DOTE 6635: Artificial Intelligence for Business Research (Spring 2026)

Reinforcement Learning Basics

Renyu (Philip) Zhang

1

Agenda

- **Why RL and MAB**
- MDP
- Value Iteration

2

2

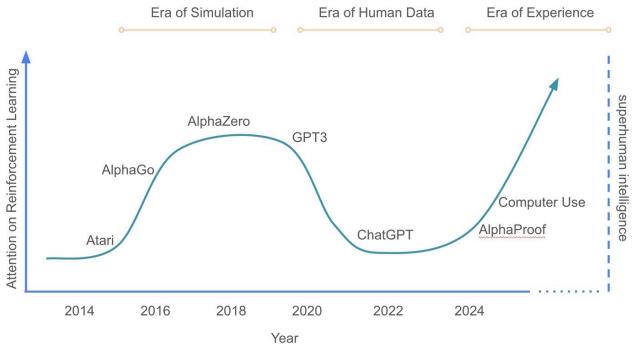
The Bitter Lesson 2.0

Welcome to the Era of Experience

David Silver, Richard S. Sutton*

Abstract

We stand on the threshold of a new era in artificial intelligence that promises to achieve an unprecedented level of ability. A new generation of agents will acquire superhuman capabilities by learning predominantly from experience. This note explores the key characteristics that will define this upcoming era.



纸上得来终觉浅
绝知此事要躬行

Prof. Richard Sutton

Prof. David Silver

https://www.youtube.com/watch?v=zzXyPGEtseI
https://storage.googleapis.com/deepmind-media/Era-of-Experience%20/The%20Era%20of%20Experience%20Paper.pdf

3

3

The Second Half of AI

Shunyu Yao
姚顺雨

The Second Half

tldr: We're at AI's halftime.

For decades, AI has largely been about developing new training methods and models. And it worked: from beating world champions at chess and Go, surpassing most humans on the SAT and bar exams, to earning IMO and IOI gold medals. Behind these milestones in the history book — DeepBlue, AlphaGo, GPT-4, and the o-series — are fundamental innovations in AI methods: search, deep RL, scaling, and reasoning. Things just get better over time.

So what's suddenly different now?

In three words: RL finally works. More precisely: RL finally generalizes. After several major detours and a culmination of milestones, we've landed on a working recipe to solve a wide range of RL tasks using language and reasoning. Even a year ago, if you told most AI researchers that a single recipe could tackle software engineering, creative writing, IMO-level math, mouse-and-keyboard manipulation, and long-form question answering — they'd laugh at your hallucinations. Each of these tasks is incredibly difficult and many researchers spend their entire PhDs focused on just one narrow slice.

Yet it happened.

So what comes next? The second half of AI — starting now — will shift focus from solving problems to defining problems. In this new era, evaluation becomes more important than training. Instead of just asking, "Can we train a model to solve X?", we're asking, "What should we be training AI to do, and how do we measure real progress?" To thrive in this second half, we'll need a timely shift in mindset and skill set, ones perhaps closer to a product manager.

Only after GPT-2 or GPT-3, it turned out that the missing piece is priors. You need powerful language pre-training to distill general commonsense and language knowledge into models, which then can be fine-tuned to become web (WebGPT) or chat (ChatGPT) agents (and change the world). **It turned out the most important part of RL might not even be the RL algorithm or environment, but the priors, which can be obtained in a way totally unrelated from RL.**

https://ysymyth.github.io/The-Second-Half/

Learning internal representations by error-propagation 59436 *
DE Rumelhart, GE Hinton, RJ Williams
Parallel Distributed Processing: Explorations in the Microstructure of ...

???

Imagenet classification with deep convolutional neural networks 171956 *
A Krizhevsky, I Sutskever, GE Hinton
Advances in neural information processing systems 25

ImageNet Large Scale Visual Recognition Challenge 49275
O Russakovsky, J Deng, H Su, J Krause, S Satheesh, S Ma, Z Huang, ...
arXiv preprint arXiv:1409.0575, 2014

Playing atari with deep reinforcement learning 16853
V Mnih, K Kavukcuoglu, D Silver, A Graves, I Antonoglou, D Wierstra, ...
arXiv preprint arXiv:1312.5602, 2013

The arcade learning environment: An evaluation platform for general agents 3844
MG Bellemare, Y Naddaf, J Veness, M Bowling
Journal of Artificial Intelligence Research 47, 253-279

Attention is all you need 168293
A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, AN Gomez, ...
Advances in neural information processing systems 30, 2017

Findings of the 2014 workshop on statistical machine translation 1332
O Bojar, C Buck, C Federmann, B Haddow, P Koehn, J Leveling, C Monz, ...
Proceedings of the ninth workshop on statistical machine translation, 12-58

Language models are few-shot learners 48882 *
T Brown, B Mann, N Ryder, M Subbiah, JD Kaplan, P Dhariwal, ...
Advances in neural information processing systems 33, 1877-1901

Superglue: A sticker benchmark for general-purpose language understanding systems 2483
A Wang, Y Pruksachatkun, N Nangia, A Singh, J Michael, F Hill, O Levy, ...
Advances in neural information processing systems 32

The second half

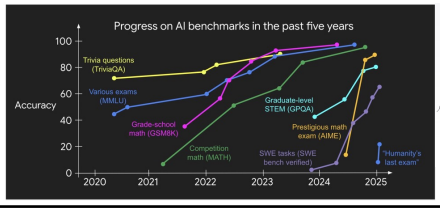
This recipe is completely changing the game. To recap the game of the first half:

- We develop novel training methods or models that hillclimb benchmarks.
- We create harder benchmarks and continue the loop.

This game is being ruined because:

- The recipe has essentially standardized and industried benchmark hillclimbing without requiring much more new ideas. As the recipe scales and generalizes well, your novel method for a particular task might improve it by 5%, while the next o-series model improve it by 30% without explicitly targeting it.
- Even if we create harder benchmarks, pretty soon (and increasingly soon) they get solved by the recipe. My colleague Jason Wei made a beautiful figure to visualize the trend well:

We should fundamentally re-think **evaluation** and make it aligned with practical utility!



推出人工智能代理"Evergreen"基准

xbench 排行榜 · 第一波

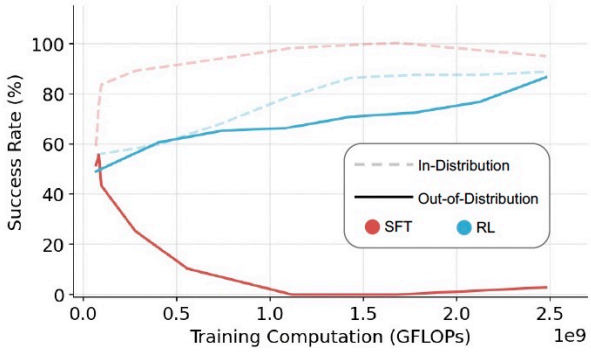
https://xbench.org/

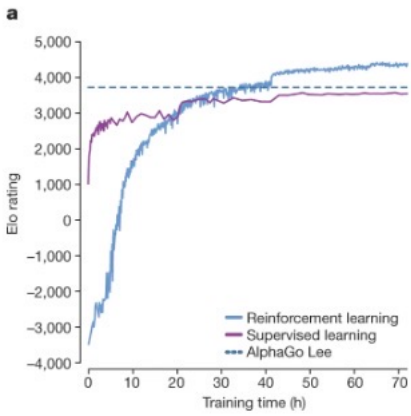
4

Why Do We Need RL?

STF Memorizes, RL Generalizes: https://arxiv.org/pdf/2501.17161; AlphaGo Zero: https://www.nature.com/articles/nature24270

You cannot beat your teacher by imitation.





Sft memorizes, rl generalizes: A comparative study of foundation model post-training

T Chu, Y Zhai, J Yang, S Tong, S Xie… - arXiv preprint arXiv …, 2025 - arxiv.org

… Despite **RL**'s superior **generalization**, we show that **SFT** is still helpful for effective **RL** training: **SFT** stabilizes … While **RL** exhibits superior **generalization** compared to **SFT**, we show that …

☆ Save 99 Cite Cited by 3 Related articles All 2 versions »

Mastering the game of go without human knowledge

D Silver, J Schrittwieser, K Simonyan, I Antonoglou… - nature, 2017 - nature.com

… expert moves, and by reinforcement **learning** from self-play. Here we introduce an algorithm … reinforcement **learning**, **without human** data, guidance or domain **knowledge** beyond **game** …

☆ Save 99 Cite Cited by 11902 Related articles All 42 versions Web of Science: 4555 »

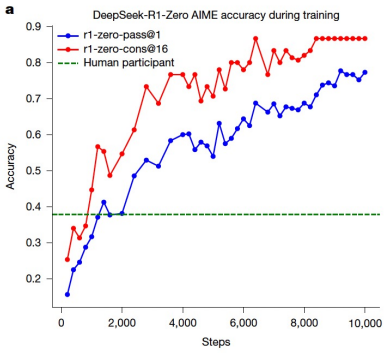
5

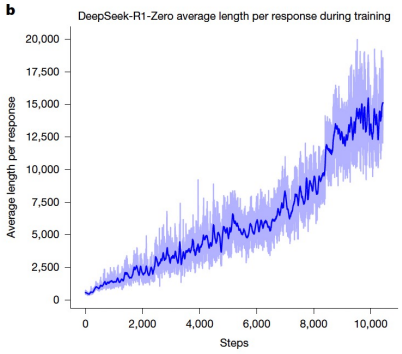
5

DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning

https://www.nature.com/articles/s41586-025-09422-z

- DeepSeek-R1, the first open sourced reasoning model, much cheaper than but as good as OpenAI-o1.
  - No reward model, just the rule-based final reward in accuracy and format (by DeepSeek-V3).
  - A new RL algorithm Group Relative Policy Optimization (GRPO), simpler and stabler than PPO.
- DeepSeek-R1-Zero, purely RL, no SFT: Self-evolving intelligence:
  - Poor readability
  - Language mix





Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a-\sqrt{a+x}} = x$ is equal to

Response: <think>
To solve the equation $\sqrt{a-\sqrt{a+x}} = x$, let's start by squaring both ···
$\left(\sqrt{a-\sqrt{a+x}}\right)^2 = x^2 \implies a - \sqrt{a+x} = x^2$.
Rearrange to isolate the inner square root term:
$(a-x^2)^2 = a+x \implies a^2 - 2ax^2 + (x^2)^2 = a+x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$
···
Wait, wait. Wait. That's an aha moment I can flag here.
Let's reevaluate this step-by-step to identify if the correct sum can be ···
We started with the equation:
$\sqrt{a-\sqrt{a+x}} = x$
First, let's square both sides:
$a - \sqrt{a+x} = x^2 \implies \sqrt{a+x} = a - x^2$
Next, I could square both sides again, treating the equation: ···
···

Video on DeepSeek-R1: https://www.bilibili.com/video/BV1YGR8YUEEd/
Coding GRPO from scratch: https://github.com/aburkov/theLMbook/blob/main/GRPO_From_Scratch_Multi_GPU_DataParallel_Qwen_2_5_1_5B_Instruct.ipynb

6

6

SL vs. RL

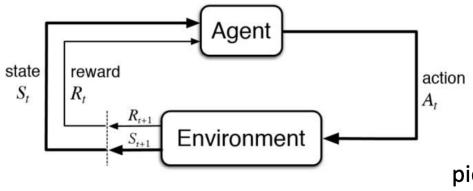
https://rail.eecs.berkeley.edu/deeprlcourse-fa23/deeprlcourse-fa23/static/slides/lec-1.pdf

supervised learning

input: $\mathbf{x}$
output: $\mathbf{y}$
data: $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}$
goal: $f_\theta(\mathbf{x}_i) \approx \mathbf{y}_i$

someone gives this to you

reinforcement learning



pick your own actions

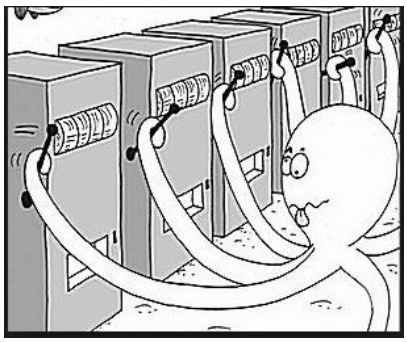
input: $\mathbf{s}_t$ *at each time step*
output: $\mathbf{a}_t$ *at each time step*
data: $(\mathbf{s}_1, \mathbf{a}_1, r_1, \ldots, \mathbf{s}_T, \mathbf{a}_T, r_T)$
goal: learn $\pi_\theta : \mathbf{s}_t \rightarrow \mathbf{a}_t$
to maximize $\sum_t r_t$

7

7

MAB: Simplest RL

https://github.com/callmespring/RL-short-course/blob/main/Lecture%201/slides.pdf



- A bandit machine has multiple arms.
- Pulling each arm yields an independent reward.
- The reward distribution is unknown.
- Limited resource, time, traffic or money.
- **Objective**: Dynamically which arm to pull at each time to maximize the expected cumulative rewards.
- **Model**:
  - The machine has K arms, $\{1,2,\ldots,K\}$
  - In period t, arm $A_t \in \{1,2,\ldots,K\}$ is pulled, with reward R_t.
  - Expected reward of arm a: $Q(a) = E[R_t | A_t = a]$, which is unknown.
  - **Objective**: maximize $\sum_{t=1}^{T} E[R_t]$

8

8

Naïve Approach: Greedy

https://github.com/callmespring/RL-short-course/blob/main/Lecture%201/slides.pdf

- **Greedy**: Estimate the expected reward of each action and choose the action with the maximum expected reward.

$$\hat{Q}_t(a) = \frac{\sum_{i=1}^{t} R_i \mathbb{I}(A_i = a)}{\sum_{i=1}^{t} \mathbb{I}(A_i = a)} \qquad A_t = \arg\max_a \hat{Q}_{t-1}(a)$$

- **Question**: Is this a good policy?
  - 2 Ads, CTR_1 = 0.2, CTR_2 = 0.15, but unknown to us.
  - After 100 rounds, due to bad luck, estimated CTR_1 = 0.1 and CTR_2 = 0.13.
  - Greedy algorithm will pick ad 2 forever afterwards.

Greedy will be locked into the wrong decision forever!

9

9

Exploitation Exploration Trade-off

https://github.com/callmespring/RL-short-course/blob/main/Lecture%201/slides.pdf

- **Exploitation**: Maximize the current reward based on the available information.
- **Exploration**: Collect more information for better future decisions.
- Balancing exploitation and exploration: epsilon-greedy.
  - With probability 1-epsilon, select the arm with the maximum expected reward.
  - With probability epsilon, select an arm uniformly random.
- Epsilon is a tunable parameter and should be high at the beginning and decrease to 0 overtime. Why?
- Choice of epsilon: $\epsilon_t = t^{-1/3} \cdot (K \log t)^{1/3}$

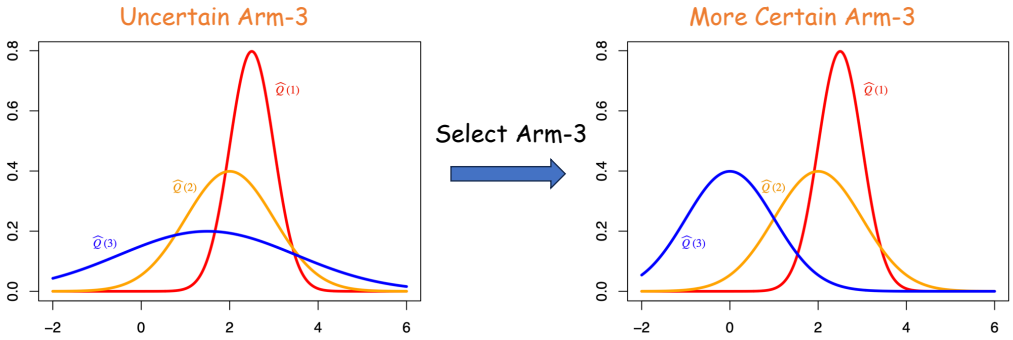
10

10

Optimism in the Face of Uncertainty (OFU)

https://github.com/callmespring/RL-short-course/blob/main/Lecture%201/slides.pdf

- The more uncertain arm should be explored with higher priority.



- Two ways to implement the OFU:
  - Upper confidence bound (UCB)
  - Thompson Sampling (TS)

11

11

UCB and TS Algorithms

https://github.com/callmespring/RL-short-course/blob/main/Lecture%201/slides.pdf
https://arxiv.org/pdf/1904.07272

UCB

- **Input**: some positive constant c, termination time T.
- **Initialization**: $t = 0$, $\widehat{Q}(a) = 0$, $\mathbb{N}(a) = 0$, for $a = 1, 2, \cdots, k$.
- **While** $t < T$:
  - **Update** t: $t \leftarrow t + 1$.
  - **UCB action selection**:

Usually, c=2

$$a^* \leftarrow \arg\max_a [\widehat{Q}(a) + \sqrt{c \log(t)/\mathbb{N}_t(a)}].$$

  - **Receive reward** R from arm a^*.
  - **Update** $\mathbb{N}(a^*)$: $\mathbb{N}(a^*) \leftarrow \mathbb{N}(a^*) + 1$.
  - **Update** $\widehat{Q}(a^*)$:

$$\widehat{Q}(a^*) \leftarrow \frac{\mathbb{N}(a^*) - 1}{\mathbb{N}(a^*)} \widehat{Q}(a^*) + \frac{1}{\mathbb{N}(a^*)} R.$$

TS (Bernoulli Reward)

- **Input**: hyper-parameters $\alpha, \beta > 0$, termination time T.
- **Initialization**: $t = 0$, $S_a = F_a = 0$, for $a = 1, 2, \cdots, k$.
- **While** $t < T$:
  - **Update** t: $t \leftarrow t + 1$.
  - **Posterior sampling**: For $a = 1, 2, \cdots, k$, sample

$$\theta_a \sim \text{Beta}(S_a + \alpha, F_a + \beta)$$

  - **Action selection**: $a^* \leftarrow \arg\max_a \theta_a$.
  - **Receive reward** R from arm a^*.
  - **Update** S_{a^*} and F_{a^*}:
    - If $R = 1$, $S_{a^*} \leftarrow S_{a^*} + 1$;
    - If $R = 0$, $F_{a^*} \leftarrow F_{a^*} + 1$.

Performance metric: Regret(t) = E[max_reward*t - cumulative reward by your algorithm].
- Minimally, we want Regret(T) = o(T), i.e., Regret(T)/T → 0 as T → +inf
- UCB and TS achieve the minimal theoretical regret.

12

12

Is MAB Really Effective in Practice?

Home > Marketing Science > Vol. 36, No. 4 >

Customer Acquisition via Display Advertising Using Multi-Armed Bandit Experiments

Eric M. Schwartz, Eric T. Bradlow, Peter S. Fader

Published Online: 20 Apr 2017 | https://doi.org/10.1287/mksc.2016.1023

Abstract

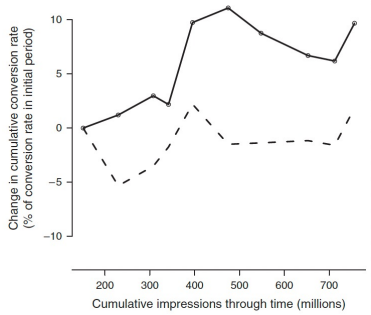
Firms using online advertising regularly run experiments with multiple versions of their ads since they are uncertain about which ones are most effective. During a campaign, firms try to adapt to intermediate results of their tests, optimizing what they earn while learning about their ads. Yet how should they decide what percentage of impressions to allocate to each ad? This paper answers that question, resolving the well-known "learn-and-earn" trade-off using multi-armed bandit (MAB) methods. The online advertiser's MAB problem, however, contains particular challenges, such as a hierarchical structure (ads within a website), attributes of actions (creative elements of an ad), and batched decisions (millions of impressions at a time), that are not fully accommodated by existing MAB methods. Our approach captures how the impact of observable ad attributes on ad effectiveness differs by website in unobserved ways, and our policy generates allocations of impressions that can be used in practice. We implemented this policy in a live field experiment delivering over 750 million ad impressions in an online display campaign with a large retail bank. Over the course of two months, our policy achieved an 8% improvement in the customer acquisition rate, relative to a control policy, without any additional costs to the bank. Beyond the actual experiment, we performed counterfactual simulations to evaluate a range of alternative model specifications and allocation rules in MAB policies. Finally, we show that customer acquisition would decrease by about 10% if the firm were to optimize click-through rates instead of conversion directly, a finding that has implications for understanding the marketing funnel.

Data is available at https://doi.org/10.1287/mksc.2016.1023.

How should advertisers allocate traffic impression to each ad to maximize customer acquisition?

TS + heterogeneous generalized linear model

Figure 3. Results Observed in the Field Experiment



Customer acquisition via display advertising using multi-armed bandit experiments

EM Schwartz, ET Bradlow, PS Fader - Marketing Science, 2017 - pubsonline.informs.org

… Beyond the actual experiment, we performed counterfactual simulations to evaluate a range … customer acquisition would decrease by about 10% if the firm were to optimize click-through …

☆ 保存 99 引用 被引用次数：437 相关文章 所有 15 个版本 Web of Science: 170 »

13

13

MAB in Action

Home > Management Science > Vol. 69, No. 7 >

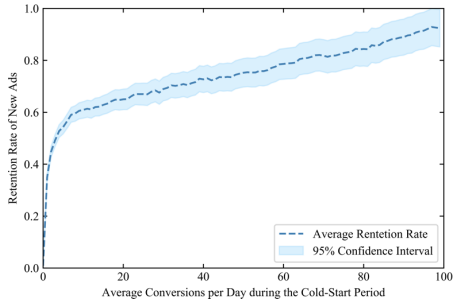
Cold Start to Improve Market Thickness on Online Advertising Platforms: Data-Driven Algorithms and Field Experiments

Zikun Ye, Dennis J. Zhang, Heng Zhang, Renyu Zhang, Xin Chen, Zhiwei Xu

Published Online: 17 Oct 2022 | https://doi.org/10.1287/mnsc.2022.4550

Abstract

Cold start describes a commonly recognized challenge for online advertising platforms: with limited data, the machine learning system cannot accurately estimate the click-through rates (CTR) of new ads and, in turn, cannot efficiently price these new ads or match them with platform users. Traditional cold start algorithms often focus on improving the learning rates of CTR for new ads to improve short-term revenue, but unsuccessful cold start can prompt advertisers to leave the platform, decreasing the thickness of the ad marketplace. To address these issues, we build a data-driven optimization model that captures the essential trade-off between short-term revenue and long-term market thickness on the platform. Based on duality theory and bandit algorithms, we develop the shadow bidding with learning (SBL) algorithms with a provable regret upper bound of $O(T^{2/3}K^{1/3}(\log T)^{1/3}d^{1/2})$, where K is the number of ads and d captures the error magnitude of the underlying machine learning oracle for predicting CTR. Our proposed algorithms can be implemented in a real online advertising system with minimal adjustments. To demonstrate this practicality, we have collaborated with a large-scale video-sharing platform, conducting a novel, two-sided randomized field experiment to examine the effectiveness of our SBL algorithm. Our results show that the algorithm increased the cold start success rate by 61.62% while compromising short-term revenue by only 0.717%. Our algorithm has also boosted the platform's overall market thickness by 3.13% and its long-term advertising revenue by (at least) 5.35%. Our study bridges the gap between the theory of bandit algorithms and the practice of cold start in online advertising, highlighting the value of well-designed cold start algorithms for online advertising platforms.



Tackle the cold-start problem from an operations lens: epsilon-Greedy + Duality + Neural Nets.

Market thickness +3.13%; ad revenue +5.35%.

Two-sided experiment design addresses interference.

Cold start to improve market thickness on online advertising platforms: Data-driven algorithms and field experiments

Z Ye, DJ Zhang, H Zhang, R Zhang… - Management …, 2023 - pubsonline.informs.org

… We study the cold start problem as a data-driven optimization model and offer efficient algorithms to tackle this challenge. Viewing the problem as ad allocation in the repeated-auction …

☆ 保存 99 引用 被引用次数：60 相关文章 所有 10 个版本 »

14

14

Home > Marketing Science > Vol. 38, No. 2 >

Dynamic Online Pricing with Incomplete Information Using Multiarmed Bandit Experiments

Kanishka Misra, Eric M. Schwartz, Jacob Abernethy

Published Online: 29 Mar 2019 | https://doi.org/10.1287/mksc.2018.1129

Abstract

Pricing managers at online retailers face a unique challenge. They must decide on real-time prices for a large number of products with incomplete demand information. The manager runs price experiments to learn about each product's demand curve and the profit-maximizing price. In practice, balanced field price experiments can create high opportunity costs, because a large number of customers are presented with suboptimal prices. In this paper, we propose an alternative dynamic price experimentation policy. The proposed approach extends multiarmed bandit (MAB) algorithms from statistical machine learning to include microeconomic choice theory. Our automated pricing policy solves this MAB problem using a scalable distribution-free algorithm. We prove analytically that our method is asymptotically optimal for any weakly downward sloping demand curve. In a series of Monte Carlo simulations, we show that the proposed approach performs favorably compared with balanced field experiments and standard methods in dynamic pricing from computer science. In a calibrated simulation based on an existing pricing field experiment, we find that our algorithm can increase profits by 43% during the month of testing and 4% annually.

Data files and the online appendix are available at https://doi.org/10.1287/mksc.2018.1129.

Dynamic online pricing with **incomplete information using multiarmed bandit experiments**
K Misra, EM Schwartz, J Abernethy - Marketing Science, 2019 - pubsonline.informs.org
… are presented with suboptimal **prices**. In this paper, we propose an alternative **dynamic price experimentation** policy. The proposed approach extends **multiarmed bandit** (MAB) …
☆ Save 99 Cite Cited by 293 Related articles All 7 versions Web of Science: 78 »

Home > Marketing Science > Vol. 42, No. 2 >

A Multiarmed Bandit Approach for House Ads Recommendations

Nicolás Aramayo, Mario Schiappacasse, Marcel Goic

Published Online: 14 Jul 2022 | https://doi.org/10.1287/mksc.2022.1378

Abstract

Nowadays, websites use a variety of recommendation systems to decide the content to display to their visitors. In this work, we use a multiarmed bandit approach to dynamically select the combination of house ads to exhibit to a heterogeneous set of customers visiting the website of a large retailer. House ads correspond to promotional information displayed on the website to highlight some specific products and are an important marketing tool for online retailers. As the number of clicks they receive not only depends on their own attractiveness but also on how attractive are other products displayed around them, we decide about complete collections of ads that capture those interactions. Moreover, as ads can wear out, in our recommendations we allow for nonstationary rewards. Furthermore, considering the sparsity of customer-level information, we embed a deep neural network to provide personalized recommendations within a bandit scheme. We tested our methods in controlled experiments where we compared them against decisions made by an experienced team of managers and the recommendations of a variety of other bandit policies. Our results show a more active exploration of the decision space and a significant increment in click-through and add-to-cart rates.

History: Olivier Toubia served as the senior editor and Hema Yoganarasimhan served as associate editor for this article.

Funding: This work was supported by the Institute for Research in Market Imperfections and Public Policy [Grant ICM IS130002]. M. Schiappacasse gratefully acknowledges partial funding by the Institute for Research in Market Imperfections and Public Policy (IS130002 ANID).

Supplemental Material: The data files and online appendix are available at https://doi.org/10.1287/mksc.2022.1378.

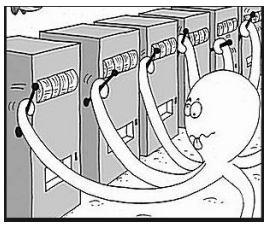
A **multiarmed bandit approach** for **house ads recommendations**
N Aramayo, M Schiappacasse, M Goic - Marketing Science, 2023 - pubsonline.informs.org
… Nowadays, websites use a variety of **recommendation** systems to decide the content to … use a **multiarmed bandit approach** to dynamically select the combination of **house ads** to exhibit …
☆ Save 99 Cite Cited by 37 Related articles All 7 versions

15

15

When Should We Use MAB in Practice?

- We face a simple enough decision-making setting with independent feedback.
- Exploitation vs. exploration is the key trade-off with very limited prior information.
- Cold start for new products, new ads, etc.



What if we need to make decisions in a more complex world?

16

16

Agenda

- Why RL and MAB
- **MDP**
- Value Iteration & Policy Iteration

17

17

Markov Process

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf

"The future is independent of the past given the present"

Definition

A state S_t is *Markov* if and only if

$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, ..., S_t]$$

Definition

A *Markov Process* (or *Markov Chain*) is a tuple $\langle \mathcal{S}, \mathcal{P} \rangle$

- $\mathcal{S}$ is a (finite) set of states
- $\mathcal{P}$ is a state transition probability matrix, $\mathcal{P}_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s]$

You need to pay attention to the ergodicity of the Markov Chain

Andrey Markov

$$\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$$

to

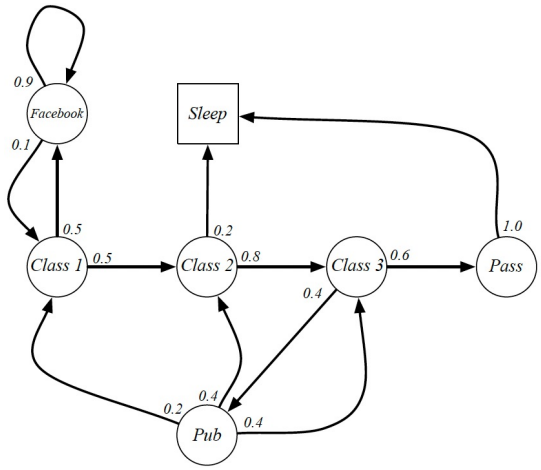
$$\mathcal{P} \quad = \textit{from} \begin{bmatrix} \mathcal{P}_{11} & \dots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{n1} & \dots & \mathcal{P}_{nn} \end{bmatrix}$$

18

18

Markov Process

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf



$$\mathcal{P} = \begin{array}{c} \\ C1 \\ C2 \\ C3 \\ Pass \\ Pub \\ FB \\ Sleep \end{array} \begin{array}{c} \begin{array}{ccccccc} C1 & C2 & C3 & Pass & Pub & FB & Sleep \end{array} \\ \begin{bmatrix} & 0.5 & & & & 0.5 & \\ & & 0.8 & & & & 0.2 \\ & & & 0.6 & 0.4 & & \\ & & & & & & 1.0 \\ 0.2 & 0.4 & 0.4 & & & & \\ 0.1 & & & & & 0.9 & \\ & & & & & & 1 \end{bmatrix} \end{array}$$

Student Markov Chain Transition Matrix

19

19

Markov Reward Process

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf

A Markov reward process is a Markov chain with values.

Definition

A *Markov Reward Process* is a tuple $\langle \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

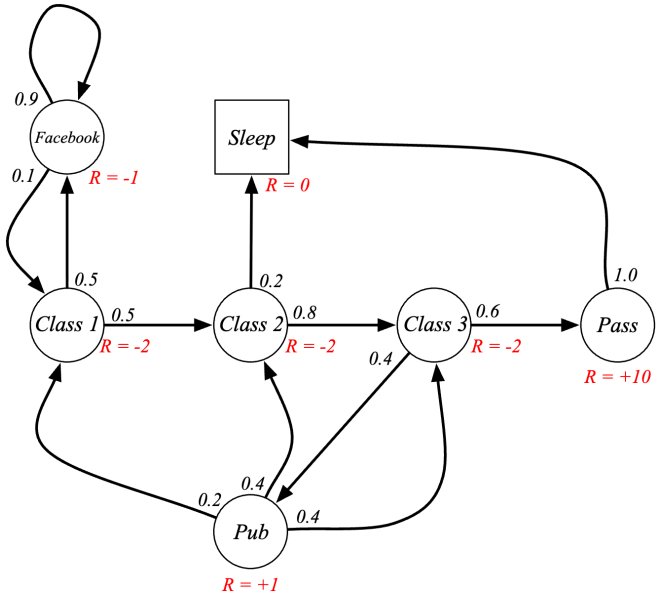
- $\mathcal{S}$ is a finite set of states
- $\mathcal{P}$ is a state transition probability matrix, $\mathcal{P}_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s]$
- $\mathcal{R}$ is a reward function, $\mathcal{R}_s = \mathbb{E}[R_{t+1} \mid S_t = s]$
- γ is a discount factor, $\gamma \in [0, 1]$

20

20

Student Markov Reward Process

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf



21

21

(Discounted) Return and Value

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf

Definition

The *return* G_t is the total discounted reward from time-step t.

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

The value function $v(s)$ gives the long-term value of state s

Definition

The *state value function* $v(s)$ of an MRP is the expected return starting from state s

$$v(s) = \mathbb{E}[G_t \mid S_t = s]$$

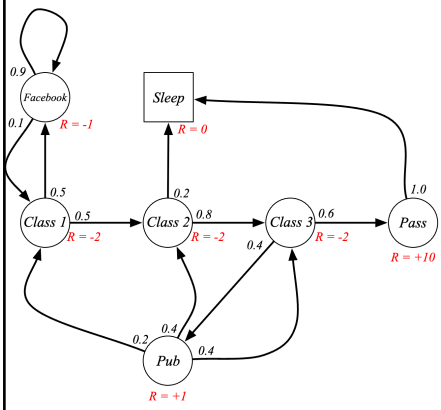
What does it mean if γ is close to 0 or close to 1?

22

22

Student MRP Returns

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf



Sample returns for Student MRP:
Starting from $S_1 = \text{C1}$ with $\gamma = \frac{1}{2}$

$$G_1 = R_2 + \gamma R_3 + \ldots + \gamma^{T-2} R_T$$

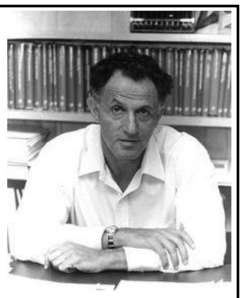
C1 C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 10 * \frac{1}{8} = -2.25$
C1 FB FB C1 C2 Sleep	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} = -3.125$
C1 C2 C3 Pub C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 1 * \frac{1}{8} - 2 * \frac{1}{16} \ldots = -3.41$
C1 FB FB C1 C2 C3 Pub C1 …	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} \cdots = -3.20$
FB FB FB C1 C2 C3 Pub C2 Sleep	

23

23

Bellman Equation for MRP

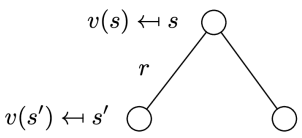
https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf



Richard Bellman

- The value function consists of two parts:
  - Immediate reward R_{t+1}
  - Discounted value of successor state $\gamma v(S_{t+1})$

$$\begin{aligned} v(s) &= \mathbb{E}[G_t \mid S_t = s] \\ &= \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s\right] \\ &= \mathbb{E}\left[R_{t+1} + \gamma\left(R_{t+2} + \gamma R_{t+3} + \ldots\right) \mid S_t = s\right] \\ &= \mathbb{E}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right] \\ &= \mathbb{E}\left[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s\right] \end{aligned}$$



$$v(s) = \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} v(s')$$

Bellman equation in matrix form:

$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_n \end{bmatrix} + \gamma \begin{bmatrix} \mathcal{P}_{11} & \ldots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{11} & \ldots & \mathcal{P}_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}$$

$$v = \mathcal{R} + \gamma \mathcal{P} v \longrightarrow v = (I - \gamma \mathcal{P})^{-1} \mathcal{R}$$

- Computational complexity: $O(n^3)$
- Direct solution is only possible for small-scale MRPs.

24

24

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf

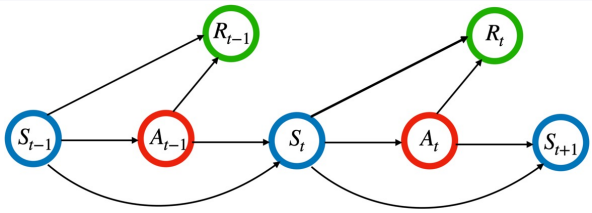
A Markov decision process (MDP) is a Markov reward process with decisions. It is an *environment* in which all states are Markov.

Definition

A *Markov Decision Process* is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathcal{P}$ is a state transition probability matrix, $\mathcal{P}^{a}_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$
- $\mathcal{R}$ is a reward function, $\mathcal{R}^{a}_{s} = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$
- γ is a discount factor $\gamma \in [0, 1]$.

Markov Decision Process

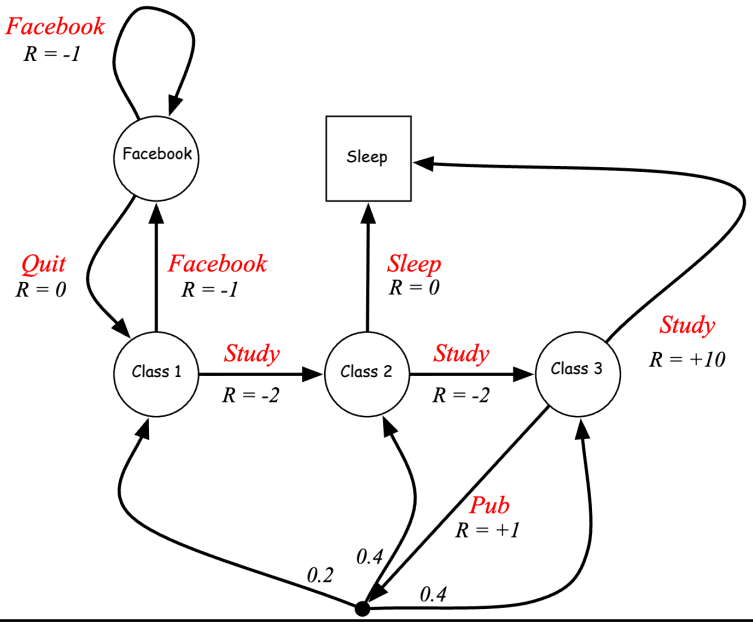


25

25

Student Markov Decision Process

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf



26

26

Markovian Policy

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf

Definition

A *policy* π is a distribution over actions given states,

$$\pi(a|s) = \mathbb{P}[A_t = a \mid S_t = s]$$

- A policy π fully defines the behavior of an agent.
- MDP policies depend on the current state, but not the history.
- So, policies are stationary (i.e., time-independent), $A_t \sim \pi(.|S_t)$

- Given an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ and a policy π
- The state sequence $S_1, S_2, \ldots$ is a Markov process $\langle \mathcal{S}, \mathcal{P}^\pi \rangle$
- The state and reward sequence $S_1, R_2, S_2, \ldots$ is a Markov reward process $\langle \mathcal{S}, \mathcal{P}^\pi, \mathcal{R}^\pi, \gamma \rangle$
- where

$$\mathcal{P}^\pi_{s,s'} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}^a_{ss'}$$
$$\mathcal{R}^\pi_s = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}^a_s$$

27

27

Value Functions

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf

Definition

The *state-value function* $v_\pi(s)$ of an MDP is the expected return starting from state s, and then following policy π

$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$$

Definition

The *action-value function* $q_\pi(s, a)$ is the expected return starting from state s, taking action a, and then following policy π

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$$

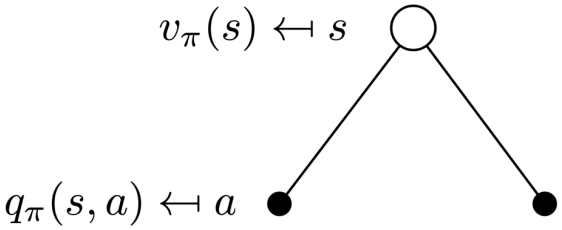
28

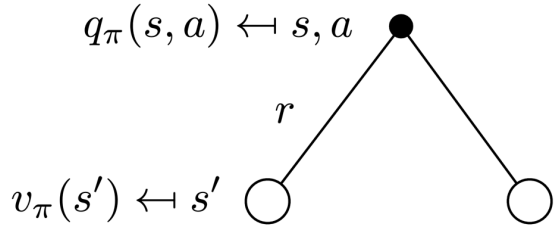
28

Bellman Equations

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf

$v_\pi(s) = \mathbb{E}_\pi\left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s\right]$ $q_\pi(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\right]$





$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) q_\pi(s,a)$$

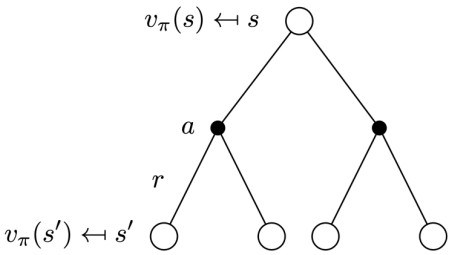
$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s')$$

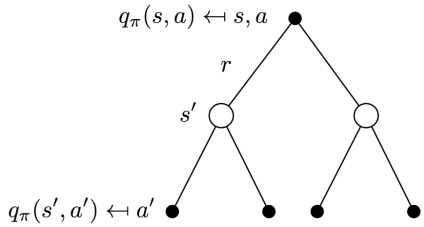
29

29

Bellman Equations

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf





$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s') \right)$$

$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a'|s') q_\pi(s', a')$$

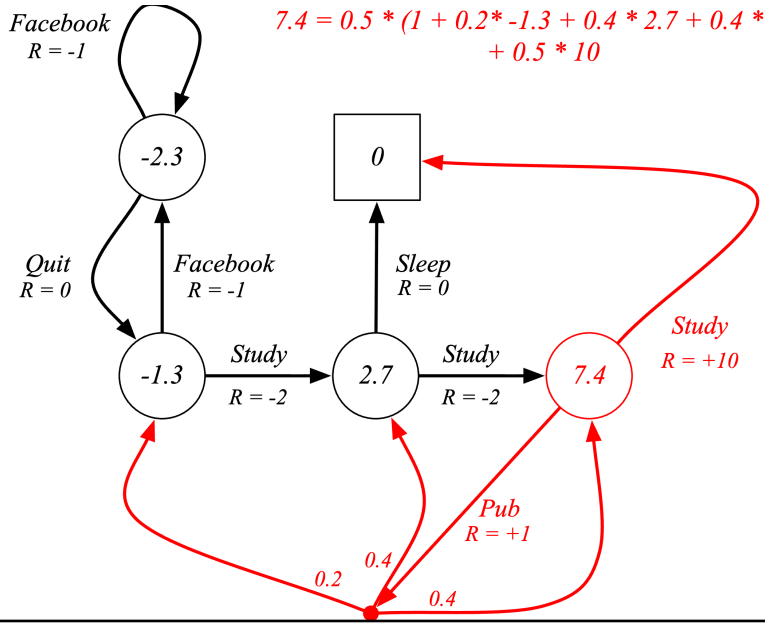
In matrix forms: $v_\pi = \mathcal{R}^\pi + \gamma \mathcal{P}^\pi v_\pi \quad v_\pi = (I - \gamma \mathcal{P}^\pi)^{-1} \mathcal{R}^\pi$

30

30

Bellman Equations for Student MDP

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf

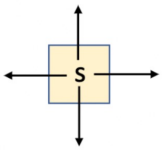


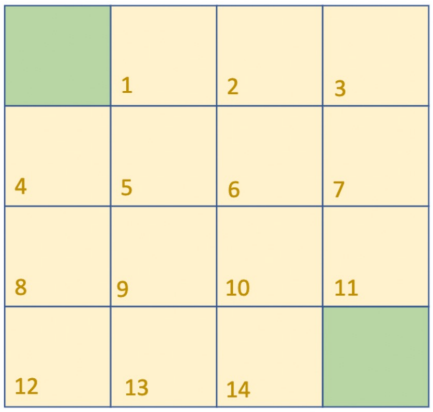
31

31

GridWorld Example

https://github.com/callmespring/RL-short-course/blob/main/Lecture%202/slides.pdf





terminal state

- Deterministic environment
- Action: {up, down, right, left}
- Reward = -1 for each move.

What would be the value function of the uniformly random policy?

$$\pi(n|\cdot) = \pi(s|\cdot) = \pi(w|\cdot) = \pi(e|\cdot) = 0.25$$

32

32

GridWorld Value Function

https://github.com/callmespring/RL-short-course/blob/main/Lecture%202/slides.pdf

By **symmetry** and Bellman equation,

0	v_1 1	v_2 2	v_3 3
v_1 4	v_5 5	v_6 6	v_2 7
v_2 8	v_6 9	v_5 10	v_1 11
v_3 12	v_2 13	v_1 14	0

$$v_1 = \frac{1}{4}(-1+v_2)+\frac{1}{4}(-1+v_5)+\frac{1}{4}(-1+0)+\frac{1}{4}(-1+v_1)$$
$$v_2 = \frac{1}{4}(-1+v_3)+\frac{1}{4}(-1+v_6)+\frac{1}{4}(-1+v_1)+\frac{1}{4}(-1+v_2)$$
$$v_3 = \frac{1}{4}(-1+v_3)+\frac{1}{4}(-1+v_2)+\frac{1}{4}(-1+v_2)+\frac{1}{4}(-1+v_3)$$
$$v_5 = \frac{1}{4}(-1+v_6)+\frac{1}{4}(-1+v_6)+\frac{1}{4}(-1+v_1)+\frac{1}{4}(-1+v_1)$$
$$v_6 = \frac{1}{4}(-1+v_2)+\frac{1}{4}(-1+v_5)+\frac{1}{4}(-1+v_5)+\frac{1}{4}(-1+v_2)$$

$$\Rightarrow (v_1, v_2, v_3, v_5, v_6) = (-14, -20, -22, -18, -20)$$

33

33

Policy Evaluation Algorithm

https://github.com/callmespring/RL-short-course/blob/main/Lecture%202/slides.pdf

- **Input**: a policy π, a threshold parameter $\epsilon > 0$
- **Initialization**: $V(s) = 0$ for any $s \in \mathcal{S}$
- **Repeat**:

  $\Delta \leftarrow 0$

  For each $s \in \mathcal{S}$

  $\nu \leftarrow V(s)$

  $V(s) \leftarrow \sum_{a \in \mathcal{A}} \pi(a|s)\left[\mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a V(s')\right]$

  $\Delta \leftarrow \max(\Delta, |\nu - V(s)|)$

  until $\Delta < \epsilon$
- **Output** V

34

34

Policy Evaluation Algorithm

https://github.com/callmespring/RL-short-course/blob/main/Lecture%202/slides.pdf

$k = 0$

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

$k = 1$

0.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	0.0

$k = 2$

0.0	-1.7	-2.0	-2.0
-1.7	-2.0	-2.0	-2.0
-2.0	-2.0	-2.0	-1.7
-2.0	-2.0	-1.7	0.0

$k = 3$

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

$k = 10$

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

$k = \infty$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0

Figure: Value functions at each iteration

35

35

Optimal Value Function

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf

Definition

The *optimal state-value function* $v_*(s)$ is the maximum value function over all policies

$$v_*(s) = \max_\pi v_\pi(s)$$

The *optimal action-value function* $q_*(s, a)$ is the maximum action-value function over all policies

$$q_*(s, a) = \max_\pi q_\pi(s, a)$$

- The optimal value function specifies the best possible performance in the MDP.
- An MDP is "solved" once we know the optimal value function.

36

36

Existence of Optimal Policy

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf

- Define: $\pi \geq \pi'$ if $v_\pi(s) \geq v_{\pi'}(s), \forall s$

Theorem

For any Markov Decision Process

- *There exists an optimal policy π_* that is better than or equal to all other policies, $\pi_* \geq \pi, \forall \pi$*
- *All optimal policies achieve the optimal value function, $v_{\pi_*}(s) = v_*(s)$*
- *All optimal policies achieve the optimal action-value function, $q_{\pi_*}(s,a) = q_*(s,a)$*

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \underset{a \in \mathcal{A}}{\text{argmax}}\ q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

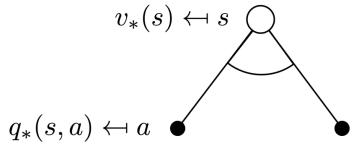
- There is always a deterministic optimal policy for any MDP.

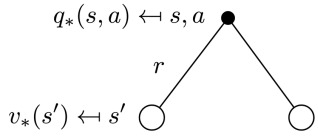
37

37

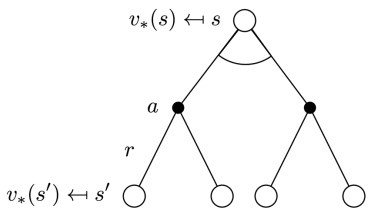
Bellman Optimality Equations

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf

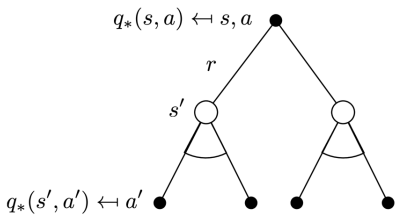




$$v_*(s) = \max_a q_*(s,a)$$



$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s')$$



$$v_*(s) = \max_a \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s')$$

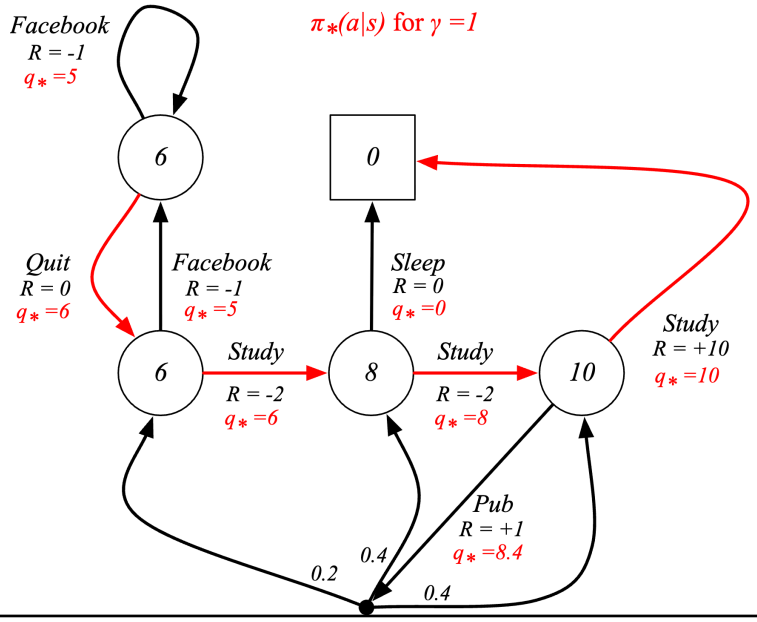
$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \max_{a'} q_*(s',a')$$

38

38

Optimal Policy for Student MDP

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf



39

39

Remarks on MDP Formulation

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf

- Always make sure rewards are determined by current states and actions.
- Always make sure the future states only depend on the current state.
  - This is not true in general. When it is not true, encode history into states.
- If there is an absorbing state, we model it as an infinite horizon undiscounted problem ($\gamma = 1$) until reaching the absorbing state.
  - E.g., search problems.
- So far we assume we know all information: state space S, transition probability P, action space A, and reward function R.
  - In general RL, some, or even all, of these assumptions is relaxed.
  - People often call MDP with all such information vs. RL where some information is absent.

40

40

Solving MDP

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf

- Traditional dynamic programming methods:
  - Value iteration
  - Policy iteration
- Traditional RL methods:
  - Sarsa
  - Q-learning
- Modern DRL methods:
  - DQN
  - TRPO
  - PPO
  - GRPO
  - Many more……

41

41

Agenda

- Why RL and MAB
- MDP
- **Value Iteration & Policy Iteration**

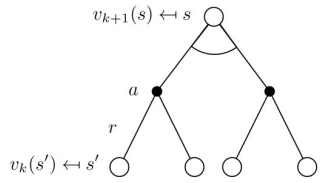
42

42

From Bellman to Value Iteration

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-3-planning-by-dynamic-programming-.pdf

- Recall the Bellman optimality condition: $v_*(s) = \max_a \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s')$
- Iteratively update the optimal policy based on the Bellman optimality condition:



$$v_{k+1}(s) = \max_{a \in \mathcal{A}} \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_k(s') \right)$$

$$\mathbf{v}_{k+1} = \max_{a \in \mathcal{A}} \mathcal{R}^a + \gamma \mathcal{P}^a \mathbf{v}_k$$

- **Initialization**: $\boldsymbol{V}(s) = \mathbf{0}$, $\pi(s) \in \mathcal{A}$ arbitrarily for any $s \in \mathcal{S}$
- **Repeat**:
  - $\Delta \leftarrow \mathbf{0}$
  - **For each** $s \in \mathcal{S}$
    - $\nu \leftarrow \boldsymbol{V}(s)$
    - $\boldsymbol{V}(s) \leftarrow \max_{a \in \mathcal{A}} \left[\mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a \boldsymbol{V}(s') \right]$ (Bellman optimality operator)
    - $\Delta \leftarrow \mathbf{max}(\Delta, |\nu - \boldsymbol{V}(s)|)$
  - **until** $\Delta < \epsilon$
- **Output**: optimal deterministic policy given by

$$\pi^{opt}(s) = \arg\max_{a \in \mathcal{A}} \left[\mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a \boldsymbol{V}^{\pi^{\text{opt}}}(s') \right].$$

- Bellman optimality operator is a contraction mapping if $\gamma < 1$.

43

43

New Results in Value Iteration

QUANTITATIVE ECONOMICS
JOURNAL OF THE ECONOMETRIC SOCIETY

Original Articles | Open Access

Strong convergence and dynamic economic models

Robert L. Bray

First published: 11 February 2019 | https://doi.org/10.3982/QE833

PDF TOOLS SHARE

Abstract

Morton and Wecker (1977) stated that the value iteration algorithm solves a dynamic program's policy function faster than its value function when the limiting Markov chain is ergodic. I show that their proof is incomplete, and provide a new proof of this classic result. I use this result to accelerate the estimation of Markov decision processes and the solution of Markov perfect equilibria.

Strong convergence and dynamic economic models
RL Bray - Quantitative Economics, 2019 - Wiley Online Library
Morton and Wecker (1977) stated that the value iteration algorithm solves a dynamic program's policy function faster than its value function when the limiting Markov chain is ergodic. I …
☆ 保存 99 引用 被引用次数：14 相关文章 所有 20 个版本 »

Home > Management Science > Vol. 65, No. 10 >

Markov Decision Processes with Exogenous Variables

Robert L. Bray

Published Online: 24 Apr 2019 | https://doi.org/10.1287/mnsc.2018.3158

Abstract

I present two algorithms for solving dynamic programs with exogenous variables: endogenous value iteration and endogenous policy iteration. These algorithms are always at least as fast as relative value iteration and relative policy iteration, and they are faster when the endogenous variables converge to their stationary distributions sooner than the exogenous variables.

This paper was accepted by Yinyu Ye, optimization.

Markov decision processes with exogenous variables
RL Bray - Management Science, 2019 - pubsonline.informs.org
… Markov decision process will eventually forget the current action as its state variables … The agent's actions influence the endogenous variable but not the exogenous variable. Specifically…
☆ 保存 99 引用 被引用次数：26 相关文章 所有 9 个版本 »

44

44

Policy Iteration

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-3-planning-by-dynamic-programming-.pdf

- Policy Evaluation:

$$v_{k+1}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_k(s') \right)$$

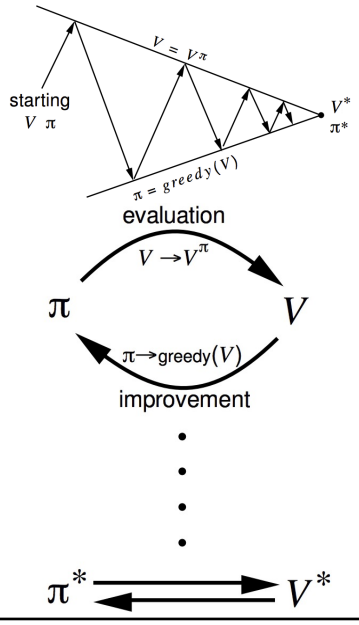
- (Greedy) Policy Improvement: $\pi'(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, q_\pi(s, a)$

$$q_\pi(s, \pi'(s)) = \max_{a \in \mathcal{A}} q_\pi(s, a) \geq q_\pi(s, \pi(s)) = v_\pi(s)$$

$$\begin{aligned} v_\pi(s) &\leq q_\pi(s, \pi'(s)) = \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s\right] \\ &\leq \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s\right] \\ &\leq \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 q_\pi(S_{t+2}, \pi'(S_{t+2})) \mid S_t = s\right] \\ &\leq \mathbb{E}_{\pi'}\left[R_{t+1} + \gamma R_{t+2} + \ldots \mid S_t = s\right] = v_{\pi'}(s) \end{aligned}$$

- Once the improvement process stops, we find the optimal policy that satisfied the Bellman optimality.

$$q_\pi(s, \pi'(s)) = \max_{a \in \mathcal{A}} q_\pi(s, a) = q_\pi(s, \pi(s)) = v_\pi(s)$$



45

45

Policy Iteration Algorithm

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-3-planning-by-dynamic-programming-.pdf

- **Initialization**: $V(s) = 0$, $\pi(s) \in \mathcal{A}$ arbitrarily for any $s \in \mathcal{S}$
- **Repeat**:
  - $\Delta \leftarrow 0$
  - **For each** $s \in \mathcal{S}$
    - $\nu \leftarrow V(s)$
    - $V(s) \leftarrow \sum_{a \in \mathcal{A}} \pi(a|s) \left[\mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a V(s') \right]$
    - $\Delta \leftarrow \max(\Delta, |\nu - V(s)|)$
  - **until** $\Delta < \epsilon$
- **policystable** $\leftarrow$ **True**
- **For each** $s \in \mathcal{S}$:
  - $b \leftarrow \pi(s)$
  - $\pi(s) \leftarrow \arg\max_a [\mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a V(s')]$
  - **If** $b \neq \pi(s)$ **then policystable** $\leftarrow$ **False**
- **If policystable, then Return** π, **else** go to bullet point #2

Bullet point #2

Policy iteration also enjoys exponential convergence.

$$\|Q^\star - Q^{\pi_{k+1}}\|_\infty \leq \gamma \|Q^\star - Q^{\pi_k}\|_\infty$$

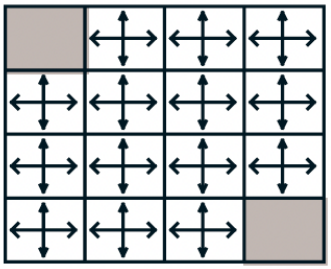
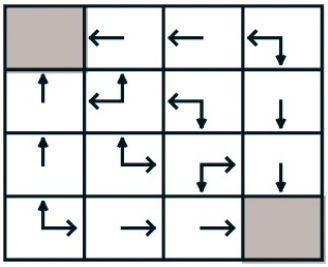
https://nanjiang.cs.illinois.edu/files/cs542/note1.pdf

46

46

Policy Iteration in Small GridWorld

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-3-planning-by-dynamic-programming-.pdf

	v_π	π	
Iteration 1	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0		Random Policy
Iteration 2	0.0 -14. -20. -22. -14. -18. -20. -20. -20. -20. -18. -14. -22. -20. -14. 0.0		Optimal Policy

47

47

Value Iteration vs. Policy Iteration

https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-2-mdp.pdf

- Value iteration:
  - Pro: each iteration is fast
  - Con: Convergence is slow and asymptotic (i.e., it depends on the parameter epsilon)
- Policy iteration:
  - Pro: convergence is guaranteed in finite number of iterations (often very small in practice)
  - Con: policy evaluation in each iteration may be slow and expensive
- Practically, policy iteration is faster than value iteration in general.
- If the discount factor γ is close to 1, both methods are slow.

48

48

Another Perspective: Linear Programming

https://nanjiang.cs.illinois.edu/files/cs542/note1.pdf

- The MDP problem can be formulated as the following LP, where T is the Bellman optimality operator:

Primal Form The primal LP is:

$$\min_{V \in \mathbb{R}^S} d_0^\top V$$
$$\text{s.t. } V \geq \mathcal{T}V.$$

- This LP has |S| decision variables and |S*A| linear constraints: $V(s) \geq R(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)}[V(s')],\ \forall a \in \mathcal{A}.$
- The optimal solution to the LP is the optimal value function to MDP:

Proposition 5. *$V \geq \mathcal{T}V \Rightarrow V \geq V^\star$, and $V^\star$ is feasible.*

Dual Form Taking the dual of the primal LP yields:

$$\max_{d \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}, d \geq \mathbf{0}} d^\top R$$
$$\text{s.t. } [\textstyle\sum_{a\in\mathcal{A}} d(s,a)]_{s\in\mathcal{S}} = \gamma P^\top d + (1-\gamma) d_0.$$

Policy $\pi(a|s) = \dfrac{d(s,a)}{\sum_{a' \in \mathcal{A}} d(s,a')}$

49

Estimation of Single-Agent Dynamic Model

Econometrica, Vol. 80, No. 5 (September, 2012), 2213–2230

NOTES AND COMMENTS

CONSTRAINED OPTIMIZATION APPROACHES TO ESTIMATION OF STRUCTURAL MODELS

BY CHE-LIN SU AND KENNETH L. JUDD[1]

Estimating structural models is often viewed as computationally difficult, an impression partly due to a focus on the nested fixed-point (NFXP) approach. We propose a new constrained optimization approach for structural estimation. We show that our approach and the NFXP algorithm solve the same estimation problem, and yield the same estimates. Computationally, our approach can have speed advantages because we do not repeatedly solve the structural equation at each guess of structural parameters. Monte Carlo experiments on the canonical Zurcher bus-repair model demonstrate that the constrained optimization approach can be significantly faster.

KEYWORDS: Structural estimation, dynamic discrete choice models, constrained optimization.

Constrained optimization approaches to estimation of structural models

CL Su, KL Judd - Econometrica, 2012 - Wiley Online Library

Estimating structural models is often viewed as computationally difficult, an impression partly due to a focus on the nested fixed-point (NFXP) approach. We propose a new constrained optimization approach for structural estimation. We show that our approach and the NFXP algorithm solve the same estimation problem, and yield the same estimates. Computationally, our approach can have speed advantages because we do not repeatedly solve the structural equation at each guess of structural parameters. Monte Carlo …

☆ 保存 99 引用 被引用次数：586 相关文章 所有 21 个版本 »

Combining the sample-averaged augmented log-likelihood function as the objective function and the integrated Bellman equations as constraints yields the following constrained optimization problem:

$$(18) \qquad \max_{(\theta, \mathrm{EV})} \frac{1}{M}\mathcal{L}(\theta, \mathrm{EV})$$
$$\text{subject to} \quad \mathrm{EV} = T(\mathrm{EV}, \theta).$$

It follows from Proposition 1 that the maximum-likelihood estimation problem defined by (16) and the constrained optimization problem (18) are mathematically equivalent. Consequently, NFXP and MPEC yield the same parameter estimates. Dubé, Fox, and Su (2012) proved the equivalence in solutions to the first-order conditions of NFXP and MPEC in the context of random-coefficients demand estimation. Their results and proof can be applied directly to the single-agent dynamic models considered here as well.

$$\mathcal{L}(\theta, \mathrm{EV}) = \sum_{i=1}^{M}\sum_{t=2}^{T} \log[P(d_t^i|x_t^i;\theta)] + \sum_{i=1}^{M}\sum_{t=2}^{T} \log[p_3(x_t^i|x_{t-1}^i, d_{t-1}^i;\theta_3)]$$
$$= \sum_{i=1}^{M}\sum_{t=2}^{T} \log\left(\frac{\exp[\nu(x_t^i, d_t^i, \theta) + \beta\,\mathrm{EV}(x_t^i, d_t^i)]}{\sum_{d'\in\{0,1\}} \exp[\nu(x_t^i, d', \theta) + \beta\,\mathrm{EV}(x_t^i, d')]}\right) + \sum_{i=1}^{M}\sum_{t=2}^{T} \log[p_3(x_t^i|x_{t-1}^i, d_{t-1}^i;\theta_3)].$$

50