Thompson Sampling Tutorial
“The rediscovery of a swiss army knife”

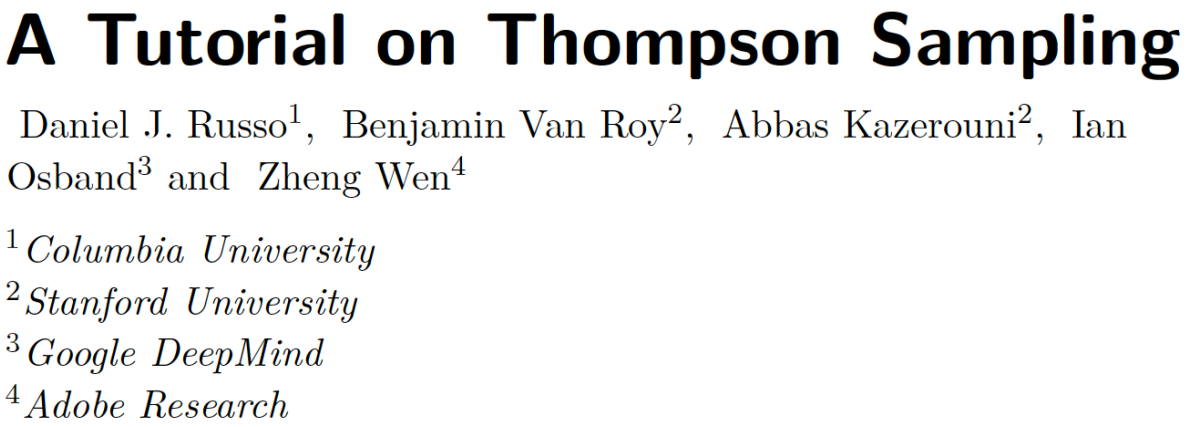
DATA science SUMMIT

Hanan Shteingart, PhD
Playtika

Alternative Name for the Lecture ☺

"Squeezing a 96-page review into a 25 min talk"

Based on Russo, D. J., Van Roy, B., Kazerouni, A., Osband, I., & Wen, Z. (2018). **A tutorial on Thompson sampling**. *Foundations and Trends® in Machine Learning*, *11*(1), 1-96.



A Tutorial on Thompson Sampling

Daniel J. Russo[1], Benjamin Van Roy[2], Abbas Kazerouni[2], Ian Osband[3] and Zheng Wen[4]

[1] *Columbia University*
[2] *Stanford University*
[3] *Google DeepMind*
[4] *Adobe Research*

https://web.stanford.edu/~bvr/pubs/TS_Tutorial.pdf
https://github.com/iosband/ts_tutorial

The Rediscovery of Thompson Sampling

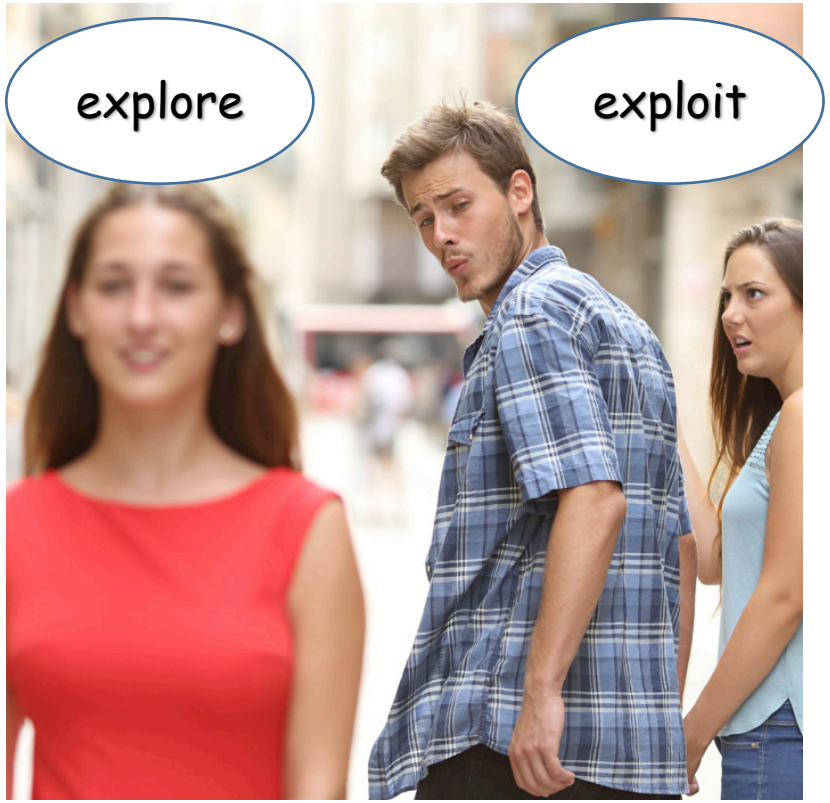
TS was first proposed in 1933 for allocating experimental effort arising in clinical trials

The algorithm was largely ignored in the academic literature until recently (although it was independently rediscovered several times in the interim)

After 8 decades… surge of interest. Especially because of (Chapelle and Li, 2011; Scott, 2010) which displayed the algorithm's strong empirical performance

Why Thompson Sampling?

- Algorithm for **online decision** problems
- **Actions** are taken **sequentially** in a manner that must **balance between <u>exploiting</u> and <u>exploring</u>**
- Addresses a **broad range** of problems in a computationally efficient manner and is therefore enjoying **wide** use
  - Bernoulli bandit problems, shortest path problems, product recommendation, assortment, active learning with neural networks, and reinforcement learning in Markov decision processes



Sorry if this image offends someone, it's just a joke…

Image from http://nymag.com/intelligencer/2017/08/antonio-guillem-the-photographer-behind-guy-looking-back-meme.html

Applications & Companies

That's why I was interested in it…

Revenue management (Ferreira et al., 2015)	Marketing (Schwartz et al., 2017)	web site optimization (Hill et al., 2017)	Monte Carlo tree search (Bai et al., 2013)	A/B testing (Graepel et al., 2010)
Internet advertising (Graepel et al., 2010; Agarwal, 2013; Agarwal et al., 2014)	Recommendation systems (Kawale et al., 2015)	hyperparameter tuning (Kandasamy et al., 2018)	arcade games (Osband et al., 2016a)	Adobe, Amazon (Hill et al., 2017)
Facebook, Google (Scott, 2010;Scott, 2015)	LinkedIn (Agarwal, 2013; Agarwal et al., 2014)	Microsoft (Graepel et al., 2010)	Netflix	Twitter

Why Do We Need Marketing in Playtika?

- The leaky bucket model for B2C companies

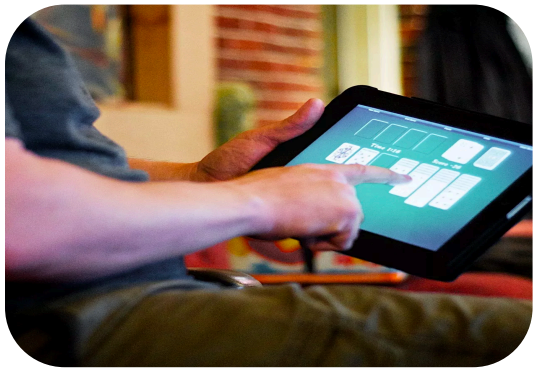
User Acquisition

Retention & Monetization

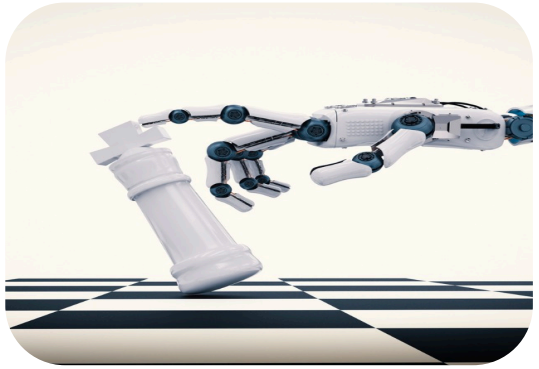
Churn Prevention and Retargeting

https://www.amazon.com/Your-Business-Leaky-Bucket-Millions-ebook

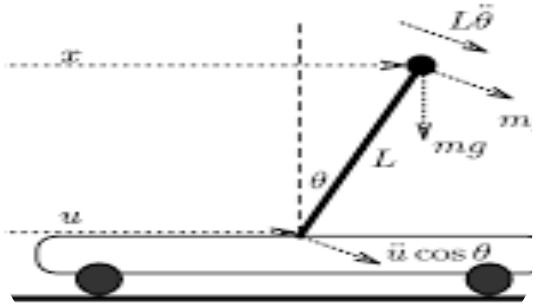
Fields of DS in Playtika (for example)



Player prediction

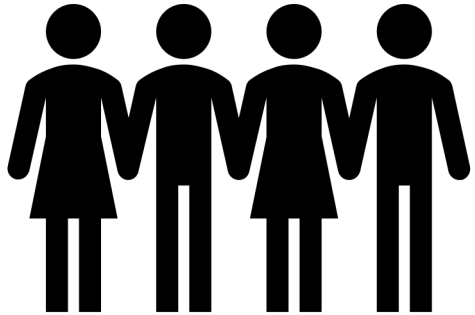


Artificial Player



Optimization & Control, e.g. User Aqcquisiton

Recommendation & Personalization



Several Teams
x10 of data scientists
x10 of data engineers

6 TB/day
30M unique users /month

Multi Arm Bandit (MAB) Problem

- Imagine a gambler (octopus?!?!) going into a casino with different slot machine.
- Tradeoff: Pull the lever with the best historical payoffs or explore other levers to gain knowledge?
- A naïve approach (sample then decide) is wasteful.

Image from https://towardsdatascience.com/reinforcement-learning-demystified-exploration-vs-exploitation-in-multi-armed-bandit-setting-be950d2ee9f6

The Bernoulli Case

Model

- *K* actions, reward is success or a failure.
- The success probabilities *P(1, .., K)* are unknown to the agent, but are fixed over time.
- The objective, is to maximize the cumulative number of successes over *T* periods, where *T* is relatively large compared to the number of arms *K*.

Real World Application: Publisher side

- The “arms” in this problem might represent different banner ads
- A success is associated either with a click on the ad, or with a conversion
- The parameters are click through or conversion rate

Small-scale experiments are now a core tool at most leading Internet companies.

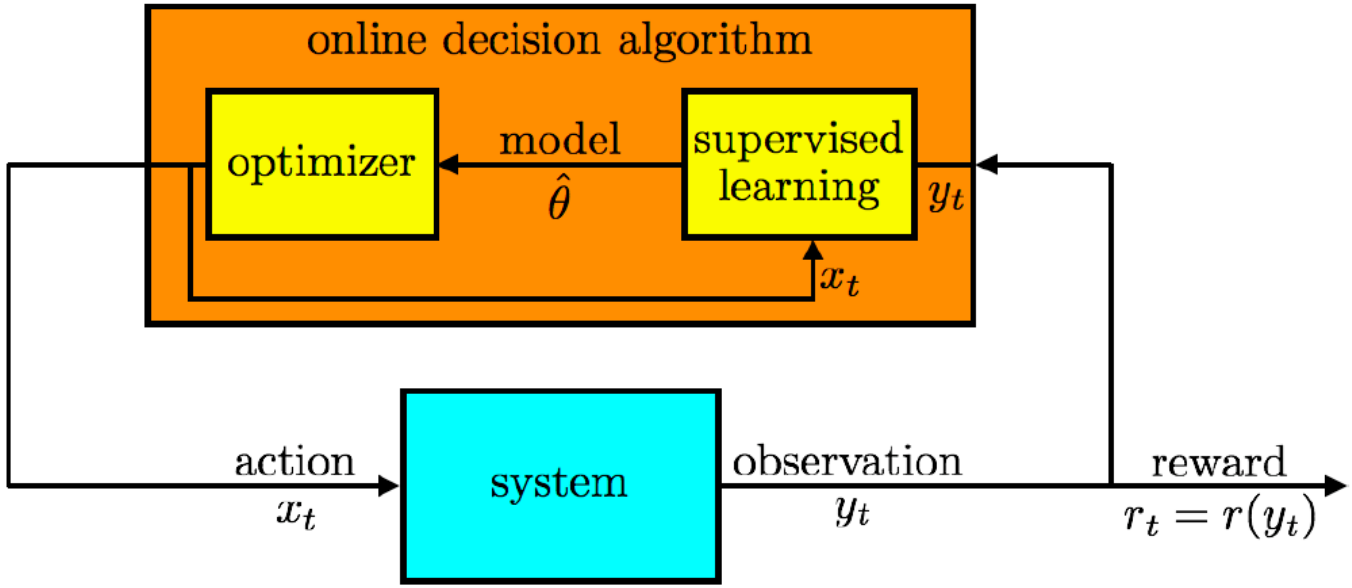
http://ad-tech.com/

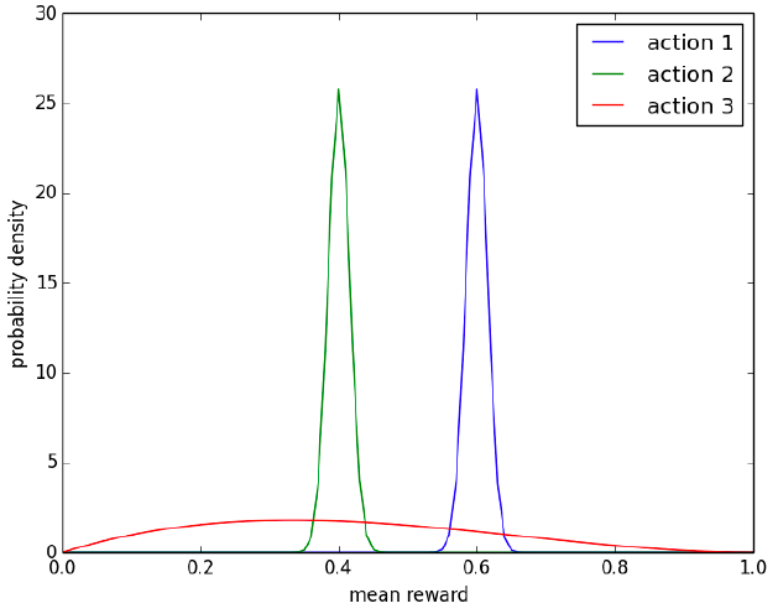
How Is this different from Supervised Learning?

- Supervised machine learning learn **passively** from **historical** data
- Yet in MAB agents **drive the generation of their own training data** (e.g. through interacting with users).
- As a result, there is enormous potential benefit in the design of algorithms that **not only learn from past data**, but also **explore systemically** to generate **useful data** that **improves future performance**

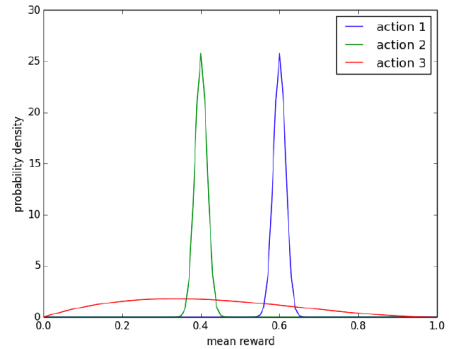
Why does Greedy Approach Fail?

- "A shortcoming of the greedy approach, which can severely curtail performance, is that it does not actively explore"





Why does Dithering Fail?



- randomly perturbing actions that would be selected by a greedy algorithm.
- ε-greedy exploration - greedy action with probability $1 - \varepsilon$ and otherwise selects an action uniformly at random
- wastes resources by failing to “write off” actions regardless of how unlikely they are to be optimal
- TS provides an alternative to dithering that more intelligently allocates exploration effort…

Thompson Sampling Main Idea

- Probability Matching: ”Optimism in face of uncertainty”
- Probability to select an action is equal to the probability this action is optimal
- By sampling actions according to the posterior probability that they are optimal, the algorithm continues to sample all actions that could plausibly be optimal, while shifting sampling away from those that are unlikely to be optimal.
- **Roughly speaking, the algorithm tries all promising actions while gradually discarding those that are believed to underperform.**

$$\Pr(a|x) = \Pr(argmax_{a'} r(x, a') = a)$$

Example: Revisit the Bernoulli MAB

- K actions ~ Bernoulli with fixed unknown probability $\theta = (\theta_1, ..., \theta_K)$
- Assume prior is beta $$p(\theta_k) = \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k)\Gamma(\beta_k)} \theta_k^{\alpha_k - 1}(1 - \theta_k)^{\beta_k - 1}$$
- posterior distribution is also beta (conjugate prior) with parameters that can be updated according to a simple rule

$$(\alpha_k, \beta_k) \leftarrow \begin{cases} (\alpha_k, \beta_k) & \text{if } x_t \neq k \\ (\alpha_k, \beta_k) + (r_t, 1 - r_t) & \text{if } x_t = k. \end{cases} \qquad \hat{\theta}_k = \alpha_k / (\alpha_k + \beta_k)$$

Compare with Bernoulli Greedy

Algorithm 1 BernGreedy(K, α, β)

```
1:  for t = 1, 2, ... do
2:      #estimate model:
3:      for k = 1, ..., K do
4:          θ̂_k ← α_k / (α_k + β_k)
5:      end for
6:
7:      #select and apply action:
8:      x_t ← argmax_k θ̂_k
9:      Apply x_t and observe r_t
10:
11:     #update distribution:
12:     (α_{x_t}, β_{x_t}) ← (α_{x_t} + r_t, β_{x_t} + 1 − r_t)
13: end for
```

expectation

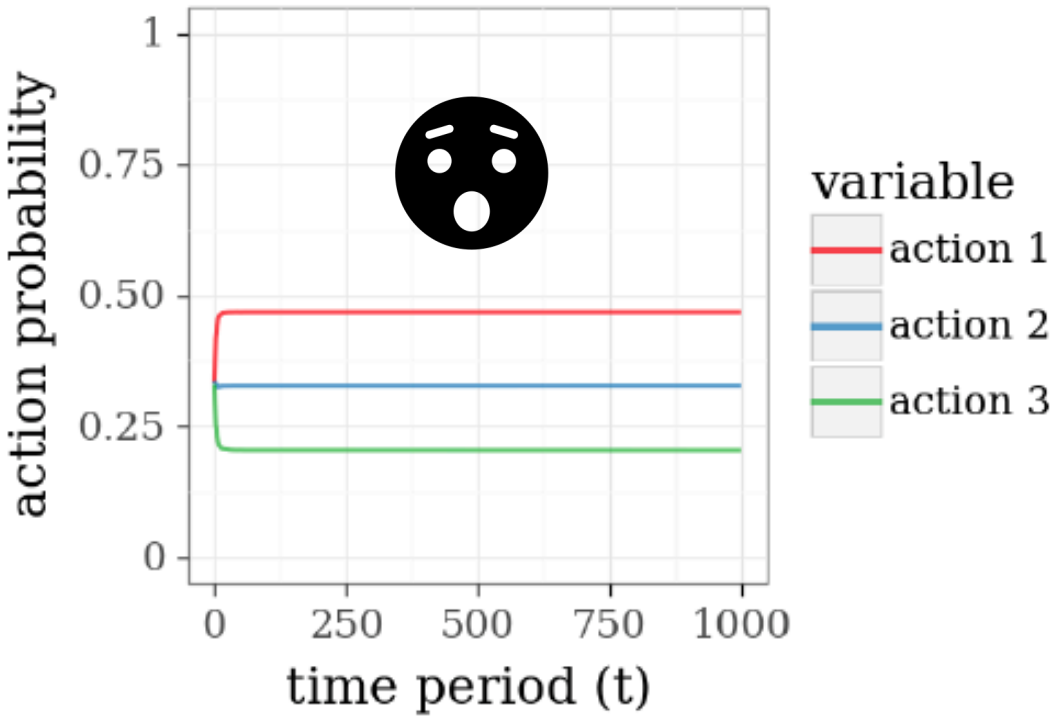
Algorithm 2 BernTS(K, α, β)

```
1:  for t = 1, 2, ... do
2:      #sample model:
3:      for k = 1, ..., K do
4:          Sample θ̂_k ~ beta(α_k, β_k)
5:      end for
6:
7:      #select and apply action:
8:      x_t ← argmax_k θ̂_k
9:      Apply x_t and observe r_t
10:
11:     #update distribution:
12:     (α_{x_t}, β_{x_t}) ← (α_{x_t} + r_t, β_{x_t} + 1 − r_t)
13: end for
```

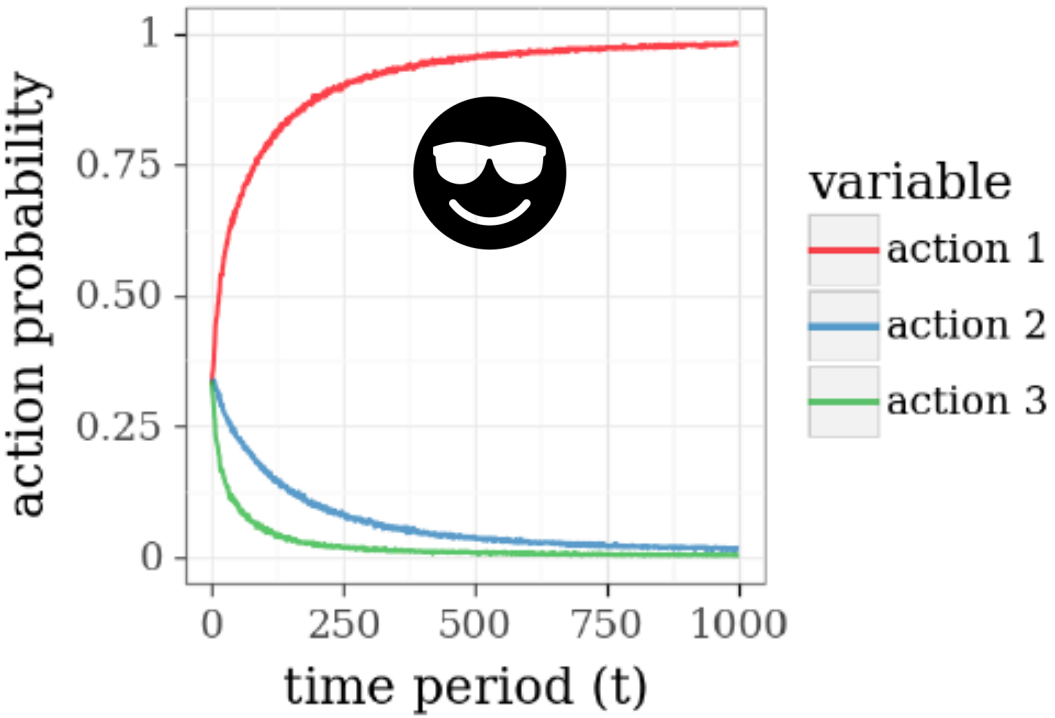
P(best)

Results of Simulation

$\theta_1 = 0.9,\ \theta_2 = 0.8,\ \text{and}\ \theta_3 = 0.7$



(a) greedy algorithm



(b) Thompson sampling

Results based on 10,000 independent simulations of each algorithm each over 1,000 time periods

Beyond Bernoulli - Generalized TS

Algorithm 4 Thompson$(\mathcal{X}, p, q, r)$

for $t = 1, 2, \ldots$ **do**
    #sample model:
    Sample $\hat{\theta} \sim p$

    #select and apply action:
    $x_t \leftarrow \text{argmax}_{x \in \mathcal{X}} \mathbb{E}_{q_{\hat{\theta}}}[r(y_t)|x_t = x]$
    Apply x_t and observe y_t

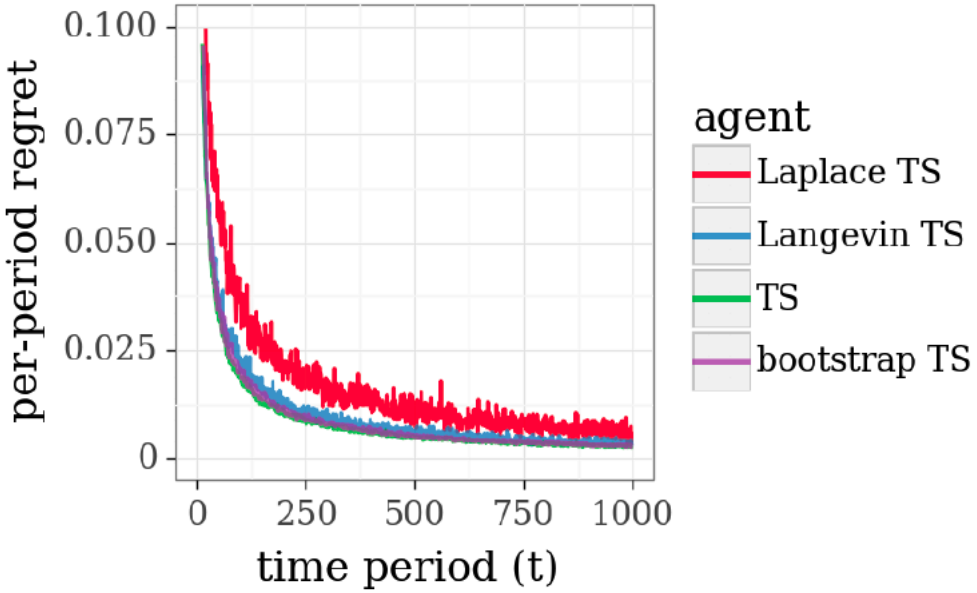
    #update distribution:
    $p \leftarrow \mathbb{P}_{p,q}(\theta \in \cdot | x_t, y_t)$
end for

Approximation

- Many practical contexts call for more complex models for which exact Bayesian inference is computationally intractable.
- Fortunately, there are reasonably **efficient** and **accurate** methods that can be used to **approximately sample from posterior distributions**:
  - Gibbs sampling
  - Langevin Monte Carlo
  - sampling from a Laplace approximation
  - **Bootstrap**

Bootstrap TS

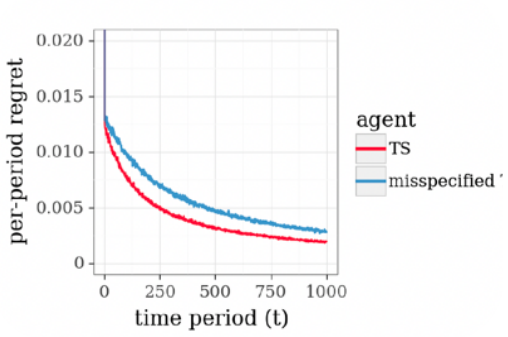
- The method generates a **hypothetical history** which is **made up** of t − 1 action-observation pairs, each sampled uniformly with replacement from $\boldsymbol{H_{t-1}} \rightarrow \widehat{\boldsymbol{H}}_{\boldsymbol{t-1}} = \{(\widehat{\boldsymbol{x}}_{\boldsymbol{1}}, \widehat{\boldsymbol{y}}_{\boldsymbol{1}}), \ldots, (\widehat{\boldsymbol{x}}_{\boldsymbol{t-1}}, \widehat{\boldsymbol{y}}_{\boldsymbol{t-1}})\}$
- We then maximize the likelihood of estimator under the hypothetical history
- there is a lack of theoretical justification fc bootstrap approaches or even understanding of whether there are nontrivial problem classes for which they are guaranteed to perform well. Yet see [1]



(a) Bernoulli bandit

[1] Eckles, D., & Kaptein, M. (2014). Thompson sampling with the online bootstrap. arXiv preprint arXiv:1410.4009.

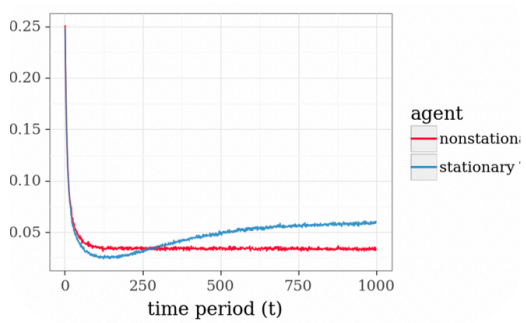
Practical Considerations



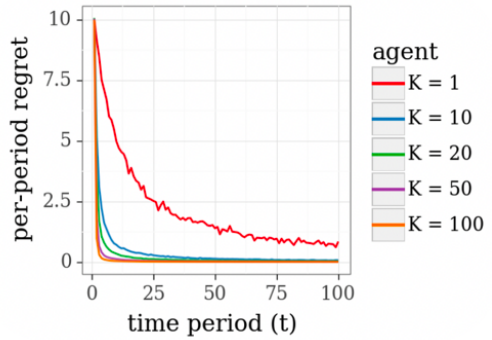
Priors – coherent vs. mis-specified prior

$$p_\theta(\cdot|x_t, z_t)$$

Context – additional information



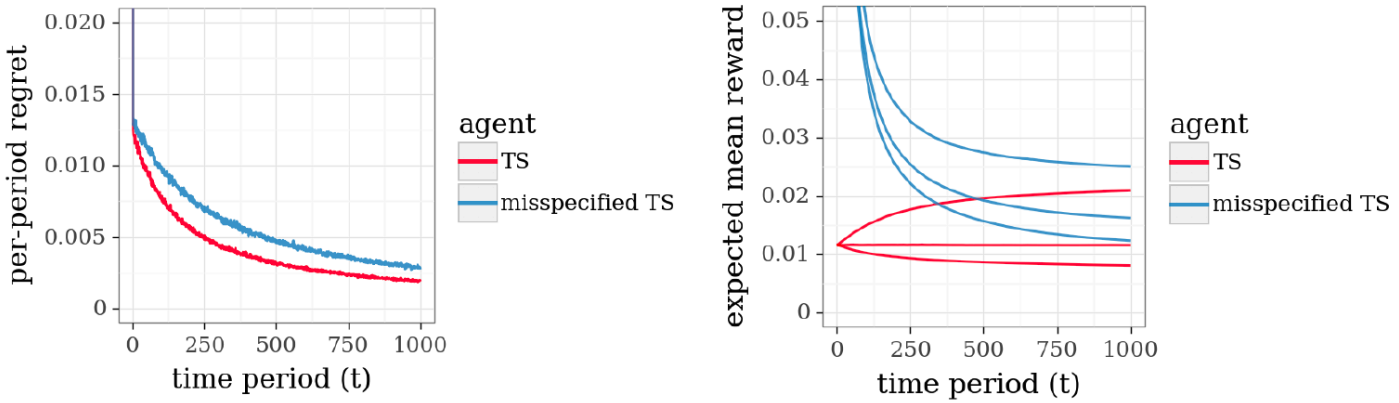
Non stationarity – environment changes over time



Concurrency – ability to select multiple actions per step

Practical Considerations (1. using Priors)

- In particular, this prior represents no understanding of the context, ignoring any useful knowledge from past experience. Taking knowledge into account reduces what must be learned and therefore reduces the time it takes for or TS to identify the most effective ads.



(a) regret **(b)** expected mean rewards

Figure 6.2: Comparison of TS for the Bernoulli bandit problem with coherent versus misspecified priors.

2. contextual online decision problems

- In such problems, the response yt to action xt
- also depends on an independent random variable zt that the agent
- observes prior to making her decision. In such a setting, the conditional
- distribution of yt takes the form $p(\cdot|x_t, z_t)$.

3. Non-Stationarity

- In many practical applications,
- the agent faces a nonstationary system, which is more appropriately
- modeled by time-varying parameters 1, 2, . . ., such that the response
- yt to action xt is generated according to pt(·|xt).
- With minor modification, TS remains an effective
- approach so long as model parameters change little over durations that
- are sufficient to identify effective actions

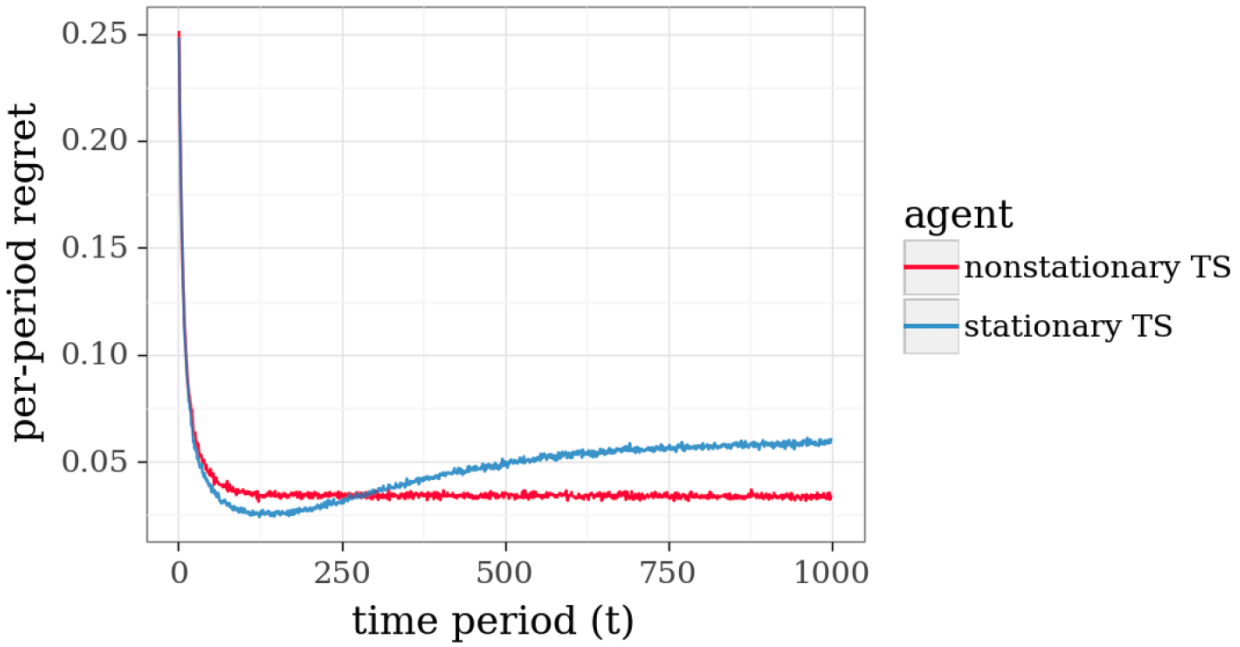
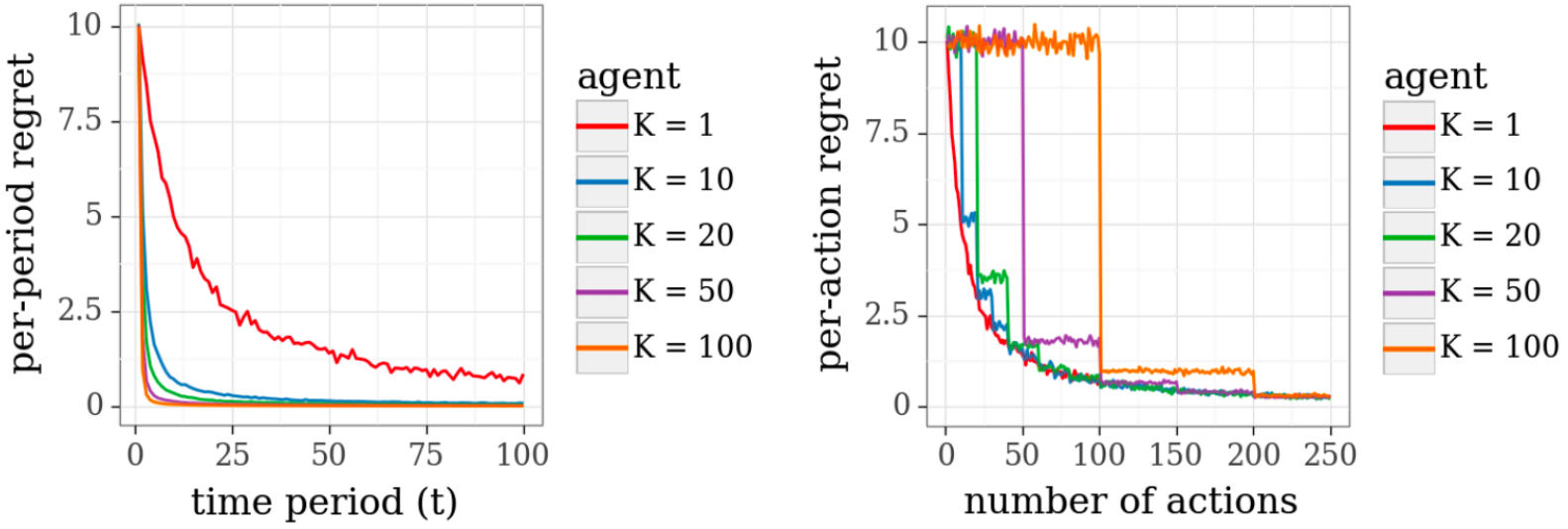


Figure 6.3: Comparison of TS versus nonstationary TS with a nonstationary Bernoulli bandit problem.

4. Concurrent

- concurrence plays an important role in web services, where at any time, a system may experiment by providing 6.4. Concurrence 47 different versions of a service to different users

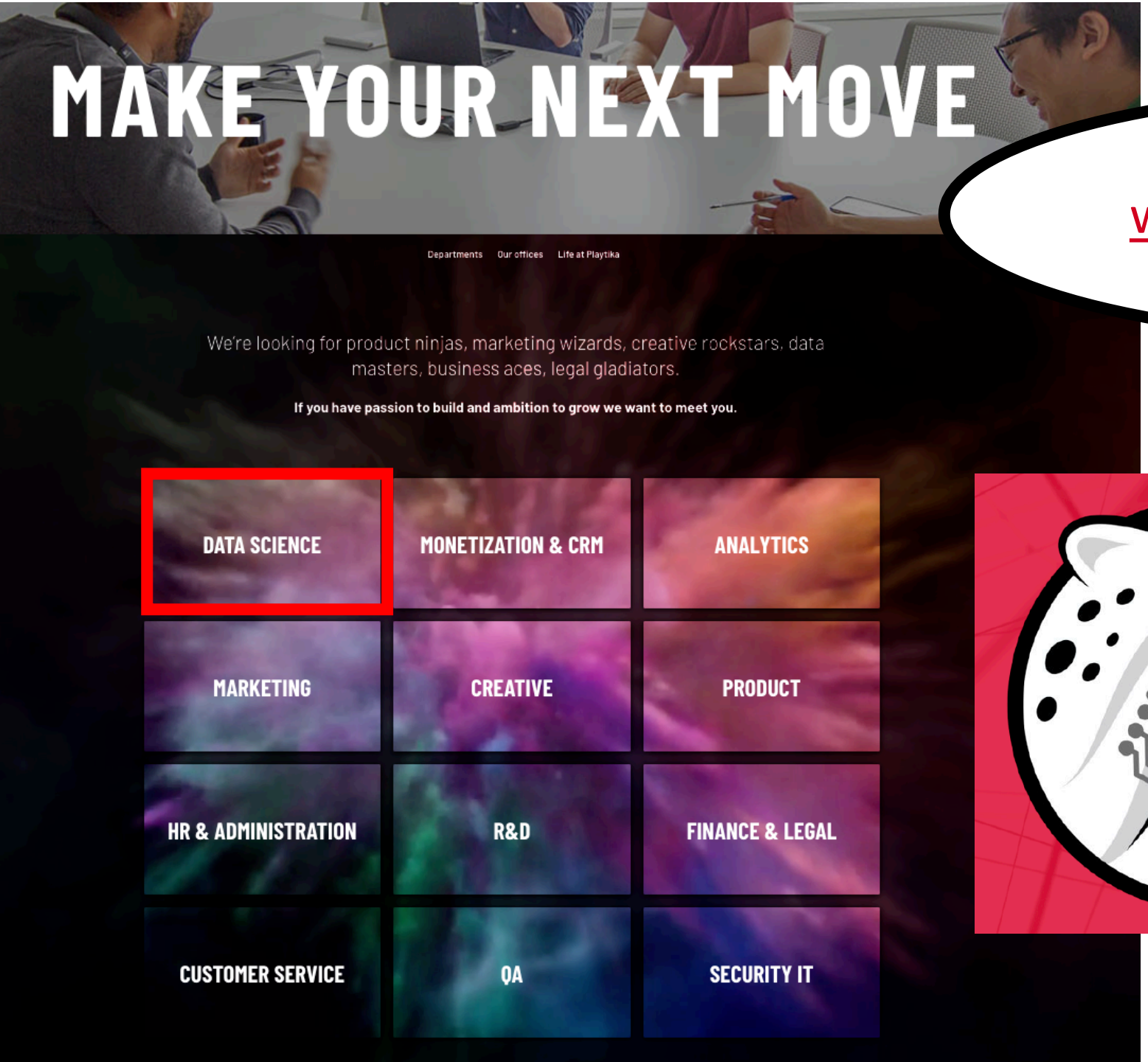


(a) per-action regret over time **(b)** per-action regret over actions

Figure 6.4: Performance of concurrent Thompson sampling.

When does it fails?

- Problems that do not Require Exploration
- Problems that do not Require Exploitation
- Time Sensitivity
- Problems Requiring Careful Assessment of Information Gain



www.playtika.com/careers