NEURAL INFORMATION PROCESSING SYSTEMS

VIRGINIA TECH

WAYNE STATE UNIVERSITY

Taming Heavy-Tailed Losses in Adversarial Bandits and the Best-of-Both-Worlds Setting

Duo Cheng[1], Xingyu Zhou[2], Bo Ji[1]

[1]Department of Computer Science, Virginia Tech

[2]Department of Electrical and Computer Engineering, Wayne State University

This work is supported in part by the NSF grants under CNS-2312833, CNS-2312835, and CNS-2153220, the Commonwealth Cyber Initiative (CCI), and Nokia Corporation

Multi-armed Bandits (MAB)

Multi-armed Bandits (MAB)

Given a **time horizon** T and a **fixed arm set** $\{1, \ldots, K\} := [K]$, a learning algorithm 👤 performs the following interaction with the environment 🌍

Multi-armed Bandits (MAB)

Given a **time horizon** T and a **fixed arm set** $\{1,\ldots,K\} := [K]$, a learning algorithm 👤 performs the following interaction with the environment 🌍

In round $t = 1,\ldots,T$:

Multi-armed Bandits (MAB)

Given a **time horizon** T and a **fixed arm set** $\{1,\ldots,K\} := [K]$, a learning algorithm 👤 performs the following interaction with the environment 🌍

In round $t = 1,\ldots,T$:

1. 👤 —— **plays arm** $a_t \in [K]$ ——→ 🌍

Multi-armed Bandits (MAB)

Given a **time horizon** T and a **fixed arm set** $\{1,\ldots,K\} := [K]$, a learning algorithm 👤 performs the following interaction with the environment 🌍

In round $t = 1,\ldots,T$:

1. 👤 —— **plays arm** $a_t \in [K]$ ——→ 🌍
2. 🌍 **determines losses** $\{\ell_{t,i}\}_{i\in[K]}$

Multi-armed Bandits (MAB)

Given a **time horizon** T and a **fixed arm set** $\{1,\ldots,K\} := [K]$, a learning algorithm 👤 performs the following interaction with the environment 🌍

In round $t = 1,\ldots,T$:

1. 👤 → **plays arm** $a_t \in [K]$ → 🌍
2. 🌍 **determines losses** $\{\ell_{t,i}\}_{i\in[K]}$
3. 👤 ← **reveals loss** ℓ_{t,a_t} ← 🌍

Multi-armed Bandits (MAB)

Given a **time horizon** T and a **fixed arm set** $\{1,\ldots,K\} := [K]$, a learning algorithm 👤 performs the following interaction with the environment 🌍

In round $t = 1,\ldots,T$:

1. 👤 **plays arm** $a_t \in [K]$ → 🌍
2. 🌍 **determines losses** $\{\ell_{t,i}\}_{i\in[K]}$
3. 👤 ← **reveals loss** ℓ_{t,a_t} 🌍

- **Bounded case**: $\{\ell_{t,i}\}_{i\in[K]} \in [0,1]^K$

Multi-armed Bandits (MAB)

Given a **time horizon** T and a **fixed arm set** $\{1,\ldots,K\} := [K]$, a learning algorithm 👤 performs the following interaction with the environment 🌍

In round $t = 1,\ldots,T$:

1. 👤 **plays arm** $a_t \in [K]$ → 🌍
2. 🌍 **determines losses** $\{\ell_{t,i}\}_{i\in[K]}$
3. 👤 ← **reveals loss** ℓ_{t,a_t} 🌍

- **Bounded case**: $\{\ell_{t,i}\}_{i\in[K]} \in [0,1]^K$
- **Heavy-tailed case** (this work):

Multi-armed Bandits (MAB)

Given a **time horizon** T and a **fixed arm set** $\{1,\ldots,K\} := [K]$, a learning algorithm 👤 performs the following interaction with the environment 🌍

In round $t = 1,\ldots,T$:

1. 👤 **plays arm** $a_t \in [K]$ → 🌍
2. 🌍 **determines losses** $\{\ell_{t,i}\}_{i\in[K]}$
3. 👤 ← **reveals loss** ℓ_{t,a_t} 🌍

- **Bounded case**: $\{\ell_{t,i}\}_{i\in[K]} \in [0,1]^K$
- **Heavy-tailed case** (this work):
  1. 🌍 determines distributions $\{P_{t,i}\}_{i\in[K]}$ s.t. $\mathbb{E}_{\ell_{t,i}\sim P_{t,i}}[|\ell_{t,i}|^{1+v}] \le u^{1+v}$ for every arm $i \in [K]$ and some **known** $u > 0, v \in (0,1]$

Multi-armed Bandits (MAB)

Given a **time horizon** T and a **fixed arm set** $\{1,\ldots,K\} := [K]$, a learning algorithm performs the following interaction with the environment

In round $t = 1,\ldots,T$:

1. **plays arm** $a_t \in [K]$ →
2. **determines losses** $\{\ell_{t,i}\}_{i\in[K]}$
3. ← **reveals loss** ℓ_{t,a_t}

- **Bounded case**: $\{\ell_{t,i}\}_{i\in[K]} \in [0,1]^K$
- **Heavy-tailed case** (this work):
  1. determines distributions $\{P_{t,i}\}_{i\in[K]}$ s.t. $\mathbb{E}_{\ell_{t,i}\sim P_{t,i}}[|\ell_{t,i}|^{1+v}] \leq u^{1+v}$ for every arm $i \in [K]$ and some **known** $u > 0, v \in (0,1]$
  2. $\ell_{t,i}$ is drawn from $P_{t,i}$ for every arm i

Multi-armed Bandits (MAB)

Given a **time horizon** T and a **fixed arm set** $\{1,\ldots,K\} := [K]$, a learning algorithm 👤 performs the following interaction with the environment 🌍

In round $t = 1,\ldots,T$:

1. 👤 **plays arm** $a_t \in [K]$ → 🌍
2. 🌍 **determines losses** $\{\ell_{t,i}\}_{i\in[K]}$
3. 👤 ← **reveals loss** ℓ_{t,a_t} 🌍

- **Bounded case**: $\{\ell_{t,i}\}_{i\in[K]} \in [0,1]^K$
- **Heavy-tailed case** (this work):
  1. 🌍 determines distributions $\{P_{t,i}\}_{i\in[K]}$ s.t. $\mathbb{E}_{\ell_{t,i}\sim P_{t,i}}[|\ell_{t,i}|^{1+v}] \le u^{1+v}$ for every arm $i \in [K]$ and some **known** $u > 0, v \in (0,1]$
  2. $\ell_{t,i}$ is drawn from $P_{t,i}$ for every arm i

Objective of 👤: minimizing (pseudo-)**regret**

$$R_T := \sum_{t=1}^{T} \left(\mu_{t,a_t} - \mu_{t,i^*}\right) \text{ with } \mu_{t,i} := \mathbb{E}_{\ell_{t,i}\sim P_{t,i}}[\ell_{t,i}] \text{ and } i^* := \operatorname{argmin}_{i\in[K]} \sum_{t=1}^{T} \mu_{t,i}$$

The Best of Both Worlds (BOBW) in Heavy-tailed MAB

The Best of Both Worlds (BOBW) in Heavy-tailed MAB

Depending on how loss distributions are determined, 🌍 is categorized as:

The Best of Both Worlds (BOBW) in Heavy-tailed MAB

Depending on how loss distributions are determined, 🌍 is categorized as:

- (Oblivious) **Adversarial** regime: loss distributions $\{P_{t,i}\}_{t\in[T],i\in[K]}$ are determined arbitrarily ahead of time with full knowledge on 👤

The Best of Both Worlds (BOBW) in Heavy-tailed MAB

Depending on how loss distributions are determined, 🌍 is categorized as:

- (Oblivious) **Adversarial** regime: loss distributions $\{P_{t,i}\}_{t \in [T], i \in [K]}$ are determined arbitrarily ahead of time with full knowledge on 👤
- **Stochastic** regime: the loss distribution of each arm is fixed over time, i.e., $P_{1,i} = \ldots = P_{T,i}, \forall i \in [K]$

The Best of Both Worlds (BOBW) in Heavy-tailed MAB

Depending on how loss distributions are determined, 🌍 is categorized as:

- (Oblivious) **Adversarial** regime: loss distributions $\{P_{t,i}\}_{t\in[T],i\in[K]}$ are determined arbitrarily ahead of time with full knowledge on 👤
- **Stochastic** regime: the loss distribution of each arm is fixed over time, i.e., $P_{1,i} = \ldots = P_{T,i}, \forall i \in [K]$

BOBW: one single 👤 ensures both optimal $O(T^{\frac{v}{1+v}})$ regret in adv. regime and $O(\log T)$ regret in sto. regime

Algorithm	Adversarial	Stochastic
Lower bound [Bubeck et al., 13]	$\Omega(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	$\Omega(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$
Robust UCB [Bubeck et al., 13]	N/A	$O(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$
HT-INF [Huang et al., 22]	$O(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	$O(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$

“Truncated Non-negative Losses” Assumption

“Truncated Non-negative Losses” Assumption

The **BOBW** guarantee of **HT-INF** [Huang et al., 22] requires the following “**Truncated Non-negative Losses**” (***TNL***) assumption (with $1[\,\cdot\,]$ denoting the indicator function):

$$\mathbb{E}_{\ell_{t,i^*} \sim P_{t,i^*}}[\ell_{t,i^*} \cdot 1[\,|\ell_{t,i^*}| > M]] \geq 0, \forall M > 0,\ t \in [T]\,.$$

"Truncated Non-negative Losses" Assumption

The **BOBW** guarantee of **HT-INF** [Huang et al., 22] requires the following "**Truncated Non-negative Losses**" (***TNL***) assumption (with $1[\,\cdot\,]$ denoting the indicator function):

$$\mathbb{E}_{\ell_{t,i^*}\sim P_{t,i^*}}[\ell_{t,i^*} \cdot 1[\,|\,\ell_{t,i^*}\,| > M]] \geq 0, \forall M > 0,\ t \in [T]\,.$$

Without ***TNL***, there is **NO** 👤 with optimal regret even **solely** for the adv. regime

“Truncated Non-negative Losses” Assumption

The **BOBW** guarantee of **HT-INF** [Huang et al., 22] requires the following “**Truncated Non-negative Losses**” (***TNL***) assumption (with $1[\,\cdot\,]$ denoting the indicator function):

$$\mathbb{E}_{\ell_{t,i^*}\sim P_{t,i^*}}[\ell_{t,i^*} \cdot 1[\,|\ell_{t,i^*}| > M]] \geq 0, \forall M > 0,\ t \in [T]\,.$$

Without ***TNL***, there is **NO** 👤 with optimal regret even **solely** for the adv. regime

Handling heavy tails without ***TNL*** would further allow us to enjoy:

“Truncated Non-negative Losses” Assumption

The **BOBW** guarantee of **HT-INF** [Huang et al., 22] requires the following “**Truncated Non-negative Losses**” (***TNL***) assumption (with $1[\,\cdot\,]$ denoting the indicator function):

$$\mathbb{E}_{\ell_{t,i^*}\sim P_{t,i^*}}[\ell_{t,i^*}\cdot 1[\,|\ell_{t,i^*}|>M]]\geq 0, \forall M>0,\ t\in[T]\,.$$

Without ***TNL***, there is **NO** 👤 with optimal regret even **solely** for the adv. regime

Handling heavy tails without ***TNL*** would further allow us to enjoy:

1. The first optimal regret in the adv. regime when observed losses are **contaminated** by the **Huber** model

“Truncated Non-negative Losses” Assumption

The **BOBW** guarantee of **HT-INF** [Huang et al., 22] requires the following “**Truncated Non-negative Losses**” (***TNL***) assumption (with $1[\,\cdot\,]$ denoting the indicator function):

$$\mathbb{E}_{\ell_{t,i^*}\sim P_{t,i^*}}[\ell_{t,i^*}\cdot 1[\,|\ell_{t,i^*}|>M]]\geq 0, \forall M>0,\ t\in[T]\,.$$

Without ***TNL***, there is **NO** 👤 with optimal regret even **solely** for the adv. regime

Handling heavy tails without ***TNL*** would further allow us to enjoy:

1. The first optimal regret in the adv. regime when observed losses are **contaminated** by the **Huber** model
2. The first **BOBW** regret when losses are protected under **pure Local Differential Privacy (LDP)**

Key Question

In heavy-tailed MAB, are there any fundamental barriers to the worst-case optimal regret in the **adversarial** regime and the **BOBW** guarantee?

Main Results of This Work

Algorithm	Adversarial	Stochastic	*TNL*-free	High-prob.
Lower bound [Bubeck et al., 13]	$\Omega(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	$\Omega(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$	N/A	N/A
Robust UCB [Bubeck et al., 13]	N/A	$O(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$	✅	✅
HT-INF [Huang et al., 22]	$O(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	$O(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$	❌	❌
OMD-LB-HT **(This work)**	$\tilde{O}(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	N/A	✅	✅
SAO-HT **(This work)**	$\tilde{O}(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	$O(\frac{K\log(K)(\log T)^4}{(\Delta)^{1/v}})$	✅	✅

$\Delta := \min_{i:\Delta_i>0} \Delta_i$

Main Results of This Work

Algorithm	Adversarial	Stochastic	***TNL***-free	High-prob.
Lower bound [Bubeck et al., 13]	$\Omega(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	$\Omega(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$	N/A	N/A
Robust UCB [Bubeck et al., 13]	N/A	$O(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$	✅	✅
HT-INF [Huang et al., 22]	$O(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	$O(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$	❌	❌
OMD-LB-HT **(This work)**	$\tilde{O}(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	N/A	✅	✅
SAO-HT **(This work)**	$\tilde{O}(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	$O(\frac{K\log(K)(\log T)^4}{(\Delta)^{1/v}})$	✅	✅

$\Delta := \min_{i:\Delta_i>0} \Delta_i$

Main Results of This Work

Algorithm	Adversarial	Stochastic	***TNL*-free**	High-prob.
Lower bound [Bubeck et al., 13]	$\Omega(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	$\Omega(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$	N/A	N/A
Robust UCB [Bubeck et al., 13]	N/A	$O(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$	✅	✅
HT-INF [Huang et al., 22]	$O(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	$O(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$	❌	❌
OMD-LB-HT **(This work)**	$\tilde{O}(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	N/A	✅	✅
SAO-HT **(This work)**	$\tilde{O}(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	$O(\frac{K\log(K)(\log T)^4}{(\Delta)^{1/v}})$	✅	✅

$\Delta := \min_{i:\Delta_i>0} \Delta_i$

Main Results of This Work

Algorithm	Adversarial	Stochastic	*TNL*-free	High-prob.
Lower bound [Bubeck et al., 13]	$\Omega(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	$\Omega(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$	N/A	N/A
Robust UCB [Bubeck et al., 13]	N/A	$O(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$	✅	✅
HT-INF [Huang et al., 22]	$O(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	$O(\sum_{i:\Delta_i>0} \frac{\log T}{(\Delta_i)^{1/v}})$	❌	❌
OMD-LB-HT **(This work)**	$\tilde{O}(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	N/A	✅	✅
SAO-HT **(This work)**	$\tilde{O}(uK^{\frac{1}{1+v}}T^{\frac{v}{1+v}})$	$O(\frac{K\log(K)(\log T)^4}{(\Delta)^{1/v}})$	✅	✅

$\Delta := \min_{i:\Delta_i>0} \Delta_i$

High-prob. bound:
- is stronger than expected bound
- implies high-prob. bound even in the **adaptive** adv. regime 😈 (in which loss distributions could depend on the history)

Concluding Remarks

Concluding Remarks

1. In heavy-tailed MAB, we achieve the first optimal adv. guarantee and the first **BOBW** guarantee without ***TNL*** assumption

Concluding Remarks

1. In heavy-tailed MAB, we achieve the first optimal adv. guarantee and the first **BOBW** guarantee without ***TNL*** assumption

2. By relaxing ***TNL***, we also achieve the first optimal adv. guarantee in the **Huber contamination** model and the first **BOBW** guarantee under pure **LDP**

Concluding Remarks

1. In heavy-tailed MAB, we achieve the first optimal adv. guarantee and the first **BOBW** guarantee without ***TNL*** assumption

2. By relaxing ***TNL***, we also achieve the first optimal adv. guarantee in the **Huber contamination** model and the first **BOBW** guarantee under pure **LDP**

3. All the guarantees above hold **with high probability**, and hence have the potential to handle **adaptive adversaries** 😈