Utilitarian Algorithm Configuration

NEURAL INFORMATION PROCESSING SYSTEMS

Devon R. Graham[1]
Kevin Leyton-Brown[1]
Tim Roughgarden[2,3]

1 UBC 2 3 a16zcrypto

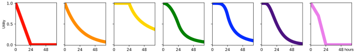
Setup

- Algorithms $i = 1, \ldots, n$
- Input instances $j = 1, 2, 3, \ldots$
- T_{ij}, runtime of i on j
- $u(T_{ij}) \in [0, 1]$, utility from running i on j
- $F_i(t) := \Pr_j(T_{ij} \leq t)$, runtime CDF
- $U_i := \mathbb{E}_j[u(T_{ij})]$, expected utility
- $\Delta_i := \max_{i'} U_{i'} - U_i$, optimality gap

Objective

- Find algorithm i^* with small optimality gap.
- Existing procedures optimize runtime.
- Our procedure optimizes utility.

Utility Function Examples



Generic Procedure

- Repeat...
  1. Choose an algorithm i.
  2. Run i on an input j for up to κ seconds.
- ... until stopping condition reached.

*The first algorithm configuration procedure to optimize **utility** instead of runtime.*

*An **anytime** procedure that requires minimal parameter-setting from the user.*

*Comes with non-trivial, **input-dependent** theoretical guarantees that improve with time.*

Scan for full paper.

[1] Graham, Devon R., Kevin Leyton-Brown, and Tim Roughgarden. "Formalizing preferences over runtime distributions." International Conference on Machine Learning. PMLR, 2023.
[2] Even-Dar, Eyal, Shie Mannor, and Yishay Mansour. "PAC bounds for multi-armed bandit and Markov decision processes." COLT 2002 Sydney, Australia, July 8–10, 2002.
[3] Mannor, Shie, and John N. Tsitsiklis. "The sample complexity of exploration in the multi-armed bandit problem." Journal of Machine Learning Research 5 Jun (2004): 623-648.

Error from Sampling

- Classic result from Bandits literature [2,3].
- Sampling introduces estimation error.
- Necessary and sufficient to take enough samples m that:

$$\sqrt{\frac{\text{logterm}}{m}} \leq \max\{\Delta_i, \epsilon\}$$

- Intuition: a large enough sample will be representative of the true mean

Error from Capping

- New, input-dependent result.
- Capping introduces error by censoring observations.
- Necessary and sufficient to take samples at a captime κ_i large enough that:

$$u(\kappa_i)(1 - F_i(\kappa_i)) \leq \Delta_i + \epsilon$$

- Intuition: we don't need to know about the tail if it contributes very little to expected utility.

Utilitarian Procrastination

- Anytime, adaptive procedure.
- Input-dependent bounds: m and κ_i only need to be large enough that:

$$\sqrt{\frac{\text{logterm}}{m}} + u(\kappa_i)(1 - F_i(\kappa_i)) \leq \max\{\Delta_i, \epsilon\}$$