Learning in Continuous Action Spaces through Sequential Monte Carlo Methods" by A. Lazaric et al.: `https://papers.nips.cc/paper/3318-reinforcement-learning-in-continuous-action-spaces-through-sequential-monte-carlo-methods.pdf`
- "Continuous control with deep reinforcement learning" by T. P. Lillicrap et al.:
  `https://arxiv.org/abs/1509.02971`

Some keyphrase to look for: "reinforcement learning", "agent with continuous action space", "reinforcement learning in auctions", "reinforcement learning in markets"

4.2 Code

In this subsection some useful code repositories are found for the development of code. Since Matlab, python, Java and R are widely used, the most relevant repositories for these languages are provided:

Python will also be shown at the course and it will be the easiest language to get support from the course organizers:

- for plotting: plotly[2] , holoviews[3] , matplotlib[4]

[2] `https://plot.ly/graphing-libraries/`
[3] `http://holoviews.org/`
[4] `https://matplotlib.org/3.1.1/index.html`

- for number crunching: numpy[5], scipy[6], scikit[7], pytorch[8], tensorflow[9]
- for reinforcement learning: tensorforce[10] , stable-baselines[11] and gym[12].
- for deep learning: keras[13].

The repository with the python code for the project is found at:
`https://github.com/asikist-ethz/market_rl`

Matlab library for reinforcement learning (untested): Reinforcement Learning Agents Toolbox[14]. For other languages, like R and Java ask Thomas directly for pointers at relevant libraries.

4.3 Data

Data from past experiments with people on several market settings can be found here:
`https://osf.io/gu62n/`
The link contains more information regarding the data. It is not mandatory to use the data, still some nice experimental setting can be evaluated by doing so.

5 Possible Directions

The above setting can be extended in several directions. For the course project the main focus is the following:

- Modelling goal: Why do we use agent based modelling?
  - Do the agents need to optimize something?
  - Do we need to observe the agent interactions, identify emerging behaviors and try to explain a phenomenon?

[5] https://numpy.org/
[6] https://www.scipy.org/
[7] https://scikit-learn.org/stable/
[8] https://pytorch.org/
[9] https://www.tensorflow.org/versions/r1.14/api_docs/python/tf
[10] https://github.com/tensorforce/tensorforce
[11] https://github.com/hill-a/stable-baselines
[12] https://gym.openai.com/
[13] https://keras.io/
[14] https://ch.mathworks.com/help/reinforcement-learning/index.html?s_tid=srchtitle

  - Something else?

- Agent design: What considerations are taken into account during agent design? What is the explanation or intuition behind these choices?

Some possible directions that can be pursued (optionally and any combination of those directions):

- Do we implement a single agent or a multiple-agent scenario?
- Can we reproduce behavior observed in data or expected from theory?
- Can we optimize agent performance (e.g. profits)?
- How is agent behavior and performance affected on different market settings, e.g. asymmetric groups, e.g. more sellers than buyers, different information settings etc.
- Can we break some of the assumptions made in the previous sections? How does this affect agent learning and behavior?
- Is knowledge transfer possible? E.g. do agents that learn in a subset of settings achieved expected or optimal behaviors in unseen settings?

6 General Comments

Some random wisdom accumulated over the years:

- Further collaboration can be pursued after the end of the course.
- The goal is not always to optimize or increase performance, but also to simulate, experiement and observe.
- Do not rush to apply a custom/learning model. There might be fully explainable formal mechanism that explains the agent behavior out there.
- Sometimes explaining why something does not work may be more important than showing that something else works.

7 Acknowledgements

The project description is written by Thomas Asikis with support from Dr. Heinrich Nax, Barbara Ikica and Dr. Antulov-Fantulin.

Nomenclature

a	The ask a seller agent submits to sell a product.
b	The bid a buyer agent submits to purchase a product.
r^+	Available budget to a buyer agent.
i	A buyer agent.
r^-	Cost for a seller agent.
d	The deal price for a matching pair of a seller and a buyer.
o	An agent observation.
j	A seller agent.