

Hands-on Reinforcement Learning for RecSys – from Bandits to Offline RL with Ray RLlib

Kourosh Hakhamaneshi – kourosh@anyscale.com
Christy Bergman – christy@anyscale.com

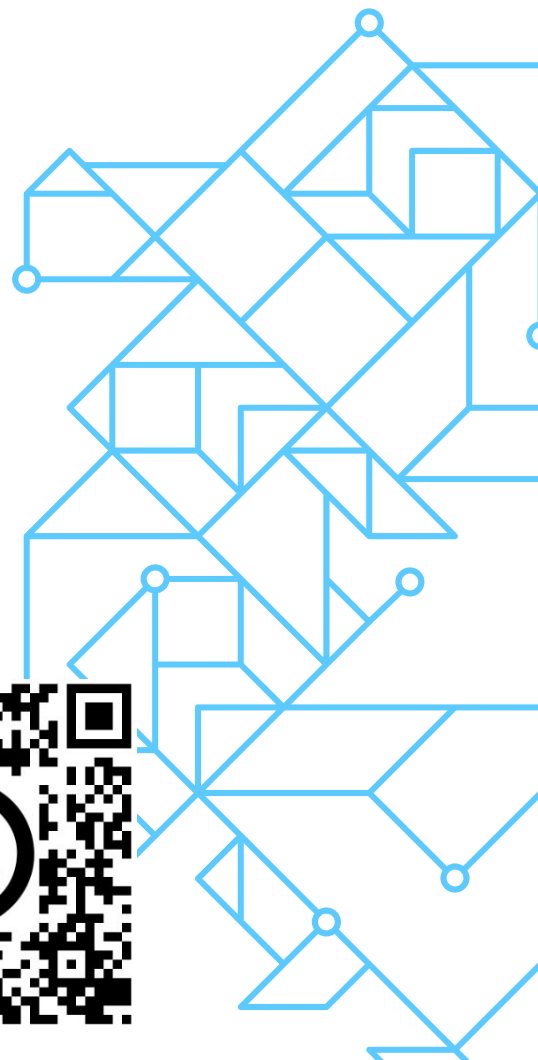




Few Important URLs

Keep these URLs open in your browser tabs

- **GitHub:** https://bit.ly/rllib_recsys_2022_github
- Q & A Doc: https://bit.ly/rllib_recsys_2022-qa
- Logins+passwords: https://bit.ly/rllib_recsys-logins
- Anyscale: console.anyscale.com
- Tutorial Survey: https://bit.ly/rllib_recsys_2022

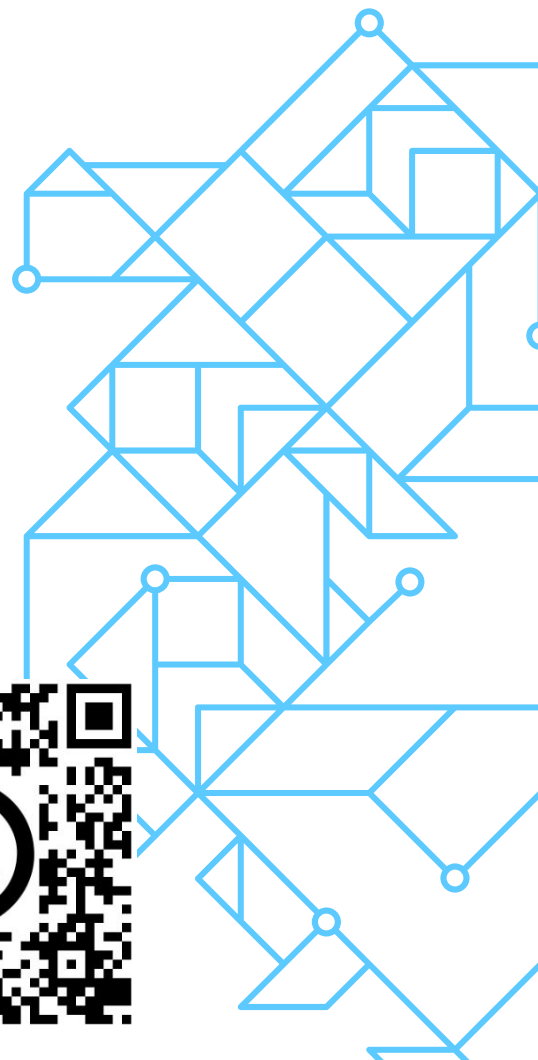




Few Important URLs

Keep these URLs open in your browser tabs

- GitHub: https://bit.ly/rllib_recsys_2022_github
- **Q & A Doc:** https://bit.ly/rllib_recsys_2022-qa
- Logins+passwords: https://bit.ly/rllib_recsys-logins
- Anyscale: console.anyscale.com
- Tutorial Survey: https://bit.ly/rllib_recsys_2022

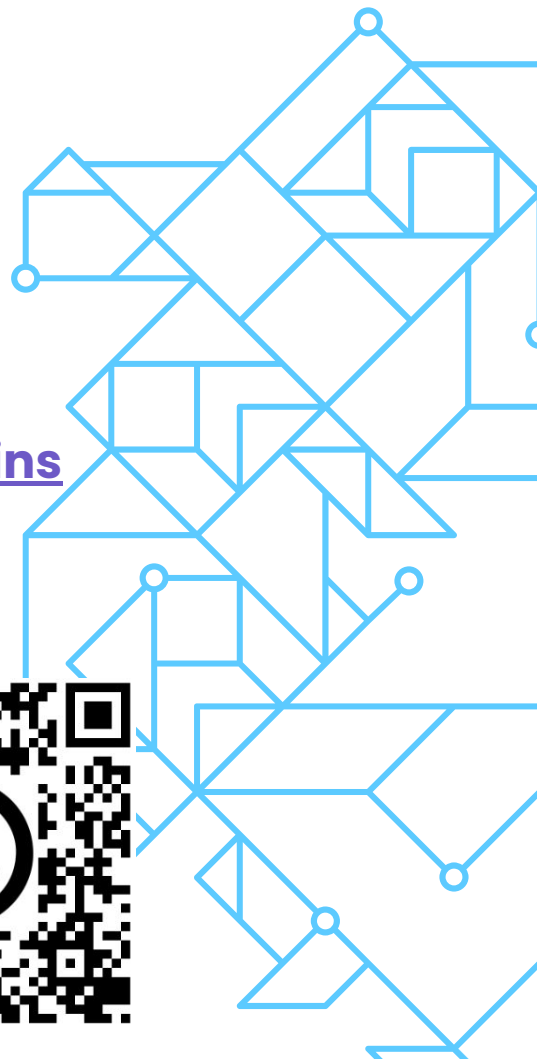




Few Important URLs

Keep these URLs open in your browser tabs

- GitHub: https://bit.ly/rllib_recsys_2022_github
- Q & A Doc: https://bit.ly/rllib_recsys_2022-qa
- **Logins+passwords:** https://bit.ly/rllib_recsys-logins
- **Anyscale:** console.anyscale.com
- Tutorial Survey: https://bit.ly/rllib_recsys_2022

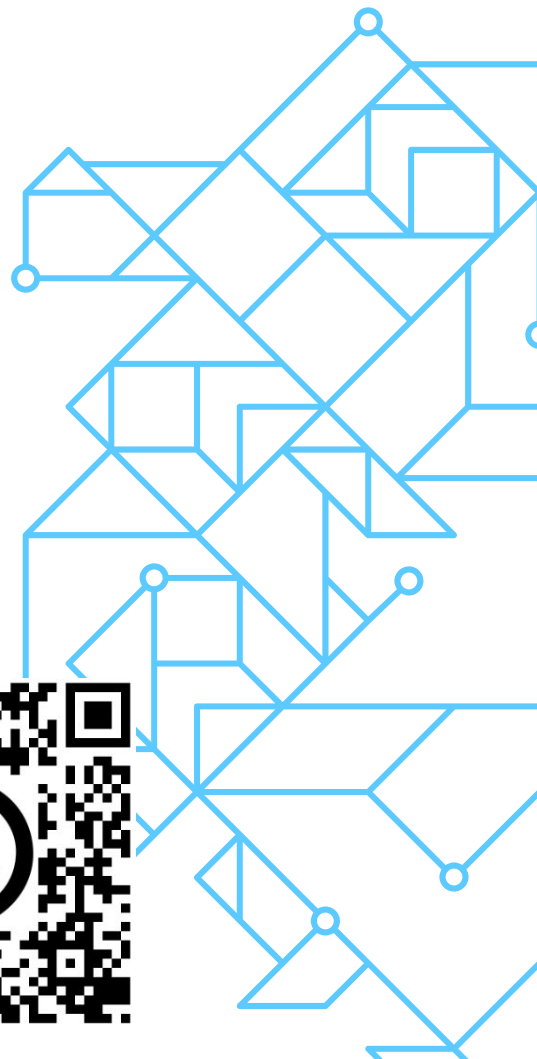




Few Important URLs

Keep these URLs open in your browser tabs

- GitHub: https://bit.ly/rllib_recsys_2022_github
- Q & A Doc: https://bit.ly/rllib_recsys_2022-qa
- Logins+passwords: https://bit.ly/rllib_recsys-logins
- Anyscale: console.anyscale.com
- **Tutorial Survey:** https://bit.ly/rllib_recsys_2022



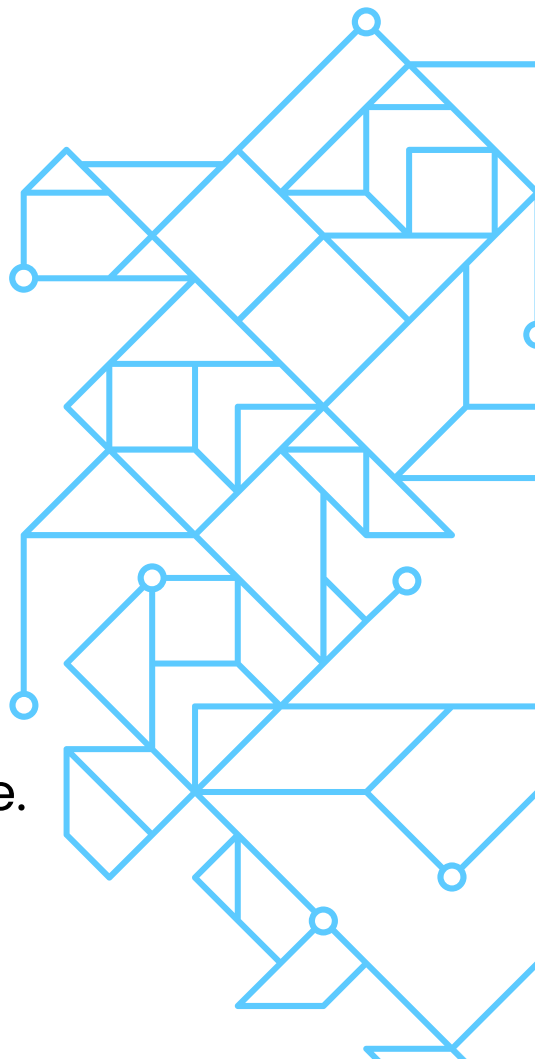


\$whoami (Christy)

- AI/ML DevAdvocate @Anyscale.
- Previously: AI/ML Solutions Architect at AWS, before that data scientist real-time fraud detection

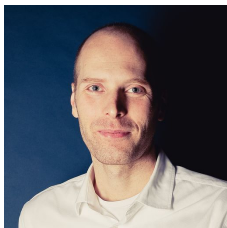
\$whoami (Kourosh)

- ML Engineer working on RL and RLib @Anyscale.
- Previously: PhD student at UC Berkeley working on RL in Robotics and design optimization





RL Team @ Anyscale



Sven



Jun



Avnish



Artur



Kourosh



Christy
(devAdvocate)





Anyscale

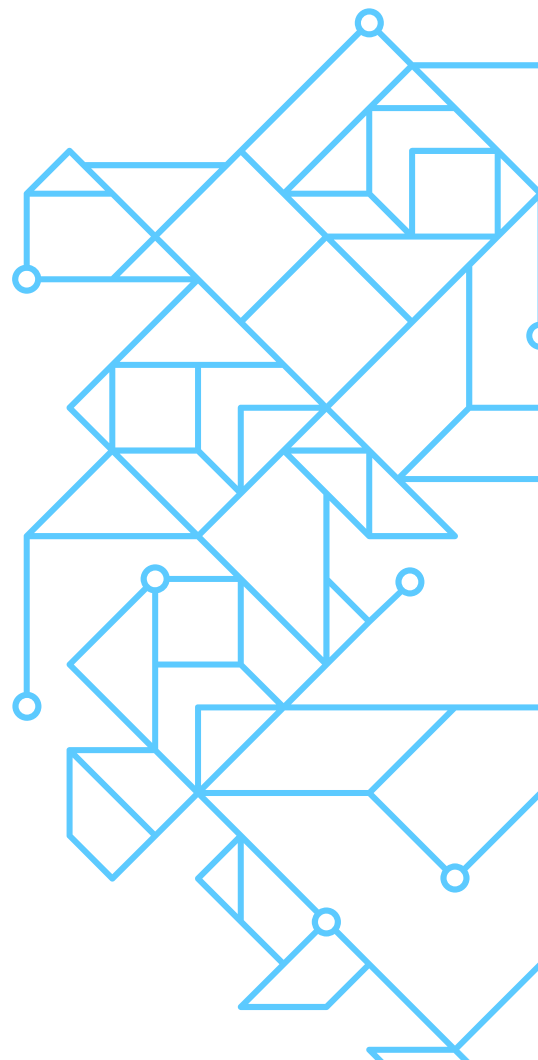
Who we are: Original creators of Ray, a unified framework for scalable, distributed computing. Part of that framework are our libraries for ML and data processing.

What we do: Scalable compute for AI and Python

Why we do it: Scaling is a necessity, scaling is hard; make distributed computing easy and simple for all developers.



Some of RLLib's Industry Users



Overview of the tutorial

- Brief intro RL
- Brief intro RecSys
 - + Traditional Approaches
 - + Defining RecSys as an RL problem
- Online RL vs Offline RL
- Hands-on coding with python notebooks and scripts
- Thank you! Connect with us!

Goals – Understand:

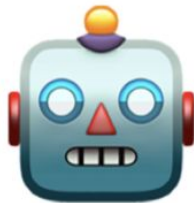
- What are the advantages of using RL in RecSys?
- What are the pros and cons of offline RL in practical scenarios?

Overview of the tutorial

- 4 min: Welcome
- 5 min: Very brief intro RL
- 5 min: Very brief intro RecSys
 - + Machine learning (ML) approach
 - + Challenges with current ML approach
 - + Map RecSys problem into MDP for RL
- 5 min: Intro Online RL vs Offline RL
- 1 hour: Hands-on with Google Colab
 - + 15min: Introduction to the environment
 - + 10min: Run baselines, bandit, and RL algorithm
 - + 5min: Conclusion so far TODO ADD slide with results
 - + 10min: Run offline RL on expert, random, greedy data
 - + 5min: Conclusion so far TODO ADD slide with results
 - + 5min: Deploy a policy to production using Ray Serve



Brief intro RL



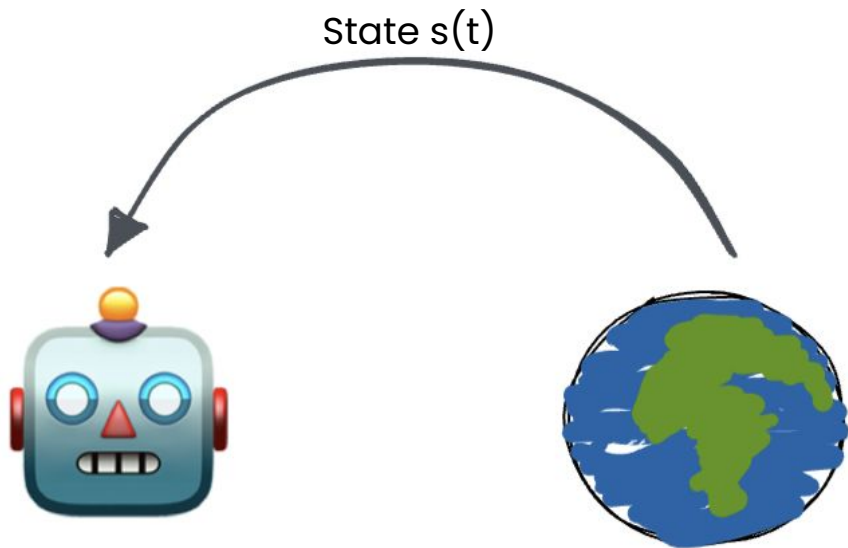
Agent



Environment

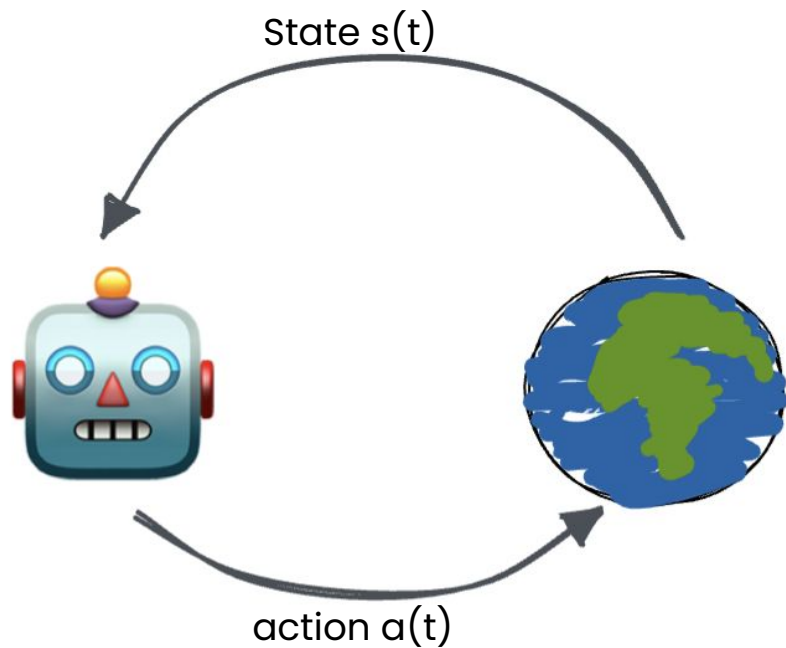


Brief intro RL



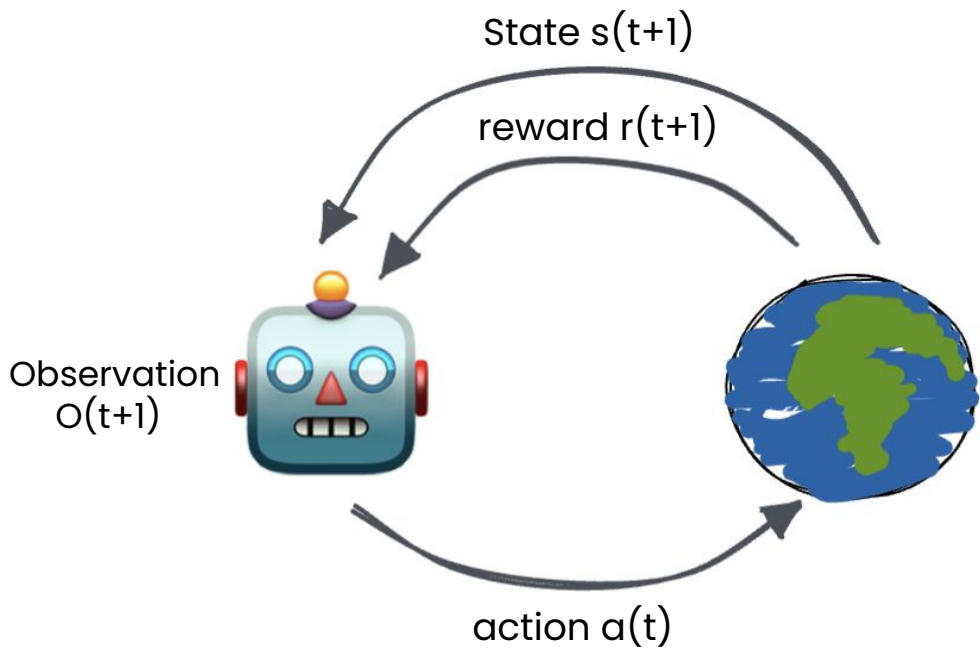


Brief intro RL



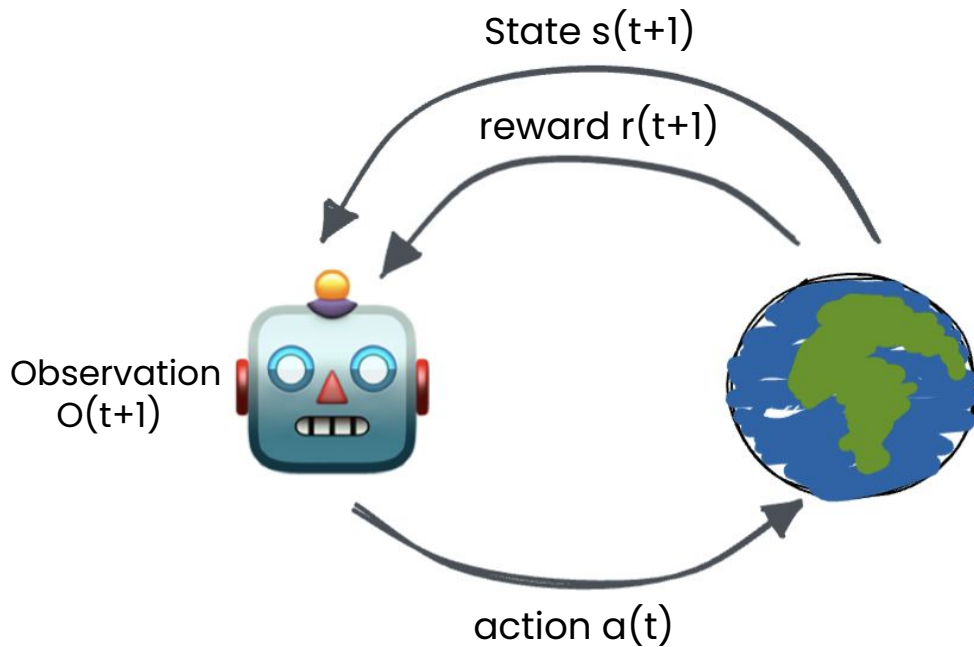


Brief intro RL





Brief intro RL



Algorithm

$$\max_{\pi} \mathbb{E}_{\pi} \left[\sum_t r(s_t, a_t) \right]$$

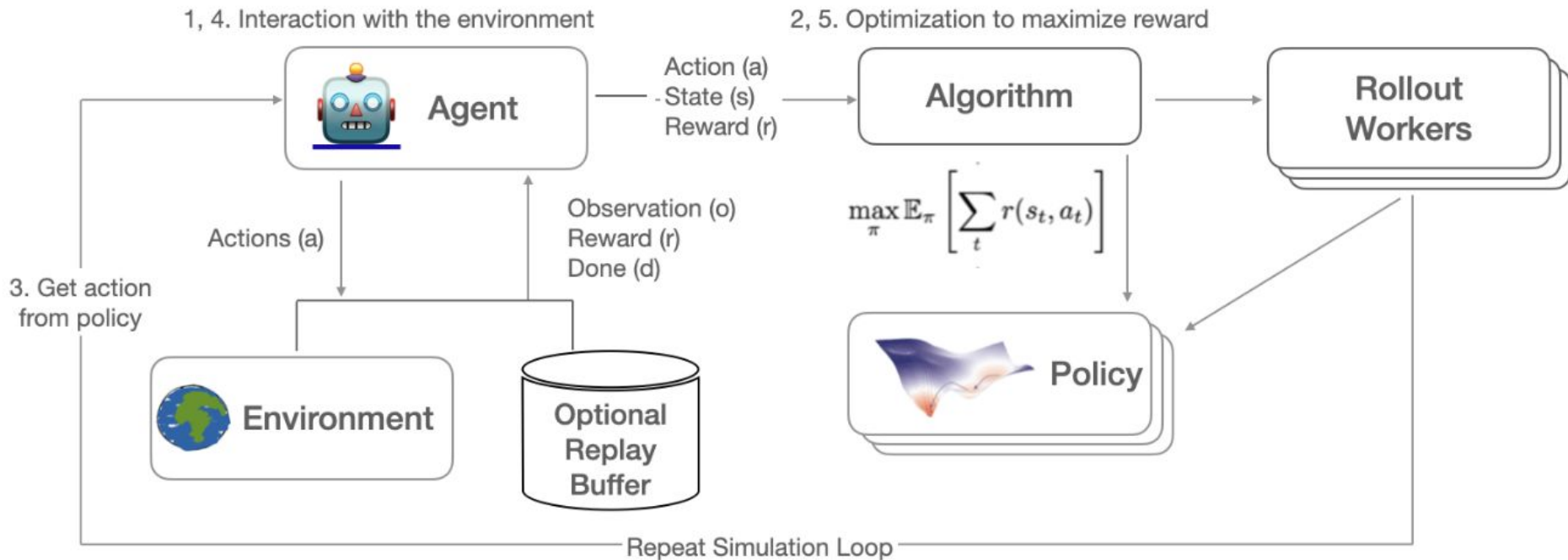


Policy

π

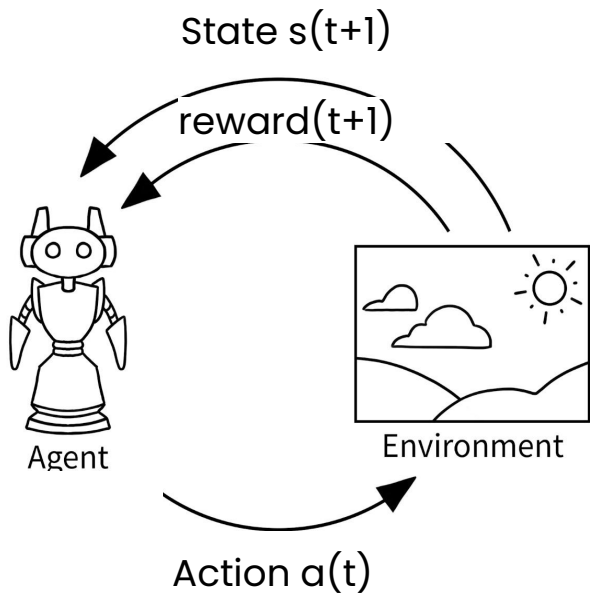


Brief intro RL





Brief intro RL



Conversation between an agent and an environment.

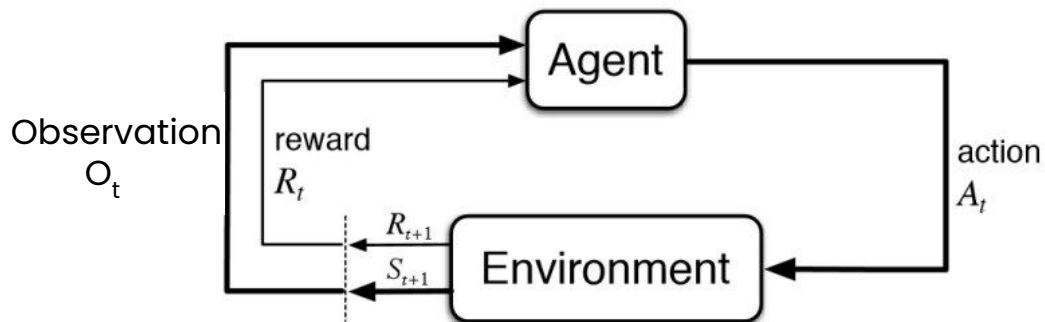
Learning objectives:

- Maximize sum of rewards.
- Learn from delayed reward.
- Proper exploration to maximally learn



Brief intro RL formalization

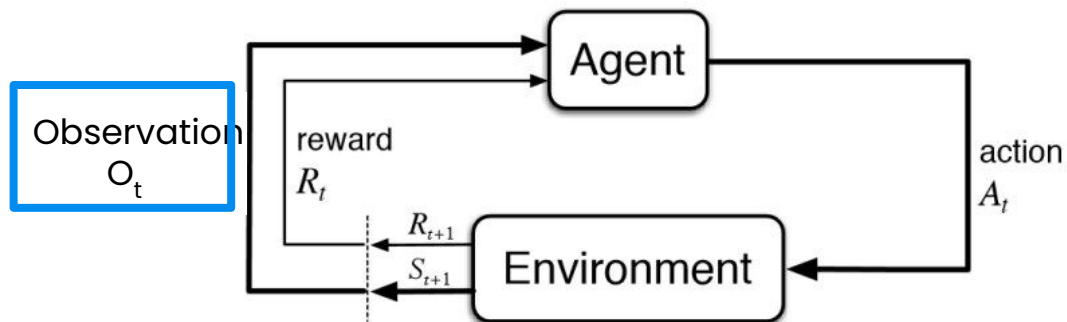
(S, A, P, R, γ)





Brief intro RL formalization

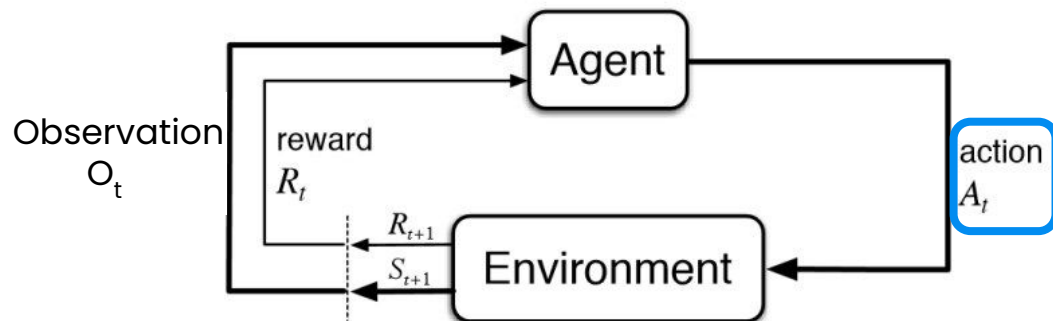
$$(S, A, P, R, \gamma)$$





Brief intro RL formalization

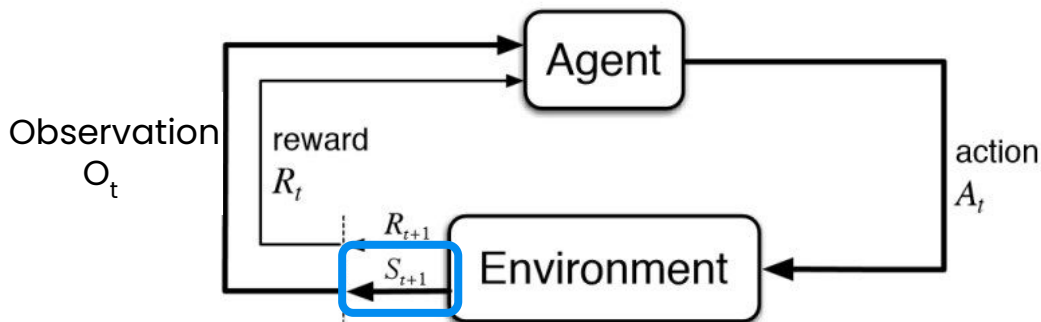
$$(S, A, P, R, \gamma)$$





Brief intro RL formalization

$$(S, A, \boxed{P}, R, \gamma)$$



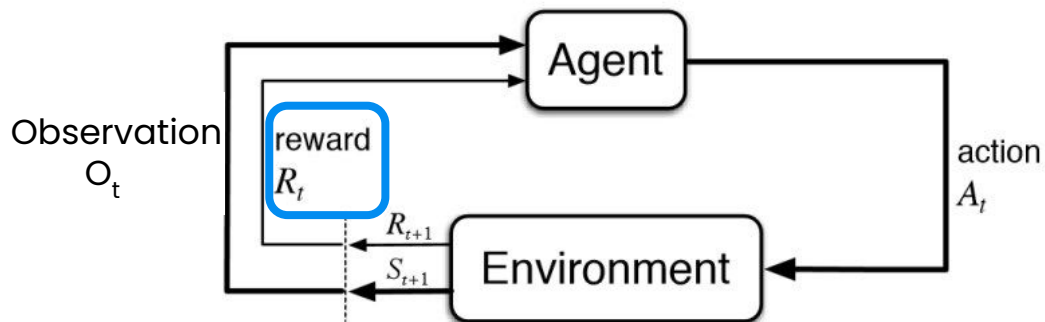
$$S_0 \sim \mathcal{P}(.)$$

$$S_{t+1} \sim \mathcal{P}(.|S_t, A_t)$$



Brief intro RL formalization

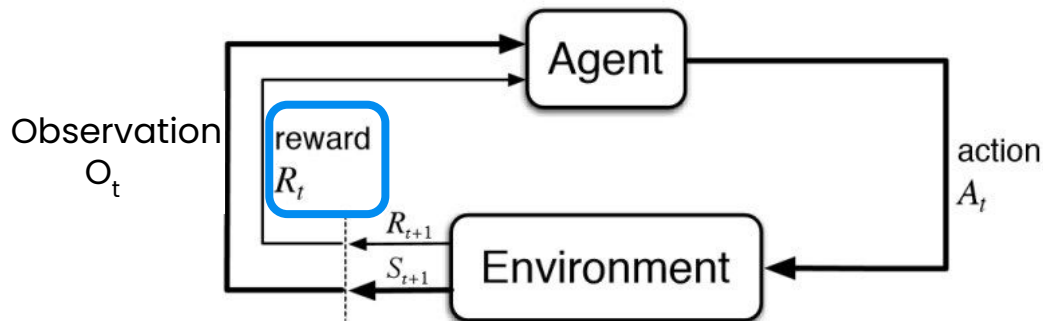
$$(S, A, P, R, \gamma)$$





Brief intro RL formalization

$$(S, A, P, R, \gamma)$$

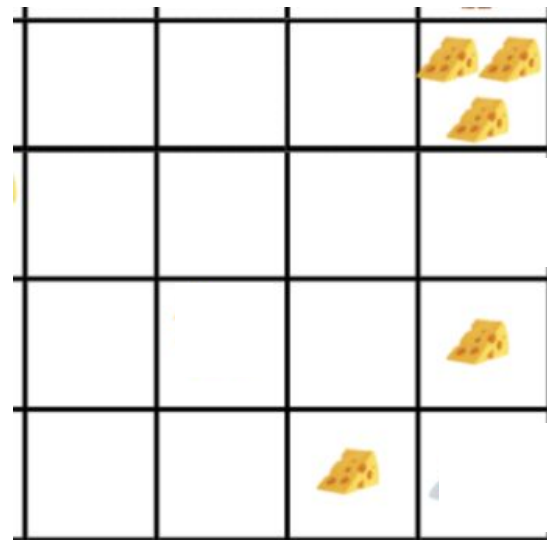


$$R(\tau) = \sum_t \gamma^t R_t$$



Discount factor γ in RL

- If $\gamma = 0$, the algorithm considers **1-step rewards only**.
- If $\gamma = 1$, the algorithm considers all future rewards equally.





Brief intro RecSys

Companies want to recommend content.



ML: Pointwise recommendations.

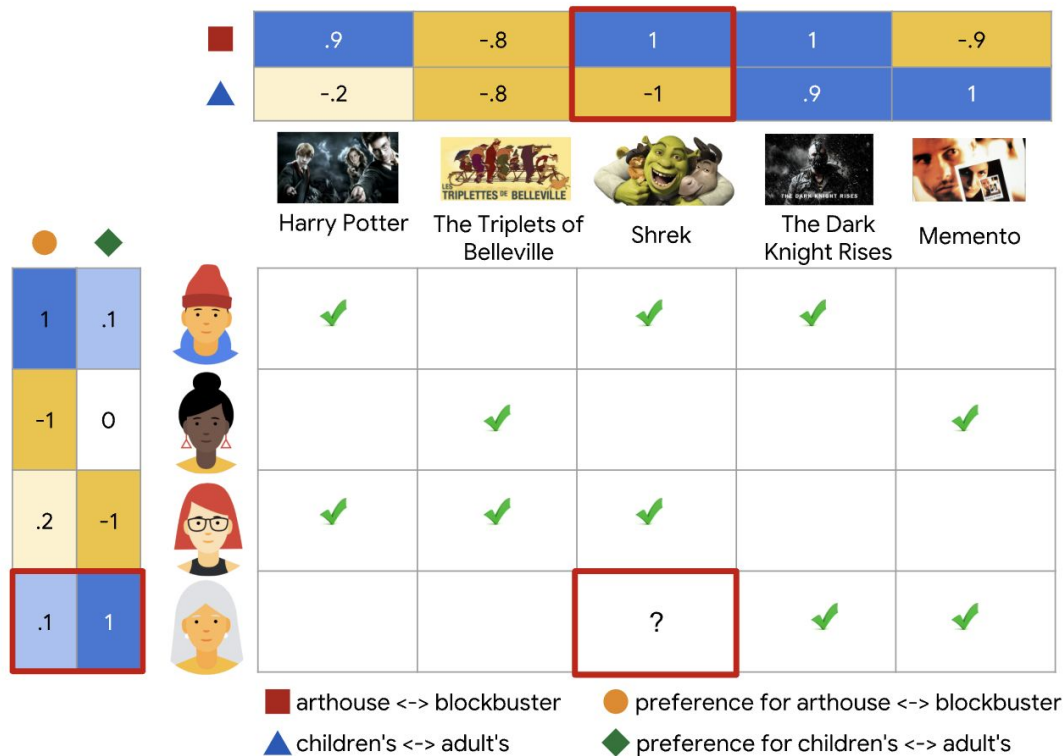


RL: Combine pointwise recommendations with session based data.





Brief intro RecSys ML





Challenges with traditional ML in RecSys

- Traditional ML (collaborative filtering) models are **static with respect to time**.
 - Ignores the **sequence of interactions** with a given user.
- Static models can be:
 - Too short-sighted and **miss out on Long-term, delayed rewards**
 - **Overlook important and changing user intents** or business conditions such as seasonality or promotional campaigns



New way: RL in RecSys

- By taking each user's session history as a sequence of decisions, **the RecSys problem can be converted into a sequential decision-making problem.**
 - $\Pr[S(t+1)=s_{t+1} \mid A(t)=a_t, S(t)=s_t, A(t-1)=a_{t-1}, \dots S(0)=s_0]$
- A stochastic process is a **Markov Decision Process (MDP)** if the **values at time t depend only on the values at time t-1.**
 - $Q_{\pi}(s, a) = E_{\pi} \left[\sum_{j=0}^T \gamma^j r_{t+j+1} \mid S_t = s, A_t = a \right]$
- **RL has become the de-facto ML approach for solving MDPs.**



New way: RL in RecSys

- By taking each user's session history as a sequence of decisions, **the RecSys problem can be converted into a sequential decision-making problem.**



New way: RL in RecSys

- By taking each user's session history as a sequence of decisions, **the RecSys problem can be converted into a sequential decision-making problem.**
 - **documents** = items to be recommended
 - **States** = item features, user features



New way: RL in RecSys

- By taking each user's session history as a sequence of decisions, **the RecSys problem can be converted into a sequential decision-making problem.**
 - **documents** = items to be recommended
 - **States** = item features, user features
 - **Actions** = recommended items



New way: RL in RecSys

- By taking each user's session history as a sequence of decisions, **the RecSys problem can be converted into a sequential decision-making problem.**
 - **documents** = items to be recommended
 - **States** = item features, user features
 - **Actions** = recommended items
 - **Rewards** = long term satisfaction (explicit or implicit)



New way: RL in RecSys

- By taking each user's session history as a sequence of decisions, **the RecSys problem can be converted into a sequential decision-making problem.**
 - **documents** = items to be recommended
 - **States** = item features, user features
 - **Actions** = recommended items
 - **Rewards** = long term satisfaction (explicit or implicit)
 - **Gamma** = 0 (bandits) or 1 (RL)



New way: RL in RecSys

- By taking each user's session history as a sequence of decisions, **the RecSys problem can be converted into a sequential decision-making problem.**
 - **documents** = items to be recommended
 - **States** = item features, user features
 - **Actions** = recommended items
 - **Rewards** = long term satisfaction (explicit or implicit)
 - **Gamma** = 0 (bandits) or 1 (RL)
 - **Agent** = user or customer receiving recommendations
 - **Env** = Google's RecSim (wrapped as Gym env)
 - **Algorithm** = RLib algorithm



RL Environment: Delayed Rewards & Long Term Satisfaction (LTS)

top 20 candidates



1



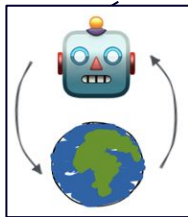


RL Environment: Delayed Rewards & Long Term Satisfaction (LTS)

top 20 candidates



1



Recommendation
1 (action)

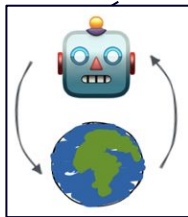


2



RL Environment: Delayed Rewards & Long Term Satisfaction (LTS)

top 20 candidates

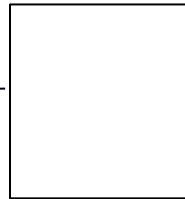


2

Recommendation
1 (action)



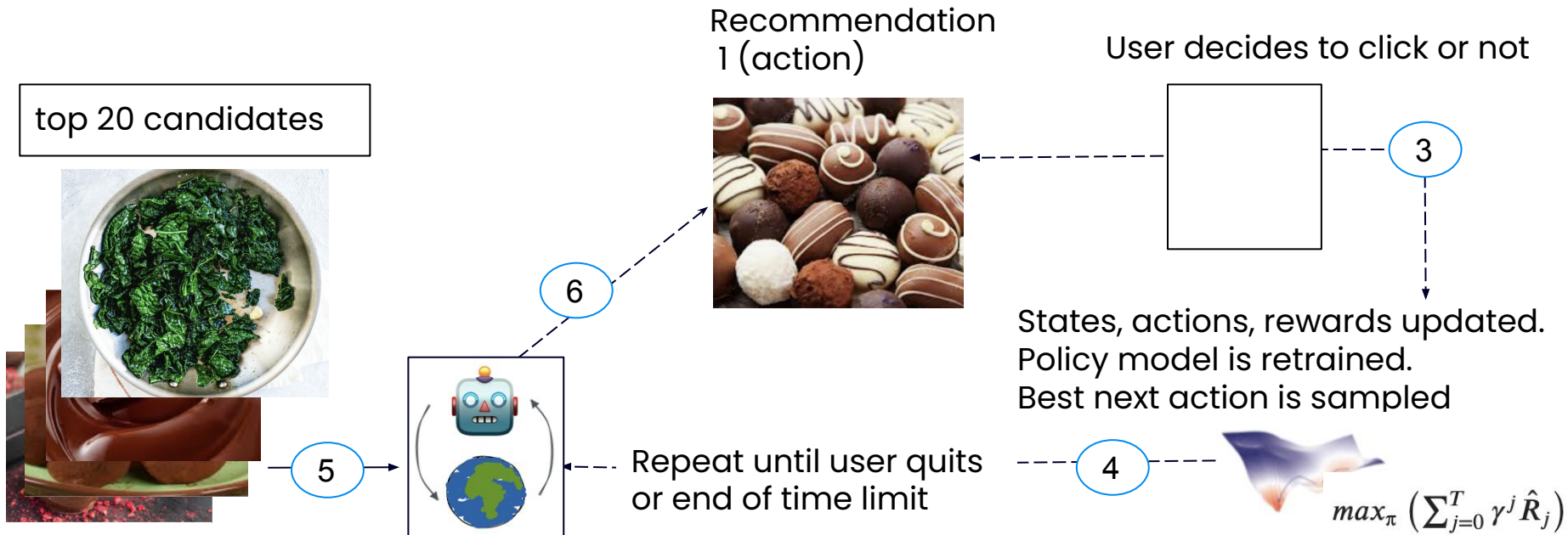
User decides to click or not



3

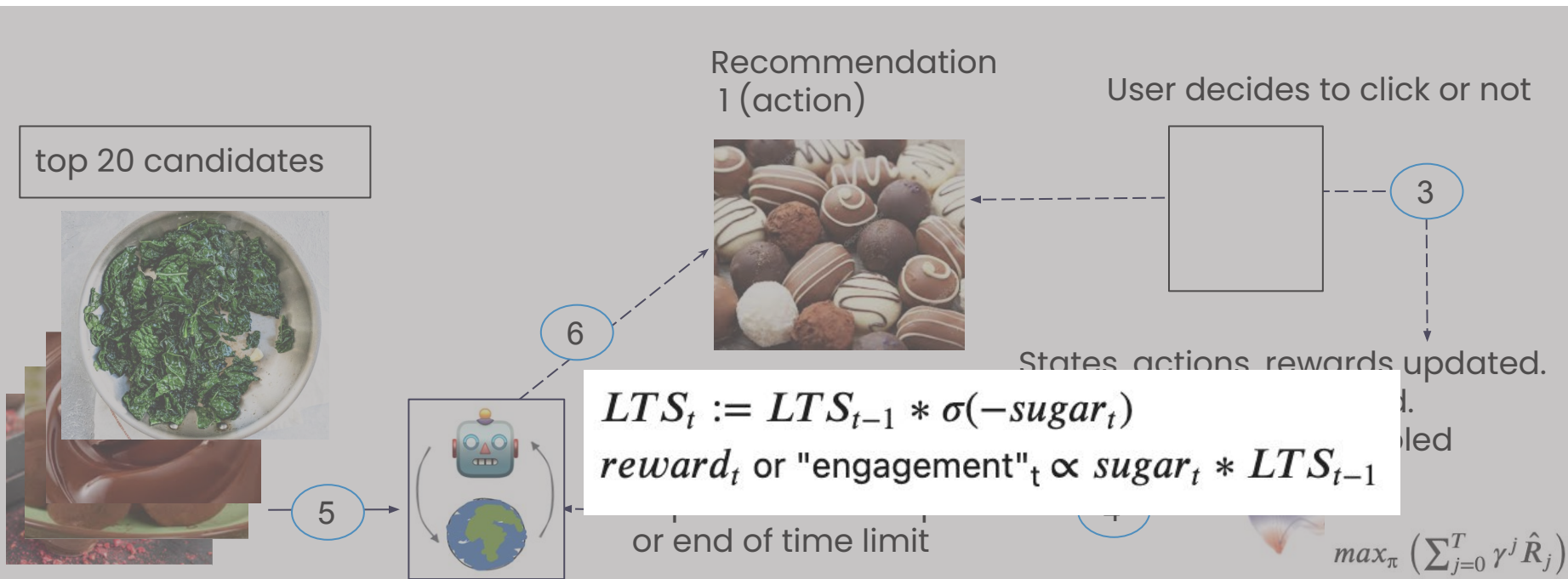


RL Environment: Delayed Rewards & Long Term Satisfaction (LTS)



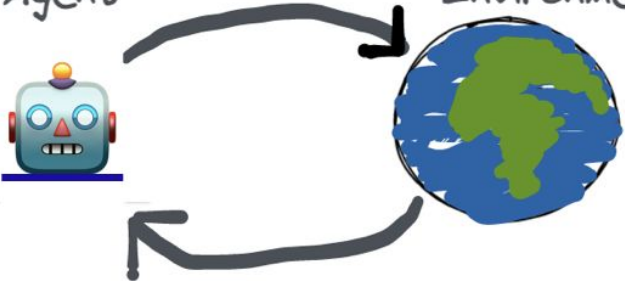


RL Environment: Delayed Rewards & Long Term Satisfaction (LTS)



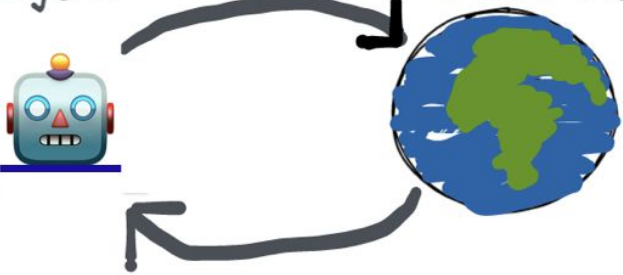
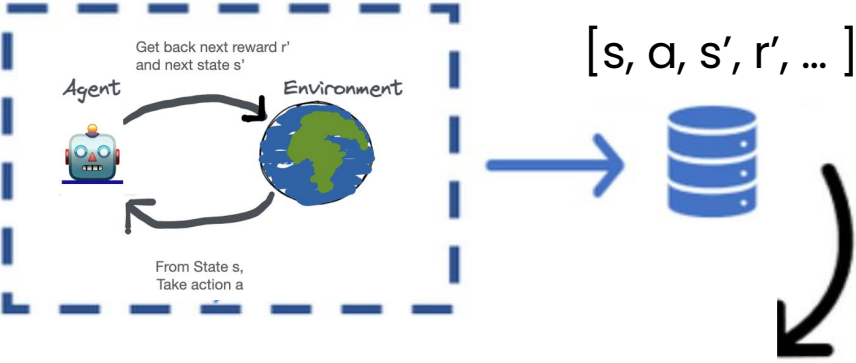


Online RL vs Offline RL

Online RL	Batch / Offline RL
<p data-bbox="253 401 562 464">Get back next reward r' and next state s'</p> <p data-bbox="106 475 222 529">Agent</p> <p data-bbox="550 475 788 518">Environment</p>  <p data-bbox="291 840 471 903">From State s, Take action a</p> <p>The diagram illustrates the Online Reinforcement Learning loop. On the left, a small blue robot icon labeled 'Agent' is shown. On the right, a globe icon labeled 'Environment' is shown. A curved arrow points from the Agent to the Environment, and another curved arrow points from the Environment back to the Agent, forming a cycle.</p>	

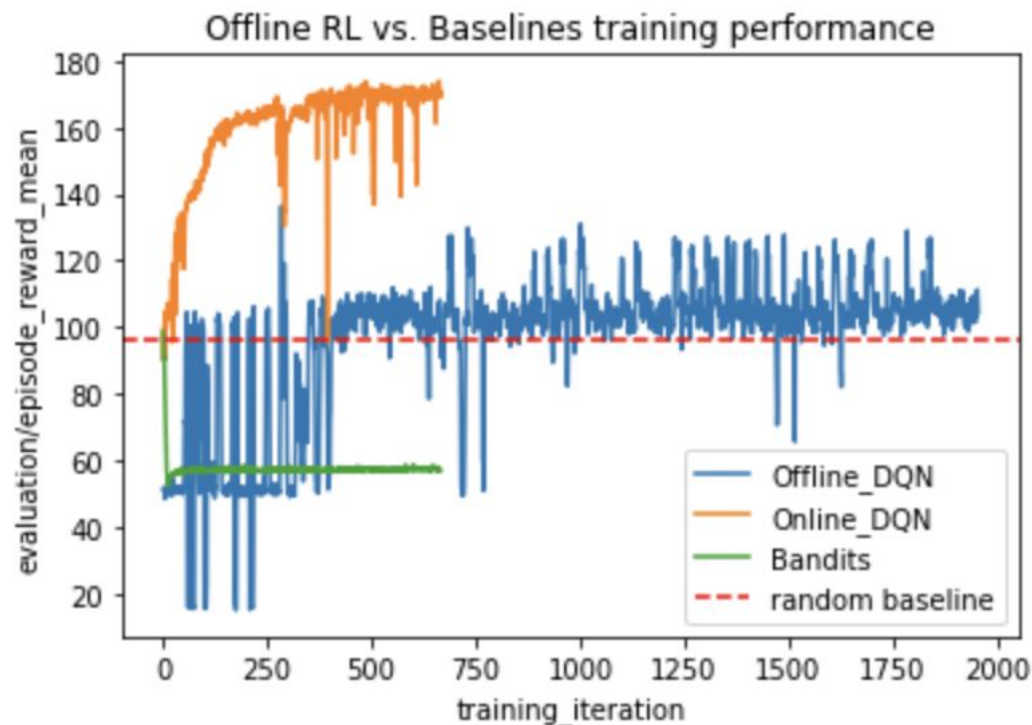


Online RL vs Offline RL

Online RL	Batch / Offline RL
<p data-bbox="253 401 562 467">Get back next reward r' and next state s'</p> <p data-bbox="108 478 224 527">Agent</p> <p data-bbox="548 478 788 516">Environment</p>  <p data-bbox="295 841 469 904">From State s, Take action a</p>	<p data-bbox="1035 445 1190 478">Get back next reward r' and next state s'</p> <p data-bbox="962 483 1029 511">Agent</p> <p data-bbox="1184 483 1300 511">Environment</p>  <p data-bbox="1054 669 1145 702">From State s, Take action a</p> <p data-bbox="1489 429 1760 483">$[s, a, s', r', \dots]$</p>



Sample result from notebook





Your Anyscale Cluster

1. Claim username/password at https://bit.ly/rlib_recsys-logins
 - a. Update the “**Status**” column to “**Claimed**”
2. Go to Console: <http://console.anyscale.com/>
3. Enter username (for the email) and password



Scale your application from
your laptop to the cloud

Get started

Work email

1

john@acme.com

2

3

Next



Your Anyscale Cluster

anyscale

- Home
- Projects
- Interactive sessions
- Jobs
- Services
- Clusters**
- Configurations

Clusters

+ Create **Start** **Terminate** **Archive**

Search names Cluster status Created by Include archived

<input type="checkbox"/>	Name	Status ↓	Active resources	Cost ? ↑↓	Cluster environment
<input type="checkbox"/>	cluster-0	Terminated	None	\$0.80	ray_tutorial_app_config_allentest200:1

4

5



Your Anyscale Cluster

anyscale

Home

Projects

Interactive sessions

Jobs

Services

Clusters

Configurations

Ray-Tutorial > cluster-0

click

Jupyter

Dashboard

Grafana

Terminate

About this cluster

Status	ID	Created by
Active (auto-suspend in 2880 minutes)	ses_QSpdJDjX3pu4Xz93iD9Sb7p	yinhaonan55+200@gmail.com
Created at	Access	Project
Jul 18, 2022, 2:13:41 PM	Only admins and you can view and edit	Ray-Tutorial

Resource usage

CPU	Object store memory	GPU
0 utilized / 8 running	0 B utilized / 6.87 GiB running	-
Cost since last start	Cost since creation	
\$0.80	\$0.80	




Your Anyscale Cluster

/


Name	Last Modified
academy	3 days ago

Launcher


Notebook




Python 3 (ipykernel)




Console




Python 3 (ipykernel)



Terminal



Text File



Markdown File

click



\$ ray@ip-10-0-104-35: ~/Ray X

```
(base) ray@ip-10-0-104-35:~/Ray-Tutorial$ ls
```

academy

```
(base) ray@ip-10-0-104-35:~/Ray-Tutorial$ cd academy
```

```
(base) ray@ip-10-0-104-35:~/Ray-Tutorial/academy$ git pull
```

```
remote: Enumerating objects: 121, done.
```

```
remote: Counting objects: 100% (121/121), done.
```

```
remote: Compressing objects: 100% (78/78), done.
```

```
remote: Total 108 (delta 53), reused 70 (delta 29), pack-reused 0
```

```
Receiving objects: 100% (108/108), 4.76 MiB | 4.13 MiB/s, done.
```

```
Resolving deltas: 100% (53/53), completed with 8 local objects.
```

```
From https://github.com/anyscale/academy
```

```
    a5ee457..405730b  main      -> origin/main
```




Your Anyscale Cluster


/

Name	Last Modified
academy	3 days ago


click

Launcher


Notebook



Python 3
(ipykernel)




Console




Python 3
(ipykernel)

\$ _


Other



Terminal




Text File







Markdown File











Your Anyscale Cluster

 File Edit View Run Kernel Tabs Settings Help

 +   

/ academy /

Name	Last Modified
 advanced-ray	5 hours ago
 images	5 hours ago
 ray-cluster-launcher	5 hours ago
 ray-crash-course	5 hours ago
 ray-project	5 hours ago
 ray-rlib	5 hours ago
 ray-serve	5 hours ago

 **click**



Your Anyscale Cluster

The screenshot shows the Anyscale Ray console interface. At the top is a menu bar with options: File, Edit, View, Run, Kernel, Tabs, and Settings. Below the menu is a toolbar with icons for creating a new folder, opening a folder, uploading, and refreshing. The main area displays the current path as `/ academy / ray-rlib /`. Below this is a table with two columns: "Name" and "Last Modified". The table contains one entry: a folder named `acm_recsys_tutorial_2022` which was modified "6 minutes ago". This folder name is highlighted with an orange rectangular box. An orange arrow points from the word "click" below the box to the folder name.

Name	Last Modified
acm_recsys_tutorial_2022	6 minutes ago

click



Your Anyscale Cluster

Filter files by name

/ ... / ray-rlib / acm_recsys_tutorial_2022 /


Name	Last Modified
images	3 hours ago
rlib_recsim	3 hours ago
saved_runs	3 hours ago
slides	3 hours ago
solutions	an hour ago
tutorial_scripts	3 hours ago
00_anyscale_acm_recsys_tutorial_tabl...	a Sep 16, 2022 5:05 PM
01_intro_gym_and_rllib_optional.ipynb	11 hours ago
• 02_anyscale_acm_recsys_tutorial.ipynb	3 hours ago
README.md	3 hours ago
requirements.txt	3 hours ago

02_anyscale_acm_recsys_...

Code

Python 3 (ipykern)

© 2019–2022, Anyscale. All Rights Reserved



Main Tutorial Notebook for RLlib ACM RecSys 2022

Learning objectives

In this this tutorial, you will learn about:

- [Defining a MDP for recommendation system using gym API](#) -15 min
- [Online RL - Bandits](#) -15 min
- [Online RL - DQN](#) -15 min
- [Break](#) -5 min
- [Offline RL](#) -15 min

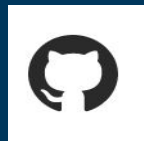
Open notebook "02_..."

Thank you.

We would love to connect with you!



Twitter – @anyscalecompute | @raydistributed



GitHub – <https://github.com/ray-project/ray>



Slack – <https://www.ray.io/community>



Discuss – <https://discuss.ray.io/>

class **Algorithm**(tune.Trainable)

WorkerSet
(trainer.workers)

“local worker”
class **RolloutWorker**

Policy Map

Pol1

Mo
del

Pol2

Mo
del

```
config.rollouts(  
    num_rollout_workers=0  
)
```

@ray.remote
class **RolloutWorker**

@ray.remote
class **RolloutWorker**

Scalability (e.g.
num_workers=100)

@ray.remote
class **RolloutWorker**

Policy Map

Pol1

Model

Pol2

Model

```
config.rollouts(  
    num_rollout_workers > 0  
)
```

Sampler

Vector Env

Ag1

Ag2

```
config.rollouts(  
    num_envs_per_worker  
)
```