Machine Learning (2019) 108:687–715
https://doi.org/10.1007/s10994-018-5758-5

Corruption-tolerant bandit learning

Sayash Kapoor[1] · **Kumar Kshitij Patel**[1] · **Purushottam Kar**[1]

Received: 28 January 2018 / Accepted: 9 August 2018 / Published online: 29 August 2018

Abstract
We present algorithms for solving multi-armed and linear-contextual bandit tasks in the face of adversarial corruptions in the arm responses. Traditional algorithms for solving these problems assume that nothing but mild, e.g., i.i.d. sub-Gaussian, noise disrupts an otherwise clean estimate of the utility of the arm. This assumption and the resulting approaches can fail catastrophically if there is an observant adversary that corrupts even a small fraction of the responses generated when arms are pulled. To rectify this, we propose algorithms that use recent advances in robust statistical estimation to perform arm selection in polynomial time. Our algorithms are easy to implement and vastly outperform several existing UCB and EXP-style algorithms for stochastic and adversarial multi-armed and linear-contextual bandit problems in wide variety of experimental settings. Our algorithms enjoy minimax-optimal regret bounds, as well as can tolerate an adversary that is allowed to corrupt upto a universally constant fraction of the arms pulled by the algorithm.

Keywords Robust learning · Online learning · Bandit algorithms

1 Introduction

The recent years have witnessed a surge in the applications of online learning, especially those of explore-exploit techniques such as multi-armed bandits and linear-contextual bandits, to online recommendation (Li et al. 2010), online advertising (Chakrabarti et al. 2008), web analytics (Tang et al. 2013), crowdsourcing (Padmanabhan et al. 2016), and even mobile health (Tewari and Murphy 2017). The result has been a diverse and rich literature, accompanied by a deep understanding of how these algorithms work on large-scale data. However, the point of application of these techniques to real-world data throws up several unforeseen

Editors: Jesse Davis, Elisa Fromont, Derek Greene, and Bjorn Bringmann.

✉ Purushottam Kar
purushot@cse.iitk.ac.in

Sayash Kapoor
sayash@cse.iitk.ac.in

Kumar Kshitij Patel
kishinmh@cse.iitk.ac.in

[1] Indian Institute of Technology Kanpur, Kanpur, India

challenges, such as those of scale and data quality. In particular, when working with consumer/user data, it is inadvisable to assume clean theoretical models for data to hold ground beyond a point. Some concrete examples are outlined below.

Click fraud via malware: malware present on user systems can be used to effectively sabotage an advertisement campaign run by a competitor by suppressing clicks on the ads pertaining to that campaign, causing a typical online advertising platform to reject those ads from consideration.

Fake reviews and ratings via automated bots: automated bots can alternatively be used to artificially boost products by posting fake reviews or simulating clicks on a compromised website, which can cause recommendation platforms to get tricked into giving those products more visibility.

Transient socio-political effects: for companies that employ celebrity brand ambassadors, actions taken by those ambassadors in their personal lives can often adversely affect brand popularity (Times, 2015) and cause a large number of users to post negative reviews or downgrade their ratings in a short period. This can adversely affect the functioning of recommendation systems, as well as the experience of users unconcerned with the event, in the short term.

Outlier behavior: not all data corruption need be malicious or even intended, but may nevertheless adversely affect the functioning of the decision making systems operating on that data. For example, in mobile health applications, temporary issues with the mobile device or mobile connectivity may cause the algorithm to conclude that a patient has become unresponsive and then target that patient more aggressively, which may adversely affect patient cooperation.

Multi-armed and linear-contextual bandits algorithms are two of the most popularly used techniques in recommendation and advertising settings. If executed in the above settings with data corruption, these bandit algorithms will encounter corrupted arm rewards/responses and their performance may degrade.

Now, note that in all the settings mentioned above, the corruptions/aberrations to the data are sparse, and sometimes even transient. For example, it is reasonable to assume that only a fraction of clicks can be suppressed by malware or be synthesized by bots. Even in the mobile health and brand-ambassador examples, the effects of data corruption are transient, hence sparse when viewed as a fraction of long-term data. Thus, a direct solution to the problems mentioned above would be to make these bandit algorithms *robust* to sparse corruptions in arm responses.

The recent years have indeed seen a resurgence of interest in developing algorithms that are resilient to data corruption. We will review these shortly. These contemporary lines of work trace their origin at least half a century back to the area of *robust statistics* (Huber 1964; Tukey 1960; Maronna et al. 2006). However, recent works have focused more on developing robust algorithms that are scalable and efficient, whereas classical works usually paid scant attention to scalability.

In our work, we develop online learning algorithms for two settings, namely multi-armed and linear-contextual bandit problems, that are tolerant to sparse corruptions in the arm responses that they receive. Our algorithms enjoy minimax-optimal regret bounds in the face of fully adaptive adversaries, as well as vastly outperform several existing approaches to both stochastic, as well as adversarial multi-armed and linear-contextual bandit problems, in

experiments. We believe our results come at an opportune moment, at a time when scalable robust algorithms are being actively investigated, as are online algorithms.

1.1 Organization

We address two bandit settings and present a total of three new algorithms. In Sect. 2, we give a brief overview of bandit literature, and discuss related work from three areas: adversarial bandits, robust algorithms, and heavy-tailed bandits. In Sect. 3, we introduce the notation we use in the rest of the paper.

In Sect. 4, we discuss the multi-armed bandits (MAB) setting that is popular when the set of actions is small and fixed, e.g., in web analytics and mobile health. We introduce two algorithms RUCB- MAB and RUCB- TUNE for this setting.

In Sect. 5, we discuss linear contextual bandits, a more general setting which allows arms to be parametrized, as well as the set of available arms to change from time step to time step. This is most applicable in online advertising and recommendation settings where the set of available ads/products may change across time. We introduce RUCB- LIN for this setting.

In Sect. 6, we perform extensive experimentation, comparing our proposed algorithms against stochastic bandit algorithms such as UCB, KL-UCB, UCBV and many others, adversarial bandit algorithms such as EXP3 and SAO, and algorithms for heavy-tailed bandits from Medina and Yang (2016). We conclude with an overview of interesting directions for future work in Sect. 7.

2 Related works and our contributions

Literature on bandits is too vast to be surveyed here. Starting with the early work of Auer et al. (2002a) on multi-armed bandits (MAB), the field has seen progress in linear bandits (Abbasi-Yadkori et al. 2011), contextual bandits (Chu et al. 2011), as well as applications to recommendation (Li et al. 2010), advertising (Chakrabarti et al. 2008), web analytics (Tang et al. 2013), crowdsourcing (Padmanabhan et al. 2016), and mobile health (Tewari and Murphy 2017).

The three lines of work that relate most closely to ours are (1) those on adversarial bandits where arm rewards/responses need not be stochastic at all, (2) those on developing corruption-resilient learning and estimation algorithms, and (3) those on bandits that suffer heavy-tailed albeit still stochastic and non-adversarial noise (since these algorithms are also sometimes referred to as "robust"). We review all three lines of work below and clarify our contributions in context.

2.1 Adversarial bandits

Given the presence of an adversary in our setting, it is tempting to utilize algorithms designed to work with non-stochastic arm reward assignments. There does exist a large body of work on EXP-style algorithms starting with Auer et al. (2002b), namely EXP3 for multi-armed bandits and EXP4 for linear contextual bandits, as well as variants such as EXP3++ (Bubeck and Slivkins 2012) and SAO (Seldin and Slivkins 2014), that are indeed able to offer sub-linear regret even if all (not just a fraction of) arm responses are chosen by an adversary.

This in itself is too pessimistic a view given that we have observed in Sect. 1 that in real-life settings, it is reasonable to expect only a fraction of the arm responses to be corrupted.

Moreover, their attractive regret bounds notwithstanding, there is a price to pay for using EXP-style algorithms. Indeed, most recent works on adversarial bandits (Bubeck and Slivkins 2012; Lykouris et al. 2018; Seldin and Slivkins 2014) focus only on multi-armed bandits and not linear-contextual bandits. This is possibly because EXP-style algorithms (such as EXP4) rapidly become infeasible to execute in practice for linear-contextual bandits.

However, we propose RUCB- LIN, a practical and efficient algorithm for linear-contextual bandits that can tolerate adversarial corruptions. Moreover, we also experimentally compare to EXP3 and SAO in the MAB setting and show that our proposed algorithms RUCB- MAB and RUCB- TUNE outperform it. We also note that from a theoretical standpoint, the regret bounds offered by EXP-style algorithms do not compare directly to the *pseudo-regret* style bounds prevalent for stochastic bandits that we provide for our algorithms.

The recent work of Lykouris et al. (2018) deserves special mention since it considers a problem setting similar to ours wherein the adversarial corruption is not rampant. Our work is independent and indeed, our algorithms and analyses differ significantly from those of Lykouris et al. Their work considers only multi-armed bandits whereas we consider multi-armed bandits as well as the more challenging case of linear-contextual bandits. Indeed, arm elimination, the strategy adopted by Lykouris et al., cannot be reliably practiced in contextual settings where the set of available "arms" may change arbitrarily from time step to time step. Moreover, in experiments, we find that RUCB- MAB and RUCB- TUNE beat strategies such as SAO that also use a form of arm-elimination.

From a theoretical standpoint, Lykouris et al. do not explicitly model the fraction of arm responses that are corrupted but instead consider the total amount of corruption introduced by the adversary during the entire online process, say $\mathcal{C}_{\text{tot}}$. Their regret bounds are of the form $\mathcal{C}_{\text{tot}} \cdot K \cdot \log^2(KT) \cdot \sum_{i \neq i^*} \frac{1}{\Delta_i}$ where K is the number of arms, i^* is the optimal arm, Δ_i is the sub-optimality in arm i and T is the time horizon. Since we can have $\mathcal{C}_{\text{tot}} = \Omega(T)$ if a constant fraction of responses are corrupted, it is not desirable that the regret bound have $\mathcal{C}_{\text{tot}}$ and the number of arms K in a multiplicative union.

In contrast, we explicitly model the fraction η of arm responses that are corrupted and offer regret bounds of the form (see Theorem 2) $\bar{R}_T(\text{RUCB- MAB}) \leq \sum_{i \neq i^*} \frac{\log T}{\Delta_i} + \eta \cdot B \cdot T$ where B is an upper bound on the corruption magnitudes. Note that the term $\eta \cdot B \cdot T$ plays the same role as $\mathcal{C}_{\text{tot}}$ does for Lykouris et al. Also notice that in our bound, this term is completely independent of the number of arms and that our bound is additive in this term, not multiplicative.

2.2 The best of both worlds?

Given the wide gap between settings with stochastic arm responses and those with adversarial responses, there has been interest in developing algorithms that can seamlessly address both: offer a superior $\log T$ regret bound if all arm responses are stochastic and regress to a more conservative $\sqrt{T}$ bound if arm responses are adversarial. Existing works achieve this either by starting out optimistically assuming a stochastic setting and then switching to EXP-style policies upon detecting signs of adversarial behavior, e.g., SAO (Bubeck and Slivkins 2012), or else carefully tuning EXP-style policies so as to offer $\log T$ regret if arm responses are completely stochastic, e.g., EXP3++ (Seldin and Slivkins 2014).

Experimentally, we compare to both SAO and EXP3 and find that RUCB- MAB and RUCB- TUNE outperform both. From a theoretical standpoint, we too can provide "best-of-both-worlds" style guarantees for RUCB- MAB and RUCB- LIN (see Theorems 2, 7). This is because our bounds for both, multi-armed as well as linear contextual bandits, gracefully

upgrade to minimax-optimal bounds for stochastic bandits if the corruption rate η goes to zero. $\eta = 0$ is the case when there is no malicious adversary and all rewards are truly stochastic. Thus, we are indeed able to recover the "best of the stochastic world".

Moreover, we offer minimax-optimal regret bounds even if a bounded fraction of the arm responses are corrupted, thus offering the "best of the adversarial world" too. Our bounds cannot handle a totally rampant adversary that, for example, corrupts all the rewards, i.e., when $\eta \to 1$. This is because our algorithms are robust versions of UCB whereas "best-of-both-worlds" style results typically choose EXP3 as the base algorithm but this choice has drawbacks as discussed earlier.

2.3 Robust learning and estimation algorithms

Robust algorithms have recently attracted a lot of attention in several areas of machine learning, signal processing, and algorithm design. Some prominent applications for which robust algorithms have been investigated are statistical estimation (Diakonikolas et al. 2018; Lai et al. 2016), optimization (Charikar et al. 2017), principal component analysis (Candès et al. 2009), regression (Bhatia et al. 2015; Chen et al. 2013; Nguyen and Tran 2013) and classification (Feng et al. 2014).

Our algorithms make novel use of recent advances in robust estimation techniques viz moment estimation (Lai et al. 2016) and linear regression (Bhatia et al. 2015). However, these adaptations are not immediate or trivial, especially for linear bandit settings where the proof progression of OFUL-style analyses has to be adapted in a novel way to accommodate the complex estimation steps carried out by robust linear regression algorithms.

2.4 Heavy-tailed bandits

There has been recent interest in developing bandit algorithms where the arm responses are samples from heavy-tailed distributions such as the works of Bubeck et al. (2013), Medina and Yang (2016), Padmanabhan et al. (2016). A point of confusion may arise here since these algorithms are also sometimes referred to as "robust" algorithms. However, crucial differences exist in our problem setting that makes these results inapplicable directly.

We note that in heavy-tailed settings, arm responses are still generated from a static distribution. However, in our problem setting, there will be an adaptive adversary which need not follow any predeclared distribution heavy-tailed or otherwise, when introducing corruptions. For example, our experiments consider an adversary that flips the sign of the response of an arm to make that arm seem unnaturally good or bad. Heavy-tailed distributions cannot model such a sentient and malicious adversary and as such, existing analyses do not apply.

Thus, works on heavy-tailed bandits do not apply in our setting. We nevertheless experimentally compare to these algorithms and show that our proposed algorithm RUCB-LIN outperforms them. Moreover, our algorithms tolerate as much as a constant fraction of corrupted responses, e.g., $\eta \cdot n$ out of a total of n responses for some constant $\eta > 0$, whereas in heavy-tailed analyses, due to assumptions made on the arm distributions, often only a logarithmic number of the total responses, e.g., $\log n$, come from "the tail", a fact often exploited by these analyses.

Another work of interest is that of Gajane et al. (2018), which considers privacy-preserving bandit algorithms. To achieve privacy-preservation, the algorithm transforms the arm responses using a known and invertible stochastic corruption process. However, there

is no external malicious adversary in this process and the reward transformations are indeed known to the algorithm.

3 Notation

We will denote vectors using boldface lower case Latin or Greek letters, e.g., $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ and $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$. The ith component of a vector $\mathbf{x}$ will be denoted as $\mathbf{x}_i$. Upper case Latin letters will be used to denote random variables and matrices, e.g., A, X, I.

$[n]$ will denote the set of natural numbers $\{1, 2, \ldots, n\}$. We will use the shorthand $\{v_i\}_S$ to denote the set $\{v_i : i \in S\}$. In particular $\{v_i\}_{[n]}$ will denote the set $\{v_1, \ldots, v_n\}$. $\mathbb{I}\{\cdot\}$ will denote the indicator operator signaling the occurrence of an event, i.e., $\mathbb{I}\{E\} = 1$ if event E takes place and $\mathbb{I}\{E\} = 0$ otherwise. The expectation of a random variable X will be denoted by $\mathbb{E}[X]$.

Given a matrix $X \in \mathbb{R}^{d\times n}$ and any set $S \subset [n]$, we let $X_S := [\mathbf{x}_i]_{i\in S} \in \mathbb{R}^{d\times|S|}$ denote the matrix whose columns correspond to entries in the set S. Also, for any vector $\mathbf{v} \in \mathbb{R}^n$ we use the notation $\mathbf{v}_S$ to denote the $|S|$-dimensional vector consisting of those components that are in S. We use the notation $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$ to denote, respectively, the smallest and largest eigenvalues of a square symmetric matrix M.

4 Robust multi-armed bandits

In this section, we will discuss the classical multi-armed bandit, introduce various adversary models and present the rUCB- MAB and rUCB- Tune algorithms.

4.1 Problem setting

The K-armed bandit problem is characterized by an ensemble of K distributions $\boldsymbol{\nu} = \{\nu_1, \ldots, \nu_K\}$ over reals, one corresponding to each *arm*, with corresponding means $\boldsymbol{\mu} = \{\mu_1, \ldots, \mu_K\} \in \mathbb{R}^K$. At each time step, the player selects and *pulls* an arm $I_t \in [K]$ guided by some arm-selection strategy π. In response, a *reward* $r_t \in \mathbb{R}$ is generated (see below for details). Let $\mathcal{H}^t = \{I_1, r_1, \ldots, I_{t-1}, r_{t-1}, I_t\}$ denote the past history of the plays, $i^* \in \arg\max_{i\in[K]} \mu_i$ denote an arm with the highest expected reward, $\mu^* = \mu_{i^*}$ denote the highest expected reward, $\Delta_i = \mu^* - \mu_i$ denote the sub-optimality of arm i, and $\Delta_{\min} := \min_{\Delta_i>0} \Delta_i$ denote the sub-optimality of the closest competitor to the best arm(s).

Problem Setting 1 Adversarial Multi-armed Bandits

for $t = 1, 2, 3..$ **do**
  Player plays an arm, $I_t \in [K]$
  Adversary tosses a coin, $z_t = \text{Ber}(\eta) \in \{0, 1\}$
  Adversary chooses a corruption ζ_t
  Clean reward is generated $r_t^* \sim \nu_{I_t}$ conditioned on $\mathcal{H}^t$
  Player receives reward, $r_t = \mathbb{I}\{z_t = 0\} \cdot r_t^* + \mathbb{I}\{z_t = 1\} \cdot \zeta_t$
end for

4.2 Adversary model

In the stochastic setting, after the player pulls the arm I_t at time t, the reward is generated (conditioned on $\mathcal{H}^t$) from the distribution ν_{I_t} so that $\mathbb{E}\left[r_t \mid \mathcal{H}^t\right] = \mu_{I_t}$. Thus, in this "clean" setting, the reward obtained for an arm is always an unbiased estimate of its mean reward. Previous works such as those of Bubeck et al. (2013), Medina and Yang (2016) have studied settings where the distributions ν_i are heavy-tailed. However, we are more interested in cases where occasionally, the reward that is generated for the played arm is not the one received by the player at all, for applications to click fraud and other settings.

Several adversary models are prevalent in literature. To present the essential aspects of our methods, we choose a simple *stochastic* adversary model for the first discussion. We will consider a much more powerful *fully adaptive* adversary in the next section on linear-contextual bandits. We note that although algorithms for heavy-tailed bandits can handle stochastic adversaries, we will be able to handle polynomially many corruptions and, as we point out later, we can modify our algorithms to handle adaptive adversaries in this setting itself as well.

Let η denote the *corruption rate*. A stochastic adversary closely follows the progress of the arm pulls and reward generation. At each time step t, after the algorithm has decided to pull an arm I_t, the adversary first decides whether to corrupt this arm pull or not by performing a Bernoulli trial $z_t \in \{0, 1\}$ with bias η, i.e., if $\mathcal{H}^t = \{I_1, z_1, r_1, \ldots, I_{t-1}, z_{t-1}, r_{t-1}, I_t\}$, then $\mathbb{E}\left[z_t \mid \mathcal{H}^t\right] = \eta$. Then it generates a corruption ζ_t arbitrarily but independent of $\mathcal{H}^t$. After this, the "clean reward" is generated in the classical manner satisfying $\mathbb{E}\left[r_t^* \mid \mathcal{H}^t\right] = \mu_{I_t}$ and the reward received by the player is calculated as follows

$$r_t = \mathbb{I}\{z_t = 0\} \cdot r_t^* + \mathbb{I}\{z_t = 1\} \cdot \zeta_t.$$

Let B denote the largest magnitude of any corruption, i.e., $|\zeta_t| \leq B$. This bound B need not be known to the learner. Note that we allow the adversary to generate the corruption completely arbitrarily and that too *after* it is known which arm will be pulled. This allows the adversary to give different corruptions if it knows that the best arm is being played, i.e., $I_t = i^*$ than if a non-best arm is being played. We will later study more powerful adversarial models where the adversary can choose to corrupt the arm pull and even decide the corruption *after* the clean reward r_t^* has been generated and even in a manner *dependent* on the complete history $\mathcal{H}^t$.

4.3 Notions of regret

In classical bandit learning, the goal of the algorithm is to minimize regret or alternatively, maximize the cumulative reward $\sum_{t=1}^{T} r_t$ accumulated over the entire play of T rounds. However, in our corrupted setting, this, may not be the most appropriate. To address this, we consider two notions of regret.

The first notion, which we simply refer to as *Regret* in this paper, captures how the expected cumulative reward actually received by algorithm compares to the expected cumulative reward that it could have gotten had it only played the best arm again and again and had there been no adversary to corrupt those fictional arm pulls. We define this notion for an algorithm over a sequence of T plays as

$$\bar{R}_T(\pi) = \sum_{t=1}^{T} \mu^* - \mathbb{E}\left[r_t\right] = \mu^* \cdot T - \mathbb{E}\left[\sum_{t=1}^{T} r_t\right].$$

However, one may complain that this notion of regret is unfair since it pits uncorrupted rewards of the best arm against the corrupted rewards of the arms that are played. To address this concern, we also look at the notion of *Uncorrupted Regret*, defined below, which is a more fair comparison since it compares expected uncorrupted rewards of the arms played with those of the best arm:

$$\bar{R}_T^*(\pi) = \sum_{t=1}^{T} \mu^* - \mathbb{E}\left[r_t^*\right] = \mu^* \cdot T - \mathbb{E}\left[\sum_{t=1}^{T} r_t^*\right].$$

We note that this notion exactly corresponds to the popular notion of *pseudo-regret* which looks at the expected performance of a single best arm *in hindsight*.

4.4 A minimax regret lower bound

The presence of an adversary (even a stochastic one) can make life difficult for a player. Indeed, consider a setting where $\mu^* > 0$ and we have an adversary that, whenever allowed to, corrupts the reward to a default value of $r_t = 0$. For this simple setting, even for the optimal policy that always plays $I_t \equiv i^*$, the expected regret is still $\bar{R}_T = \eta\mu^* \cdot T$. The following result demonstrates this crisply by establishing a minimax regret lower bound for the stochastic adversary model.

Theorem 1 *Let $K > 1$ and $T \geq K - 1$. Then for any policy π, and any constant $c \in (0, 1)$, there exists an MAB instance characterized by K distributions $\boldsymbol{\nu} = \{\nu_1, \ldots, \nu_K\}$ all of which are Gaussian with unit variance and means that lie in the the interval $[0, 1]$, i.e., $\nu_i = \mathcal{N}(\mu_i, 1)$ where $\mu_i \in [0, 1]$, and a stochastic adversary with corruption rate η such that*

$$\bar{R}_T(\pi) \geq \frac{1}{27}\sqrt{(K-1)T} + c\eta \cdot T.$$

Algorithm 1 rMEst: Robust Mean

Input: Set $S = \{x_i\}_{[n]}$
Output: An estimate of mean(S)
1: **return** median(S)

Algorithm 2 rVUCB: Robust Variance Upper Confidence Bound

Input: Set $S = \{x_i\}_{[n]}$, horizon estimate T, upper bound $\eta_0 < 1/2$ on corruption rate
Output: A UC estimate of var(S)
1: Let $y_j = \frac{(x_j - x_{j+\lceil n/2 \rceil})^2}{2}$ for $j \in [\lfloor n/2 \rfloor]$
2: Let $\tilde{\sigma} \leftarrow \text{median}(y_1, y_2, \ldots, y_{\lfloor n/2 \rfloor})$
3: Let $c \leftarrow D\left(\eta_0^{1/2} + \left(\eta_0 + \sqrt{\frac{\log T}{n}}\right)^{3/4}\right)$ //see (3)
4: **return** $\tilde{\sigma}/(1 - \min\{2\eta_0, c\})$

Algorithm 3 RUCB- MAB: A Robust Algorithm for MABs

Input: Upper bound σ_0 on reward variances
1: Initialization: Play each arm $i \in [K]$ once
2: **for** $t = K+1, K+2, \ldots, T$ **do**
3: $\quad \tilde{\mu}_{i,t} \leftarrow \text{RMEST}\,(R_i(t))$ //median
4: $\quad$ Play arm $I_t = \underset{i\in[K]}{\arg\max}\ \tilde{\mu}_{i,t} + \sqrt{\frac{\log t}{T_i(t)}}\, e\sigma_0$
5: **end for**

Algorithm 4 RUCB- TUNE: A Tuned Robust Algorithm for MABs

Input: Upper bound η_0 on corruption rate
1: Initialization: Play each arm $i \in [K]$ once
2: **for** $t = K+1, K+2, \ldots, T$ **do**
3: $\quad \tilde{\mu}_{i,t} \leftarrow \text{RMEST}\,(R_i(t))$ //median
4: $\quad \tilde{\sigma}_{i,t} \leftarrow \text{RVUCB}\,(R_i(t))$ //variance UCB
5: $\quad$ Play arm
$\quad I_t = \underset{i\in[K]}{\arg\max}\ \tilde{\mu}_{i,t} + \left[\eta_0 + \sqrt{\frac{\log t}{T_i(t)}}\right] e\tilde{\sigma}_{i,t}$
6: **end for**

4.5 RUCB-MAB: a minimax-optimal robust algorithm for MAB

For any arm $i \in [K]$ let $I_i(t) := \{\tau < t : I_\tau = i\}$ denote the set of past time steps when arm i was pulled, let $T_i(t) := |I_i(t)|$ denote the number of times the arm was pulled in the past, let $R_i(t) := \{r_\tau : \tau \in I_i(t)\}$ denote the (possibly corrupted) rewards that were received by this arm so far, and let $\tilde{\mu}_{i,t} := \text{median}(R_i(t))$ denote the median of these rewards.

The RUCB- MAB algorithm described in Algorithm 3 builds upon the classic UCB algorithm by Auer et al. (2002a). At every step it computes an upper confidence estimate of the mean of every arm $i \in [K]$ and pulls the arm with the highest estimate. However, it makes a two crucial changes to the classical estimate.

Whereas UCB uses the mean and a simple agnostic variance term to construct its upper confidence bound, RUCB- MAB uses the median, and a variance-aware estimate (notice the use of a variance upper bound σ_0 in the algorithm) to construct its upper confidence bound. This helps overcome the confounding effects of the adversarial rewards that may be present in the sets $R_i(t)$. We show that RUCB- MAB enjoys the following regret bound for Gaussian reward distributions.

Theorem 2 *When executed on a collection of K arms with Gaussian reward distributions $v_i \equiv \mathcal{N}(\mu_i, \sigma_i)$ with $\sigma_i \le \sigma_0$ and a stochastic adversary with a corruption rate $\eta \le \frac{\Delta_{\min}}{4e\sigma_0}$ and $|\zeta_t| \le B$, the* RUCB- MAB *algorithm ensures a gap-dependent regret bound*

$$\bar{R}_T(\text{RUCB- MAB}) \le C \sum_{i\neq i^*} \frac{\sigma_0^2 \ln T}{\Delta_i} + \eta \cdot (\mu^* + B)T,$$

as well as a gap-agnostic regret bound

$$\bar{R}_T(\text{RUCB- MAB}) \le C'\sqrt{KT \ln T} + \eta \cdot (\mu^* + B)T,$$

for constants C, C' clarified in the proof. Moreover, in the stochastic setting with no adversary, i.e., $\eta = 0$, we recover the following regret bounds

$$\bar{R}_T(\text{RUCB- MAB}) \leq C \sum_{i \neq i^*} \frac{\sigma_0^2 \ln T}{\Delta_i},$$

$$\bar{R}_T(\text{RUCB- MAB}) \leq C' \sqrt{KT \ln T}.$$

We note that for $\eta = 0$ we indeed recover minimax-optimal regret bounds for stochastic bandits. Also note that if $\eta = \Omega(1)$, Theorem 1 rules out sub-linear regret bounds for any algorithm and hence the linear regret offered by Theorem 2 is no surprise. However, it is also important to note that for small values of η such as $\eta \approx \frac{1}{T^a}$ for $a > 0$, which still allow as many as T^{1-a} number of the samples to be corrupted, RUCB- MAB actually gets sub-linear regret $T^{\max\{0.5, 1-a\}}$.

However, below we establish a much stronger, sub-linear uncorrupted regret guarantee for RUCB- MAB. This shows that RUCB- MAB is able to identify the best arm after sub-linearly many pulls and incur vanishing regret thereafter.

Theorem 3 *When executed on a collection of K arms with Gaussian reward distributions $\nu_i \equiv \mathcal{N}(\mu_i, \sigma_i)$ with $\sigma_i \leq \sigma_0$ and a stochastic adversary with a corruption rate $\eta \leq \frac{\Delta_{\min}}{4e\sigma_0}$, the* RUCB- MAB *algorithm ensures an uncorrupted regret bound*

$$\bar{R}^*_T(\text{RUCB- MAB}) \leq C' \sqrt{KT \ln T}.$$

Improving the upper bound on η Theorem 2 requires the corruption rate to be bounded as $\eta \leq \frac{\Delta_{\min}}{4e\sigma_i}$ which may be very stringent if $\Delta_{\min} = \min_{\Delta_i > 0} \Delta_i$ is very small. Although the need to assume such bounds on the corruption rate is very common in robust learning and robust statistics literature (Bhatia et al. 2015; Diakonikolas et al. 2018) and represents the *breakdown* point of the algorithm, we can improve this upper bound on η to a problem-independent, universal constant.

To do so, a standard sieve is applied by separating arms that satisfy $\Delta_i > 4e\eta\sigma_i$ (for which Theorem 2 itself applies) and those that do not (for which $\Delta_i \leq 4e\eta\sigma_0$). The total regret due to the second set of arms cannot exceed $4e\eta\sigma_0 T$. Bounding the regret separately for these arms gives us the following regret bound which puts a much milder requirement on η.

Corollary 1 *If initialized with $\sigma_0 = \max_i \sigma_i$ with the corruption rate satisfying $\eta \leq 1/4$,* RUCB- MAB *incurs a regret,*

$$\bar{R}_T(\text{RUCB- MAB}) \leq C(1-\eta)\sqrt{KT \ln T} + \eta \cdot (\mu^* + B)T + 4e\eta\sigma_0 T.$$

We note that the constraint $\eta < 1/4$ involves a universal constant and is required to satisfy the requirements for the results of Lai et al. (2016) to hold. Note that even this new regret bound becomes sub-linear if $\eta = o(1)$ such as $\eta = 1/\sqrt{T}$. We note that all the above results can be extended to several useful non-Gaussian, and indeed heavy-tailed distributions including those studied by Bubeck et al. (2013). This is because Lai et al. (2016, Theorem 1.3) show that the median estimator, with some modifications, is able to recover the mean faithfully for general distributions with bounded fourth moments.

4.6 RUCB-TUNE: robust tuned MABs

The RUCB- MAB algorithm assumes access to a uniform bound on the variances of the different arms. In their early work itself, Auer et al. (2002a) noticed that performing variance estimation can greatly boost the accuracies of the estimation procedure. This intuition was taken up by Audibert et al. (2007) who developed algorithms that automatically tune to the

variance of the arms. We present one such "tuned" algorithm for the MAB settings with adversarial corruptions.

The robust estimates are not as straightforward in this case, as most variance estimates available in literature are *relative* estimates whereas the UCB framework works primarily with estimates which incur bounded additive error. To handle this, we propose a novel variance upper confidence bound algorithm RVUCB based on a robust variance estimation technique by Lai et al. (2016).

The RVUCB estimator turns out to be crucial for the regret bound to be established. For sake of simplicity, we present the regret bound for Gaussian reward distributions but remind the reader that these results readily extend to several interesting families of non-Gaussian and heavy distributions with minor changes to the procedure. This is because the underlying result of Lai et al. (2016, Theorem 1.3) can be adapted to show that median-based mean and variance estimation techniques do work for non-Gaussian, heavy-tailed distributions too.

Theorem 4 *When executed on a collection of K arms with Gaussian reward distributions $\nu_i \equiv \mathcal{N}(\mu_i, \sigma_i)$ and a stochastic adversary with a corruption rate $\eta \leq \frac{\Delta_{\min}}{4e\sigma_i}$, the* RUCB- TUNE *algorithm, when executed with a setting $\eta_0 \geq \eta$, ensures a regret bound*

$$\bar{R}_T(\text{RUCB- TUNE}) \leq C(1-\eta)\sqrt{KT\ln T} + \eta_0 \cdot (\mu^* + B)T,$$

for a constant C clarified in the proof.

Note that RUCB- TUNE requires an estimate of an upper bound η_0 the corruption rate in order to operate. This can be done in practice via an (online) grid search. In our experiments, we did not find RUCB- TUNE to be sensitive to imprecise setting of η_0. As before, we can introduce two improvements: show a truly sub-linear uncorrupted regret bound for the RUCB- TUNE algorithm, and remove the constraint on the corruption rate $\eta \leq \frac{\Delta_{\min}}{4e\sigma_i}$, here as well.

Theorem 5 *When executed on a collection of K arms with Gaussian reward distributions $\nu_i \equiv \mathcal{N}(\mu_i, \sigma_i)$ and a stochastic adversary with a corruption rate $\eta \leq \frac{\Delta_{\min}}{4e\sigma_i}$, the* RUCB- TUNE *algorithm, when executed with a setting $\eta_0 \geq \eta$, ensures an uncorrupted regret bound*

$$\bar{R}^*_T(\text{RUCB- TUNE}) \leq C'\sqrt{KT\ln T}.$$

Corollary 2 *When executed on a collection of K arms with Gaussian reward distributions $\nu_i \equiv \mathcal{N}(\mu_i, \sigma_i)$ and a stochastic adversary with a corruption rate $\eta \leq 1/4$, the* RUCB- TUNE *algorithm, when executed with a setting $\eta_0 \geq \eta$, ensures a regret bound*

$$\bar{R}_T(\text{RUCB- TUNE}) \leq C(1-\eta)\sqrt{KT\ln T} + \eta_0 \cdot (\mu^* + B)T + 4e\eta\sigma_{\max}T,$$

where $\sigma_{\max} = \max_i \sigma_i$. Note that RUCB- TUNE *does not require knowledge of $\sigma_{\max}$.*

Before concluding, we note that RUCB- MAB and RUCB- TUNE can be made robust against stronger, adaptive adversaries, that can decide their corruptions based on the entire history of the play rather than independently of it, by replacing the simple median-based estimators with more detailed, convex optimization-based estimators of Diakonikolas et al. (2018, 2016). However, these algorithms, as well as their analyses are much more intricate, and we defer these to future work.

5 Robust linear contextual bandits

In this section, we discuss the linear contextual bandit problem under a much stronger adversary model and present the RUCB- LIN algorithm.

5.1 Problem setting

The stochastic linear contextual bandit framework (Abbasi-Yadkori et al. 2011; Li et al. 2010) extends to settings where every arm $\mathbf{a}$ is parametrized by a vector $\mathbf{a} \in \mathbb{R}^d$ (abusing notation). However, the set of all arms is potentially infinite, and moreover, not all arms may be available at every time step.

At each time step t, the player receives a set of n_t arms (called *contexts*) $A_t = \left\{\mathbf{x}^{t,1}, \ldots, \mathbf{x}^{t,n_t}\right\} \subset \mathbb{R}^d$. These are the only arms that can be pulled in this round. A good example from the advertising world is a limited number of items that are available for display at the moment the user arrives at the website. Items that are not available cannot be displayed to the user at that time instant. The set, as well as the number n_t of contexts available can vary from time step to time step. The player selects and pulls an arm $\hat{\mathbf{x}}^t \in A_t$ as per its arm selection policy. In response, a reward r_t is generated. Let $\mathcal{H}^t = \left\{A_1, \hat{\mathbf{x}}^1, r_1, \ldots, A_{t-1}, \hat{\mathbf{x}}^{t-1}, r_{t-1}, A_t, \hat{\mathbf{x}}^t\right\}$.

5.2 Adversary model

In the stochastic linear bandit setting, the reward is generated using a *model vector* $\mathbf{w}^* \in \mathbb{R}^d$ (that is unknown to the algorithm) as follows: $r_t = \left\langle \mathbf{w}^*, \hat{\mathbf{x}}^t \right\rangle + \epsilon_t$, where ϵ_t is a *noise* value that is typically assumed to be (conditionally) centered and σ-sub-Gaussian, i.e., $\mathbb{E}\left[\epsilon_t \mid \mathcal{H}^t\right] = 0$ (centering), as well as for some $\sigma > 0$, for any $\lambda > 0$, we have $\mathbb{E}\left[\exp(\lambda\epsilon_t) \mid \mathcal{H}^t\right] \leq \exp(\lambda^2\sigma^2/2)$ (sub-Gaussianity).

Here we consider an *adaptive adversary* that is able to view the on-goings of the online process and at any time instant t, *after* observing the history $\mathcal{H}^t$ and the "clean" reward value, i.e., $\left\langle \mathbf{w}^*, \hat{\mathbf{x}}^t \right\rangle + \epsilon_t$, able to add a corruption value b_t to the reward. For notational uniformity, we will assume that for time instants where the adversary chooses not to do anything, $b_t = 0$. Thus, the final reward to the player at every time step is $r_t = \left\langle \mathbf{w}^*, \hat{\mathbf{x}}^t \right\rangle + \epsilon_t + b_t$. For sake of simplicity we will assume that, for some $B > 0$, the final (possibly corrupted) reward presented to the player satisfies $r_t \in [-B, B]$ almost surely.

Note that this is a much more powerful adversary than the stochastic adversary we looked at earlier. This adversary is allowed to look at previous rewards and arm pulls, as well as the currently pulled arm and its clean reward before deciding if to corrupt and if so, by how much. There are no independence restrictions on this adversary. The only constraint we place is that at no point in the online process, should the adversary have corrupted more than an η fraction of the observed rewards. Formally, let $G_t = \{\tau < t : b_\tau = 0\}$ and $B_t = \{\tau < t : b_\tau \neq 0\}$ denote the "good" and "bad" time instances. We insist that $|B_t| \leq \eta \cdot t$ for all t.

Problem Setting 2 Adversarial Linear Bandits

for $t = 1, 2, 3..$ **do**
  Player receives a set of contexts $A_t = \left\{\mathbf{x}^{t,1}, \ldots, \mathbf{x}^{t,n_t}\right\} \subset \mathbb{R}^d$
  Player plays an arm, $\hat{\mathbf{x}}^t \in A_t$
  Clean reward is generated $r_t^* = \left\langle \mathbf{w}^*, \hat{\mathbf{x}}^t \right\rangle + \epsilon_t$ conditioned on $\mathcal{H}^t$
  Adversary chooses a corruption b_t after inspecting $\hat{\mathbf{x}}^t$, r_t^* and $\mathcal{H}^t$
  while making sure that $|\tau \leq t : b_\tau \neq 0| \leq \eta \cdot (t+1)$.
  Player receives reward, $r_t = r_t^* + b_t$
end for

5.3 Notion of regret

The goal of the algorithm is to maximize the cumulative reward it receives over the time steps $\sum_{t=1}^{T} r_t$. However, a more popular technique of casting this objective is in the form of *cumulative pseudo regret*. At time t, let $\mathbf{x}^{t,*} = \arg\max_{\mathbf{x} \in A_t} \langle \mathbf{w}^*, \mathbf{x} \rangle$ be the arm among the available contexts that yields the highest expected (uncorrupted) reward. The cumulative pseudo regret of a policy π is defined as follows

$$\bar{R}_T(\pi) = \sum_{t=1}^{T} \left\langle \mathbf{w}^*, \mathbf{x}^{t,*} \right\rangle - \mathbb{E}\left[r_t\right].$$

Note that unlike the MAB case, the best arm here may change across time-steps. For sake of simplicity, we assume that $\|\mathbf{w}^*\|_2 \leq 1$, and $\|\mathbf{x}\|_2 \leq 1$ almost surely for all $\mathbf{x} \in A_t$ for all t. We postpone introducing and analysing a notion of *uncorrupted regret*, as we did for multi-armed bandits, to future work.

Note that the regret lower bound in Theorem 1 applies to the linear bandit setting as well due to a reduction of the MAB problem to the linear bandit problem (let $d = K$ where K is the number of arms in the MAB problem, $\mathbf{w}_i^* = \mu_i$ and contexts $A_t \subseteq \{\mathbf{e}_1, \ldots, \mathbf{e}_d\}$ where $\mathbf{e}_i$ are canonical vectors). Thus, any policy for linear bandits under an adversary must incur regret at least $\Omega\left(\eta \cdot T\right)$ which rules out sub-linear regret bounds for robust linear bandits if $\eta = \Omega\left(1\right)$.

5.4 RUCB-LIN: a robust algorithm for linear contextual bandits

We use the notation $\|\mathbf{x}\|_M = \sqrt{\mathbf{x}^\top M \mathbf{x}}$ for a vector $\mathbf{x} \in \mathbb{R}^d$ and a matrix $M \in \mathbb{R}^{d \times d}$. The RUCB- LIN algorithm is described in Algorithm 5 and builds upon the OFUL algorithm (Abbasi-Yadkori et al. 2011) for linear contextual bandits. At every step, the algorithm performs an estimation $\mathbf{w}^t$ of the true model vector $\mathbf{w}^*$, as well as creates a *confidence set* to explicate the region of uncertainty. At prediction time, it uses the *Optimism in the Face of Uncertainty* principle to select an arm to pull.

However, unlike OFUL that uses a simple ridge regression estimator for $\mathbf{w}^t$ and a direct ellipsoidal confidence set constructed using all arms pulled so far, RUCB- LIN needs to do a much more refined job. Neither can it use a simple estimator due to the adaptive adversarial

Algorithm 5 RUCB- LIN: A Robust Algorithm for Linear Contextual Bandits

Input: Upper bound σ_0 on the noise sub-Gaussian parameter, upper bound η_0 on the corruption rate, tolerance ϵ, horizon T

for $t = 1, 2, \ldots, T$ **do**
  Receive set of arms A_t
  Play arm $\hat{\mathbf{x}}^t = \arg\max_{\mathbf{x} \in A_t, \mathbf{w} \in C_{t-1}} \langle \mathbf{x}, \mathbf{w} \rangle$
  Receive reward r_t
  $\hat{\mathbf{w}}^t \leftarrow \text{TORRENT}\left(\left\{\hat{\mathbf{x}}^i, r_i\right\}_{[t]}, \eta_0, \epsilon\right)$
  $\hat{G}_t \leftarrow \left\{\tau \leq t : \left|r_\tau - \left\langle \hat{\mathbf{w}}^t, \mathbf{x}^\tau \right\rangle\right| \leq \sigma_0 \log T\right\}$
  $M_t \leftarrow \sum_{\tau \in \hat{G}_t} \mathbf{x}^\tau (\mathbf{x}^\tau)^\top$
  $\bar{\mathbf{w}}^t \leftarrow M_t^{-1} X_{\hat{G}_t} \mathbf{y}_{\hat{G}_t}$
  $C_t \leftarrow \{\mathbf{w} : \left\|\mathbf{w} - \bar{\mathbf{w}}^t\right\|_{M_t} \leq \sigma_0 \sqrt{d \log T} + \eta B T\}$
end for

Algorithm 6 The TORRENT algorithm for Robust Regression (Bhatia et al. 2015)

Input: Training data $\{\mathbf{x}_i, y_i\}, i = 1 \ldots n$, thresholding parameter η_0, tolerance ϵ

1: $\mathbf{w}^0 \leftarrow \mathbf{0}$, $S_0 = [n]$, $t \leftarrow 0$, $\mathbf{r}^0 \leftarrow \mathbf{y}$

2: **while** $\left\|\mathbf{r}^t_{S_t}\right\|_2 > \epsilon$ **do**

3: $\quad \mathbf{w}^{t+1} \leftarrow \arg\min_{\mathbf{w}} \sum_{i \in S_t} (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle)^2$

4: $\quad \mathbf{r}^{t+1}_i \leftarrow \left(y_i - \left\langle \mathbf{w}^{t+1}, \mathbf{x}_i \right\rangle\right)$

5: $\quad$ // BOT chooses the $(1 - \eta_0)n$ points with the smallest residual $\left|\mathbf{r}^{t+1}_i\right|$

$\quad S_{t+1} \leftarrow \text{BOT}(\mathbf{r}^{t+1}, (1 - \eta_0)n)$

6: $\quad t \leftarrow t + 1$

7: **end while**

8: **return** $\mathbf{w}^t$

corruptions, nor can it use all arms pulled so far in its confidence ball creation. We describe how to overcome these challenges below.

For model estimation, we chose the TORRENT algorithm of Bhatia et al. (2015). Even though there are several approaches to robust regression (Chen et al. 2013; Nguyen and Tran 2013), we chose TORRENT since it is simple to implement yet offers guarantees against an adaptive adversary. This method requires a technical condition called *subset regularity* to be satisfied which we will address shortly.

Given the model estimate, RUCB- LIN performs a pruning step and constructs a confidence set, which, as we shall see, has a noise removal effect. It lets in previously pulled arms whose rewards were not corrupted but stops those which experienced severe corruptions. We note that step 8 in Algorithm 6, although inexpensive, was not found to greatly affect the performance of the algorithm. However, including this step makes our analysis much more convenient.

RUCB- LIN is extremely simple to implement and scales to large problems with ease. Extensions of RUCB- LIN to high dimensional settings where the model $\mathbf{w}^*$ is sparse are possible by using high-dimensional variants of TORRENT. However, we postpone these to future work. Before presenting the regret analysis, we first address the *subset regularity* condition required by TORRENT.

5.5 Data hardness

Given the powerful adaptive adversary model in our setting, it would not be possible to make much headway unless we have some niceness in the problem structure given to us. More specifically, if the set of arms A_t that are supplied to us at each step is skewed (for instance, if they are chosen by the adversary as well), then we cannot hope to do much. To prevent this, we require the set of contexts to satisfy some regularity conditions. We note that there exist past works in linear bandit settings, such as those of Gentile et al. (2014, 2017), that do place restrictions on the context sets. The following notion of subset regularity succinctly captures the notion of a well-conditioned set of arms being presented during the course of the play. In the following, for $n > 0$, $\gamma \in (0, 1]$, let $\mathcal{S}_\gamma = \{S \subset [n] : |S| = \gamma \cdot n\}$ denote the set of all subsets of S of size $\gamma \cdot n$.

Definition 1 (*SSC and SSS properties* Bhatia et al. 2015) A matrix $X \in \mathbb{R}^{d \times n}$ satisfies the *Subset Strong Convexity Property* (resp. *Subset Strong Smoothness Property*) at level γ with strong convexity constant λ (resp. strong smoothness constant Λ) if we have:

$$\lambda \leq \min_{S \in \mathcal{S}_\gamma} \lambda_{\min}(X_S X_S^\top) \leq \max_{S \in \mathcal{S}_\gamma} \lambda_{\max}(X_S X_S^\top) \leq \Lambda.$$

Definition 2 (*Subset regularity*) A sequence of context sets $A_1, A_2, \ldots, A_T$ satisfies the $(\eta, \{\lambda_t\}_{[T]}, \{\Lambda_t\}_{[T]}, T_0)$ subset regularity property if for some $T_0 > 0$, for every $t \geq T_0$, and every possible choice of $\mathbf{x}^\tau \in A_\tau$ for $\tau = 1, \ldots, t$, the matrix $[\mathbf{x}^1 \mathbf{x}^2 \ldots \mathbf{x}^t] \in \mathbb{R}^{d \times t}$ satisfies the SSC and SSS properties at level η with constants λ_t and Λ_t respectively.

Note that the $(1-\eta, \{\lambda_t\}_{[T]}, \{\Lambda_t\}_{[T]}, T_0)$ subset regularity property helps ensure that after enough, i.e., T_0 iterations have passed, at every time step $t \geq T_0$, no matter which arms we have chosen till now, and no matter which of those arms have had their responses corrupted by the adversary (so long as only an η fraction of the total number of arms pulled till now have been corrupted), the matrix of arm vectors whose responses were not corrupted has bounded eigenvalues. Such a property is immensely helpful in performing robust regression in the face of an adaptive adversary. As Bhatia et al. comment, such a condition is in some sense necessary if there is no restriction on which arms the adversary may corrupt. Recall that the stochastic adversary in the previous section had less power as the arms to corrupt were decided on the basis of a Bernoulli trial.

Satisfying subset regularity It might be worrisome as to how a property such as Subset Regularity may be satisfied. However, it turns out that if the arm sets A_t are generated i.i.d. (conditioned on the history) from some sub-Gaussian distribution over $\mathbb{R}^d$ then the property is satisfied with high probability for a value T_0 that has only poly-logarithmic dependence on T. To avoid notational clutter we show this result below for the case when contexts are drawn from the standard multivariate Gaussian distribution but stress that similar results do hold for all sub-Gaussian distributions as well. Indeed, the reader may refer to the work of Bhatia et al. (2015) for proofs of such results in the batch setting which can be extended to the online setting using the technique used to prove Lemma 1.

Lemma 1 *For any $\eta > 0$, and each round t, suppose the context vectors $A_t = \left\{\mathbf{x}^{t,1}, \ldots, \mathbf{x}^{t,n_t}\right\}$ are generated i.i.d. (conditioned on n_t and past history $\mathcal{H}^t$) from the standard multivariate normal distribution $\mathcal{N}(\mathbf{0}, I_{d\times d})$. Let $n_t = \mathcal{O}(1)$ for all t. Then with probability at least $1-\delta$, the sequence $A_1, A_2, \ldots, A_T$ satisfies the $(\eta, \{\lambda_t\}_{[T]}, \{\Lambda_t\}_{[T]}, T_0)$ subset regularity property with $T_0 \geq \mathcal{O}\left(\log^2\left(\frac{Td}{\delta}\right)\right)$. Moreover, with the same confidence, we have $\lambda_t \geq t/4 - \mathcal{O}\left(\log(T/\delta) + \sqrt{T\log(T/\delta)}\right)$, as well as $\Lambda_t \leq t/4 + \mathcal{O}\left(\log(T/\delta) + \sqrt{T\log(T/\delta)}\right)$.*

We are now ready to prove the regret bound for RUCB- LIN. The proof hinges on a crucial confidence ellipsoid result which does not follow directly from existing works, e.g., that of Abbasi-Yadkori et al. (2011), since existing works never have to selectively throw away points due to them being corrupted. Since RUCB- LIN does perform such a pruning step, we have to prove this result afresh.

Theorem 6 *For any $\delta, \eta > 0$, if the sequence of context sets is generated such that it satisfies the two subset regularity properties $(\eta, \{\lambda_t\}_{[T]}, \{\Lambda_t\}_{[T]}, T_0)$ and $(1-\eta, \left\{\tilde{\lambda}_t\right\}_{[T]}, \left\{\tilde{\Lambda}_t\right\}_{[T]}, T_0)$ such that $\frac{\Lambda_t}{\tilde{\lambda}_t} \leq \frac{1}{16}$ for all $t \geq T_0$, then for all $t \geq T_0$,*

$$\left\|\mathbf{w}^* - \bar{\mathbf{w}}^t\right\|_{M_t} \leq \sigma_0\sqrt{d\log T} + \eta B \cdot T,$$

where M_t is obtained after the pruning step (see Algorithm 5 Steps 6-9).

The above result at first glance seems weaker than that for OFUL by Abbasi- Ypadkori et al. (2011, Theorem 2) that offers a radius logarithmic in the horizon $\sqrt{d\log T}$ whereas

Theorem 6 offers $\sqrt{d \log T} + \eta \cdot T$. This is no accident and simply another confession that even an algorithm that does have complete knowledge of the model $\mathbf{w}^*$, cannot achieve sub-linear regret, given the regret lower bound.

Theorem 6 gives a formal reasoning for this. Since corruptions abound, RUCB- LIN can never decrease the size of its confidence ball for fear of excluding $\mathbf{w}^*$. However, notice that for small values of $\eta \approx 1/\sqrt{T}$, the radius of the ball used in Theorem 6 does shrink to $\sqrt{d \log T} + \eta \cdot \sqrt{T}$, while still allowing $\sqrt{T}$ corruptions. We now state a regret bound for RUCB- LIN.

Theorem 7 *If the sequence of context sets is generated (conditionally) such that it satisfies the* $(\eta, \{\lambda_t\}_{[T]}, \{\Lambda_t\}_{[T]}, T_0)$ *and* $(1-\eta, \left\{\tilde{\lambda}_t\right\}_{[T]}, \left\{\tilde{\Lambda}_t\right\}_{[T]}, T_0)$ *subset regularity properties such that* $\frac{\Lambda_t}{\tilde{\lambda}_t} \leq \frac{1}{16}$ *for all* $t \geq T_0$*, then* RUCB- LIN *ensures*

$$\mathbb{E}\left[\bar{R}_T(\text{RUCB- LIN})\right] \leq C \cdot d\sqrt{T \log T} + \eta B \cdot T,$$

for a constant C clarified in the proof. Moreover, in the stochastic setting with no adversary, i.e., $\eta = 0$*,* RUCB- LIN *ensures* $\mathbb{E}\left[\bar{R}_T(\text{RUCB- LIN})\right] \leq C \cdot d\sqrt{T \log T}$.

Breakdown point analysis If we are generating arms from a standard Gaussian distribution, then $\frac{\Lambda_t}{\tilde{\lambda}_t} \leq \frac{1}{16}$ can be ensured, for instance, when $\eta < \frac{1}{100}$ (Bhatia et al. 2015). Also note that for small values of η such as $\eta \approx \frac{1}{T^a}$ for $a > 0$, which still allow as many as T^{1-a} number of the samples to be corrupted, RUCB- LIN actually gets sub-linear regret $T^{\max\{0.5, 1-a\}}$. We note that we have not attempted to optimize constants such as 1/100 in the above result. In practice, we find RUCB- LIN to be able to tolerate very well upto 10–15% of arm pulls being corrupted.

6 Experiments

We discuss the experimental design and results for RUCB- MAB/RUCB- TUNE and RUCB- LIN here. The experiments show that these algorithms are robust to corruptions and significantly outperform other UCB-style algorithms.[1]

6.1 Robust multi-armed bandit experiments

We compare the empirical performance of RUCB- MAB and RUCB- TUNE against several algorithms for stochastic, adversarial, and "best-of-both-world" bandits.

Data For each arm i, the arm means were sampled as $\mu_i \sim \mathcal{U}(0, 1)$ and the arm variances as $\sigma_i \sim \mathcal{U}(0, 1)$. The arm rewards were sampled for each arm from $\mathcal{N}(\mu_i, \sigma_i)$. Experiments were run with the number of arms set to 100 and 10, and for 1100 and 11,000 iterations respectively.

Adversary The corruptions were generated by conducting Bernoulli trials with bias η. If given a chance to corrupt an arm, our adversary offered a zero reward if the selected arm was the best arm and a corrupted reward of $\frac{s}{\eta}$ if the selected arm was not the best arm. We used $s = 0.04$ to prevent the adversary from rewarding the bad arms too much and hence violating the goodness order of the arms. We note that while other adversary models are

[1] Code and datasets for our experiments are available at https://github.com/purushottamkar/rUCB.

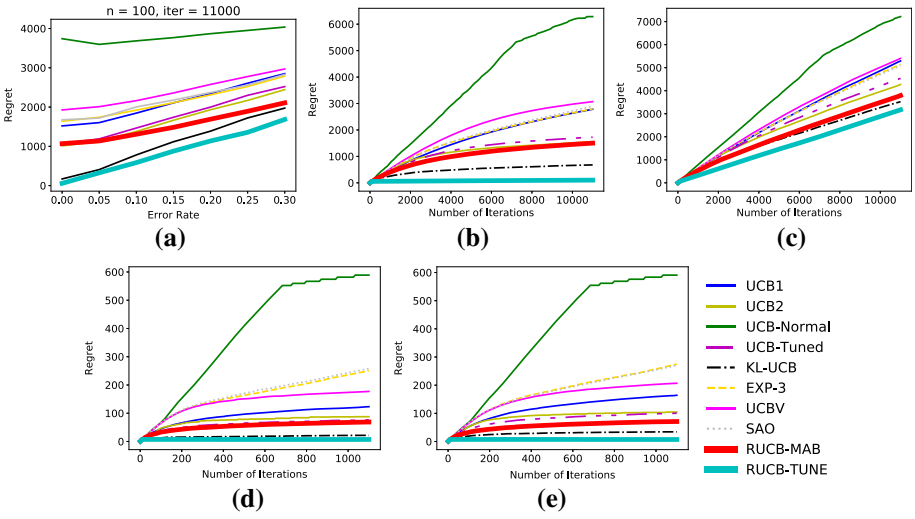

Fig. 1 Variation in regret $\bar{R}_T$ for various algorithms across time T and error η. **a** $\bar{R}_T$ versus η; $K = 100$. **b** $\bar{R}_T$ versus T; $\eta = 0.1$ $K = 100$. **c** $\bar{R}_T$ versus T; $\eta = 0.3$ $K = 100$. **d** $\bar{R}_T$ versus T; $\eta = 0$ $K = 10$. **e** $\bar{R}_T$ versus T; $\eta = 0.1$ $K = 10$

indeed possible, we believe the adversary model used here does not unfairly benefit any particular algorithm.

Algorithms We tested RUCB- MAB and RUCB- TUNE against a large number of Upper Confidence Bound algorithms popular in literature including KL-UCB (Garivier and Cappé 2011), UCB1, UCB2, UCB-Normal, UCB-Tuned (Auer et al. 2002a) and UCB-V (Audibert et al. 2009). The last three algorithms estimate the variance of the arms, while UCB-Normal is an algorithm specially designed for cases when the reward distributions are normal. We tuned the value of the α parameter in UCB2 as suggested by Auer et al. (2002a) and found $\alpha = 0.14$ to work well. We also run tests against the EXP3 and SAO algorithms (Bubeck and Slivkins 2012) which offer regret bounds in adversarial and best-of-both-world settings. We set a default value of $\sigma_0 = 1$ as the upper bound on standard deviations for RUCB- MAB.[2] For EXP3 we tuned the γ value and found it to be optimal at about 0.2. The variant of UCB-V used was taken from the original work of Audibert et al. (2007), with the constants and exploration function as suggested by the authors. For finding the median in an online fashion, we used a two heaps, which allowed us to get $\mathcal{O}(\log n)$ time complexity for finding the median at each time step. This made the algorithm very efficient for extensive experiments.

Evaluation metric We compare the regret $\bar{R}_T$ and uncorrupted regret $\bar{R}^*_T$ for all algorithms. All results are averaged over 50 repetitions of the same experiment.

Results The results are shown in Figs. 1 and 2. We observe that while RUCB- MAB performs poorly when compared to UCB2 and UCB-Tuned for low values of error rate, it quickly overtakes them with an increase in error rate. On the other hand, RUCB- TUNE enjoys much lower regret than all other algorithms as the number of iterations and the corruption rate increase. However, for the zero corruption case, the performance is very closely followed by KL-UCB. We credit this result to the fact that the exploration term estimates are typically lower for RUCB- TUNE which reduces performance for such small number of arms. For the case of uncorrupted regret, results are similar. As evident in both the graphs, the slope of

[2] This value can be further improved by tuning the parameter.

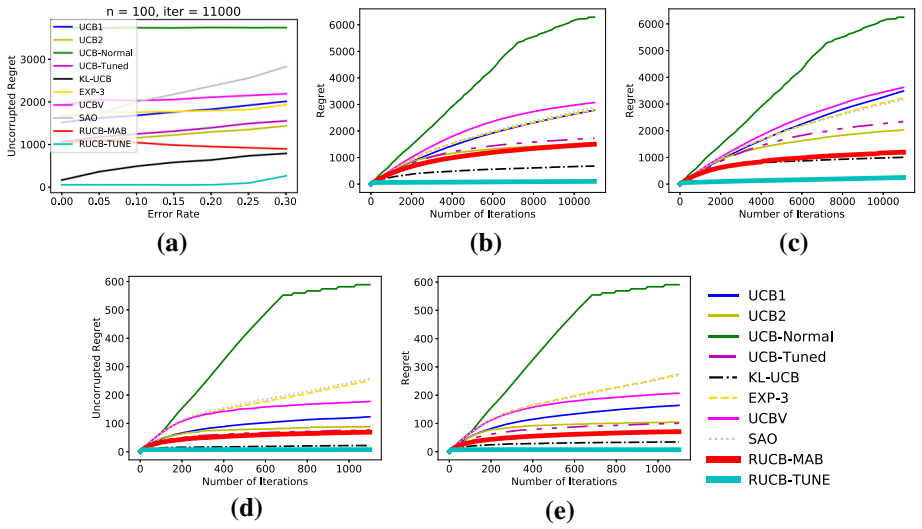

Fig. 2 Variation in uncorrupted regret $\bar{R}_T^*$ for various algorithms across time T and error η. **a** $\bar{R}_T^*$ versus η; $K = 100$. **b** $\bar{R}_T^*$ versus T; $\eta = 0.1$ $K = 100$. **c** $\bar{R}_T^*$ versus T; $\eta = 0.3$ $K = 100$. **d** $\bar{R}_T^*$ versus T; $\eta = 0$ $K = 10$. **e** $\bar{R}_T^*$ versus T; $\eta = 0.1$ $K = 10$

regret versus iterations (or regret vs. the corruption rate) decreases as we plot the uncorrupted rewards.

It is interesting to note that we outperform EXP3 and SAO in this setting, since neither is able to reconcile the fact that not all, but only a fraction of arms are corrupted by the adversary, and end up choosing arms as though every pull were corrupted. Variance estimating algorithms (UCB-Normal, UCB-Tuned, UCB-V, RUCB- TUNE) perform better than those that don't estimate variance. Overall, it seems that RUCB- MAB, RUCB- TUNE work well even for high corruption rates with hundreds of arms, which is a setting of interest.

6.2 Robust linear contextual bandit experiments: comparison with LINUCB

We also compare the empirical performance of RUCB- LIN with LINUCB across error rates, the dimension of the context vectors, and the magnitude of corruption.

Data The true model vector $\mathbf{w}^* \in \mathbb{R}^d$ was chosen to be a random unit norm vector with $d = 10$. The arms at each time-step were sampled as $\mathbf{x}^{t,i} \sim \mathcal{N}(0, I_d)$, and the reward for the selected arm was generated as $y_i = \langle \mathbf{w}^*, \mathbf{x}_i \rangle + \epsilon_i$ where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. All experiments used $n_t = 50$ arms being generated afresh at each time step, a corruption rate of $\eta = 0.1$, $d = 10$, and the scale of the corruptions to be $c_t = 10$, unless stated otherwise. All results reported are averaged over 50 repetitions of the same experiment.

Adversary The corruptions were generated as $b_t = -r_t^* - c_t \cdot \langle \mathbf{w}^*, \mathbf{x}^{t,*} \rangle$, where $\mathbf{x}^{t,*}$ is the best possible arm and c_t is the magnitude of corruption. We note that while other adversary models are indeed possible, we believe the adversary model used here does not unfairly benefit any particular algorithm.

Algorithms We compared RUCB- LIN to LINUCB (Abbasi-Yadkori et al. 2011) and used the TORRENT-FC implementation by Bhatia et al. (2015).

Evaluation metric We measured regret $\bar{R}_T$ and uncorrupted regret $\bar{R}_T^*$ over 1000 iterations.

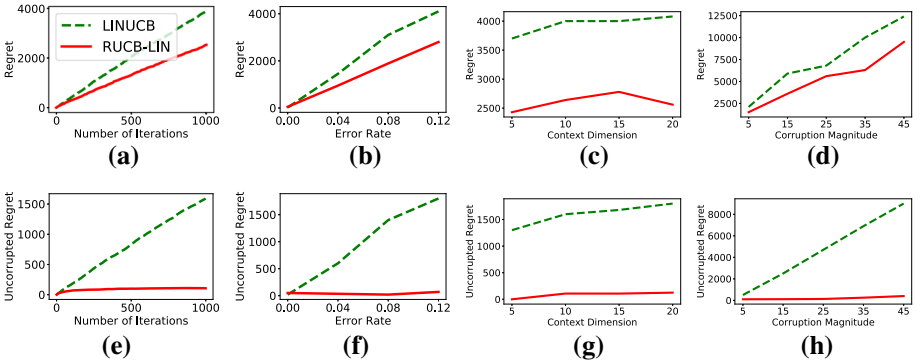

Fig. 3 Variation of regret $\bar{R}_T$ and uncorrupted regret $\bar{R}^*_T$ incurred by RUCB- LIN and LINUCB against time T, error rate η, dimension of the context vector d and the magnitude of corruption introduced c_t. Note that while LINUCB has a slight edge over RUCB- LIN when the error $\eta = 0$, RUCB- LIN overtakes LINUCB by a large margin in presence of adversarial corruption. **a** $\bar{R}_T$ versus T. **b** $\bar{R}_T$ versus η. **c** $\bar{R}_T$ versus d. **d** $\bar{R}_T$ versus c_t. **e** $\bar{R}^*_T$ versus T. **f** $\bar{R}^*_T$ versus η. **g** $\bar{R}^*_T$ versus d. **h** $\bar{R}^*_T$ versus c_t

Results Figure 3 shows that RUCB- LIN incurs much lower regret as compared to LINUCB as the corruption rate increases. While LINUCB has a slight edge in the case without corruptions, it quickly starts losing out to RUCB- LIN when the error rate increases. A more interesting result is in the case of uncorrupted regret. From the graph of uncorrupted regret plotted against time we can see the true gains RUCB- LIN has over LINUCB. While the LINUCB algorithm continues to incur linearly increasing uncorrupted regret with time, RUCB- LIN eventually converges to the best model vector. The ability of RUCB- LIN to retrospectively mark points as corrupted allows it to make increasingly better decisions as the number of iterations increases, since it can identify the correct model vector. LINUCB is not able to determine the correct model vector.

6.3 Robust linear bandit experiments: comparison with heavy-tailed methods

In this section, we compare empirical performance of RUCB- LIN with the algorithms for heavy-tailed bandits proposed by Medina and Yang (2016):

- CR- TRUNC- 1 represents the *Confidence-Region* algorithm of Medina and Yang (2016) (Algorithm 1 therein) with the *Truncation* estimator defined in the paper, and parameter $\alpha_t = \sqrt{t}$. We found no significant improvement in performance of CR- TRUNC- 1 even upon carefully tuning the exponent of t in α_t.
- CR- TRUNC- 2 represents our alternate implementation of the same algorithm which offers better empirical performance. While CR- TRUNC- 1 has truncation levels that increase with time as $\mathcal{O}\left(\sqrt{t}\right)$, for CR- TRUNC- 2 we fix the truncation levels to be a constant value $\alpha = 20$, set equal to the largest magnitudes any uncorrupted reward could take. This amounts to giving CR- TRUNC- 2 an unfair advantage by revealing to it the optimal truncation level.
- CR- MOM represents the *Mini-Batch Confidence Region* algorithm of Medina and Yang (2016) (Algorithm 3 therein) which uses the median of means estimator defined in the paper. We run this algorithm with $\delta = 0.1$ and $r = 10 \approx T^{1/3}$.

Executing the CR- MOM algorithm requires a modification to the experimental setup. Recall that our algorithms are presented with a set of available arms (contexts) at each step

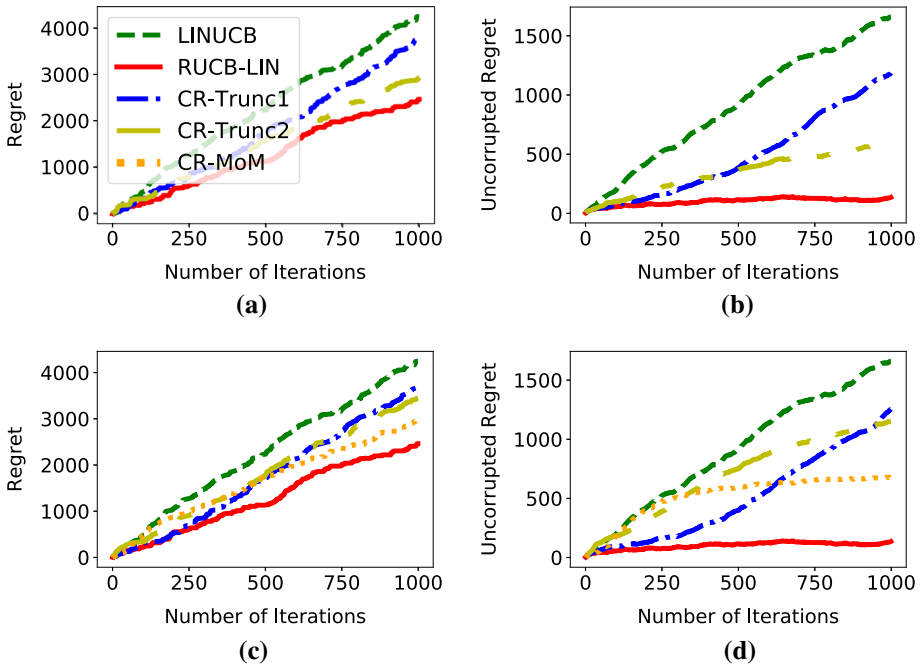

Fig. 4 Time evolution of regret $\bar{R}_T$ and uncorrupted regret $\bar{R}^*_T$ of RUCB- LIN, LINUCB, CR- TRUNC- 1, CR- TRUNC- 2 and CR- MOM. Note that while **a**, **b** are run for the original experimental setting defined in the text, **c**, **d** require a slightly different experimental setting to allow for a comparison with CR- MOM (see Sect. 6.3) for details. **a** $\bar{R}_T$ versus T, dynamic contexts. **b** $\bar{R}^*_T$ versus T, dynamic contexts. **c** $\bar{R}_T$ versus T, static contexts. **d** $\bar{R}^*_T$ versus T, static contexts

and only those arms can be pulled. However, the CR- MOM algorithm likes to pull the same arm repeatedly, in order to take the median of means of the observed pulls. To satisfy this need, we ensured that the context set stayed constant at all time steps, i.e., the same set of arms was available for pulls at all steps which allowed CR- MOM repeated pulls of the same arm. Thus, whereas the experimental setup remains the same as Sect. 6.2 for Fig. 4a, b, the change for Fig. 4c, d in that we do not change the set of arms at each time-step, with the rest of the experiment setting same as Sect. 6.2.

Results In Fig. 4a, b, we observe that RUCB- LIN maintains its lead. Both CR- TRUNC- 1 and CR- TRUNC- 2 are unable to discern the true model vector (as evidenced by their uncorrupted regret $\bar{R}^*_T$ increasing linearly with time). Figure 4c, d similarly showcase RUCB- LIN maintaining its lead. However, given enough iterations, CR- MOM is able to recover the true model vector, despite performing poorly in the cold-start region. This is because CR- MOM needs to collect repeated pulls of arms in order to get discern the true rewards from the corrupted rewards set by the adversary. This leads to poor performance in the beginning, but it does eventually converge to the true model vector.

7 Discussion and future work

In this work, we reported three algorithms – RUCB- MAB, RUCB- TUNE and RUCB- LIN to address the task of corruption-tolerant bandit learning in the multi-armed and linear-contextual settings. All our algorithms are extremely scalable and easy to implement and

enjoy crisp and tight regret bounds, as well as superior performance to a wide range of competitor methods in experiments.

Using more powerful estimators, e.g., those by Diakonikolas et al. (2016, 2018) within RUCB- MAB and RUCB- TUNE should offer stronger results, albeit at the cost of making the algorithms more expensive. Extending the analysis for RUCB- MAB to non-Gaussian distributions and deriving high probability regret bounds [as Lykouris et al. (2018) do] would be interesting. For RUCB- LIN, extending the algorithm to high-dimensional settings as well as deriving sub-linear uncorrupted regret bounds by making additional assumptions on the corruption rate η (as we did in Theorem 3 for RUCB- MAB) would be useful.

From an applications standpoint, it is of interest to apply RUCB- MAB and RUCB- LIN to recommendation settings. As our experiments indicate, these algorithms tend to outperform existing methods not only when corruptions abound, but also in when there is no adversary present. This may put RUCB- LIN in an advantageous position wherein it is able to neglect non-adversarial variations in user behavior to capture the core user profile. The applications to settings where we suspect click-fraud or other malicious behavior are of course, immediate.

Acknowledgements The authors would like to thank the reviewers and editors for pointing out several relevant works, as well as helping improve the presentation of the paper. S.K. is supported by the National Talent Search Scheme under the National Council of Education, Research and Training (Ref. No. 41/X/2013-NTS). K.K.P. thanks Honda Motor India Pvt. Ltd. for an award under the 2017 Y-E-S Award program. P.K. is supported by the Deep Singh and Daljeet Kaur Faculty Fellowship and the Research-I foundation at IIT Kanpur, and thanks Microsoft Research India and Tower Research for research grants.

A Proofs from Sect. 4

Proof of Theorem 1 Fix a policy π and let the reward distributions be Gaussians with unit variance $\mathbf{u}_i = \mathcal{N}(\mu_i, 1)$. Let $\Delta > 0$ be a constant to be determined later. Given a constant $c \in (0, 1)$, consider two settings, one where the vector of the arm means is $\boldsymbol{\mu} = \{c + \Delta, c, c, \ldots, c\} \in \mathbb{R}^K$ for the K arms and the other where the arm means are $\boldsymbol{\mu}' = \boldsymbol{\mu} + 2\Delta \cdot \mathbf{e}_j$ where $\mathbf{e}_j = (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{R}^K$ is the jth canonical vector. The coordinate j will be decided momentarily.

Clearly, in the first setting, the first arm is the best and in the second setting the jth arm is the best. In both settings, the adversary acts simply by assigning a (corrupted) reward of 0 whenever it gets a chance to corrupt an arm pull. Clearly such an adversary is a stochastic adversary.

Let $T_i(T, \pi)$ denote the number of times the player obeying a policy π pulls the ith arm in a sequence of T trials. Also, for any $\boldsymbol{\mu} \in \mathbb{R}^K$, policy π and $T > 0$, define $\mathbb{P}_{\boldsymbol{\mu},\pi,\eta,T}$ to be the distribution induced on the history $\mathcal{H}^T$ by the action of policy π on the arms with mean rewards as given by the vector $\boldsymbol{\mu}$ and the adversary described above with corruption rate η (a cleaner construction of the distribution $\mathbb{P}_{\boldsymbol{\mu},\pi,\eta,T}$ is possible by properly defining filtrations but we avoid that to keep the discussion focused).

Also let $\mathbb{E}_{\boldsymbol{\mu},\pi,\eta,T}$ denote expectations taken with respect to $\mathbb{P}_{\boldsymbol{\mu},\pi,\eta,T}$ and let $\bar{R}_T(\pi, \boldsymbol{\mu}, \eta)$ denote the expected regret with respect to the same. Also define

$$j := \arg\min_{i \neq 1} \mathbb{E}_{\boldsymbol{\mu},\pi,\eta,T}[T_i(T, \pi)],$$

and use this to define $\boldsymbol{\mu}' = \boldsymbol{\mu} + 2\Delta \cdot \mathbf{e}_j$. Note that j is taken to be the suboptimal arm in the first setting least likely to be played by the policy π when interacting with the arms with means $\boldsymbol{\mu}$ and the adversary. Given the above, it is easy to see that since

$$\bar{R}_T(\pi, \boldsymbol{\mu}, \eta) = \Delta \cdot \sum_{i=2}^{K} \mathbb{E}_{\boldsymbol{\mu},\pi,\eta,T}[T_i(T,\pi)] + c\eta \cdot T,$$

we have

$$\begin{aligned} \bar{R}_T(\pi, \boldsymbol{\mu}, \eta) &\geq \mathbb{P}_{\boldsymbol{\mu},\pi,\eta,T}[T_1(T,\pi) \leq T/2] \cdot \frac{T\Delta}{2} + c\eta \cdot T \\ \bar{R}_T(\pi, \boldsymbol{\mu}', \eta) &\geq \mathbb{P}_{\boldsymbol{\mu}',\pi,\eta,T}[T_1(T,\pi) > T/2] \cdot \frac{T\Delta}{2} + c\eta \cdot T \end{aligned} \tag{1}$$

We now apply the Pinkser's inequality (Tsybakov 2009)[Lemma 2.6] to get

$$\mathbb{P}_{\boldsymbol{\mu},\pi,\eta,T}\left[T_1(T,\pi) \leq \frac{T}{2}\right] + \mathbb{P}_{\boldsymbol{\mu}',\pi,\eta,T}\left[T_1(T,\pi) > \frac{T}{2}\right] \geq \exp\left[-KL(\mathbb{P}_{\boldsymbol{\mu},\pi,\eta,T} || \mathbb{P}_{\boldsymbol{\mu}',\pi,\eta,T})\right],$$

where KL stands for the Kullback-Leibler divergence. Now, applying straightforward manipulations we can get

$$KL(\mathbb{P}_{\boldsymbol{\mu},\pi,\eta,T} || \mathbb{P}_{\boldsymbol{\mu}',\pi,\eta,T}) = \mathbb{E}_{\boldsymbol{\mu},\pi,\eta,T}[T_j(T,\pi)] \cdot KL(\mathcal{N}(\mu_j, 1), \mathcal{N}(\mu_j', 1)).$$

Now, using the fact that $KL(\mathcal{N}(c,1), \mathcal{N}(c+\Delta, 1)) = 2\Delta^2$, applying an averaging argument to get $\mathbb{E}_{\boldsymbol{\mu},\pi,\eta,T}[T_i(T,\pi)] \geq \frac{T}{K-1}$, setting $\Delta = \sqrt{(K-1)/4T}$, and using the sum of the two inequalities in (1) shows that

$$\bar{R}_T(\pi, \boldsymbol{\mu}, \eta) + \bar{R}_T(\pi, \boldsymbol{\mu}', \eta) \geq \frac{2}{27}\sqrt{(K-1)T} + 2c\eta \cdot T$$

which, by an application of another averaging argument, tells us that for at least one setting $\tilde{\boldsymbol{\mu}} \in \{\boldsymbol{\mu}, \boldsymbol{\mu}'\}$, we must have

$$\bar{R}_T(\pi, \tilde{\boldsymbol{\mu}}, \eta) \geq \frac{1}{27}\sqrt{(K-1)T} + c\eta \cdot T,$$

which finishes the proof. □

Proof of Theorem 2 First of all, note that step 4 in Algorithm 3 can be seen as executing the strategy

$$I_t = \arg\max_{i \in [K]} \tilde{\mu}_{i,t} + \left(\eta + \sqrt{\frac{\log t}{T_i(t)}}\right) e\sigma_0$$

The only difference between the above expression and the one used by Algorithm 3 is an additive term $e\eta\sigma_0$ which does not change the output of the arg max operation. We next note that the corruption model considered by Lai et al. (2016) is exactly the stochastic corruption model. Next, we note that in the uni-dimensional case, the AGNOSTICMEAN algorithm presented by Lai et al. (2016, Algorithm 3) is simply the median estimator. Given this, at every time step t, Lai et al. (2016, Theorem 1.1) guarantee that with probability at least $1 - \frac{4}{t^2}$

$$\left|\mu_i - \tilde{\mu}_{i,t}\right| \leq \left(\eta + \sqrt{\frac{\log t}{T_i(t)}}\right) e\sigma_i \tag{2}$$

Now suppose we have played an arm $i \neq i^*$ enough number of times to ensure $T_i(t) \geq \frac{16e^2\sigma_0^2 \log T}{\Delta_i^2}$, then we have the following chain of inequalities

$$\begin{aligned}
\tilde{\mu}_{i,t} + \left(\eta + \sqrt{\frac{\log t}{T_i(t)}}\right) e\sigma_0 &\leq \mu_i + \left(\eta + \sqrt{\frac{\log t}{T_i(t)}}\right) e\sigma_0 + \left(\eta + \sqrt{\frac{\log t}{T_i(t)}}\right) e\sigma_i \\
&= \mu^* - \Delta_i + \left(\eta + \sqrt{\frac{\log t}{T_i(t)}}\right) e\sigma_0 + \left(\eta + \sqrt{\frac{\log t}{T_i(t)}}\right) e\sigma_i \\
&\leq \mu^* \\
&\leq \tilde{\mu}_{i^*,t} + \left(\eta + \sqrt{\frac{\log t}{T_{i^*}(t)}}\right) e\sigma_{i^*} \\
&\leq \tilde{\mu}_{i^*,t} + \left(\eta + \sqrt{\frac{\log t}{T_{i^*}(t)}}\right) e\sigma_0
\end{aligned}$$

where the first and fourth steps follow from (2), the second step follows from the definitions, the third step uses the fact that $T_i(t)$ is large enough and $\eta_0 \leq \frac{\Delta_i}{4e\sigma_0}$, and the final step uses the fact that $\sigma_{i^*} \leq \sigma_0$ by construction.

The above shows that once an arm is pulled sufficiently many times, it will never appear as the highest upper bound estimate in the RUCB- MAB algorithm and hence will never get pulled again. This allows us to estimate, using a standard proof technique, the expected number of times each arm would be pulled, as follows

$$\begin{aligned}
\mathbb{E}[T_i(t)] &= 1 + \sum_{t=K+1}^{T} \mathbb{I}\{I_t = i\} \\
&= 1 + \mathbb{E}\left[\sum_{t=K+1}^{T} \mathbb{I}\left\{I_t = i \wedge T_i(t) \leq \frac{16e^2\sigma_0^2 \ln T}{\Delta_i^2}\right\} + \mathbb{I}\left\{I_t = i \wedge T_i(t) > \frac{16e^2\sigma_0^2 \ln T}{\Delta_i^2}\right\}\right] \\
&\leq 1 + \frac{16e^2\sigma_0^2 \ln T}{\Delta_i^2} + \sum_{t=K+1}^{T} \mathbb{P}\left[I_t = i \wedge T_i(t) > \frac{16e^2\sigma_0^2 \ln T}{\Delta_i^2}\right] \\
&= 1 + \frac{16e^2\sigma_0^2 \ln T}{\Delta_i^2} + \sum_{t=K+1}^{T} \mathbb{P}\left[I_t = i \mid T_i(t) > \frac{16e^2\sigma_0^2 \ln T}{\Delta_i^2}\right] \mathbb{P}\left[T_i(t) > \frac{16e^2\sigma_0^2 \ln T}{\Delta_i^2}\right] \\
&\leq 1 + \frac{16e^2\sigma_0^2 \ln T}{\Delta_i^2} + \sum_{t=K+1}^{T} \frac{16}{t^2} \\
&\leq \frac{16e^2\sigma_0^2 \ln T}{\Delta_i^2} + 35,
\end{aligned}$$

where in the first step, we use the fact that initially, each arm gets played once in a round-robin fashion in step 1 of Algorithm 3. We now have

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} r_t\right] &= \mathbb{E}\left[\sum_{i=1}^{K}\sum_{t=1}^{T} r_t \mathbb{I}\{I_t = i\}\right] \\
&= \sum_{i=1}^{K}\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{E}\left[r_t \mathbb{I}\{I_t = i\} \mid \mathcal{H}^t\right] \mathbb{I}\{I_t = i\}\right]
\end{aligned}$$

$$\geq \sum_{i=1}^{K}\sum_{t=1}^{T}(1-\eta)\mu_i \mathbb{E}\left[\mathbb{I}\{I_t = i\}\right] - B\eta \cdot T$$
$$= (1-\eta)\sum_{i=1}^{K}\mu_i \mathbb{E}\left[T_i(t)\right] - B\eta \cdot T$$

Combining with the previous bound on $\mathbb{E}\left[T_i(t)\right]$ and using $\eta > 0$ gives us the gap-dependent regret bound

$$\bar{R}_T(\text{RUCB- MAB}) \leq \sum_{i \neq i^*} \frac{16e^2\sigma_0^2 \ln T}{\Delta_i} + 35\Delta_i + \eta \cdot (\mu^* + B)T$$

To convert to the gap-agnostic form claimed in Theorem 2, we simply use the Cauchy-Schwartz inequality as follows

$$\bar{R}_T(\text{RUCB- MAB}) = (1-\eta)\mu^* \cdot T - \mathbb{E}\left[\sum_{t=1}^{T} r_t\right] + \eta \cdot (\mu^* + B)T$$
$$= (1-\eta)\sum_{i=1}^{K}\Delta_i \mathbb{E}\left[T_i(t)\right] + \eta \cdot (\mu^* + B)T$$
$$\leq (1-\eta)\sqrt{\sum_{i=1}^{K}\Delta_i^2 \mathbb{E}\left[T_i(t)\right]}\sqrt{\sum_{i=1}^{K}\mathbb{E}\left[T_i(t)\right]} + \eta \cdot (\mu^* + B)T$$
$$= (1-\eta)\sqrt{16e^2\sigma_0^2 KT\ln T + 35T\sum_{i=1}^{K}\Delta_i^2} + \eta \cdot (\mu^* + B)T,$$

which establishes the result. □

Proof *(Sketch of Theorem 3)* Notice that the proof of Theorem 2 shows that once a suboptimal arm is pulled sufficiently many times, it will never appear as the highest upper bound estimate in the RUCB- MAB algorithm and hence will never get pulled again. Hereon, the standard analysis applies.

$$\mathbb{E}\left[\sum_{t=1}^{T} r_t^*\right] = \mathbb{E}\left[\sum_{i=1}^{K}\sum_{t=1}^{T} r_t^* \mathbb{I}\{I_t = i\}\right]$$
$$= \sum_{i=1}^{K}\sum_{t=1}^{T}\mathbb{E}\left[\mathbb{E}\left[r_t^* \mathbb{I}\{I_t = i\} \mid \mathcal{H}^t\right]\mathbb{I}\{I_t = i\}\right]$$
$$= \sum_{i=1}^{K}\sum_{t=1}^{T}\mu_i \mathbb{E}\left[\mathbb{I}\{I_t = i\}\right]$$
$$= \sum_{i=1}^{K}\mu_i \mathbb{E}\left[T_i(t)\right]$$

Notice that this result relies on the assumption that the corruption rate is bounded $\eta \leq \frac{\Delta_{\min}}{4e\sigma_0}$. □

Proof of Corollary 1 The proof of Theorem 2 assures us that for arms that satisfy $\Delta_i > 4e\sigma_0\eta_0$ we have

$$\mathbb{E}[T_i(t)] \leq \frac{16e^2\sigma_0^2 \ln T}{\Delta_i^2} + 35$$

The total contribution to the regret due to these arms is already bounded by Theorem 2 as

$$\sum_{i:\Delta_i > 4e\sigma_0\eta_0} \Delta_i \cdot \mathbb{E}[T_i(t)] \leq C(1-\eta)\sqrt{KT\ln T} + \eta \cdot (\mu^* + B)T$$

For arms that do not satisfy the above condition, i.e., for whom we have $\Delta_i \leq 4e\sigma_0\eta_0$, the above does not apply. However, notice that the total contribution to the regret due to these arms can be at most

$$\sum_{i:\Delta_i \leq 4e\sigma_0\eta_0} \Delta_i \cdot \mathbb{E}[T_i(t)] \leq 4e\sigma_0\eta_0 \sum_{i:\Delta_i \leq 4e\sigma_0\eta_0} \mathbb{E}[T_i(t)] \leq 4e\sigma_0\eta_0 T,$$

since we must have $\sum_{i:\Delta_i \leq 4e\sigma_0\eta_0} T_i(T) \leq T$. Combining the two results gives us the claimed bound. Notice that no assumptions are made regarding $\Delta_{\min}$ in this proof. □

Proof of Theorem 4 In this case, we notice that the in the uni-dimensional case, the CovarianceEstimation algorithm proposed by Lai et al. (2016, Algorithm 4) is simply Step 1 and Step 2 of the rVUCB algorithm (see Algorithm 2). Given this, at every time step t, Lai et al. (2016, Theorem 1.5) guarantee that with probability at least $1 - \frac{4}{t^2}$

$$\left|\sigma_i - \tilde{\sigma}_{i,t}\right| \leq D\left(\eta^{1/2} + \left(\eta + \sqrt{\frac{\log t}{T_i(t)}}\right)^{3/4}\right)\sigma_i, \tag{3}$$

for some constant D, which establishes, with probability at least $1 - \frac{4}{t^2}$, that

$$\sigma_i \leq \tilde{\sigma}_{i,t}/(1-c),$$

where $c = D\left(\eta^{1/2} + \left(\eta + \sqrt{\frac{\log t}{T_i(t)}}\right)^{3/4}\right)$. To avoid a divide-by-zero error, we set a maximum bound 2η on c and assume that $\eta < 1/2$. This establishes that the algorithm rVUCB does indeed provide a high confidence upper bound on the variance of the distributions.

After noticing this, the rest of the analysis is routine. Given that an arm $i \neq i^*$ has been pulled enough number of times to ensure that we have $T_i(t) \geq \max\left\{\frac{16e^2\sigma_i^2(1+p)\log T}{\Delta_i^2}, \frac{\log T}{\eta^2}\right\}$, where $p = D(\sqrt{\eta} + (2\eta)^{3/4})$, we have the following chain of inequalities

$$\begin{aligned}
\tilde{\mu}_{i,t} + \left(\eta_0 + \sqrt{\frac{\log t}{T_i(t)}}\right)e\tilde{\sigma}_{i,t} &\leq \mu_i + \left(\eta_0 + \sqrt{\frac{\log t}{T_i(t)}}\right)e\tilde{\sigma}_{i,t} + \left(\eta + \sqrt{\frac{\log t}{T_i(t)}}\right)e\sigma_i \\
&= \mu^* - \Delta_i + \left(\eta_0 + \sqrt{\frac{\log t}{T_i(t)}}\right)e\tilde{\sigma}_{i,t} + \left(\eta + \sqrt{\frac{\log t}{T_i(t)}}\right)e\sigma_i \\
&\leq \mu^*
\end{aligned}$$

$$\leq \tilde{\mu}_{i^*,t} + \left(\eta + \sqrt{\frac{\log t}{T_{i^*}(t)}}\right) e\sigma_{i^*}$$
$$\leq \tilde{\mu}_{i^*,t} + \left(\eta_0 + \sqrt{\frac{\log t}{T_{i^*}(t)}}\right) e\tilde{\sigma}_{i^*,t}$$

where the first step follows from (2), the second step follows from the definitions, the third step uses the fact that $T_i(t)$ is large enough and η_0 is small enough, and the final step uses (3) and the fact that $\eta \leq \eta_0$ by definition. The above shows that once an arm is pulled sufficiently many times, it will never appear as the highest upper bound estimate in the RUCB- TUNE algorithm and hence will never get pulled again. The rest of the proof is routine now. □

B Proofs from Sect. 5

Proof *(Sketch of Lemma 1)* The proof is similar to that of previous results by Gentile et al. (2014, Lemma 2) and Gentile et al. (2017, Lemma 1). We need only show the result for one specific value of t and one specific subset $S \subset [t]$, $|S| = (1-\eta)\cdot|S|$. The result then follows from first a union bound over all subsets, as is done by Bhatia et al. (2015), and then a union bound over all $t \leq T$ which imposes an additional logarithmic factor.

For a fixed $\mathbf{z} \in \mathbb{R}^d$, and any $t \in [T]$, Gentile et al. (2014, Claim 1) show that

$$\mathbb{E}\left[\min_{k\in\{1,\ldots,n_t\}} (\mathbf{z}^\top \mathbf{x}^{t,k})^2 \mid n_t\right] \geq 1/4,$$

since we have assumed for sake of simplicity that the arms are being sampled from a standard Gaussian. A similar result holds for general sub-Gaussian distributions too. Now for any subset $S \subset [t]$, the proof then continues as in the analysis of Gentile et al. (2014, Lemma 2) by using optional skipping and setting up a Freedman-style matrix tail bound to get, as a consequence of the above, the following high-confidence estimate, holding with probability at least $1-\delta$,

$$\min_{\substack{\tau\in S\\ k_\tau\in\{1,\ldots,n_\tau\}}} \lambda_{\min}\left(\sum_{\tau\in S} \mathbf{x}^{\tau,k_\tau}(\mathbf{x}^{\tau,k_\tau})^\top\right) \geq B\left(|S|, \frac{\delta}{2d}\right), \tag{4}$$

where

$$B(T,\delta) = T/4 - 8\left(\log(T/\delta) + \sqrt{T\,\log(T/\delta)}\right).$$

Continuing with the union bounds as described above finishes the proof. □

Proof of Theorem 6 To avoid clutter, we will replace $\hat{G}_t$ by G in the following. Let $\boldsymbol{\epsilon}_G$ and $\mathbf{b}_G$ denote the noise and corruption values in those time instances so that $\mathbf{r}_G = X_G^\top \mathbf{w}^* + \boldsymbol{\epsilon}_G + \mathbf{b}_G$. Note that $M_t = X_G X_G^\top$. We have

$$\begin{aligned}\bar{\mathbf{w}}^t &= (X_G X_G^\top)^{-1} X_G^\top \mathbf{r}_G \\ &= (X_G X_G^\top)^{-1} X_G^\top (X_G^\top \mathbf{w}^* + \boldsymbol{\epsilon}_G + \mathbf{b}_G) \\ &= \mathbf{w}^* + (X_G X_G^\top)^{-1} X_G^\top (\boldsymbol{\epsilon}_G + \mathbf{b}_G)\end{aligned}$$

Now, following the proof technique of Abbasi-Yadkori et al. (2011) requires us to bound $\|X_G(\boldsymbol{\epsilon}_G + \mathbf{b}_G)\|_{M_t}$. Using the fact that $M_t = X_G X_G^\top$ gives us

$$\|X_G(\boldsymbol{\epsilon}_G + \mathbf{b}_G)\|_{M_t} \leq \|X_G \boldsymbol{\epsilon}_G\|_{M_t} + \|X_G \mathbf{b}_G\|_{M_t}.$$

Let $G_t = \{\tau \leq t : b_\tau = 0\}$ be the set of clean points till time t. Since the results of Bhatia et al. (2015, Theorem 10) ensure that the output of TORRENT satisfies $\left\|\hat{\mathbf{w}}^t - \mathbf{w}^*\right\|_2 \leq \mathcal{O}(\sigma_0)$, we are assured with probability at least $1 - \frac{1}{t^2}$ that $G_t \subseteq \hat{G}_t$. Thus, we get

$$\begin{aligned}
\|X_G \boldsymbol{\epsilon}_G\|_{M_t}^2 &= \boldsymbol{\epsilon}_G^\top X_G (X_G X_G^\top)^{-1} X_G^\top \boldsymbol{\epsilon}_G \\
&= \boldsymbol{\epsilon}_{G_t}^\top X_{G_t} (X_G X_G^\top)^{-1} X_{G_t}^\top \boldsymbol{\epsilon}_{G_t} \\
&\leq \boldsymbol{\epsilon}_{G_t}^\top X_{G_t} (X_{G_t} X_{G_t}^\top)^{-1} X_{G_t}^\top \boldsymbol{\epsilon}_{G_t}
\end{aligned}$$

where the second step follows from the fact that we can canonically define $\epsilon_\tau = 0$ for the corrupted time instances, i.e., if $\tau < t$ and $\tau \notin G_t$) by setting $b_t = b_t + \epsilon_t$, and the last step uses the fact that $G_t \subset G$. However, the quantity $\boldsymbol{\epsilon}_{G_t}^\top X_{G_t} (X_{G_t} X_{G_t}^\top)^{-1} X_{G_t}^\top \boldsymbol{\epsilon}_{G_t}$ can be bounded by $\sigma_0 \sqrt{d \log T}$ using the self normalized martingale inequality by Abbasi-Yadkori et al. (2011, Theorem 1) as it is the set of uncorrupted points to which standard results keep applying. The second quantity $\|X_G \mathbf{b}_G\|_{M_t}$ can be similarly bounded by using the fact that $\|\mathbf{b}_G\|_0 \leq 2\eta \cdot t$ and since $\left\|\hat{\mathbf{w}}^t - \mathbf{w}^*\right\|_2 \leq \mathcal{O}(\sigma_0)$ by Bhatia et al. (2015, Theorem 10), any so, corrupted points τ that may have landed into the set $\hat{G}_t$ must satisfy $|b_\tau| \leq \sigma_0 \sqrt{\log T}$. This finishes the proof. Note that the last argument $|b_\tau| \leq \sigma_0 \sqrt{\log T}$ reveals that the pruning step is indeed a noise-removal step. It prunes away any arm which had its reward excessively corrupted. □

Proof of Theorem 7 The proof is mostly routine and follows the proof of a similar result by Abbasi-Yadkori et al. (2011, Theorem 3). Let us define $(\hat{\mathbf{x}}^t, \tilde{\mathbf{w}}^t) = \underset{\mathbf{x} \in A_t}{\arg\max} \underset{\mathbf{w} \in C_{t-1}}{\arg\max} \langle \mathbf{x}, \mathbf{w} \rangle$. Then

$$\begin{aligned}
\mathbb{E}\left[\left\langle \mathbf{w}^*, \mathbf{x}^{t,*} \right\rangle - r_t \mid \mathcal{H}^t\right] &\leq (1-\eta)\left(\left\langle \mathbf{w}^*, \mathbf{x}^{t,*} \right\rangle - \left\langle \mathbf{w}^*, \hat{\mathbf{x}}^t \right\rangle\right) \\
&\quad + \eta\left(\left\langle \mathbf{w}^*, \mathbf{x}^{t,*} \right\rangle + B\right) \\
&\leq (1-\eta)\left(\left\langle \tilde{\mathbf{w}}^t, \hat{\mathbf{x}}^t \right\rangle - \left\langle \mathbf{w}^*, \hat{\mathbf{x}}^t \right\rangle\right) + \eta\left(\left\langle \mathbf{w}^*, \mathbf{x}^{t,*} \right\rangle + B\right) \\
&= (1-\eta)\left\langle \tilde{\mathbf{w}}^t - \mathbf{w}^*, \hat{\mathbf{x}}^t \right\rangle + \eta\left(\left\langle \mathbf{w}^*, \mathbf{x}^{t,*} \right\rangle + B\right) \\
&= (1-\eta)\left(\left\langle \tilde{\mathbf{w}}^t - \bar{\mathbf{w}}^t, \hat{\mathbf{x}}^t \right\rangle - \left\langle \mathbf{w}^* - \bar{\mathbf{w}}^t, \hat{\mathbf{x}}^t \right\rangle\right) \\
&\quad + \eta\left(\left\langle \mathbf{w}^*, \mathbf{x}^{t,*} \right\rangle + B\right) \\
&\leq (1-\eta)\left\|\hat{\mathbf{x}}^t\right\|_{M_t^{-1}} \left(\left\|\tilde{\mathbf{w}}^t - \bar{\mathbf{w}}^t\right\|_{M_t} + \left\|\mathbf{w}^* - \bar{\mathbf{w}}^t\right\|_{M_t}\right) \\
&\quad + \eta(1 + B),
\end{aligned}$$

Now, the SSC properties guarantee $\lambda_{\min}(M_t) = \Omega(t)$ which gives us $\left\|\hat{\mathbf{x}}^t\right\|_{M_t^{-1}} \leq \mathcal{O}\left(\frac{1}{\sqrt{t}}\right)$. This finishes the proof upon using Theorem 6 and simple manipulations. □

References

Abbasi-Yadkori, Y., Pal, D., & Szepesvari, C. (2011). Improved algorithms for linear stochastic bandits. In *Proceedings of the 25th annual conference on neural information processing systems (NIPS)*.

Audibert, J.-Y., Munos, R., & Szepesvári, C. (2007). Tuning bandit algorithms in stochastic environments. In *Proceedings of the 18th international conference on algorithmic learning theory (ALT)*.

Audibert, J.-Y., Munos, R., & Szepesvári, C. (2009). Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, *410*(19), 1876–1902.

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, *47*, 235–256.

Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. (2002b). The nonstochastic multiarmed bandit problem. *SIAM Journal of Computing*, *31*(1), 48–77.

Bhatia, K., Jain, P., & Kar, P. (2015). Robust regression via hard thresholding. In *Proceedings of the 29th annual conference on neural information processing systems (NIPS)*.

Bubeck, Sébastian., & Slivkins, A. (2012). The best of both worlds: stochastic and adversarial bandits. In *Proceedings of the 25th annual conference on learning theory (COLT)*.

Bubeck, S., Cesa-Bianchi, N., & Lugosi, G. (2013). Bandits with heavy tail. *IEEE Transaction on Information Theory*, *59*(11), 7711–7717.

Candès, E. J., Li, X., & Wright, J. (2009). Robust principal component analysis? *Journal of the ACM*, *58*(1), 1–37.

Chakrabarti, D., Kumar, R., Radlinski, F., & Upfal, E. (2008). Mortal multi-armed bandits. In *Proceedings of the 21st international conference on neural information processing systems (NIPS)*.

Charikar, M., Steinhardt, J., & Valiant, G. (2017). Learning from untrusted data. In *Proceedings of the 49th annual ACM SIGACT symposium on theory of computing (STOC)* (pp. 47–60).

Chen, Y., Caramanis, C., & Mannor, S. (2013). Robust sparse regression under adversarial corruption. In *Proceedings of the 30th international conference on machine learning (ICML)*.

Chu, W., Li, L., Reyzin, L., & Schapire, R. (2011). Contextual bandits with linear payoff functions. In *Proceedings of the 14th international conference on artificial intelligence and statistics (AISTATS)*.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., & Stewart, A. (2016). Robust estimators in high dimensions without the computational intractability. In *Proceedings of the 57th IEEE annual symposium on foundations of computer science (FOCS)*.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., & Stewart, A. (2018). Robustly learning a gaussian: Getting optimal error, efficiently. In *Proceedings of the twenty-ninth annual acm-siam symposium on discrete algorithms (SODA)* (pp. 2683–2702).

Feng, J., Xu, H., Mannor, S., & Yan, S. (2014). Robust logistic regression and classification. In *Proceedings of the 28th annual conference on neural information processing systems (NIPS)*.

Gajane, P., Urvoy, T., & Kaufmann, E. (2018). Corrupt bandits for preserving local privacy. In *Proceedings of the 29th international conference on algorithmic learning theory (ALT)*.

Garivier, A., & Cappé, O. (2011). The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory (COLT)*.

Gentile, C., Li, S., Kar, P., Karatzoglou, A., Zappella, G., & Etrue, E. (2017). On context-dependent clustering of bandits. In *Proceedings of the 34th international conference on machine learning (ICML)*.

Gentile, C., Li, S., & Zappella, G. (2014). Online clustering of bandits. In *Proceedings of the 31st international conference on machine learning (ICML)*.

Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, *35*(1), 73–101.

Lai, K. A., Rao, A. B., & Vempala, S. (2016). Agnostic estimation of mean and covariance. In *Proceedings of the 57th IEEE annual symposium on foundations of computer science (FOCS)*.

Li, L., Chu, W., Langford, J., & Schapire, R. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international world wide web conference (WWW)*.

Lykouris, T., Mirrokni, V., & Leme, R. P. (2018). Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th annual ACM SIGACT symposium on theory of computing (STOC)* (pp. 114–122).

Maronna, R. A., Martin, R. D., & Yohai, V. J. (2006). *Robust statistics: Theory and methods*. New York: Wiley.

Medina, A. M., & Yang, S. (2016). No-regret algorithms for heavy-tailed linear bandits. In *Proceedings of the 33rd international conference on machine learning (ICML)*.

Nguyen, N. H., & Tran, T. D. (2013). Exact recoverability from dense corrupted observations via ℓ_1-minimization. *IEEE Transactions on Information Theory*, *59*(4), 2017–2035.

Padmanabhan, D., Bhat, S., Garg, D., Shevade, S. K., & Narahari, Y. (2016). A robust UCB scheme for active learning in regression from strategic crowds. In *Proceedings of the international joint conference on neural networks (IJCNN)*.

Seldin, Y., & Slivkins, A. (2014). One practical algorithm for both stochastic and adversarial bandits. In *Proceedings of the 31st international conference on machine learning (ICML)*.

Tang, L., Rosales, R., Singh, A. P., & Agarwal, D. (2013). Automatic Ad format selection via contextual bandits. In *Proceedings of the 22nd ACM international conference on information and knowledge management (CIKM)*.

Tewari, A., & Murphy, S. A. (2017). *Mobile health, chapter From Ads to interventions: Contextual bandits in mobile health* (pp. 495–517). New York: Springer.

The Hindustan Times. #Appwapsi: Snapdeal gets blowback from Aamir Khan controversy, Nov 24, (2015). https://www.hindustantimes.com/india/appwapsi-snapdeal-gets-blowback-from-aamir-khan-controversy/story-N3HwOObJ0WMe9vz7GjXFBO.html. Accessed July 15, 2018.

Tsybakov, A. B. (2009). *Introduction to nonparametric estimation*. New York: Springer.

Tukey, J. W. (1960). A survey of sampling from contaminated distributions. *Contributions to Probability and Statistics*, *2*, 448–485.