Monitoring Reliability and Robustness of Agents for Dynamic Pricing in different (Re-)Commerce Markets

Monitoring der Zuverlässigkeit und Robustheit von Agenten zur dynamischen Bepreisung in unterschiedlichen (Re-)Commerce Märkten

Nikkel Mollenhauer

Universitätsbachelorarbeit
zur Erlangung des akademischen Grades

Bachelor of Science
(B. Sc.)

im Studiengang IT-Systems Engineering
eingereicht am 30. Juni 2022 am Fachgebiet
Enterprise Platform and Integration Concepts
der Digital-Engineering-Fakultät der Universität Potsdam

Gutachter	Dr. Rainer Schlosser
Betreuer	Johannes Huegle
	Alexander Kastius

Abstract

Zusammenfassung

Contents

Demo Chapter

Sadly, if you cite your own publications, they will appear in the bibliography. Thus, make sure to cite your papers with yourself as one of the authors.

This chapter shows off some of the basic formats of this thesis. Many packages are included in order for you to be able to start immediately without having to manually add all of the important things. The features deemed most important are now presented.

We now state a theorem and restate it later on again. Have a look at the source code in order to see how the theorem is written. Many macros are used, and all of them can be used without using math mode explicitly. Note that we can refer to inequality (0.1) as an inequality through the magic of an option in its label.

▶ **Theorem 0.1 (Variable Drift).** Let $(\mathcal{F}_t)_{t\in\mathbf{N}}$ be a filtration, $(X_t)_{t\in\mathbf{N}}$ be a random process over $\mathbf{R}_0^+$ adapted to $\mathcal{F}$, $x_{\min} > 0$, and let $T = \inf\{t \mid X_t < x_{\min}\}$. Additionally, let D denote the smallest real interval that contains at least all values $x \geq x_{\min}$ that, for all $t \leq T$, any X_t can take. Furthermore, suppose that

1. $X_0 \geq x_{\min}$ and that
2. there is a monotonically increasing function $h\colon D \to \mathbf{R}^+$ such that, for all $t < T$, we have $X_t - \mathrm{E}[X_{t+1} \mid \mathcal{F}_t] \geq h(X_t)$.

Then

$$\mathrm{E}[T \mid \mathcal{F}_0] \leq \frac{x_{\min}}{h(x_{\min})} + \int_{x_{\min}}^{X_0} \frac{1}{h(z)} \mathrm{d}z\ . \tag{0.1}$$

◀

Please shift your attention to Figure 1. This reference was created using the package cleveref, which knows in what environment the label is defined in. This way, you can easily change a theorem into a lemma, and the name of the reference will be adjusted automatically. A wrapfigure like **??** is referenced just like a normal figure.

Of course, you can also use tables in a fancy style. See, for example, **??** . This document already contains packages in order to also handle larger tables. Hence, it is possible to use tables spanning multiple pages or to rotate a page into landscape in order to fit in a wider table.

(a) This is the caption of the subfigure that displays the logo of the HPI.

(b) This is the caption of the subfigure that displays the logo of the UP.

Figure 1: These are the two logos featured on the title page. Figure 1 (a) belongs to the HPI, whereas Figure 1 (b) belongs to the UP.

Before we continue, consider the following obvious theorem. We conjecture that it also holds for $n = 2$.

▶ **Theorem 0.2.** Let $a, b, c, n \in \mathbf{N}^+$ with $n > 2$. Then

$$a^n + b^n \neq c^n \, . \qquad ◀$$

Since the proof is straightforward, it is omitted. Nonetheless, we present a proof in order to show off the proof environment.

Proof of Theorem 0.2. Unfortunately, there is too little space in this PDF for the proof. ■

You can have very expressive and fancy enumerations from the package enumitem. Again, we can easily reference an item like item (i).

(i) The labels of the items can be nicely chosen.

(ii) Note how the labels are left-aligned. This does not look good but should demonstrate what is easily possible.

We can even interrupt this enumeration and easily resume it immediately.

(iii) We continue where we left off.

Recall that Theorem 0.1 was as follows:

▶ **Theorem 0.1 (Variable Drift).** Let $(\mathcal{F}_t)_{t \in \mathrm{N}}$ be a filtration, $(X_t)_{t \in \mathrm{N}}$ be a random process over $\mathbf{R}_0^+$ adapted to $\mathcal{F}$, $x_{\min} > 0$, and let $T = \inf\{t \mid X_t < x_{\min}\}$. Additionally, let D denote the smallest real interval that contains at least all values $x \geq x_{\min}$ that, for all $t \leq T$, any X_t can take. Furthermore, suppose that

Text	**Number**
This is some text. Thus, it is left-aligned.	**0**
Numbers are right-aligned.	**1**
The numbers are formatted in bold using the package array.	**2**

1. $X_0 \geq x_{\min}$ and that
2. there is a monotonically increasing function $h\colon D \to \mathbf{R}^+$ such that, for all $t < T$, we have $X_t - \mathrm{E}[X_{t+1} \mid \mathcal{F}_t] \geq h(X_t)$.

Then

$$\mathrm{E}[T \mid \mathcal{F}_0] \leq \frac{x_{\min}}{h(x_{\min})} + \int_{x_{\min}}^{X_0} \frac{1}{h(z)} \mathrm{d}z \,. \tag{0.1}$$

◀

Note that the reference above still refers to the first occurrence of the theorem. However, the theorem is repeated without any noise. That is, it is identical to the other occurrence.

From the next page on, other than a warp figure and some filler text, there is not much more to see. Thank you very much for taking your time and reading so far. I hope you got an impression of what this template is capable of. Have fun using it, and create a great thesis!

1 Introduction

This thesis builds upon the bachelors project "Online Marketplace Simulation: A Testbed for Self-Learning Agents" of the Enterprise Platform and Integration Concepts research group at the Hasso-Plattner-Institute. Therefore, the project will be referenced and all examples and experiments will have been conducted using its framework.

1.1 Objective of the Thesis

This thesis introduces ways to monitor the *Reliability* and *Robustness* of different agents (rule-based as well as trained using various Reinforcement-Learning (RL) approaches) tasked with dynamically pricing products in a Circular Economy marketplace. Since the terms *Reliability* and *Robustness* can be interpreted differently depending on context and personal experience, we will define our usage in the Reliability and Robustness section. Following the term definitions, we will give a short introduction ans explanation of what a Circular Economy market is (The Circular Economy model) as well as what Reinforcement Learning is and how we employ the technique in our framework (Using the simulated marketplace to train agents).

1.1.1 Reliability and Robustness

1. *Reliability*: With *Reliability*, we describe the ability of an agent to be able to transfer knowledge of a certain type of marketplace and/or against a certain opponent over to a different scenario. If Agent A performs well against Agent B on marketplace M, does it perform the same against Agent C on marketplace M, or against Agent B on marketplace N?

2. *Robustness*: *Robustness* is the property that describes how well an agent performs over a longer period of time. In a real-world marketplace, consistency is key to success, so finding profitability outliers and their causes are a central part of evaluating an agent's Robustness.

1.2 Introduction to the Recommerce platform

1.2.1 The Circular Economy model

The main goal of the aforementioned bachelors project was to develop a realistic online marketplace that simulates a Circular Economy. A market is most commonly referred to as being a "Circular Economy" if it includes the three activities of reduce, reuse and recycle [KRH17]. This means that while in a classical Linear Economy market each product is being sold once at its *new price* and after use being thrown away, in a Circular Economy, recycling and thereby waste reduction is a major focus. In our project, we modelled this by adding two additional price channels, *re-buy price* and *used price*, to the pre-existing *new price* of a product.

The *re-buy price* is defined as the price a vendor is willing to pay a customer to buy back a used product, while the *used price* is defined as the price the vendor sets for products they previously bought back and now want to sell alongside new products (whose price is defined by the *new price*).

From now on, we will refer to the Circular Economy market simply as *the market*.

1.2.2 Using the simulated marketplace to train agents

After the initial market was modelled the goal was to train agents using different reinforcement-learning algorithms to dynamically set prices on this marketplace, both in monopolistic scenarios as well as in competition with rule-based vendors which set prices following a strict set of pre-defined rules. These rules can range from simply undercutting the lowest competitor's price to more advanced techniques such as price-fixing and -gouging.

Is the following sentence a footnote?

find out if we have such competitors

Reinforcement-learning agents are trained through a process of trial-and-error. They interact with the market through an observable state and an action which influences the following state. Figure 1.1 illustrates the RL-model in the context of our market. The goal of the agent is to maximize the so-called reinforcement signal, which in our case is the profit the agent made during the last cycle, since we want to train agents to maximize profits on real markets. By observing which prices lead to which reinforcement signal, the agents get more profitable over the course of training.

Create a *nicer* diagram with the three prices

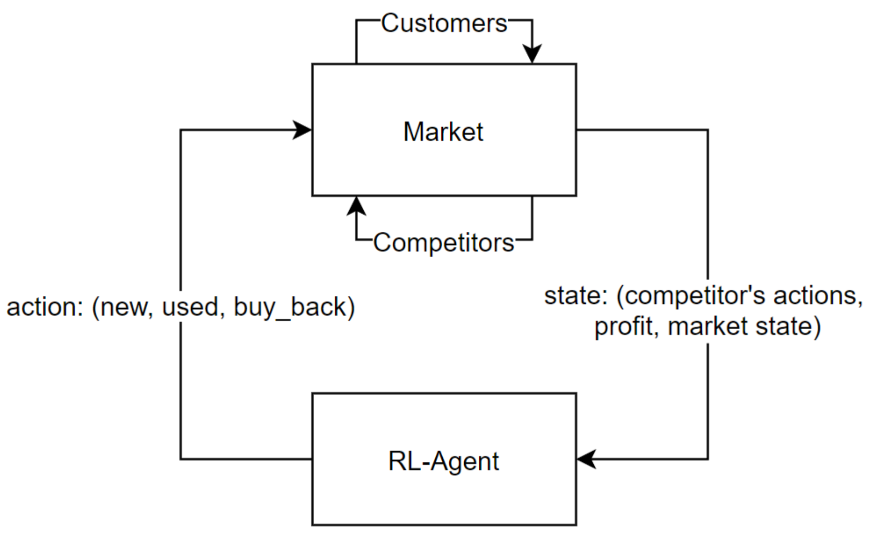
Customers
Market
Competitors
action: (new, used, buy_back)
state: (competitor's actions, profit, market state)
RL-Agent

Figure 1.1: The standard reinforcement-learning model in the context of our market.

2 Related Work

2.1 Approaches to evaluating RL-agents

2.1.1 ...on the fly (while training)

2.1.2 ...after training has finished

3 What makes a good agent?

In this section we want to take a look at what defines an objectively "good" agent, focusing on the dynamic pricing aspect and how our recommerce marketplaces were built to simulate real markets as realistically as possible. This section will be split into two parts - first, we will take a look at Reinforcement Learning Agents, to then try and transfer as much knowledge as possible to the rulebased competitors we specifically built for the framework.

3.1 Overview of market components

3.1.1 Focus on how agents make profit etc.

3.2 How realistic the market is/how realistic it can be

3.2.1 Restrictions for evaluation arising from this

4 Approaches to monitoring agents

In this section we will take a look at the different approaches we took to monitoring agents in our framework, explaining the reasons why we chose to implement specific features and how they help us in determining an agents strengths and weaknesses.

4.1 When to monitor what

Our workflow (which will be explained in more detail in The *recommerce* workflow) can generally be split into two parts when it comes to monitoring and evaluation of agents. In the future, when talking about the *workflow* we will be talking about the process of configuring and starting a training session, where a Reinforcement-Learning agent is being trained on a specific marketplace against competitors. The *workflow* also includes the subsequent collection of data used to evaluate the agent's performance. We are also introducing the term of the *complete agent* in this section, which will be used to refer to both Reinforcement-Learning agents that have been fully trained as well as rule-based agents which do not need training.

1. During training: Having data available as soon as possible without having to wait for a long training session to end is crucial to an efficient workflow. Our framework enables us to analyze and visualize data while a training session is still running . This enables us to filter out agents with sub-par performance prematurely. This can be done using user-defined threshholds or on the basis of previoulsly collected data (e.g. only keeping agents that are performing better than at least 50% of other agents after the same amount of training in comparable scenarios.)

2. On complete agents: After a training session has finished we have a complete and final set of data available for an agent, which enables us to perform more thorough and reliable tests. These can include simulating runs of a marketplace to gather data on the agent's performance in different scenarios and against different competitors, or running a static analysis of the agent's policy in different market states. The tools available for trained agents are in the same way also usable on rule-based agents.

Implement this feature. It should at least be able to temporarily halt the training to start an intermediate monitoring session

Also implement this feature. Allow users to set rules for when a training session should be terminated if the agent is not performing well at a certain point. Also, it should be able to give the program a set of datapoints and have it set rules if possible

In the following sections, we will take a look at all of the tools our framework provides for monitoring agents, distinguishing between the two general types of monitoring mentioned above. The goal of this section is to give a short overview of each tool, how and why they were implemented and what value they offer to the framework as a whole and to the analysis of agent reliability and robustness in particular. We will also discuss features that are currently not available, explaining how they could benefit the entire workflow or enrich the overall experience.

4.2 Monitoring during a training session

When talking about monitoring agents during a training session, we are of course talking about Reinforcement-Learning agents, since Rule-Based agents always perform the same and cannot be trained. Monitoring agents while they are still being trained enables us to be more closely connected to the training process. Ultimately, the goal of such monitoring tools is to be able to predict the estimated "quality" of the final trained agent as reliably as possible while the training is still going. Users must however be careful when interpreting the results of monitoring tools that work on "incomplete" agents, we will go more into this in Interpreting the results.

Is this a footnote? This info should be obvious, but maybe should be mentioned for completeness?

4.2.1 Tensorboard

The *Tensorboard* is an external tool from the from the Reinforcement-Learning library *Tensorflow*[1]. With just a few lines of code a training session can be connected to a Tensorboard. We are then able to pass any number of parameters and metrics we deem interesting or useful to the Tensorboard, which then offers visualizations for each of them, updating live as the training progresses. Though the Tensorboard does not interpret data in any way, it is an immensely useful tool for quickly and easily recording data and offering a first rough comparison of competitors in the market.

Question for the tutors: Do I give an example of such code?

Create some sample diagrams. Perhaps these can be re-used in the workflow section, so they could perhaps be moved to the Appendix for reference?

4.2.2 Live-monitoring

Unlike the Tensorboard, the monitoring tools summarised under the term *live-monitoring* were completely and from the ground up built by our team. For most of the visualizations, the *matplotlib*[2] library was used. Live-monitoring aims to be

1 https://www.tensorflow.org/
2 https://matplotlib.org/stable/index.html

more configurable and in-depth with the metrics it offers than the aforementioned Tensorboard. During a training session, "intermediate" models, as we will call them, are being saved after a set number of episodes. These models contain the current policy of the agent and can be used the same as models of complete agents, the only difference being the quality of the agent, as those with a lower amount of trained episodes generally perform worse. These intermediate models can then be used by a range of monitoring tools available to us. Since the models only contain the current policy of an agent but not the history of states and actions preceding the model, we need to run separate simulations on these models to be able to analyze and evaluate them. For this, we utilize our *agent-monitoring* toolset, which will be explained in more detail in the Agent-monitoring section.

make the current live-monitoring so that graphs are actually being created while the training is running, not like it is now with graphs only being created afterwards

Have we introduced the concept of episodes before? Should a Glossary be introduced, or do we do this stuff in the introductions?

Does this need a citation?

4.3 Monitoring complete agents

The following tools offer a wide range of functionalities for monitoring complete agents. We can use these tools to both determine the reliability and robustness of one certain agent, but also to compare different agents, either during a joined monitoring session or by comparing results of monitoring runs on identically parametrized market situations.

4.3.1 Agent-monitoring

Using the *Agent-monitoring* toolset, users can configure a custom market simulation, using the following parameters:

1. Episodes: This parameter decides how many independent simulations are run in sequence. At the start of each episode, the market state will be reset. Within an episode, each vendor runs through 50 timesteps, during each of which a price is set (depending on the chosen economy type, this can include a rebuy price for used items) and a set number of customers interact with the vendors.

2. Plot interval : A number of diagram types enable the user to view averaged or aggregated data over a period of time. The plot interval parameter decides the size of these intervals. Smaller intervals mean more accurate but also more convoluted data points. Computational time also increases with a smaller interval size.

Currently, the plot interval is not used by anything, since the statistics-plots have been "removed" in favor of the tensorboard plots

3. Marketplace: Using this parameter, the user can set the marketplace on which the monitoring session will be run. A wide variety of scenarios is available, which are explained in .

Section reference to introduction where the scenarios are explained

During each episode and for each vendor, all actions and market events are being recorded using *watchers*.

Do I keep this new word? If yes, it should be explained at least a little bit!

4.3.2 Exampleprinter

The *Exampleprinter* is a tool meant for quickly evaluating an agent in-depth. When run, each action the monitored agent takes is being recorded, in addition to market states and events, such as the number of customers arriving and the amount of products thrown away. For certain market scenarios such as the *CircularEconomyRebuyPriceDuopoly*, which simulates a circular economy model with rebuy prices in which two vendors compete against each other, these actions and events are also being summarised as an animated graphic, where each time-step is being illustrated.

Add in a graphic here. Perhaps have two of the time-steps to "simulate" the animation

4.3.3 Policyanalyzer

The *Policyanalyzer* is our only tool which does not simulate a run of the market scenario. Instead, the tool can be used to monitor an agent's reaction to different market events. The user can decide on up to two different features to give as an input, such as the competitor's new and used prices, and the Policyanalyzer will feed all possible input combinations to the agent and record its reactions.

Diagram

4.4 Features for the future

This section is meant as a collection of ideas and approaches for tools that could further enhance the workflow, but which have not been implemented and therefore not been tested for their feasability and usefulness.

5 The *recommerce* workflow

5.1 Configuring the run

5.1.1 The webserver/Docker-API

5.2 During training

5.2.1 Choosing what to show the user

5.3 Saving models at certain stages during training

5.3.1 Live-monitoring tools (see Chapter 4)

5.4 After training/Complete agents

5.4.1 Why do we even monitor these agents?

5.4.2 Which datapoints prove to be/are most effective?

6 Interpreting the results

6.1 Graphs and diagrams are available...

6.1.1 ...comparing with other agents/models

6.1.2 ...which hyperparameters influence the results in what ways?

6.1.3 ...can we augment e.g. Grid-Search with our analysis?

6.1.4 -> Would need to make results "machine-readable" again

7 Conclusions & Outlook

Bibliography

[KRH17] Julian Kirchherr, Denise Reike and Marko Hekkert. **Conceptualizing the circular economy: An analysis of 114 definitions**. *Resources, Conservation and Recycling* 127 (2017), 221–232. ISSN: 0921-3449. DOI: https://doi.org/10.1016/j.resconrec.2017.09.005. URL: https://www.sciencedirect.com/science/article/pii/S0921344917302835 (see page 6).

Declaration of Authorship

I hereby declare that this thesis is my own unaided work. All direct or indirect sources used are acknowledged as references.

Potsdam, 13th May 2022

Nikkel Mollenhauer