Garbage In, Reward Out: Bootstrapping Exploration in Multi-Armed Bandits

Branislav Kveton, Google Research
Csaba Szepesvári, DeepMind and University of Alberta
Sharan Vaswani, Mila, University of Montreal
Zheng Wen, Adobe Research
Mohammad Ghavamzadeh, Facebook AI Research
Tor Lattimore, DeepMind

Stochastic Multi-Armed Bandit

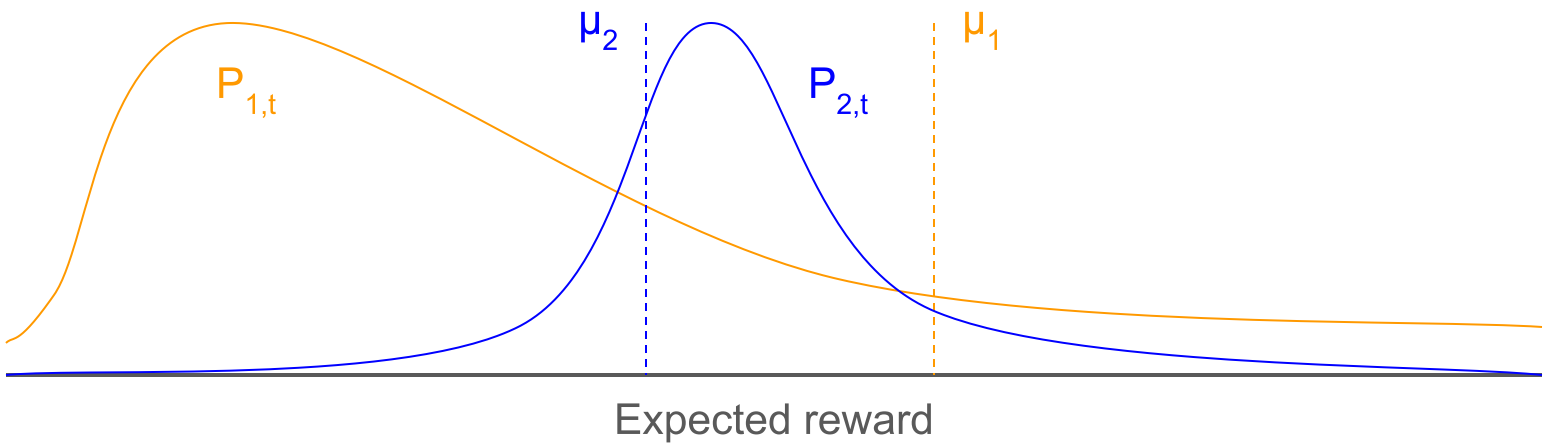
- Learning agent sequentially pulls K arms in n rounds

Arm 1 Arm 2 Arm K

- The agent pulls arm I_t in round $t \in [n]$ and observes its reward
- Reward of arm i is in [0, 1] and drawn i.i.d. from a distribution with mean μ_i
- Goal: Maximize the expected n-round reward
- Challenge: Exploration-exploitation trade-off

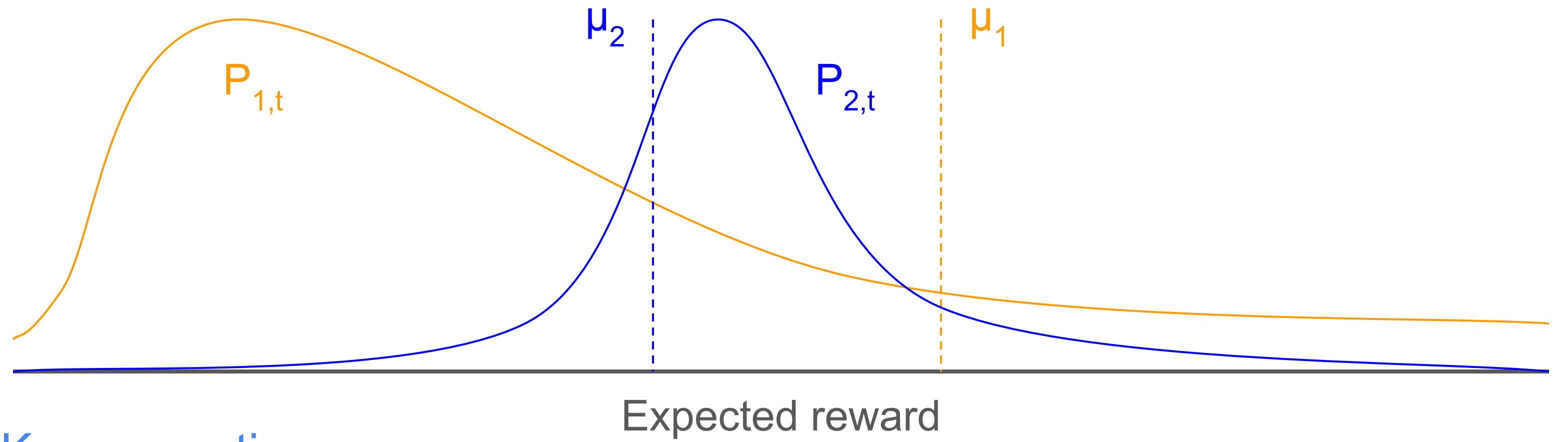
Thompson Sampling (Thompson, 1933)

- Sample $\mu_{i,t}$ from **posterior distribution $P_{i,t}$** and pull arm $I_t = \text{argmax}_i\ \mu_{i,t}$



Thompson Sampling (Thompson, 1933)

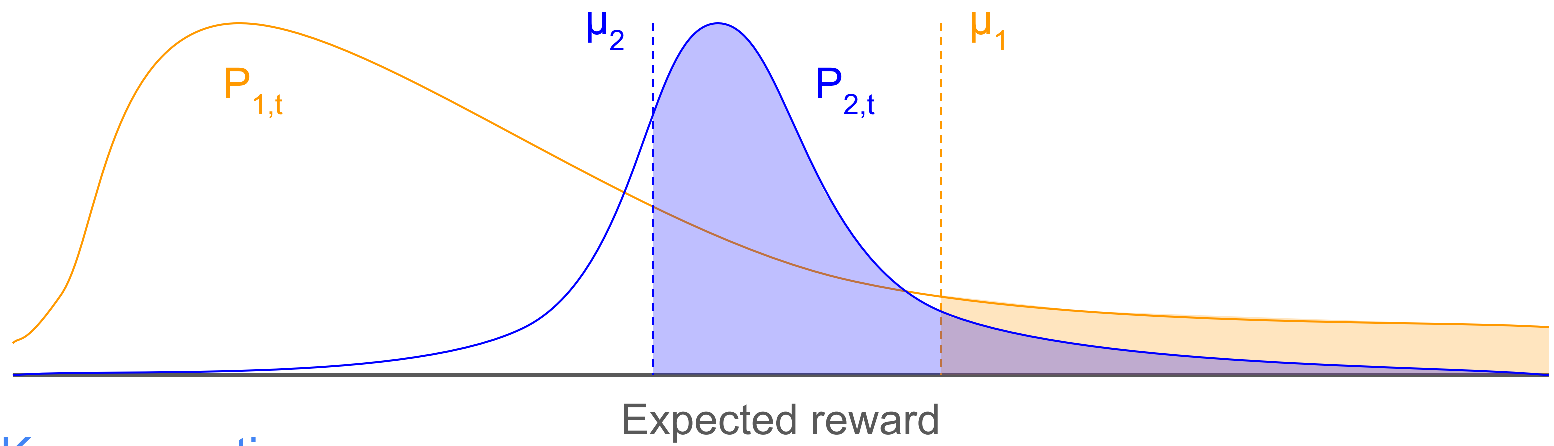
- Sample $\mu_{i,t}$ from posterior distribution $P_{i,t}$ and pull arm $I_t = \text{argmax}_i\ \mu_{i,t}$



- Key properties
  - $P_{i,t}$ concentrates at μ_i with the number of pulls

Thompson Sampling (Thompson, 1933)

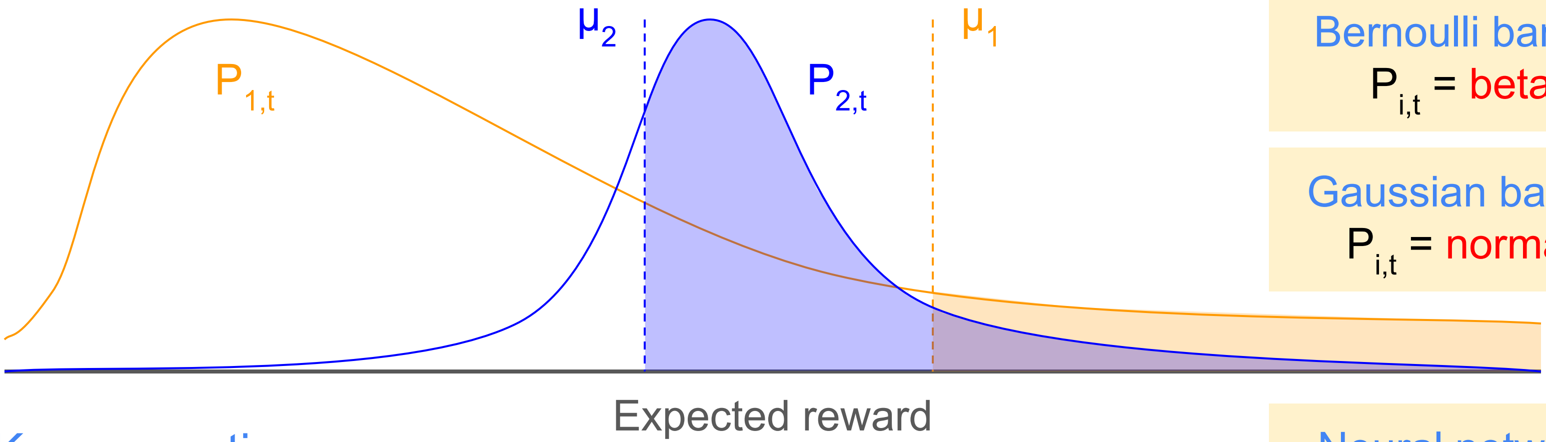
- Sample $\mu_{i,t}$ from posterior distribution $P_{i,t}$ and pull arm $I_t = \text{argmax}_i\ \mu_{i,t}$



- Key properties
  - $P_{i,t}$ concentrates at μ_i with the number of pulls
  - $\mu_{i,t}$ overestimates μ_i with a sufficient probability

Thompson Sampling (Thompson, 1933)

- Sample $\mu_{i,t}$ from posterior distribution $P_{i,t}$ and pull arm $I_t = \mathrm{argmax}_i\ \mu_{i,t}$



Bernoulli bandit
$P_{i,t}$ = beta

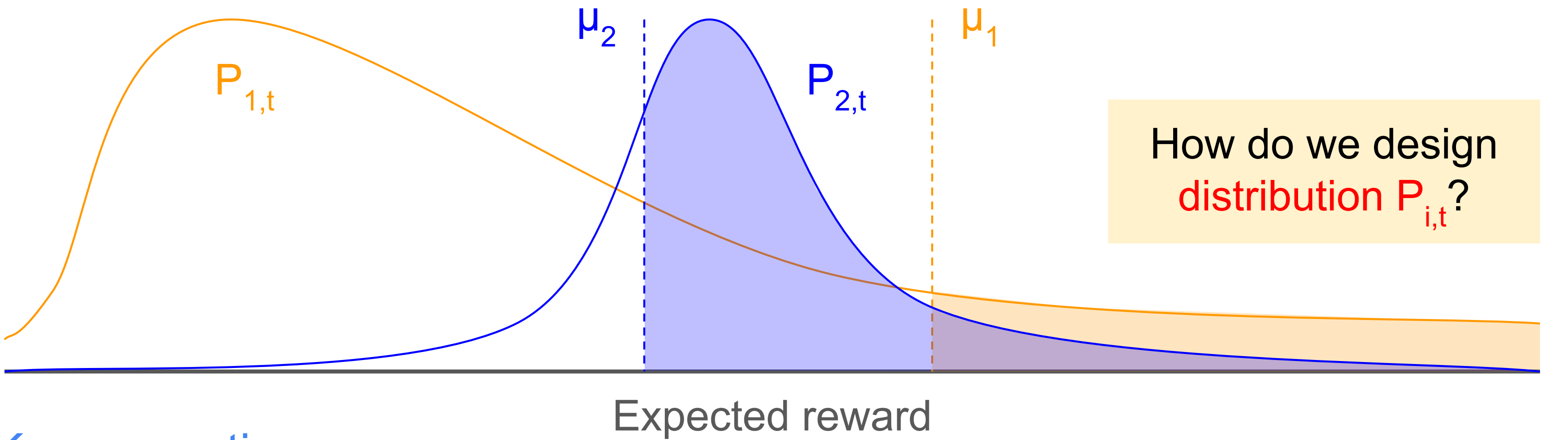
Gaussian bandit
$P_{i,t}$ = normal

- Key properties
  - $P_{i,t}$ concentrates at μ_i with the number of pulls
  - $\mu_{i,t}$ overestimates μ_i with a sufficient probability

Neural network
$P_{i,t}$ = ???

General Randomized Exploration

- Sample $\mu_{i,t}$ from ~~posterior~~ distribution $P_{i,t}$ and pull arm $I_t = \text{argmax}_i\ \mu_{i,t}$



How do we design distribution $P_{i,t}$?

- Key properties
  - $P_{i,t}$ concentrates at (scaled and shifted) μ_i with the number of pulls
  - $\mu_{i,t}$ overestimates (scaled and shifted) μ_i with a sufficient probability

Giro (Garbage In, Reward Out) with [0, 1] Rewards

- $\mu_{i,t}$ is the mean of a non-parametric bootstrap sample of the history of arm i with pseudo-rewards (garbage)

Giro (Garbage In, Reward Out) with [0, 1] Rewards

- $\mu_{i,t}$ is the mean of a non-parametric bootstrap sample of the history of arm i with pseudo-rewards (garbage)

	History		
Arm 1	0	0	
Arm 2	1	0	1

Giro (Garbage In, Reward Out) with [0, 1] Rewards

- $\mu_{i,t}$ is the mean of a non-parametric bootstrap sample of the history of arm i with pseudo-rewards (garbage)

	History	Garbage
Arm 1	0 0	0 0 1 1
Arm 2	1 0 1	0 0 0 1 1 1

Giro (Garbage In, Reward Out) with [0, 1] Rewards

- $\mu_{i,t}$ is the mean of a non-parametric bootstrap sample of the history of arm i with pseudo-rewards (garbage)

	History	Garbage	Bootstrap sample	$\mu_{i,t}$
Arm 1	0 0	0 0 1 1	1 1 1 1 0 0	2 / 3
Arm 2	1 0 1	0 0 0 1 1 1	1 1 1 1 1 0 0 0 0	5 / 9

Giro (Garbage In, Reward Out) with [0, 1] Rewards

- $\mu_{i,t}$ is the mean of a non-parametric bootstrap sample of the history of arm i with pseudo-rewards (garbage)

	History	Garbage	Bootstrap sample	$\mu_{i,t}$
Arm 1	0 0	0 0 1 1	1 1 1 1 0 0	2 / 3
Arm 2	1 0 1	0 0 0 1 1 1	1 1 1 1 1 0 0 0 0	5 / 9

- Benefits and challenges of randomized garbage
  - $\mu_{i,t}$ overestimates scaled and shifted μ_i with a sufficient probability
  - Bias in the estimate of μ_i

Contextual Giro with [0, 1] Rewards

- Straightforward generalization to complex structured problems
- $\mu_{i,t}$ is the estimated reward of arm i in a model trained on a non-parametric bootstrap sample of the history with pseudo-rewards (garbage)

History	Garbage		Bootstrap sample			$\mu_{i,t}$
$(x_1, 0)$	$(x_1, 0)$	$(x_1, 1)$	$(x_1, 0)$	$(x_1, 1)$	$(x_2, 0)$	Estimate from a learned model
$(x_2, 1)$	$(x_2, 0)$	$(x_2, 1)$	$(x_2, 1)$	$(x_2, 1)$	$(x_2, 1)$	
$(x_3, 0)$	$(x_3, 0)$	$(x_3, 1)$	$(x_3, 0)$	$(x_3, 1)$	$(x_3, 1)$	

- Giro is as general as the ε-greedy policy... but no tuning!

How to do bandits with neural networks easily?

How does Giro compare to Thompson sampling?

See you at poster #125!