УДК 004.89
МРНТИ

RESEARCH OF MULTIAGENT SYSTEM IN A DYNAMICALLY CHANGING ENVIRONMENT USING REINFORCEMENT LEARNING ALGORITHMS

A. PRENOV, A. AKSHABAYEV

[1]JSC «Kazakh-British Technical University»

Abstract: *Nowadays, betting has become one of the most well-known facilities in the modern world. Thus, there occurred a plenty of bookmakers which got profitable in the very short period of time. Sport prediction is very important and interesting problem for machine learning algorithms. Research explores the usage of one of the most mind-blowing phenomenases - the multi-agent system in the study of the world of bets. Since, Reinforcement Algorithms are the irreplaceable ones in the study of gamblings, we'll show the implementation and the meaning of the reinforcement algorithm. Study will consider the role of reinforcement algorithm used by multi-agents to determine the winners and losers. We'll examine the efficiency of a given algorithm in the obscure surroundings. Moreover, we'll show the process of transferring the data among agents and demonstrate its efficiency. Finally, we'll provide cases where this solution can be useful in terms of business, mathematics, etc.*

Keywords: *sports betting, multi-agent systems, q-learning, reward, state, action*

ДИНАМИКАЛЫҚ ӨЗГЕРІСТЕГІ ОРТАДАҒЫ МУЛЬТИАГЕНТТІК ЖҮЙЕНІ КҮШЕЙТЕ ОҚЫТУ АЛГОРИТМДЕРІ НЕГІЗІНДЕ ЗЕРДЕЛЕУ

Аңдатпа: *Қазіргі таңда спорт сайыстарына ұтыс тігу заманауи қоғамның ең танымал ойын-сауық түрлерінің біріне айналды. Спорттық бәс тігулер кең етек жая түсумен бірге, сан алуан мәселелер де бетке шыға бастады. Зерттеу, бүгінгі күннің ең өзекті терминдердің бірін, мультиагенттік жүйелердің спорт сайыстарына ұтыс тігудегі қолданысын қарастырады. Нығайту алгоритмдері құмар ойындарды зерттеудегі таптырмас құрал болғандықтан, осы алгоритмнің жүзеге асуын және қолданысын анықтап көрсетеміз. Жазылған жұмыс мултиагенттік жүйелердің нығайту алгоритмінің арқасында спорт ойындарындағы ұтыскерлер мен ұтылғандарды анықтап, есептеу тәсілін көрсетеді. Сол үшін алгоритмнің тиімділігін шынайы ортада тексеріп, нәтижелерін шығарамыз. Сондай-ақ мультиагенттердің өздерінің арасында ақпаратты қаншалықты тиімді таситынын зерттейміз. Қорыта келе, алгоритмнің қолдану аялары мен болашақтағы мүмкіншіліктері туралы жан-жақты зерделенеді.*

Түйінді сөздер: *спорт сайыстарына ұтыс тігу, мультиагенттік жүйелер, q-оқыту, олжа, жай-күйі, іс-қимыл*

ИССЛЕДОВАНИЕ МУЛЬТИАГЕНТНЫХ СИСТЕМ В ДИНАМИЧЕСКИ ИЗМЕНЯЕМОЙ СРЕДЕ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМОВ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

Аннотация: *Ставки на спорт являются одним из самых популярных видов развлечений современного мира на сегодняшний день. Вследствие чего букмекерские конторы стали неимоверно прибыльными за очень короткое время, что за собой привлекло немало проблем, о которых приводится ниже. Исследование рассматривает один из самых умопомрачительных феноменов сегодняшнего дня, то есть мультиагентные системы в мире спортивных ставок. Рассматривается реализация и смысл обучения с подкреплением, в виду того, что данный вид обучения незаменим в изучении спортивных*

ставок. Алгоритмы обучения с подкреплением незаменимы в анализе азартных игр, поэтому исследование рассмотрит роль обучения, используемых мультиагентами, чтобы определить победителей и проигравших. Протестирована эффективность алгоритма в реальной среде. Также демонстрируется процесс передачи данных среди мультиагентов и результативность процесса, сферы применения алгоритма.

Ключевые слова: *ставки на спорт, мультиагентные системы, q-обучение, награда, состояние, действие*

Introduction

First of all, I would like to introduce a problem I want to face. At the present time, there are a lot of problems caused by a betting establishments. Many families have getting divorced. There are a bunch of bankrupt, gambling addictions, anxiety and depression, even suicide risks. I personally believe that these facilities have to go under.

The disclosure of the work patterns of betting offices would turn the world upside down for all intents and purposes. There is still no precise way to attain a formula to compute the probability of win. Many researchers tried to beat the bookmakers. However, all attempts were futile.

The objective of the research is to show the work principle of one of the most profitable businesses - betting establishments and try to beat them. Relevance of this research can't be underestimated since Machine Learning is the main tool in the study of bets, sports and forecasting in general.

Target-setting

In light of technology age, most bookmakers have their own web-site. The most popular ones are olimp.kz, 1xbet.kz, profitbet.kz, tennisi.kz, etc.

I chose Kazakhstan's betting office named tennisi as the example of the bookmaker.

I was focused on a tennis due to plenty of facts. First of all, there are only two options in tennis: player 1 wins and player 2 loses or vise-versa, player 2 wins and player 1 loses. It means that computing the probability of win or lose would be much easier because we don't need to consider a draw situations. Usually game is played to 6 points. If draw occurs players continue their game adding 2 points required to win. For example, if the score is 5-5, player needs to win with 7-5.

Secondly, tennis game doesn't depend on external factors such as the weather, health of players, etc. So, we're minimizing the 'domain' of probability calculation by eliminating these

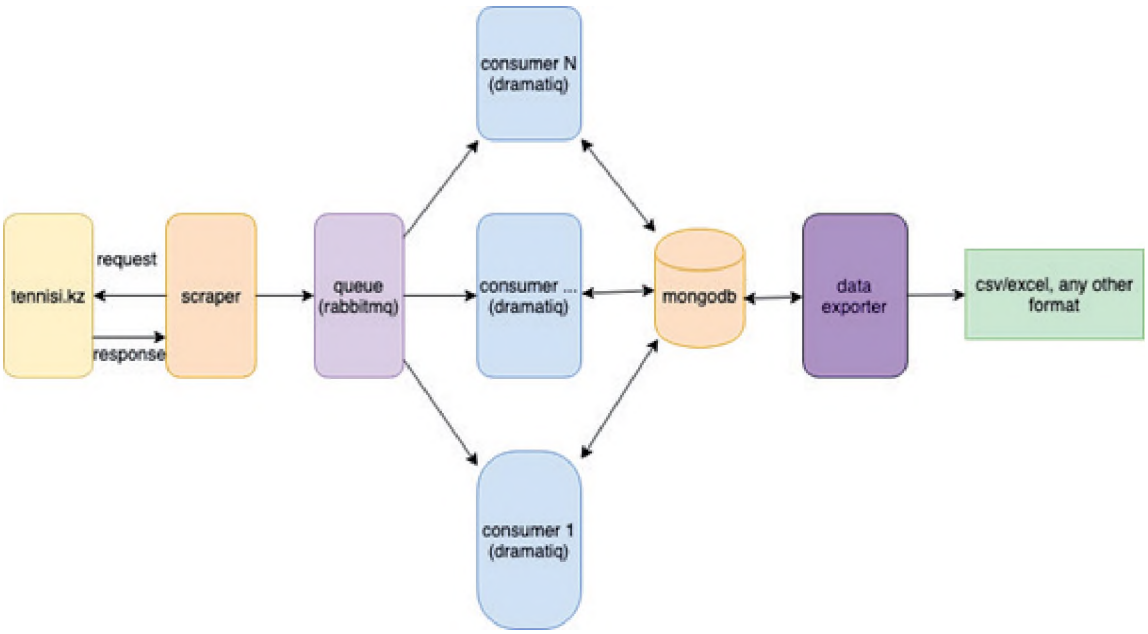

Fig. 1. Data Proccessing Proccess

factors above in order to make the process easier and convenient.

Data Processing

As you can see, scraper collects all the data from web-site and puts it into a queue. I added queue because of the potential enormous amount of data. If all requests handled immediately, it will lead to overloading of the server and the next situation might happen: one request is still loading, but the next one is already handled, and everything goes in the wrong order. Therefore, I use message-broker which will handle all requests in a right time. After queue the information is stored in a DBMS (Database Management System), MongoDB, in this case. Then, this data can be exported in a CSV (or any other) format.

Introduction to Q-Learning:

In this research I used q-learning algorithm as the reinforcement algorithm which is based on the Q function that calculates the optimal road to a goal (or just win).

The function adds new reward to maximum amount of previous reward multiplied by discount factor.

First, we need to initialize our q-function and q-table where initial values are zero. Then agent starts exploring the environment and performing actions. Each action gives certain result: negative, positive or neutral (just zero). So, agent starts learning based on empirical knowledge.

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Repeat
   $\Delta \leftarrow 0$
   For each $s \in \mathcal{S}$:
   $v \leftarrow V(s)$
   $V(s) \leftarrow \sum_{s',r} p(s', r|s, \pi(s))[r + \gamma V(s')]$
   $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   $a \leftarrow \pi(s)$
   $\pi(s) \leftarrow \text{argmax}_a \sum_{s',r} p(s', r|s, a)[r + \gamma V(s')]$
   If $a \neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return V and π; else go to 2

Fig. 2. Pseudocode for Q-Learning

Application of Q-Learning

Next action is importing the information from a database and sending it to an agent.

First, let's see what agent is. Agent is an entity that do some independent actions in order to achieve a certain goal. In addition, agent

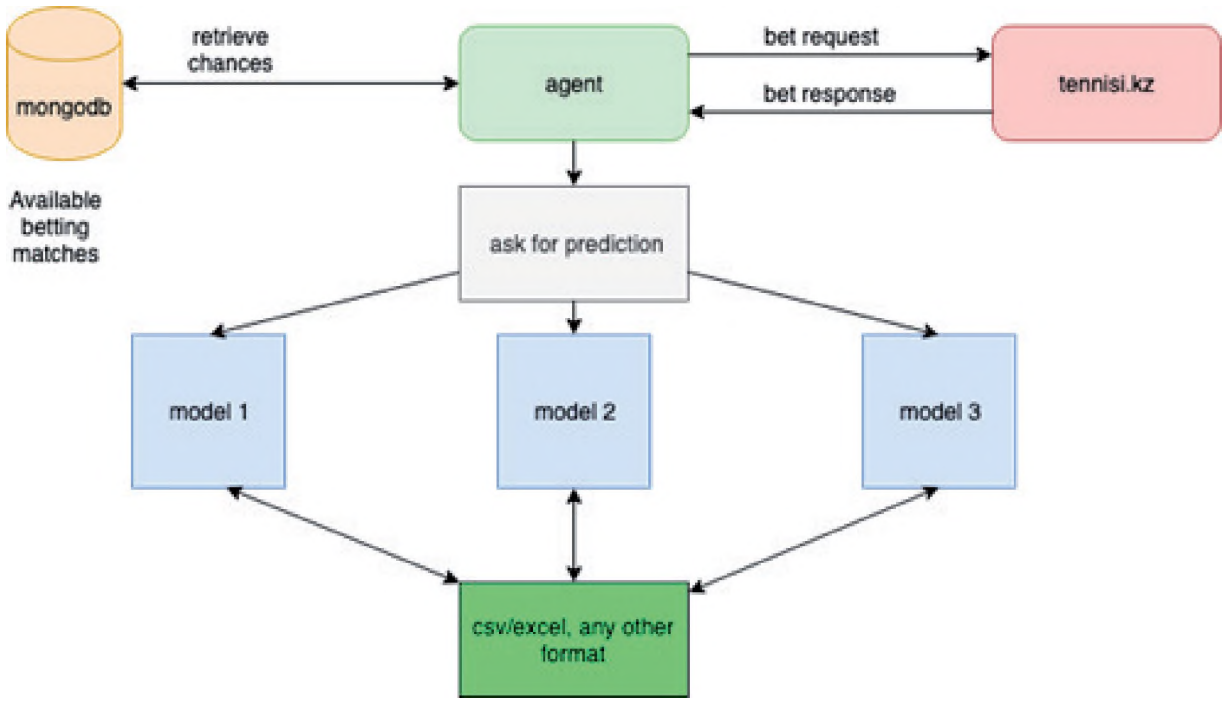

Fig. 3. Learning Algorithm

can not only use the previous downloaded and programmed knowledge, but also learn something new to reach this goal. In other words, agent can use its empirical knowledge (knowledge based on past experience).

Agent sends request to several models (we can use only q-learning based model or several independent models with different algorithms) and asks their predictions. After getting a prediction, agent submits it to bookmaker and gets the result if it's correct or not. This is how the agents learn.

The classical model of machine learning divides the data into two sets: train data set and test data set. However, nowadays we use train, validation and test data division.

Comparison

To summarize, let's compare all reinforcement learning algorithms with q-learning algorithm.

	Q-learning	Other reinforcement algorithms
Policy	Off-policy (explores the environment)	On-policy (doesn't explore)
Environment	Discrete, but can also work in continuous environment	Either discrete or continuous
Based on	Q-function	Other
Model	Model-free	Model-free
Function type	Directe	Direct, Recursive

Conclusion

To sum up, the research shows abilities of machine learning, q-learning, to be exact, in the study of the world of bets. As a consequence, the solution can be applied on many fields of game theory and discover the new opportunities in this area.

Therefore, many similar problems can be solved such as exchange rates forecasting, frauds in gambling and so on. In other words, we will have a real 'mind' which does all the calculations, analysis without any mistakes in them.

Research provides a great usage of multi-agent systems in sports betting and considers one of the large betting establishments, namely tennisi.

The prototype which is written on Python 3.7 can be used in a future investigation. Project is flexible and can be overwritten in many languages with many algorithm types of reinforcement learning.

Work discovers potential power of the q-learning in modern world with application in sports betting.

We achieved great results in this period of time. As a future work, I want to develop my ideas and improve the performance rate, minimize time and memory consuming and expand the use cases.

REFERENCES

1. Huang, S. Introduction to Various Reinforcement Learning Algorithms. Part I (Q-Learning, SARSA, DQN, DDPG). (2018) Retrieved from https://towardsdatascience.com/introduction-to-various-reinforcement-learning-algorithms-i-q-learning-sarsa-dqn-ddpg-72a5e0cb6287
2. Choudhary, A. A Hands-On Introduction to Deep Q-Learning using OpenAI Gym in Python (2019). Retrieved from https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/
3. Violante, A. Simple Reinforcement Learning: Q-Learning. (2019). Retrieved from https://towardsdatascience.com/simple-reinforcement-learning-q-learning-fcddc4b6fe56
4. Garant, D., Castro, B., Lesser, V. Accelerating Multi-agent Reinforcement Learning with Dynamic Co-learning.(2014). Cambridge: Massachusetts Institute of Technology.
5. McCabe, A., Trevathan, J. Artificial Intelligence in Sports Prediction. (2008). Retrieved from https://www.researchgate.net/publication/220841301_Artificial_Intelligence_in_Sports_Prediction
6. Prenov, A. Code Sample. Retrieved from https://github.com/aibaq/betting