Risk aversion in learning algorithms and recommendation systems *

Andreas Haupt Aroon Narayanan

July 11, 2022

Abstract

Consider online learning algorithms that simultaneously make decisions and learn from feedback. Such algorithms are widely deployed in recommendation systems for products and digital content. This article exhibits a bias of online learning algorithms towards less risky alternatives, and how it shapes demand on recommendation systems. First, we consider k-armed bandits. We prove that ε-Greedy chooses a riskless arm over a risky arm of equal expected reward with probability arbitrarily close to one. This is a consequence of undersampling of arms with bad reward estimates. Through experiments, we show that other online learning algorithms exhibit risk aversion as well. In a recommendation system environment we show that content that yields less noisy reward from users is favored by the algorithm. Combined with equilibrium forces driving strategic content creators towards content of similar expected quality, the advantage for content that is not necessarily better, just less volatile, is exaggerated.

1 Introduction

Online learning algorithms are algorithms that simultaneously make decisions and learn how to make future decisions better. They are widely used in the digital economy, in particular to help users navigate a plethora of options for products such as digital content and e-commerce products. As they are used in more and more settings of economic interest, it becomes important to understand the economic implications of their use. We demonstrate a bias that can be found both in simple (k-armed bandit) and more complex (recommendation systems) online learning algorithms: risk aversion.

To be more precise, take the simple example of providing the learning algorithm with two options—either pull lever A and get a certain payoff of 0, or pull lever B and get a stochastic payoff of either 1 or -1, distributed uniformly. At any point in time, contingent on past observations of payoffs from the pulled action (the *bandit setting*), an algorithm specifies a distribution on actions it takes next. What is the probability that the learning algorithm chooses each of the arms after T rounds of interaction? How likely is the learning algorithm to choose a deterministic arm of the same expectation as a volatile arm?

A risk averse learning algorithm would choose the arm that yields 0 reward with certainty over the other arm. We prove that a classic algorithm, ε-Greedy, chooses the deterministic arm with probability 1 in the limit (i.e. after sufficient rounds of interaction, it chooses the deterministic arm with probability close to 1). Note however that ε-Greedy is not designed to be risk-averse. It keeps an average utility received from each lever, and, chooses with some probability the arm with the highest reward or a random arm. Risk aversion appears to be an emergent property of the algorithm.

Risk aversion is more than a mere intellectual curiosity. A concrete setting in which it can play a significant role is recommendation systems, which regularly improve their recommendations using feedback from users. The choices that the recommendation algorithm takes, such as which pieces of content to show to a user next or which products to provide as an answer to a search query in ecommerce, are determined

*We thank the mathoverflow.com user fedja for a helpful reply, and David Parkes and seminar audiences at Harvard for helpful comments.

by how valuable the recommendation is deemed to be. It is conceivable that one of the choices gives a less noisy estimate of user preference because it makes it easier for the recommendation system to observe user actions when interacting with it.

Our analysis complements analyses of online learning that are based on asymptotic guarantees. In the above example, where we show perfect risk aversion of ε-Greedy, both arms have equal expected rewards and hence asymptotic regret makes no predictions. But even with two arms with slightly differing expected rewards, we are able to demonstrate that learning algorithms display consistent risk aversion in finite time. Thus, until asymptotic guarantees kick in, learning algorithms may have hitherto unrecognized biases. In a world where data is big and changes quickly, transient behaviour can become pivotal. This necessitates the study of the kind of biases that we demonstrate.

Our contributions are threefold:

1. We first prove that a classic bandit algorithm, ε-Greedy, acts like a risk-averse agent in the example above: As $T \to \infty$, it chooses the riskless arm with probability one. The proof exposes the mechanism that leads to algorithmic risk preferences: If algorithms hold estimates of the quality of their actions as ε-Greedy does, these estimates are biased because algorithms undersample arms after bad draws.

2. We simulate other algorithms to show the importance of risk aversion in finite time. We show that they also tend to choose options of lower risk, even facing arms of moderate differences in reward. As a further quantification of risk aversion, we define and determine experimentally *certainty equivalents* for non-deterministic options for each time T, i.e. the deterministic payoff an online learning algorithms would need to get to choose the deterministic option with $\frac{1}{2}$ probability after T rounds.

3. Finally, we show the effects of risk aversion in a realistic economic environment. We use a production-level recommendation system, Autorec Sedhain et al. (2015), which takes into account both that recommendation systems are not independently running bandit learners for different users, but learn across users, and high user heterogeneity. Risk aversion leads to significantly lower demand for content that gives more volatile feedback. This may be seen as a bias of an algorithm to recommend, and generate demand for content whose performance may more easily be monitored. We discuss the implications of such a bias if content creators are strategic.

The structure of the rest of the article is as follows. In section 2 we provide our online learning setup and our definition of risk aversion. We continue in section 3 with our main theoretical result regarding the risk aversion of ε-Greedy. In section 4, we provide simulations of other online learning algorithms. Our simulation of a recommendation system and a discussion of impacts for content creators can be found in section 5. We collect related literature in section 6. We summarize and discuss our results in section 7. In Appendix A, we provide a definition of risk aversion that accomodates the generality of Markov Decision Processes. Appendix B contains additional Figures.

2 Model and Definitions

In a k-armed bandit problem, a decision maker repeatedly takes an action from a finite set A, $|A| = k$. Each action a is associated to a distribution $F_a \in \Delta(\mathbb{R}), a \in A$. A *strategy* or *algorithm* used by the decision maker is a function $\pi\colon (A \times [0,1])^* \to \Delta([k])$.

An algorithm π generates (potentially random) sequences of *actions* $(a_t)_{t\in\mathbb{N}}$ and *rewards* or *payoffs* $(r_t)_{t\in\mathbb{N}}$. For each $t \in \mathbb{N}$, repeatedly, the algorithm chooses an action $a_t \sim \pi(a_1, r_1, a_2, r_2, ..., a_{t-1}, r_{t-1})$ and gets a reward $r_t \sim F_{a_t}$. We will use the shorthand notation $a_{1:t}$ for $(a_1, a_2, \dots, a_t)$ and $r_{1:t} = (r_1, r_2, \dots, r_t)$. The two main examples we consider in this article are ε-Greedy and the Upper Confidence Band algorithm.

Example 1. ϵ-Greedy ε-Greedy chooses the empirically best action with probability $1-\epsilon$, and randomizes between all the actions with probability ϵ, i.e.

$$\pi_i(a_{1:t-1}, r_{1:t-1}) = \begin{cases} \text{Unif}(\arg\max_{a\in A} \mu_a(t-1)) & \text{w.p. } 1-\varepsilon \\ \text{Unif}(A) & \text{else,} \end{cases}$$

where

$$\mu_a(t-1) \coloneqq \frac{1}{|\{1 \leq t' \leq t-1 : a_{t'} = a\}|} \sum_{1 \leq t' \leq t-1 : a_{t'} = a} r_{t'}$$

is the historical average reward. We say that if ε-Greedy takes an action maximizing $\mu_a(t-1)$ it *exploits*, otherwise, it *explores*.

Example 2. UCB Upper Confidence Band (UCB) algorithm derives an "optimistic" estimate of the mean from the empirical mean, and then maximizes this estimate. Denoting by that $T_i(t-1)$ the number of times action i has been taken until period $t-1$, the algorithm comutes at each period the quantity

$$\mathrm{UCB}_i(t-1,\delta) = \begin{cases} \infty & \text{if } T_i(t-1) = 0 \\ \hat{\mu}_i(t-1) + \sqrt{\frac{2\log(1/\delta)}{T_i(t-1)}} & \text{otherwise,} \end{cases}$$

and maximizes

$$\pi_i(A_1, r_1, A_2, r_2, ..., A_{t-1}, r_{t-1}) = \mathrm{Unif}(\arg\max_{a \in A} \mathrm{UCB}_a(t-1,\delta)).$$

We call a 2-armed bandit problem with $A = \{n, r\}$ such that $F_n = \delta_c$ (without loss of generality 0) and square-integrable F_r such that $\mathbb{E}_{r \sim F_r}[r] = c$ and $\mathrm{Var}_{r \sim F_v}[r] > 0$ a *risky choice*.

Definition 1. *An algorithm π is risk-averse if*

$$\lim_{t \to \infty} \mathbb{P}[a_t = n] > \frac{1}{2}$$

for a risky choice. Further, the algorithm is fully risk-averse if $\lim_{t\to\infty} \mathbb{P}[a_t = n] = 1$.

In Appendix A of the appendix, we give a fully general definition for Reinforcement Learning algorithms.

Note that risk aversion implies that an algorithm will choose the non-risky arm for a larger fraction of periods, in expectation. That is,

$$\lim_{t \to \infty} \frac{1}{t} \mathbb{E}\left[|\{t | a_t = n\}|\right] > \frac{1}{2}.$$

3 Risk Aversion of ε-Greedy Algorithms

We start our analysis of risk aversion with a result on ε-Greedy.

Theorem 1. *For any exploration rate $(\varepsilon_t)_{t \in \mathbb{N}}$ such that $\varepsilon_t \to 0$ and $\sum_{t=0}^{T} \varepsilon_t \to \infty$, and the risky choice between a deterministic 0-reward and a Rademacher distributed reward, ε-Greedy is fully risk-averse.*

The main intuition for this result is that many algorithms undersample actions for which they received low rewards. Riskier actions that get a low reward are "trapped" in pessimistic estimates of reward. This leads to a behaviour consistent with risk aversion over long time spans of learning. This is represented in Figure 1. In the region of advantage for the risky arm above the x-axis, the estimate moves around more, and since expected values are the same, the advantage can quickly dissipate and become negative, at which point the advantage is updated less frequently, which means that the risky arm is undersampled. This can persist for quite long, leading to risk aversion of ε-Greedy.

The theoretical tractability of our analysis in Theorem 1 relies on the fact that ε-Greedy is an index policy with expected value for each of the actions being the corresponding index. The learning dynamics of an ε-Greedy algorithm can be reduced to a random walk (the *advantage walk*), whose asymptotic behavior is well-understood. Many other algorithms can be written as index policies such as EXP3, Thompson Sampling with beta priors and Upper Confidence Band algorithms (compare Lattimore and Szepesvári (2020)); these do not have the same theoretical tractability as their updates are non-linear.

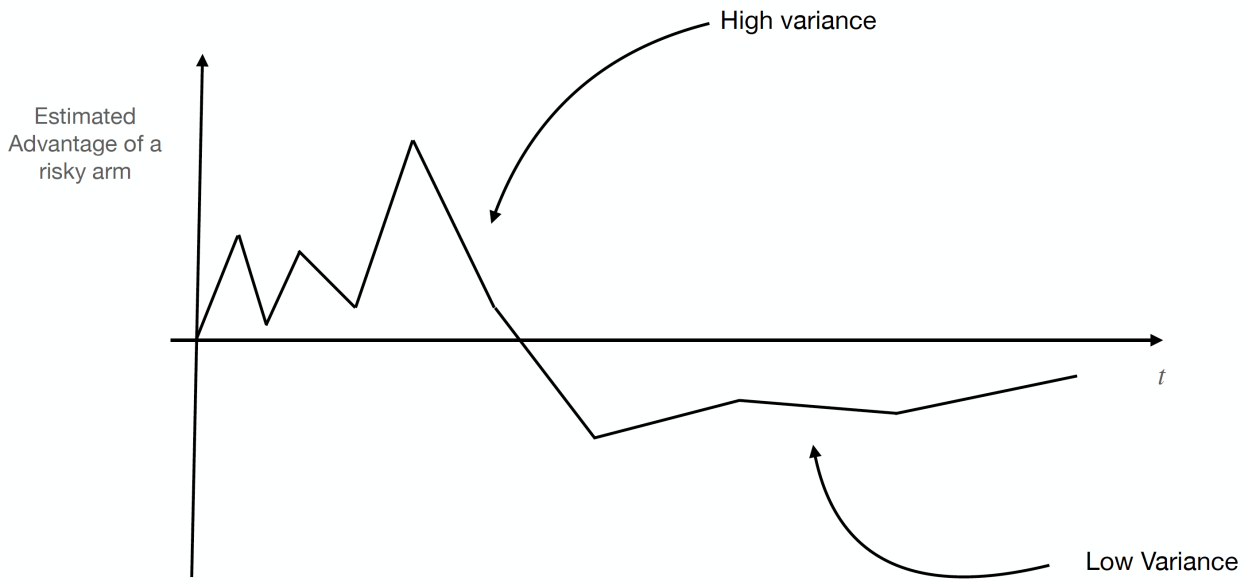


Figure 1: Main intuition for risk aversion in online algorithms. A non-uniform random walk will spend more time in places with less diffusion. Compare Appendix B in the appendix for an instantiation from a run of ε-Greedy.

Proof. First observe that ε-Greedy can be written as a stochastic process of a particularly simple form if it uses a deterministic and one non-deterministic arm:

$$\arg\max_{i\in[k]} \frac{1}{|\{t|A_t=i\}|}\sum_{t:A_t=i} r_i = \begin{cases}\{r\} & \text{if } \sum_{t:A_t=i} r_i > 0\\ \{n\} & \text{if } \sum_{t:A_t=i} r_i < 0\\ \{r,n\} & \text{if } \sum_{t:A_t=i} r_i = 0\end{cases}$$

This can be further simplified by observing $\sum_{t:A_t=i} r_i = \sum_{t=1}^{T} r_t$ as one of the arms has 0 reward. Hence, the quantities $\sum_{t=1}^{T} r_t$ are sufficient statistics for ε_t-Greedy.

Denote the sufficient statistic by $X_T = \sum_{t=1}^{T} r_t$. The transition distribution of $(X_t)_{t\in\mathbb{N}}$ is

$$\begin{aligned} X_0 &= 0\\ X_{t+1} &= \begin{cases} X_t & \text{w.p. } \frac{\varepsilon_t}{2} + (1-\varepsilon_t)(1_{X_t<0} + \frac{1}{2}1_{X_t=0})\\ X_t + x_t & \text{w.p. } \frac{\varepsilon_t}{2} + (1-\varepsilon_t)(1_{X_t>0} + \frac{1}{2}1_{X_t=0}).\end{cases}\end{aligned} \tag{1}$$

where $x \sim$ Rademacher independently across time. $(X_t)_{t\in\mathbb{N}}$ is a (lazy) random walk, which we call *advantage walk*. We first observe that the probability that this process is positive is related to ε-Greedy choosing the non-risky arm.

Claim 1. $\mathbb{P}[X_t \le 0] \to 1$ *as* $t \to \infty \implies \mathbb{P}[A_t = n] \to 1$.

Proof. Note that as $\sum_{t=0}^{T} \varepsilon_t \to \infty$, the advantage walk steps infinitely often almost surely, which means that $\mathbb{P}[X_t = 0] \to 0$.

Furthermore, note that $P[A_t = n|X_t < 0] \ge 1 - \varepsilon_t/2$. Thus, $P[A_t = n] \ge P[A_t = n, X_t < 0] = P[A_t = n|X_t \le 0]P[X_t < 0] \ge (1-\varepsilon_t/2)P[X_t < 0]$. Since $\varepsilon_t \to 0$and $P[X_t < 0] - P[X_t \le 0] \to 0$, the claim follows. □

It is hence sufficient to show $\mathbb{P}[X_t > 0] \to 0$.

Define the time since the last passing time of zero as $\tau_0^T := \max\{t \le T | X_t = 0\}$. Define $S_0^t := \sum_{t'=\tau_0^t}^t S_{t'}$, where $S_{t'} \sim \text{Bernoulli}(1 - \frac{1}{2}\varepsilon_{t'})$ the number of times the lazy random walk steps if it is positive.

Let $(H_t)_{t\in\mathbb{N}}$ be a standard random walk.

Claim 2. $\mathbb{P}[X_t > 0] = \mathbb{P}[H_{t'} > 0, t' = 1, 2, \ldots, S_0^t]$.

Proof. We have that

$$\begin{aligned} X_t > 0 &\iff X_{t'} > 0, t' = \tau_0^t + 1, \tau_0^t + 2, ..., t \\ &\iff H_{t'} > 0, t' = 1, 2, \ldots, S_0^t. \end{aligned}$$

The first line comes from the definition of τ_0^t. For the second line, note that $X_{t'} > 0$ implies that it steps S_0^t times from $t' = \tau_0^t$ to $t' = t$. This is because the risky arm is favored in this region and hence the safe arm is chosen with $\frac{1}{2}\varepsilon_{t'}$ probability, which is when the process does not step. This is equivalent to a standard random walk remaining above 0 for S_0^t periods. □

Claim 3. *As* $t \to \infty$,

$$t - \tau_0^t \to \infty$$

in probability.

Proof. We would like to show:

$$\forall c > 0, \delta > 0 : \exists t \in \mathbb{N} \forall t' \ge t : \mathbb{P}[t - \tau_0^t \le c] \le \delta.$$

Fix any c and any $\delta > 0$. Then:

$$\begin{aligned} \mathbb{P}[t - \tau_0^t \le c] &= \mathbb{P}[\exists t' \in \{t - c, t - c + 1, ..., t\} : X_{t'} = 0] \\ &\le \sum_{t'=t-c}^{t} \mathbb{P}[X_{t'} = 0] \\ &\le \sum_{t'=t-c}^{t} \Big\{ \mathbb{P}\Big[X_{t'} = 0 \Big| |\{s \in [\tau_0^{t'}, t'] | X_{s+1} - X_s \neq 0\}| \ge \kappa\Big] \\ &\quad + \mathbb{P}[|\{s \in [\tau_0^{t'}, t'] | X_{s+1} - X_s \neq 0\}| < \kappa] \Big\} \\ &\le c \max_{l \in [\kappa, T]} \binom{l}{\frac{l}{2}} 2^{-l} \\ &\quad + c\mathbb{P}\Big[\exists m \ge \frac{t - \tau_0^t}{\kappa} \wedge n \in [\tau_0^t, t - m] \Big| X_n = X_{n+1} = ... = X_{n+m}\Big] \end{aligned}$$

For the first inequality, we use the fact that the probability can be split into two, one conditioning on an event A and the other conditioning on its complement A^c, and then replace the latter with the probability of A^c. For the second, the first term just replaces each term in the sum with the largest element of the sum. The second term uses the pigeonhole principle, since the event that X_t steps at most κ times for $t - \tau_t^0$ periods is the same as saying that there is at least one continuous sequence of length $\frac{t-\tau_t^0}{\kappa}$ that does not step.

By Stirling's approximation, the first term is approximately $c \max_{l\in[\kappa,T]} \frac{1}{\sqrt{l\pi}} = \frac{c}{\sqrt{\kappa\pi}} \le \frac{c}{\sqrt{(t-c)\pi}} \to 0$.

For the second term,

$$
\begin{aligned}
&c\mathbb{P}\Big[\exists m \geq \frac{t-\tau_0^t}{\kappa} \wedge n \in [\tau_0^t, t-m] \Big| X_n = X_{n+1} = ... = X_{n+m}\Big] \\
&\leq c \prod_{s' \in \{s, s+1, .., s+\frac{t}{\kappa}\}} (1-\varepsilon_{s'}) \\
&\leq c \exp \sum_{s' \in \{s, s+1, .., s+\frac{t}{\kappa}\}} \varepsilon_{s'}
\end{aligned}
$$

which goes to zero given that $\sum_t \varepsilon_t = \infty$. □

Claim 4. $S_0^t \xrightarrow[t\to\infty]{\mathbb{P}} \infty$.

Proof. Fix $c, \delta > 0$. By (3), we can choose t' such that for any $t \geq t$, $t - \tau_0^t > 2c$ with probability at least $\delta/2$. Choose t large enough such that $\varepsilon_{t''}/2 \leq \kappa \coloneqq 2(\delta/\binom{2c}{c})^{1/c}$ for $t' \geq t - 2c$, which is possible as $\varepsilon_t \to 0$. Then, as at most c zero draws of the Bernoulli random variable need to happen between $t-c$ and t, whose probability is bounded by κ, we can bound

$$
\begin{aligned}
\mathbb{P}[S_0^t \leq c] &\leq \delta/2 + \mathbb{P}[S_0^t \leq c | t - \tau_0^t > 2c] \\
&\leq \delta/2 + \binom{2c}{c}\kappa^c \\
&= \delta.
\end{aligned}
$$

This concludes the proof. □

Claim 5. *For any* $x \in \mathbb{N}_{\geq 0}$

$$
\mathbb{P}[H_t = y, H_{t'} > 0, t' = 1, 2, \ldots, t] = \frac{y\mathbb{P}[|H_t| = y]}{t}.
$$

and therefore

$$
\mathbb{P}[H_{t'} > 0, t' = 1, 2, \ldots, t] = \frac{\mathbb{E}[|H_t|]}{t}.
$$

Proof. Suppose that $H_t = y > 0$. Let $N_t(x, y)$ be the number of ways to get from $(0, x)$ to (t, y). Note that the event $E = \{H_t = y, H_{t'} > 0, t' = 1, 2, \ldots, t\}$ has happened iff the random walk stays on the same side of 0 in the interval $[1, t]$. Let N denote the number of ways to do this, and $\pi = \mathbb{P}(E|S_t = y)$. Then $\pi = \frac{N}{N_t(0,y)}$, but also $\pi = \frac{y}{n}$ by the Reflection Principle. As a result, the total number of ways is $\frac{y}{t}N_t(0, y)$, and each has $\frac{1}{2}(t + y)$ rightward steps and $\frac{1}{2}(t - y)$ leftward steps. Therefore

$$
\mathbb{P}[H_t = y, H_{t'} > 0, t' = 1, 2, \ldots, t] = \frac{y}{t}N_t(0, y)p^{\frac{1}{2}(t+y)}q^{\frac{1}{2}(t-y)} = \frac{y\mathbb{P}[|H_t| = y]}{t}
$$

Summing over y gives the second equation. □

The asymptotics of the absolute value of a random walk are well understood:

$$
\lim_{t\to\infty} \frac{\mathbb{E}[|H_t|]}{\sqrt{t}} = \sqrt{\frac{2}{\pi}}.
$$

See, e.g., Weisstein (2002), and references therein. This implies for large enough t that

Claim 6. *There is a constant* $C > 0$ *such that* $\mathbb{P}[H_{t'} > 0, t' = 1, 2, \ldots, t] \leq Ct^{-\frac{1}{2}}$.

Let $\delta > 0$. Set $c \coloneqq \frac{\sqrt{2}}{\sqrt{\delta}}$ and $\delta' \coloneqq \frac{\delta}{2}$. We find that with probability at least $1 - \delta' = \frac{\delta}{2}$, $S_0^t > c = (2/\varepsilon)^{1/2}$. In particular,

$$\begin{aligned}
\mathbb{P}[X_t > 0] &\leq \mathbb{P}[t - \tau_0^t - S_0^t \leq (2/\varepsilon)^{1/2}] \\
&\quad + \mathbb{P}[t - \tau_0^t - S_0^t > (2/\varepsilon)^{1/2}]\mathbb{E}[X_t \leq 0 | t - \tau_0^t - S_0^t > (2/\varepsilon)^{1/2}] \\
&\leq \frac{\varepsilon}{2} + (1 - \frac{\varepsilon}{2})\mathbb{E}[H_{t-\tau_0^t - S_0^t} \leq 0 | t - \tau_0^t - S_0^t > (2/\varepsilon)^{1/2}] \\
&= \frac{\varepsilon}{2} + (1 - \frac{\varepsilon}{2})\mathbb{E}[H_1, H_2, \dots H_{t-\tau_0^t - S_0^t} \leq 0 | t - \tau_0^t - S_0^t > (2/\varepsilon)^{1/2}] \\
&\leq \frac{\varepsilon}{2} + (1 - \frac{\varepsilon}{2})\mathbb{E}[H_1, H_2, \dots H_{\lfloor (2/\varepsilon)^{1/2} \rfloor} \leq 0] \\
&\leq \frac{\varepsilon}{2} + \mathbb{E}[H_1, H_2, \dots H_{\lfloor (2/\varepsilon)^{1/2} \rfloor} \leq 0] \\
&\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.
\end{aligned}$$

This demonstrates convergence. □

The reader might wonder regarding extensions of the theorem to more general arm distributions and exploration rates. An extension to a non-deterministic arm with a more general distributions is possible and leads to more involved stochastics in the application of the reflection principle.

On the other hand, the proof does not straightforwardly generalize to two arms ordered in second-order stochastic dominance, e.g., normally distributed reward distributions with identical expectation, but different standard deviation. The reason for this is that the proof technique relies on the fact that the *unnormalized* sum of rewards is a sufficient state for the algorithm. For any two arms that are ordered in stochastic dominance, the normalization would need to be taken into account.

The assumptions on exploration are standard conditions guaranteeing that ε-Greedy is a no-regret algorithm, but mainly reduce the complexity of out proof. If $\varepsilon \not\to 0$, our definition of full risk aversion is not satisfied as the algorithm might explore even in the limit $t \to \infty$. If risk aversion is defined as choosing the non-risky arm *in periods where it exploits*, the theorem holds true. The assumption $\sum_{t=0}^{T} \varepsilon_t \to \infty$ simplifies the proof at several points, but is also not necessary for the result to hold.

4 Risk Attitudes of Other Learning Algorithms

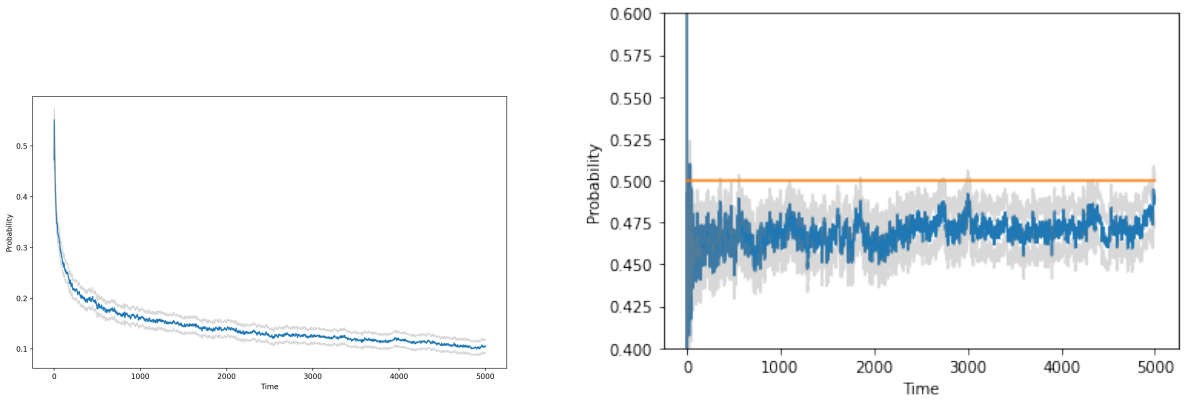
We use simulations to complement our theory by analysing an algorithm that has both theoretical optimality guarantees Lattimore and Szepesvári (2020) and practical advantages, Schrittwieser et al. (2020): Upper Confidence Band. As it does not possess ε-Greedy's theoretical tractability, we resort to simulation to provide insights on its risk attitudes.

4.1 Risky Choices

We first consider ε_t-Greedy (with exploration rate $\varepsilon_t = t^{-\frac{1}{2}}$) and Upper Confidence Band (with optimism term $\delta = (1 + t\ln(t)^2)^{-1})^1$ taking the risky choice between a 0-reward arm and a Rademacher distributed arm. We estimate the mean probability of choosing a particular arm by averaging 5,000 runs of ε-Greedy and UCB for periods $t = 1, 2, \dots, 5,000$. We use a Savitzky-Golay filter of length 9 and order 3 to smoothen the outcomes. The results are shown in Figure 2.

As established by our theoretical result, ε-Greedy behaves fully risk averse, choosing the risky arm with probability converging to zero. We find that UCB has an imperfect risk aversion, plateauing at a choice of about 46.5% for the risky arm.

[1]These choices lead to optimal convergence guarantees, see Auer et al. (2002) for ε-Greedy and Lattimore and Szepesvári (2020) for Upper Confidence Band.



(a) ε-Greedy

(b) Upper Confidence Band

Figure 2: The probability of choosing the risky arm over time with no bias. Gray lines demarcate 95% confidence intervals. In the Upper Confidence Band plot, an orange line shows the 0.5 probability line.

4.2 Unequal Expected Rewards

Second, we show that this risk behaviour exists for a finite number of periods even when the arms do not have the same expected reward. We consider biases favoring the risky arm from $b = 0.1$ to 1. This means that we are comparing an arm with deterministic reward $-b$ to a Rademacher-distributed arm. Figure 3 shows the results for ε-Greedy and Upper Confidence Band.

For small biases, risk aversion can persistent for quite a large number of time periods, for 0.2 bias for almost the entire 500 simulated periods. For bigger biases, the effect is, as to be expected, small. (Note that ε-Greedy shows a larger effect than Upper Confidence Band—for larger biases, Upper Confidence Band has a higher probability of choosing the risky arm.)

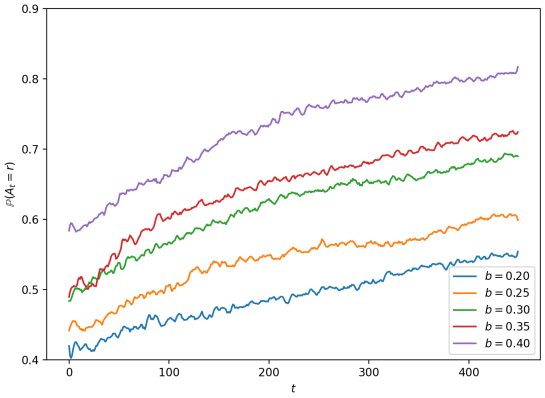
4.3 Certainty Equivalents

Figure 4 give us an idea of the *certainty equivalents* for the algorithms for different times. The heatmaps show the probability of choosing the risky arm for each pair (b, t), and highlight the pairs (b, t) such that $\mathbb{P}[A_t = r]$ is closest to $\frac{1}{2}$. At such (b, t), the algorithm is indifferent between the two arms, so that the bias at that specific time period corresponds to a notion of certainty equivalent for the algorithm for that time period. As can be seen from the heatmap, after fewer rounds the algorithms require a higher certainty equivalent, meaning that they become less risk averse over time.[2] Certainty equivalents are consistently positive, which is what our theory of risk aversion predicts. Note that, as seen in Figure 3, Upper Confidence Band is significantly less risk averse than ε-Greedy, and has lower certainty equivalents.

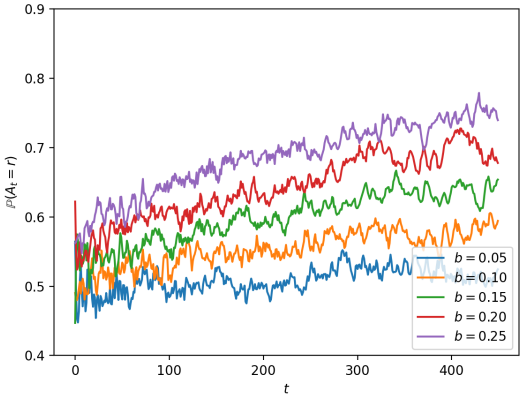
5 Impact of Risk Preferences of Recommendation Systems

A major application of bandit algorithms is in recommendation systems. For example, streaming sites make content recommendations based on user profile and content desirability, or e-commerce platforms personalize search results based on prior user interactions. In this section, we make the connection of recommendation systems with online learning algorithms more formal in subsection 5.1, present an economic simulation environment in subsection 5.2 and discuss results and supply-side implications in subsection 5.3 and subsection 5.4, respectively.

[2]This is in line with predictions of vanishing regret.

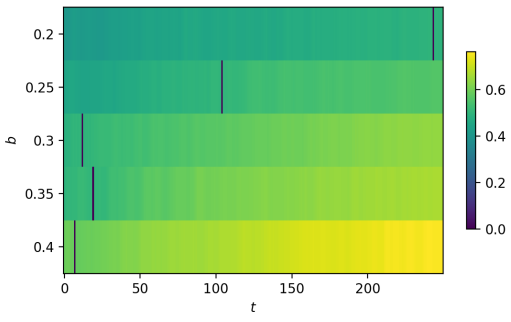


(a) ε-Greedy

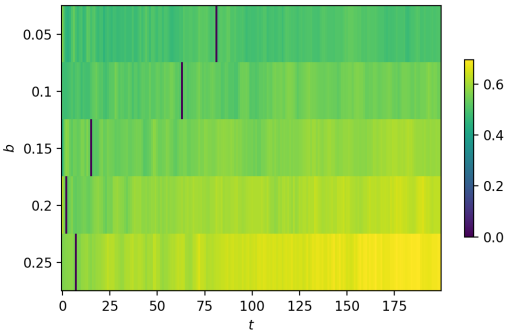


(b) Upper Confidence Band

Figure 3: Probability of choosing the risky arm over time



(a) ε-Greedy



(b) Upper Confidence Band

Figure 4: Certainty Equivalents

5.1 Recommendation Systems as Online Learning Algorithms

Many recommendation systems use a model to predict how much a user will like a particular item of content, and then recommend it.

To make their predictions, the recommendation system tries to learn a model mapping users i and item j to a predicted utility this agent would derive, $u(i,j)$. Assuming some class of functions $f(i,j;\theta)$, a typical formulation for this task is

$$\arg\min_\theta \sum_{(i,j) \text{ observed}} \|u(i,j) - f(i,j;\theta)\|_1 + \lambda\|\theta\|_2, \tag{2}$$

where $\|\bullet\|_1, \|\bullet\|_2$ are norms, and "(i,j) observed" means that the system has observed a reaction from user i to item j. While the first term tries to match observed ratings, the second term tries to favour simple models. One example of the class of functions f is the class of bilinear functions,

$$f(i,j;\theta) = u_i^T v_j, \quad u_i, v_j \in \mathbb{R}^d,$$

where $\theta = ((u_i)_{i\in[n]}, (v_j)_{j\in[k]}$ are characteristic vectors of the content used. Other models replace the function with more complex function classes, e.g., neural networks.

To recommend content items, the system approximately maximizes $f(i,j;\theta)$: It chooses with a probability increasing in $f(i,j;\theta)$ a piece of content j for user i to show next. One example for such a choice is softmax recommendation, which, conditional on θ recommends content i to user j with probability

$$\frac{1}{\sum_{l\in A}\exp(f(i,j;\theta))}\exp(f(i,j;\theta)). \tag{3}$$

The recommendation algorithm alternates between solving (2) and recommending to users according to (3).

Compare this to the bandit setting considered so far. The types of content $k \in A$ correspond to the actions the bandit can take, the function class is represented by

$$f(j;\theta) = Q_j,$$

where $\theta = (Q_j)_{j\in[k]}$ is a single value with vanishing regularization. When choosing $\lambda = 0$ and $\|\bullet\|_1$ as the l^2-norm, choosing Q_i as average historical reward becomes optimal.

Hence, our bandit model can be seen as a special case of content recommendation. While theoretical tractability as in section 3 is lost, we may resort to simulations to see whether risk aversion is significant in realistic recommendation systems.

5.2 Risk Aversion in Recommendation Systems

A set of users $[n]$ interact with a system for rounds $t = 1, 2, \ldots, T$. There are content items $j = 1, 2, \ldots, k$, each belonging to a *genres* $l = 1, 2, \ldots, L$. Genres are sampled uniformly at random, independently across items.

User i has preference for item j from genre l which is distributed according to

$$u_i(j) = \text{clip}_{[1,5]}(z_{il} + r_l\varepsilon_{ij}), \tag{4}$$

where $z_{il} \sim \text{Unif}(1,5)$, $\varepsilon_{ij} \sim N(0,1)$, and $\text{clip}_{[0,5]}(x) = \min(5, \max(0,x))$ maps a value to the interval $[1,5]$. We call $r_l \in \mathbb{R}_+$ the *volatility* of genre l. For example, genre 1 might be TV series and genre 2 movies, with movies giving more volatile recommendation feedback as they give more infrequent signals on user utility.

Each period, each user is *active* with probability 0.2. Each active user is recommended a single item according to a recommendation policy and reacts given their utility (4).

We use a recommendation system that uses a neural network to represent user preferences in (2) of the recommended item, i.e. $f(i,j;\theta)$ is a neural network with weights θ and a softmax recommendation policy

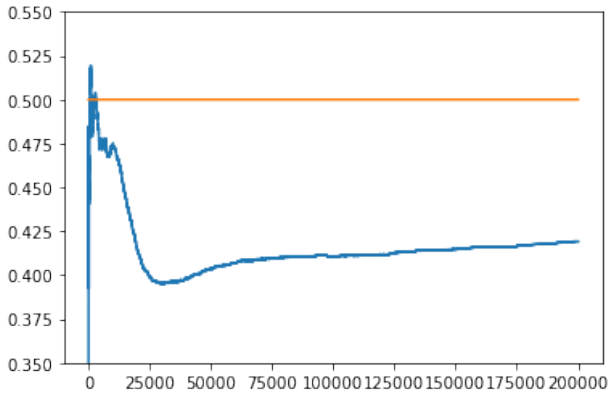


Figure 5: Proportion of risky content recommended in the first 1,000 periods of users interacting with content.

as in (3). More concretely, we use the AutoRec recommendation system Sedhain et al. (2015). Similar neural recommendation systems are used in industry, compare Covington et al. (2016). We consider in our simulation $1,000$ users, $1,700$ items, $L = 16$ genres, $r_1 = r_2 = \cdots = r_8 = 0.4$ are less volatile, $r_9 = r_{10} = \cdots = r_{18} = 0.6$ are more volatile. The system, before recommending, has 100,000 random utilities from user-item pairs, as is standard in recommendation systems, Mladenov et al. (2021); Berkeley-Reclab ([n.d.]). Note that all genres have the same expected utility for users.

We use the simulation environment RecLab for our simulations Berkeley-Reclab ([n.d.]).

5.3 Results

We plot in Figure 5 the proportion of rounds the recommendation system recommends items of genres $L = 10, 11 \ldots, 18$ to the user. The demand is significantly lower than for genres $1, 2, \ldots, 9$, which are less volatile. The relative frequency of recommendation is reduced by more than 10% compared to a half-half split that looking purely at expected rewards would predict.

The algorithm's risk aversion reduces product diversity, even if it is not in the interest of the deployer to do so: They get recommended significantly less some of the genres which have, ex ante, the same distribution of utility for an agent, hence reducing product diversity.

Risk aversion also reduces consumer surplus. If users were free to make their choice of content, the market shares of each type of genre would be 0.5. The deviation from this split means that the algorithm is choosing an outcome misaligned with consumer welfare, as, with high probability close to half of the agents prefer each of the genres.

Our result also raises questions of algorithmic fairness. The algorithm's risk bias is skewing the market share towards content that it has an inherent bias for, i.e. it is creating its own market rather than catering to the existing market. In particular, it favours content that gives more reliable estimates for consumer satisfaction, leading to an artificial preference for surveillance of users, a long-held criticism of online platforms, compare Zuboff (2015).

5.4 Strategic Content Creators

So far we assumed a fixed model for content that is available. The effects of algorithmic risk aversion may be aggravated by strategic incentives of content creators on a recommendations system, as we show in a game-theoretic model of content creation.

There are content producers in K, and content is in M. At time 0, content producers can choose to produce a piece of content $m \in A$. There are consumers $n \gg 0$ with preferences $u(i, j)$, $i = 1, 2, \ldots, n$,

$j \in M$. We assume that that user preferences are drawn i.i.d. from a distribution F, $u(i, \bullet) \sim F$. For simplicity, assume that consumer preferences are along a single dimension q of the content, where q denotes quality.[3] One example of such So if $q(m_1) \geq q(m_2)$, then $u(i, m_1) \geq u(i, m_2)$ for all i. After choosing a piece of content from M, a recommendation system recommends content to the users for many time periods. We assume that the recommendation system has no regret, compare Lattimore and Szepesvári (2020), a property that holds for ε-Greedy or Upper Confidence Band considered in this article. Assume that the recommendation system and content producer are both maximizing user utility and do not discount their payoffs; this is for example the case if user utility is a proxy for user engagement and, in turn, proportional to revenue generated through a user. Then, $\mathbb{E}u(i; m)$ is the expected that the content producers get from content profile m given that user preferences are distributed according to F. Let m^* be the profile of content that get chosen to be produced. Then, we must have the following:

Proposition 1. *For any $k, l \in K$,*

$$\mathbb{E}u(i; m_l^*) = \mathbb{E}u(i; m_k^*).$$

The proof for this is straightforward and analogous to traditional Bertrand competition. If a content producer l produces content with lower quality than another content producer k, then every consumer will prefer k's content to l's. The algorithm, being no regret, will eventually recommend only the better content and show none of the other content. Thus all content producers will pick content of the same quality. Since content is valued by consumers only along the quality dimension, the expected reward to the two content will also be the same.

Thus, in equilibrium, it is reasonable to expect that strategic content producers will choose content that is at least approximately of equal expected reward. However if content producers do not know which algorithm the recommendation system uses, or if they do not know that algorithms have inherent risk preferences, they would choose content with the same expected reward but possibly different variances in reward. Then, the recommendation system would impose its risk behaviour on the market, even if the content producers are risk neutral.

6 Related Work

Our study relates to similar studies on the impact of algorithmic decisionmaking, algorithmic confounding and mechanism design with a risk averse principal.

Simulation Studies of Economics Impacts of Algorithmic Decision-Making A closely related strand of literature studies algorithmic collusion, for example in Calvano et al. (2020); Brown and MacKay (2021); Asker et al. (2021); Hansen et al. (2021a) show that algorithms can learn to charge supra-competitive prices, and even learn punishment strategies that enforce these prices in equilibrium. Hansen et al. (2021a) show that misspecified algorithms can lead to higher prices because they overestimate their own price sensitivity. Our analysis of risk preferences of algorithms is motivated by similar emergent behavioral implications of using algorithms, but is inherently single-agent and related to recommendation systems as opposed to pricing algorithms.

Algorithmic confounding: The literature on algorithmic confounding, for example in Chaney et al. (2018), shows that recommendation systems trained on data from users already exposed to recommendation systems can increase homogeneity and decrease utility for users. In essence, the algorithms fail to take into account that their data reflects both user preferences and what the users were shown by the system. This leads it to homogenize towards popular options, which can be interpreted as the algorithm deciding in favour of "safer" options. We give a definition of risk aversion which is independent of internal reward estimates. In addition, while the literature on algorithmic confounding may reason on the bias of internal estimates, it does typically not focus on the actions an algorithm takes, which is the focus of this article.

[3]This approximation is quite restrictive yet reasonable if, for example, a mixture model for user preferences is assumed, and content creators are competing for one of the mixture distribution. In more general environments, a differentiated products model would need to be used, which is beyond the scope of this article.

Risk Preferences and Mechanism Design: Our study shows that risk aversion for an algorithm arises, connecting our work to mechanism for risk averse principals. See the survey Vasserman and Watt (2021) on implications of risk aversion in auction design. Our paper mostly aims to demonstrate that the emergence of risk aversion in learning algorithms may have economic consequences. Designing incentives with such a bias in mind is a fruitful area for future work.

7 Conclusion and Discussion

We propose a theory of risk aversion for online learning algorithms, defined as a preference for the less volatile action when faced with two actions with the same expected reward. We theoretically show that the ϵ-greedy algorithm is perfectly risk-averse. We demonstrate in simulations that the widely used Upper Confidence Band algorithm is risk averse. In a simulation of a realistic recommendation system environment, we show that content that gives more information about user preferences, even if not more desirable for users, is favoured, which is further aggravated if strategic content provision is assumed.

In this discussion, we comment on conceptual challenges of decision theory for learning algorithms, and then discuss other bandit algorithms.

7.1 Decision Theory for Algorithms

A natural extension of our work is to consider algorithmic behaviour in the form of algorithmic preferences instead of purely risk attitudes. Unfortunately, attempts to formulate a decision theory for algorithms will require significant departure from usual axiomatic preference theory, as the following examples show.

A natural notion of preference for an algorithm uses the probability of choosing a certain action.

Definition 2. *We say that an algorithm π (strictly)* prefers *action i over action j at time t, i.e. $j \succ_{\pi,t} i$, if*

$$\mathbb{P}[a_t = j] > \mathbb{P}[a_t = i].$$

Unfortunately, this preference is intransitive.

Proposition 2. *$\succ_{\varepsilon\text{-Greedy},t}$ is intransitive for $t = 3$ and small enough ε.*

A computer-aided proof shows that the lotteries

$$\begin{aligned} l_1 &= \mathbb{1}_{\{-0.01\}} \\ l_2 &= 0.51\mathbb{1}_{\{1\}} + 0.49\mathbb{1}_{\{-1\}} \\ l_1 &= 0.34\mathbb{1}_{\{1\}} + 0.33\mathbb{1}_{\{-0.02\}} + 0.33\mathbb{1}_{\{-1\}} \end{aligned}$$

lead to a preference cycle $l_1 \succ_{\varepsilon\text{-Greedy},t} l_2 \succ_{\varepsilon\text{-Greedy},t} l_3 \succ_{\varepsilon\text{-Greedy},t} l_1$.

Another deviation is that algorithms might violate conditions that are typically used to construct preferences under incomplete information. The Independence Axiom in expected utility theory, for example, requires that mixing another lottery to two lotteries shouldn't change their relative ordering. However mixing a complicated lottery to the comparison between a simple lottery and a fixed reward could make it harder for the algorithm to distinguish quickly between the two, hence making it possible to reverse the ordering between them.

As we show in this article, studying purely risk preferences allows for both theoretical and empirical insight, even if it does not generalize to a full preference theory.

7.2 Other Bandit Algorithms

We briefly comment on other bandit algorithms.

The Gittins index policy is the foremost Bayesian algorithm, known to be optimal in a wide class of problems. In a Bayesian formulation of our problem, it would in fact address many of the issues we raise in

this paper. With a correctly specified prior, i.e. with a prior that has support only among distributions with the same expected value on both arms, it would indeed be truly indifferent between the two arms, hence being risk neutral in the weak sense that we define. Further, even a small bias in favor of the risky arm, with the prior again being correctly specified to have support only on distributions with that bias as the expected value, the policy would more often than not choose the risky arm, hence also being risk neutral in the strong sense. However it is crucial to assume that the prior is exactly specified, since even a vanishing error in prior specification can make the policy behave arbitrarily. The reason many online learning algorithms are deployed is because specifying such priors is hard.

Thompson sampling is typically used for bandits with Bernoulli-distributed rewards, with a Beta prior over the Bernoulli parameters Russo et al. (2018). While this leads to a tractable formulation of the algorithm, and is widely used, the assumption that arms are Bernoulli distributed makes arms of equal expectation, yet different variance of arm rewards *impossible*, making the question of algorithmic risk aversion void. While Thompson sampling is well-defined for other classes of priors, there is little consensus in the literature on which are the right probabilistic models.

An area for future work is the inclusion of algorithms for the adversarial bandit problem like EXP3 and EXP3-IV (compare Lattimore and Szepesvári (2020)) that have found less practical use.

7.3 Avenues for Future Research

A first avenue for future work concerns the implications of risk aversion for online learning algorithms used in repeated-games environments. One example for this could be in the domain of algorithmic collusion, where firms use different price-setting algorithms. As algorithmic risk aversion seems not directly related to other observations made in this domain, e.g. correlation of algorithmic play Hansen et al. (2021b); Banchio and Mantegazza (2022), we see the application of risk aversion to such games as fruitful. Another direction is the design of algorithms with arbitrary risk properties. In particular, among the classical algorithms we used, we did not identify risk affine algorithms. Finally, our paper includes simulated results to test empirical validity of the theory. Whether the effects of risk aversion are economically relevant is ultimately an empirical question to be answered using data from deployed online learning algorithms in recommendation systems and elsewhere.

References

John Asker, Chaim Fershtman, and Ariel Pakes. 2021. Artificial Intelligence and Pricing: The Impact of Algorithm Design. *National Bureau of Economic Research Working Paper Series* No. 28535 (2021). `http://www.nber.org/papers/w28535%0Ahttp://www.nber.org/papers/w28535.pdf`

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2 (2002), 235–256.

Martino Banchio and Giacomo Mantegazza. 2022. Games of Artificial Intelligence: A Continuous-Time Approach. *arXiv preprint arXiv:2202.05946* (2022).

Berkeley-Reclab. [n.d.]. Berkeley-reclab/RecLab. `https://github.com/berkeley-reclab/RecLab`

Zach Y Brown and Alexander MacKay. 2021. *Competition in Pricing Algorithms*. Working Paper 28860. National Bureau of Economic Research. `https://doi.org/10.3386/w28860`

Emilio Calvano, Giacomo Calzolari, Vincenzo Denicolò, and Sergio Pastorello. 2020. Artificial Intelligence, Algorithmic Pricing, and Collusion. *American Economic Review* 110, 10 (October 2020), 3267–97. `https://doi.org/10.1257/aer.20190623`

Allison J. B. Chaney, Brandon M. Stewart, and Barbara E. Engelhardt. 2018. How Algorithmic Confounding in Recommendation Systems Increases Homogeneity and Decreases Utility. In *Proceedings of the 12th ACM*

Conference on Recommender Systems (Vancouver, British Columbia, Canada) *(RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 224–232. https://doi.org/10.1145/3240323.3240370

Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

Karsten T. Hansen, Kanishka Misra, and Mallesh M. Pai. 2021a. Frontiers: Algorithmic Collusion: Supra-competitive Prices via Independent Algorithms. *Marketing Science* 40, 1 (2021), 1–12. https://doi.org/10.1287/mksc.2020.1276 arXiv:https://doi.org/10.1287/mksc.2020.1276

Karsten T Hansen, Kanishka Misra, and Mallesh M Pai. 2021b. Frontiers: Algorithmic collusion: Supra-competitive prices via independent algorithms. *Marketing Science* 40, 1 (2021), 1–12.

Tor Lattimore and Csaba Szepesvári. 2020. Bandit Algorithms. *Bandit Algorithms* (2020). https://doi.org/10.1017/9781108571401

Martin Mladenov, Chih-Wei Hsu, Vihan Jain, Eugene Ie, Christopher Colby, Nicolas Mayoraz, Hubert Pham, Dustin Tran, Ivan Vendrov, and Craig Boutilier. 2021. RecSim NG: Toward Principled Uncertainty Modeling for Recommender Ecosystems. *arXiv preprint arXiv:2103.08057* (2021).

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. 2018. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning* 11, 1 (2018), 1–96.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature* 588, 7839 (2020), 604–609.

Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on World Wide Web*. 111–112.

Shoshana Vasserman and Mitchell Watt. 2021. Risk aversion and auction design: Theoretical and empirical evidence. *International Journal of Industrial Organization* 79 (2021), 102758.

Eric W Weisstein. 2002. Random Walk–1-Dimensional. *https://mathworld. wolfram. com/* (2002).

Shoshana Zuboff. 2015. Big other: surveillance capitalism and the prospects of an information civilization. *Journal of information technology* 30, 1 (2015), 75–89.

A Definition of Risk Aversion for Reinforcement Learning

In many environments with autonomous agents, the environment carries a state which might either be randomly drawn (the contextual bandit problem) or develop in response to agent actions (Markov Decision Processes), a strictly more general model.

Recall that a Markov Decision Process (MDP) is a 6-tuple $(S, A, T, R, \gamma, s_0)$.

- S is a finite set of states
- A is a finite set of actions
- $T\colon S \times A \to \Delta(S)$ is a transition function
- $R\colon S \times A \to \mathbb{R}$ is a bounded function
- $\gamma \in [0, 1)$ is a discount factor.
- an initial state s_1.

For a function

$$\pi\colon (S \times A \times \mathbb{R}_+)^* \times S \to \Delta(A),$$

we can define action and state histories

$$\begin{aligned} a_t &\sim \pi(s_{:t-1}, a_{:t-1}, s_t) \\ s_{t+1} &\sim T(s_t, a_t), \quad t = 1, 2, \dots \end{aligned}$$

Denote

$$V^\pi(s_0) = \mathbb{E}[\sum_{t=1}^{\infty} \gamma R(a_t, s_t)]$$

the value of state s_0, and define $V^\pi(s)$ for $s \in S \setminus \{s_0\}$ accordingly. Also denote by

$$\mathcal{V}^\pi(s_0) = \sum_{t=1}^{\infty} \gamma R(a_t, s_t)$$

the random variable associated to reward. We denote $V(s')$ the maximizer of $V^\pi(s')$ over all policies π. Denote

$$Q(s, a) = R(s, a) + \mathbb{E}_{s' \sim T(s,a)}[V(s')]$$

the optimal Q-function. We call a policy π *risk-averse* if for any state s that is visited infinitely often a.s.

$$\mathbb{E}[\sum_{t=1}^{\infty} \mathbb{1}_{s_t = s}] = \infty$$

and any two actions a, a' such that

$$Q(s, a) = Q(s, a')$$

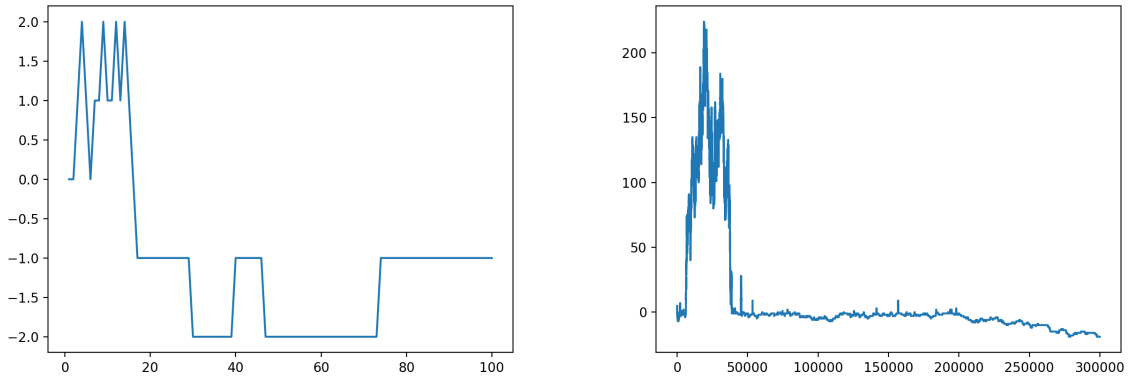
and for $\tilde{s} \sim T(s, a)$ and $\tilde{s}' \sim T(s, a')$

$$\mathcal{V}^\pi(\tilde{s}) \prec \mathcal{V}^\pi(\tilde{s}'),$$

the following holds: For the subsequence $(n_t)_{t \in \mathbb{N}}$ enumerating t such that $s_t = s$, we have that

$$\limsup_{t \to \infty} \mathbb{P}[a_t = a] < \liminf_{t \to \infty} \mathbb{P}[a_t = a']$$

This means that if the volatility of the optimal trajectory is more noisy when a particular action is taken is noisier, it will be taken less often. This generalizes to the asymptotic notion for the bandit algorithms.



(a) Short Term.

(b) Longer Term.

Figure 6: The unnormalized Q-values of a run of ε-Greedy with a 0 and a Rademacher distributed arm. In the short run, we see that the variance for positive advantage is higher than for negative advantage. In the longer run, we see that this leads to the decision-maker taking less risky actions.

In the case of contextual bandits, where $s \sim F$ is drawn i.i.d., we get a particular specialization: An algorithm is risk averse if it is risk averse for every state (also called *context*) in the support of F.

One popular online algorithm for the Markov Decision Problem in large games is Upper Confidence Trees (UCT) Schrittwieser et al. (2020). For UCT, the intuition of risk aversion outlined in section 3 holds: A part of the tree with a low-reward draw will be undersampled (in the language of upper confidence trees, subtrees might be are "purged"), suggesting to risk averse behavior.

B Additional Figures

In Figure 6, we show the unnormalized advantage of the risky arm analogously to Figure 1.