Multi-Armed Bandit and Anomaly Detection

Pouya Abbasi

Written report for ‘Academic Reading and Writing’ module
Master of Science in Mathematics
Universität Potsdam

Supervisor: Dr. Tomáš Kocák

Statement of Originality

I hereby declare that I have written this thesis independently and have not used any other aids than those indicated by me.

Name: Pouya Abbasi

Student Matriculation number: 806307

Date of submission: March 15, 2023

1 Introduction

.

1.1 Multi-Armed Bandit (MAB)

Wherever there is a need to make a decision in a complex situation and under uncertainty, one can benefit from Multi-Armed Bandit (MAB) algorithm. According to [4], ”Multi-armed bandit models are a benchmark model for learning to make decisions under uncertainty”. The MAB is a form of reinforcement learning in that an agent depending on the mission is supposed to maximize or minimize, respectively the reward or regret. For this purpose, the agent will take a strategy in order to make a compromise between exploration in order to discover new opportunities that can result in better outcomes, and exploitation to employ the best-guaranteed knowledge that currently supplies the topmost result. This makes multi-armed bandits ideal for problems where there are multiple choices or actions available but only limited information about their outcomes.

As has been mentioned in [11], roughly speaking, there are two main frameworks, *stochastic stationary bandits* and *adversarial bandits.*. The environment with stochastic stationary bandits is allowed to create rewards from a specific distribution independent from the last action of the related agent. On the other, if we remove all restrictions on the distribution of environment but know that rewards belong to a bounded set, it is called adversarial bandits. These are two extreme points of an interval and many cases can happen in between.

A classical example of the MAB is pulling the lever of a multi-lever slot machine and seeing the payoff. There are many fields that are applying the MAB algorithm including news recommendation, dynamic pricing, ad placement, tree search, resource allocation, waiting for problems, network routing, A/B testing, and medical trials, for more details please refer to [11]. A common feature in all of the above-mentioned examples is the necessity of trial and error in order to achieve the best result for the objective. Traditional methods are inefficient and sometimes impossible and this is where the MAB steps in.

As reported by [11] generally, the objective of the MAB algorithm can be quite different. Maximizing the cumulative reward over all n rounds is one of the most prevailing purposes. Another fundamental concern in the bandit area of research is to perceive the increased rate of regret as n, the number of pulling, becomes larger. Converging to the optimal arm and adjusting between exploration and exploitation is aslo an aspiration of the MAB that has been extensively discussed in [17]. In addition to them identifying anomalous (outlier) arms is a separate direction that in this paper we will talk about it.

1.2 Anomaly detection

Anomaly detection is a comprehensive concept that covers many divergent tasks including outlier, novelty, a rare event, discordant observations, exceptions, aberrations, surprises, peculiarities or contaminants detection [14, 6]. It is for exposing patterns that are not following expected behavior [6]. In most of the corpora *outlier* and *anomaly* are being used interchangeably. Anomaly has a broader definition than outliers. Outlier points out cases that have some sort of distance from the rest of the other observations. Namely, it needs to take an individual observation and measure its status in comparison to other observations. According to [9], ”An outlier is a data object that deviates significantly from the rest of the objects”.

When applying anomaly detection in MAB problems, everything will depend on the definition of anomaly. Anomalies are specific behavior that does not follow a major manner of data. They have different natures, forms, and shapes. It could happen that in one given set of batch or sequential data, different types of anomalies occur. Therefore, one single definition of anomaly can not be suitable for pointing to the other one. Hence, considering the process of anomaly detection as a set of different filters or screens that are working simultaneously, is more realistic and efficient. There are cases in that data has different sorts of anomalies and relying only on one type of detection algorithm is not enough.

The definition of the anomaly itself depends on the context and field of application. Nonetheless, when we are talking about the anomaly, we are talking about behavior in general. Thus, we are inspecting, assessing, and investigating observation as a whole and singularly simultaneously.

Whereas, there are some observations that will not be tagged as anomalies if they do not happen in a particular order. For instance, imagine we have ten normal arms $a_1, a_2, ...a_{10}$ and each of the arms is normal as long as not any of six combinations of three consecutive arms occurs i.e $a_1a_2a_1, a_1a_3a_2, a_2a_1a_3, ..$ which is an example of *collective anomalies*. Generally, nearest neighbors, either distance-based or density-based anomaly detection techniques cannot be used for collective anomalies. There is a situation where arms are not considered extreme or far from the majority of the normal set, but a sequence of pulled arms is not usual. In other world expected reward of arms lies in the so-called normal set, but since they have occurred together with or without repetitive sequence, they could be labeled as collective anomalies [6] which [2] and [14] both are blind in capturing them due to the essence of their definition of anomaly. Among all other approaches for anomaly (outlier) detection, *clustering* has substantial comparability. In the next section, some of these similarities will be introduced.

1.3 Clustering

Outlier detection has similarities same as *clustering.* As in clustering goal is to group arms based on rewards distribution. Finding the optimal number of clusters is an open field of research. However, it would not be that much unrelated if we set the number of clusters as a predefined parameter. For instance, in [15], the framework is to put dependent arms in the same cluster, where the number of arms is greater than the number of clusters. Under some conditions and assumptions, outlier detection can be considered as clustering. However, in none of the clustering lines of works, models have not been exposed to situations where there is some kind of discrepancies, anomalies, or outliers, to evaluate the performance or result of the model. Furthermore, clustering is addressing the problem of anomaly detection from the counterpart approach, i.e., unlike focusing on outliers or anomalies, the goal here is grouping arms to the number of clusters and by comparing these clusters visually or according to some particular criteria, labeling outliers.

The work [16] is a generalization of the contextual bandit. They have unrestricted some assumptions in their model; unlike other similar works where the linearity of the expected reward of arms is assumed they have loosened it has taken the situation into account where the arm's mean reward is a nonlinear function of an unknown parameter. In addition to that, in their setting, there is no need that all arms to belong to an identical parameter vector. This means only arms allocated to the same cluster share a vector of parameters. This relaxation of assumption is necessary; Otherwise inspecting arms for detecting anomalies would not be possible. They also have removed the monotonicity assumption of reward distributions. Within this framework, they have introduced a model to group arms into various clusters where arms within the same cluster have an equivalent parameter vector that explains the reward distribution of that cluster, which are linear functions of the unknown parameters. Authors in [16] have generalized the problem of clustering in five different categories, namely, Bandits with Side Observations, Contextual Bandits, MABP with Correlated Arms, Global and Regional Bandits, and Structured Bandits. Following each of them will be explained briefly.

Bandits with Side Observations: In the setting of [13], the graph of dependency represents the dependent arms; Thus, pulling an arm reveals observing the rewards of all other arms that are connected through edges. Although in [13], [5], and [3] have studied this dependency in both adversarial and stochastic settings, the above-mentioned assumption is excessively antagonistic. It is more pragmatic if we assume that arms hardly could have analogous parameters than expose their distribution. Thus, pulling an arm at most can disclose some erroneous information about the cluster of arms.

Contextual Bandits: Sometimes there is some extra information also known as features or context of arms observable. This setting is called contextual bandit and is slightly different from classical bandit models. For instance according to [12], in an online shop each product has its own vector of features. So, the recommendation machine has to offer products based on the user's vector of context such that the reward, which here is equal to the dot product of two vectors, become as close as to one.

MABP with Correlated Arms: In [15] authors have studied a condition where arms are dependent and tried to cluster arms by exploiting this assumption for discounted scenarios. They have formulated a problem for a setting where there is a slot machine with N arms that are going to be grouped into K known clusters. The dependency of arms in each cluster operates under the Bayesian framework that can be described by a known generative model with an unknown parameter. For each cluster, an expected reward and variance will be estimated. Thus, the performance of the model will be affected by the cluster characteristics. For discounted cases, they have found an MDP-based policy. In each step, this policy gives an (*index, arm*) pair for each cluster and pulls the arm from the cluster with the highest index. Additionally, they have given an error bound for a simple version for optimal policy. However, as authors in [16] have pointed out correctly, they have not provided such computation for undiscounted scenarios.

Global and Regional Bandits: Authors in [1] propose a model called "global bandits", where a known function with a universal unknown parameter generates the rewards of each arm. Hence, pulling an arm will announce inexact information about all other arms. Particularly this supposition is so prohibitive. Along these lines, in [18] and [19] have loosened the mentioned assumption and presumed only arms in same the cluster will have unique parameters. Nevertheless, still they some it makes a few heavy obligations upon the reward distributions such as "the unknown parameters that describe distributions of a single cluster are assumed to be scalar", and "the mean reward function is Holder continuous, and more importantly a monotonic function of the unknown parameter". Practically, the monotonicity condition is not very realistic.

Structured Bandits: In this very generic MAB arrangement, [10], [7], and [8] have studied problem where the rewards of each arm are mapped by a known function that depends on an unknown parameter. They suggest an optimization model for making a prediction on how many times an arm has to be pulled.

2 Outiler detection in MAB

In [2] authors have studied detecting outlier arms in a multi-armed bandit framework. For that aim, they created a model that will learn how to distinguish arms with expected rewards that are departed considerably from the majority of the other arms. Authors argue that their method is dissimilar to other existing works. Since, they capture a *group* of outlier arms with expected rewards not only greater or less than, but also in the middle of *normal arms*. To address that problem they have provided their own general definition of the outlier arm. Additionally, the authors have introduced a pulling algorithm called *GOLD* in order to hook those generic outlier arms. The algorithm makes a ”real-time neighborhood graph based on upper confidence bounds” and differs arms with anomalous habits from normal arms.

Authors refer [20] as a primary work that studied the outlier arm detection and have tried to develop and expand outlier detection. When it comes to MAB, most of the time the aim is detecting outlier arm(s) instead of solely finding outlier individual data points. However, in [20], their method simply categorizes all arms as an outlier if their expected rewards are above a threshold so-called *k-sigma rule*. This cut-off point is the mean of expected rewards of all arms plus k-times of standard deviation. They point out two main weak spots in the *k-sigma rule*; First, it defines arms as an outlier only if it exceeds the threshold. Thus, arms even with distinguishable distance from other arms will be overlooked. Second, since the distribution of expected rewards of arms is unknown, finding the optimal value for k would be difficult, and the result of outlier detection is heavily prone to having false positive or negative outputs.

To overcome the mentioned problems, authors [2] have come up with their own encompassing definition of an outlier arms group: ”instead of only focusing on the arms with an exceptionally high expected reward, an outlier arm should have an expected reward that is ‘unusually far’ from ‘most’ of the other arms”. Therefore, they have allocated a pair of parameters (ϵ, ρ). ϵ for defining a rational distance threshold to make sure that the closest distance of the outlier arm from the normal arms set is farther than the distance of arms in normal arms set from each other. ρ is to limit the smallest quantity of normal arms set, i.e., $\rho = 0.95$ means that the outlier arm will satisfy the threshold distance from at least 95% of normal arms.

Moreover, given the fact that although in a data set the number of outlier arms is unknown, the most departed arms should have been investigated first. To achieve that they have used a method of ranking arms such that outlier arms will have a higher rank in comparison to normal arms. In the following, this problem will be discussed more precisely.

2.1 Problem definition

According to authors in [2] their main goal is "to identify the outlier arms whose expected rewards are far away (significantly deviating) from most of the other arms". Because the expected value of arms is just a scalar, outliers belong to one of the two different categories. The outer category with an extreme expected reward is located upper side or lower side of normal arms, and the inner category without extreme expected rewards is located between normal arms. The nature of their general definition of outlier arms can be characterized according to three properties:

1. Regarding the expected rewards, outliers are distant from normal arms.
2. The expected reward of normal arms is so close to other normal arms that they can make the neighborhood distinguishable in comparison to outlier arms.
3. A large number of arms are normal, and they have expected rewards greater or lower than a few outlier arms.

Authors in [2] before officially describing their definition, have denoted some conventions and notions that are essential to make them understandable:

Let $\Psi = 1, ..., n$ represents the the set of n arms. Each arm comes from a different probability distribution and an unknown expected reward y_i. For any subset $\mathcal{N} \in \Psi$, each $i \in \mathcal{N}$ could have two nearest upper or lower side neighbor arms. Namely, for an upper-side neighbor:

$$\triangle_u^1(i, \mathcal{N}) = \begin{cases} 0, & \text{if } \nexists\ \acute{i} \in \mathcal{N}, y_{\acute{i}} > y_i \\ min_{\acute{i} \in \mathcal{N}, y_{\acute{i}} > y_i}(y_{\acute{i}} - y_i), & \text{otherwise} \end{cases}$$

and for a lower-side neighbor:

$$\triangle_l^1(i, \mathcal{N}) = \begin{cases} 0, & \text{if } \nexists\ \acute{i} \in \mathcal{N}, y_{\acute{i}} < y_i \\ min_{\acute{i} \in \mathcal{N}, y_{\acute{i}} < y_i}(y_i - y_{\acute{i}}), & \text{otherwise} \end{cases}$$

Then for any given i, the **neighborhood distance** is defined as following:

$$\Diamond^1(i, \mathcal{N}) = max\{\triangle_u^1(i, \mathcal{N}), \triangle_l^1(i, \mathcal{N})\}.$$

The operator $\Diamond^1$, measures the maximum distance of the arm from its upper and lower neighbors. the superscription *1* means this distance is with respect to the first closest arms from upper and lower neighborhoods. Similarly, this can extend to k-th closest upper and lower neighborhoods.

Then by means of the above concept, they describe the individual outlier arm and outlier group arms definition:

Definition 2.1 ((ϵ, ρ)-outlier arms.)**.** Given an arm $j \in \Psi$, then j is an (ϵ, ρ)-outlier arm in Ψ, if $\exists\ \mathcal{N}_u \subseteq \{i \in \Psi : y_i > y_j\}$, $\exists\ \mathcal{N}_l \subseteq \{i \in \Psi : y_i < y_j\}$ that satisfies the following two constraints:

$$\textbf{Constraint (1)} = \begin{cases} a)\ \triangle_{min}(j, \mathcal{N}_u) > (1+\epsilon) \times \Diamond^1(i, \mathcal{N}_u), & \forall i \in \mathcal{N}_u \\ b)\ \triangle_{min}(j, \mathcal{N}_u) > (1+\epsilon) \times \Diamond^1(i, \mathcal{N}_l), & \forall i \in \mathcal{N}_l \\ c)\ \triangle_{min}(j, \mathcal{N}_l) > (1+\epsilon) \times \Diamond^1(i, \mathcal{N}_u), & \forall i \in \mathcal{N}_u \\ d)\ \triangle_{min}(j, \mathcal{N}_l) > (1+\epsilon) \times \Diamond^1(i, \mathcal{N}_l), & \forall i \in \mathcal{N}_l \end{cases}$$

where $\epsilon > 0$ and $\triangle_{min}(j, \mathcal{N}_u) = min_{\acute{i} \in \mathcal{N}_u} |y_j - y_{\acute{i}}|$; and

$$\textbf{Constraint (2)} = \begin{cases} |\mathcal{N}_u| + |\mathcal{N}_l| > \rho \times n \\ |\mathcal{N}_u| = 0\ or\ |\mathcal{N}_u| > (1-\rho) \times n \\ |\mathcal{N}_l| = 0\ or\ |\mathcal{N}_l| > (1-\rho) \times n, \end{cases}$$

where $1 > \rho > 0.5$.

Hence, in practice when it comes to a real-world situation, the number of arms is more than one. The following is the extension of the definition 2.1 for covering them all:

Definition 2.2 ((ϵ, ρ)-outlier group.)**.** Given three sets of arms $\hat{\mathcal{N}} \subset \Psi$, $\mathcal{N}_u = \{i \in \Psi : y_i > max_{j \in \hat{\mathcal{N}}} y_j\}$, and $\mathcal{N}_l = \{i \in \Psi : min_{j \in \hat{\mathcal{N}}} y_j < y_i\}$ that satisfy $\hat{\mathcal{N}} \cup \mathcal{N}_u \cup \mathcal{N}_l = \Psi$, then $\hat{\mathcal{N}}$ is a (ϵ, ρ)-outlier group with respect to $\mathcal{N}_u$ and $\mathcal{N}_l$ if the two following constraints are satisfied:

$$\textbf{Constraint (1)} = \begin{cases} a)\ \triangle_{min}(j, \mathcal{N}_u) > (1+\epsilon) \times \Diamond^1(i, \mathcal{N}_u), & \forall j \in \hat{\mathcal{N}}, \forall i \in \mathcal{N}_u \\ b)\ \triangle_{min}(j, \mathcal{N}_u) > (1+\epsilon) \times \Diamond^1(i, \mathcal{N}_l), & \forall j \in \hat{\mathcal{N}}, \forall i \in \mathcal{N}_l \\ c)\ \triangle_{min}(j, \mathcal{N}_l) > (1+\epsilon) \times \Diamond^1(i, \mathcal{N}_u), & \forall j \in \hat{\mathcal{N}}, \forall i \in \mathcal{N}_u \\ d)\ \triangle_{min}(j, \mathcal{N}_l) > (1+\epsilon) \times \Diamond^1(i, \mathcal{N}_l), & \forall j \in \hat{\mathcal{N}}, \forall i \in \mathcal{N}_l \end{cases}$$

where $\epsilon > 0$ and $\triangle_{min}(j, \mathcal{N}_u) = min_{\acute{i} \in \mathcal{N}_u} |y_j - y_{\acute{i}}|$; and **Constraint (2)**: same as **Constraint (2)** in definition 2.1.

Now having the definition in hand we are ready to explore and investigate the capacities of this framework and see how comprehensive it is.

2.2 Scope and limitations of the definition

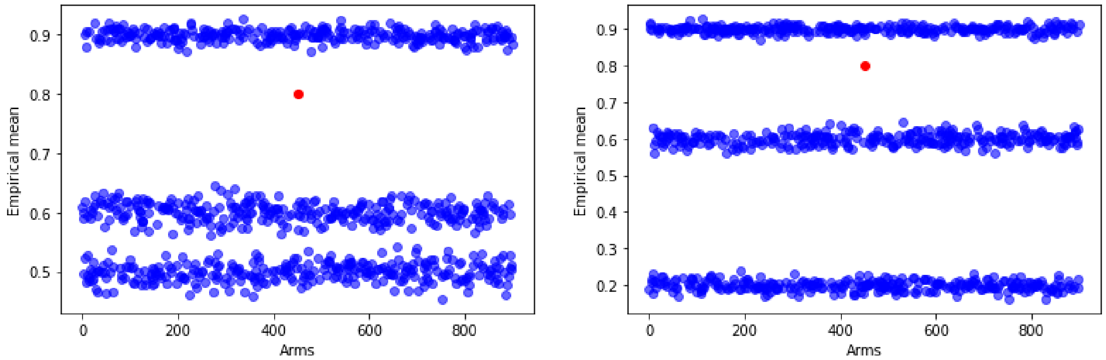
Here we will investigate cases and situations that can or cannot be captured by the definition of outlier formulated through Constraints (1) and (2) defined in 2.2. In Constraint (1), $\triangle_{min}$ means the shortest distance of outlier arm from upper $\mathcal{N}_u$ or lower $\mathcal{N}_l$ normal arms groups. Constraints (a) and (b) are for making sure that the $\triangle_{min}(j, \mathcal{N}_u)$ is $(1+\epsilon)$-times bigger than any inter-distance of arms in $\mathcal{N}_u$ and $\mathcal{N}_l$. Similarly, Constraint (d) and (e) certify the same condition for $\triangle_{min}(j, \mathcal{N}_l)$. The important point here which worth mentioning is that the $\triangle_{min}(j, \mathcal{N}_l)$ and $\triangle_{min}(j, \mathcal{N}_u)$ not only should be $(1+\epsilon)$-times greater than their corresponding lower and upper normal clusters respectively, but also they need to have a same state in comparing to the opposite normal clusters.

In other words, the constraints (b) and (d) will guarantee that the maximum or minimum distance of every neighbor arm inside opposite normal clusters are $(1+\epsilon)$-times less than $\triangle_{min}(j, \mathcal{N}_l)$ or $\triangle_{min}(j, \mathcal{N}_u)$. Constraints (b) and (d) implicitly imply that the inner deviation of opposite normal clusters should be smaller by the portion of $(1+\epsilon)$.

Particularly, identifying an arm or a group of arms as an outlier depends on the distance from the respective normal cluster, and inner-distance of the opposite cluster of arms, and the density of normal and outlier clusters. In order to label an arm or group of arms as an outlier all conditions in Constraint (1) and (2) should be satisfied at the same time. The subsets of $\mathcal{N}_u$ and $\mathcal{N}_l$ are groups of normal arms such that their size meets the requisites of Constraint (2). This proportionality will make it possible to ignore arms that are standing in a distance that results in violation of Constraints (1); If the mentioned state occurs, it is possible to remove them as long as the size of $\mathcal{N}_u$ and $\mathcal{N}_l$ stay aligned with conditions in Constraint (2).

One other interesting state as an outcome of definition and Constraints is, the possibility to construct a region of outliers. Namely, we can discover boundaries such that any arms located in the area inside that borderlines for sure can be considered an outlier. This fact has been predicted under the group of outlier arms.

Furthermore, there are blind zones where if any outliers are placed there, due to natural limitations driven by Constraints (2), those arms cannot be detected. For example, consider a set of data composed of N arms, consisting of three normal clusters C_1, C_2, and C_3, with almost equal numbers of arms e.i. $N_1 = N_2 = N_3 = N/3$, and significantly small inner-cluster distance, but a substantial inter-cluster distant from each other. In this case, identifying the outlier arm depends on the distance of normal arm clusters from each other, Figure 1a and 1b.



(a) Conditionally detectable

(b) Never detectable

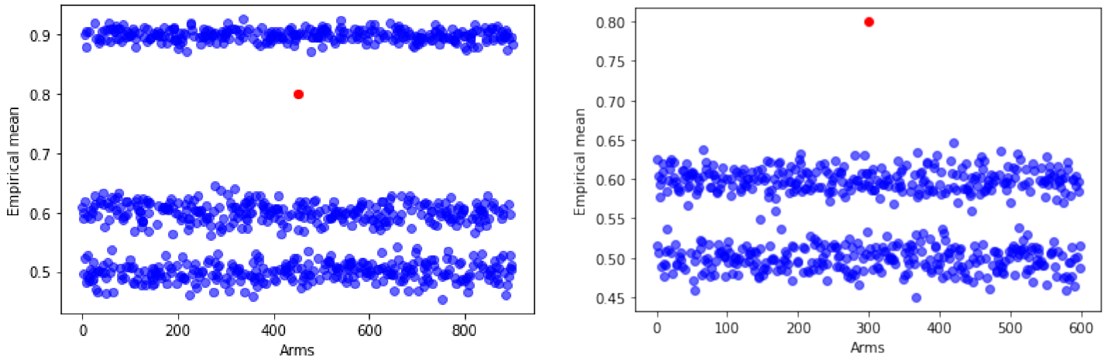
Figure 1: Intermediate outlier arm

First let us find out how the red outlier arm in a configuration as in Figure 1a, could be detected. According to the definition 2.2, with respect to outlier arm i, $\mathcal{N}_u$ is equal to or a subset of C_1, and $\mathcal{N}_l$ is equal to or subset of C_2 and C_3. Namely, $\mathcal{N}_u \subseteq \{C_1\}$ and $\mathcal{N}_l \subseteq \{C_2, C_3\}$. The size of this upper-side and lower-side normal arms respectively are $|\mathcal{N}_u| = N/3$ and $|\mathcal{N}_l| = 2N/3$. The upper-side normal arms of arm i, $\mathcal{N}_u$, is the set of arms that are very dense such that can be considered as one cluster. On the other hand, $\mathcal{N}_l$ is a set of two distinguishable clusters of normal arms.

Without losing generality, this arm cannot be labeled as an outlier as far as it has an upper-side normal arm e.i. $\mathcal{N}_u = \{C_1\}$, Figure 2a. The reason is the Constraint (2); According to that in this circumstance it needs $|\mathcal{N}_u| = 0$ *or* $|\mathcal{N}_u| > (1-\rho) \times N$. nevertheless, $|\mathcal{N}_u| = N/3 \ngeq (1-\rho) \times N$. Thus, by removing all arms of cluster c_1 from upper-side normal arms e.i putting $\mathcal{N}_u = \{\emptyset\}$, the size of $\mathcal{N}_u$ will become zero. Now, if the distance between remaining clusters C_2 and C_3 was $(1-\epsilon)$-times shorter than the distance of outlier arm i and the closest arm from $\mathcal{N}_l$, the outlier arm i can be detected as shown in Figure 2b, namely:

$$\begin{cases} \triangle_{min}(i, \mathcal{N}_l) > (1+\epsilon) \times \Diamond^1(j, \mathcal{N}_l), \quad \forall j \in C_2 \\ |\mathcal{N}_u| + |\mathcal{N}_l| = 2N/3 > \rho \times n \\ |\mathcal{N}_u| = 0 \\ |\mathcal{N}_l| = 2N/3 > (1-\rho) \times n. \end{cases}$$

Now let us explore the structure shown in Figure 1b, where the arm i cannot be detected as an outlier. In this arrangement, even by removing the entire upper-side



(a) Before removing cluster C_1

(b) After removing cluster C_1

Figure 2: Conditionally detectable intermediate outlier arm

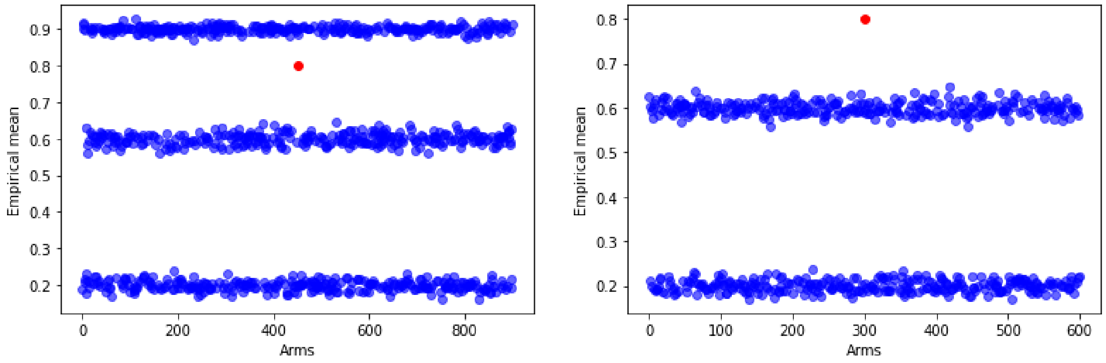
normal arms e.i. putting $\mathcal{N}_u = \{\emptyset\}$, still this outlier is invisible and not detectable.

By going through investigating all criteria in the definition 2.2 Constraint (1), we can see that except for conditions (a) and (b) none of the other conditions will be met. The reason lies in the fact that to meet these conditions the length of $\triangle_{min}(i, \mathcal{N}_l)$ should be $(1+\epsilon)$-times greater than $\Diamond^1(j, \mathcal{N}_l)$. However, it is obvious that the $\Diamond^1(j, \mathcal{N}_l)$ is equal to the distance of the cluster C_2 and C_3, which is bigger than the $\triangle_{min}(i, \mathcal{N}_l)$. This inconsistency is not even addressable by shrinking the size of $\mathcal{N}_l$ as small as the size of $\mathcal{N}_l = \{C_2\}$ or $\mathcal{N}_l = \{C_3\}$. Simply because either way although regarding the distance-wise conditions in Constraint (1) everything is perfect, Constraint (2) will be violated. If we remove any of the cluster C_2 or cluster C_3, we will have:

$$\begin{cases} \triangle_{min}(i, \mathcal{N}_l) > (1+\epsilon) \times \Diamond^1(j, \mathcal{N}_l), \quad \forall j \in C_2 \\ |\mathcal{N}_u| + |\mathcal{N}_l| = N/3 \ngeq \rho \times n \\ |\mathcal{N}_u| = 0 \\ |\mathcal{N}_l| = N/3 \ngeq (1-\rho) \times n. \end{cases}$$

which is against the criteria in Constraint (2) as displayed in Figure 3.

Under some assumptions, outlier arms with extreme expected values still could be invisible for this definition. The detection of these kinds of outlier arms depends on the length of their distance from the closest cluster to the outlier arm in comparison to the length of the distance of the central cluster from the closest cluster regarding to the outlier arm. Here, with respect to the arm i there is only one upper-side normal arms, $\mathcal{N}_u$ containing of all of cluster C_1, C_2, and C_3 and it size is $|\mathcal{N}_u| = N$. Now three cases could happen:



(a) Before removing cluster C_1

(b) After removing cluster C_1

Figure 3: Never detectable intermediate outlier

1. The distance of inter clusters lengths is $(1-\epsilon)$-times shorter than the length of arm i from the closet cluster, namely:

$$\begin{cases} \triangle_{min}(i,\mathcal{N}_u) > (1+\epsilon) \times \Diamond^1(j,\mathcal{N}_u), \quad \forall j \in C_2 \\ |\mathcal{N}_u| = N > \rho \times n \\ |\mathcal{N}_u| = N > (1-\rho) \times n, \end{cases}$$

which means the outlier arm i can be detected, as shown in Figure 4.

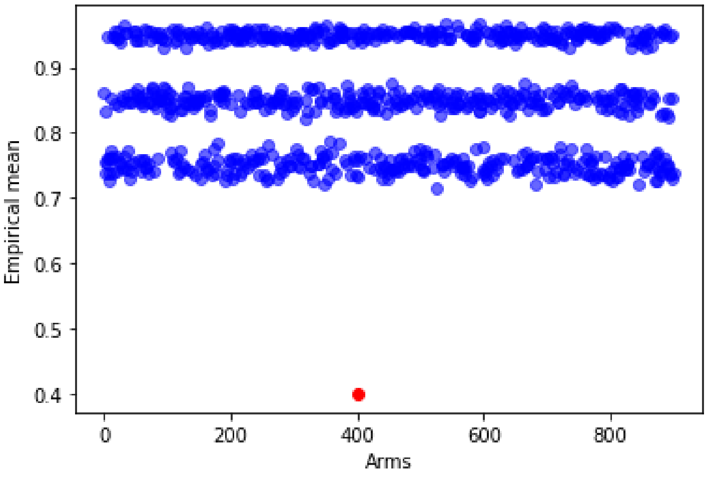
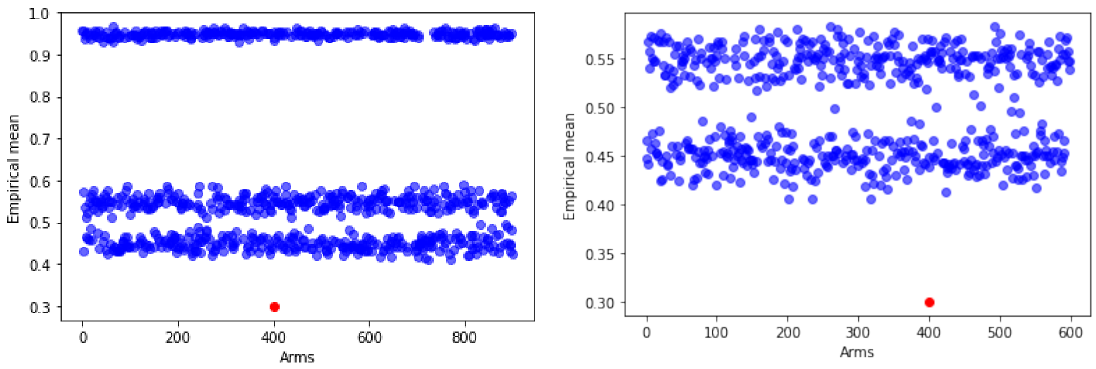


Figure 4: Detectable extreme outlier arm (red point).

2. The length of the distance between the closest and middle clusters is shorter than $(1-\epsilon)$-times the distance length of arm i from the closet cluster, but the

farthest cluster distance length from the central cluster is greater than $(1-\epsilon)$-times the gap length of arm i from the closest cluster. Since the measure of $\Diamond^1(j, \mathcal{N}_u)$ is equal to the longest distance between clusters, namely the distance size of the farthest and closest clusters with respect to outlier arm i, due to the violation of the constraint (a) in definition 2.2, the outlier arm cannot be detected as in Figure 5a. Yet, this issue can be managed by discarding the farthest cluster as the illustration in Figure 5b:

$$\begin{cases} \triangle_{min}(i, \mathcal{N}_u) > (1+\epsilon) \times \Diamond^1(j, \mathcal{N}_u), \quad \forall j \in C_2 \\ |\mathcal{N}_u| = 2N/3 > (\rho) \times n \\ |\mathcal{N}_u| = 2N/3 > (1-\rho) \times n, \end{cases}$$

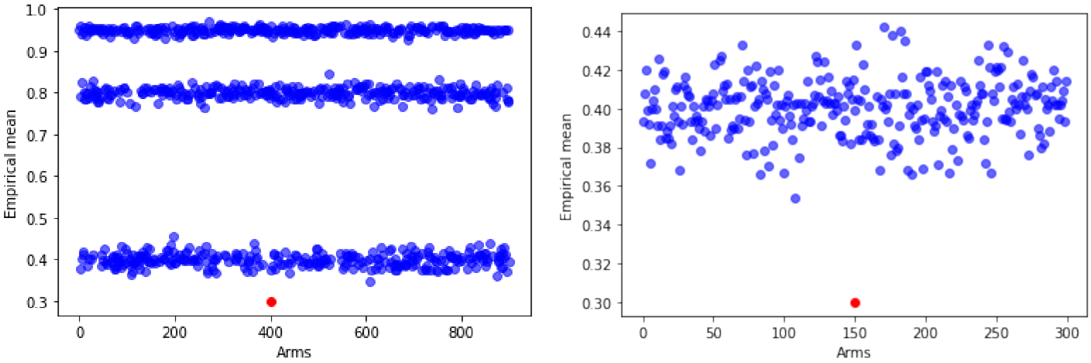


(a) Before removing furthest cluster (C_1) (b) After removing furthest cluster (C_1)

Figure 6: Outlier arm i in red is not possible to be spotted due to the effect of the furthest cluster in (a). Outlier arm i in red is detectable after removing the furthest cluster in (b).

3. The distance between the central cluster from the closest cluster is bigger than $1-\epsilon$-times the length of the arm i from the closet cluster. Similarly, since the measure of $\Diamond^1(j, \mathcal{N}_u)$ is equal to the longest distance between clusters, due to the violation of the constraint (a) in the definition 2.2, the outlier arm cannot be detected as shown in Figure 7a. Although dropping the most remote clusters will address the violation of Constraint (a), Unlike the former case, in this the outlier arm will be overlooked as shown in Figure 7b:

$$\begin{cases} \triangle_{min}(i, \mathcal{N}_u) > (1+\epsilon) \times \Diamond^1(j, \mathcal{N}_u), \quad \forall j \in C_2 \\ |\mathcal{N}_u| = N/3 \ngtr (\rho) \times n \\ |\mathcal{N}_u| = N/3 \ngtr (1-\rho) \times n, \end{cases}$$



(a)

(b)

Figure 8: Outlier arm i in red is not possible to be spotted due to the effect of the furthest cluster in (a). Outlier arm i in red is detectable after removing the furthest cluster in (b).

2.3 Pulling algorithm

Application of definition 2.2 is not efficient when there are too many arms and no prior knowledge about the ground truth. Because it gives arm same importance equally and investigates all of them. To address it and enhance the performance, a list of ranked arms Ω will be generated by a pulling algorithm 1 called Generic Outlier Detection ($GOLD$). The higher-rank elements of this set are more likely to be outlier arms.The algorithm takes ϵ, ρ, and δ as input and returns Ω, ranked list of arms as output. The ranking list is the result of the implementation of several procedures consisting of pulling arms, estimating empirical expected rewards, constructing upper confidence bound, creating a graph of neighborhood arms, forming an arms community, and updating the rank of arms in each iteration.

The mechanism and procedures of this algorithm have been supported by the following definition and theorems as preliminaries that are exactly as they are in [2].

Definition 2.3 ((ϵ, ρ)-outlier detection.)**.** Given ϵ, ρ *and* Ψ where $\epsilon > 0$ *and* $1 > \rho > 0.5$ identify a ranked list of all arms in Ψ denoted by Ω , such that for any $\hat{\mathcal{N}} \subset \Psi$, if $\hat{\mathcal{N}}$ is an (ϵ, ρ)-outlier arm group with respect to $\mathcal{N}_u$ and $\mathcal{N}_l$, it satisfies:

$$\begin{cases} rank(j) < rank(i), & \forall j \in \hat{\mathcal{N}}, \forall i \in \mathcal{N}_u \\ rank(j) < rank(i), & \forall j \in \hat{\mathcal{N}}, \forall i \in \mathcal{N}_l \end{cases}$$

where *rank(j)* is rank of arm j in Ω.

In order to practically establish this list, a few fundamental ingredients need to be introduced.

Definition 2.4 (Neighbor Arms)**.** Given two arms i and $j \in \Psi$ in the round T, they are considered as neighbor arms if:

$$|\hat{y}_i - \hat{y}_j| \leq b[\beta_i(m_i, \acute{\sigma}) + \beta_j(m_j, \acute{\sigma})]$$

where b is a coefficient function with regard to ϵ:

$$b = \frac{1 + e^{\frac{1}{16}} + \epsilon}{1 - e^{\frac{1}{16}} + \epsilon}, \quad and \quad \acute{\sigma} = \frac{6\sigma}{\pi^2 n T^2},$$

and $\beta_i(m_i, \acute{\sigma})$ is the Upper Confidence Bound (UCB) of $\hat{y}_i$ and depending on the distribution of arms are as follow:

(1) **Bounded distribution:**

$$\beta_i(m_i, \acute{\sigma}) = R\sqrt{\frac{-log(\acute{\sigma})}{2m_i}}$$

(2) **Bernoulli distribution:**

$$\beta_i(m_i, \acute{\sigma}) = Z\sqrt{\frac{\tilde{p}(1-\tilde{p})}{m_i}}, \quad \tilde{p} = \frac{\tilde{m_i}^+ + \frac{Z^2}{2}}{m_i + Z^2}, \quad and \quad Z = \mathrm{erf}^{-1}(1 - \acute{\sigma}),$$

where $\tilde{m_i}^+$ is the number of rewards that equal to 1 among m_i rewards and Z is the value of the inverse error function.

Definition 2.5 (Neighborhood Graph)**.** In round T, the neighborhood graph denoted as $G = (\Psi, E)$, is formed by Ψ where each node represents an arm and an unweighted and undirected edge exists between any pair of arms if they are neighbor arms.

Definition 2.6 (Arm Community)**.** In round T, the arm communities are formed by the connected components of G, denoted as $M = \{M_1, M_2, ..., M_k\}$ where M_i is an arm community formed by a connected component, $i = 1, ..., k$ and the size of M_i denoted by $|M_i|$ is the number of nodes in this connected component.

Inside the algorithm 1, two different terminal statuses have been defined; *Terminal status of an arm*, when they will not be pulled anymore by the algorithm, and the *terminal status of algorithm* when the number of arms at terminal status is less than $n \times (1-\rho)$. The correctness of these two statuses has been proven by the theorem 2.7 and 2.8. Later we will talk more in detail about the process of termination.

Theorem 2.7. *Give two arms $i, j \in \Psi$ and assume $\triangle_{i,j} = |y_i - y_j| > 0$. If i and j can be pulled in infinite times, i.e., $m_i \to \infty, m_j \to \infty$, then i, j will not be neighbor arms in the end.*

Theorem 2.8. *With probability at least $1-\delta$, the total number of pulls T needed to terminate for **GOLD** is bounded by:*

$$\mathrm{T} < 4\mathrm{D}_3 n \log(2\mathrm{D}_3 n) \log \sqrt{\frac{\pi_2 n}{6\delta}} + 2(n-1)$$

where:

$$\mathrm{D}_3 = \frac{2(b+1)^2 R^2}{\hat{\triangle}}$$

and

$$\hat{\triangle} = min_{i.j \in \Psi, i \neq j} |y_i - y_j|.$$

The mechanism of this algorithm consists of five steps:

Step 0: Initialization (Line 1-10).
First, before starting to pull arms, the following variables will be initialized and set to zero (Line 1-2):

- $\hat{\Psi}$, the list of arms that have reached the terminated status,
- T, number of total iteration,
- n, number of arms,
- $S[i]$, rank of each arms; it will be updated for arms during iteration,
- $\hat{y}_i$, empirical mean of arms,
- m_i, number of times that the arm i,has been pulled,
- β_i, upper confidence bound of arm i,

Then each arm will be pulled once (Lines 3-6). At the end of looping over arms, each arm's m_i will be updated to one, and T to n. Over the iteration of this loop, the $\hat{y}_i$ and $\beta_i(m_i, \acute{\delta})$ also are updated. It has to be noticed that the formula for updating $\beta_i(m_i, \acute{\delta})$ depends on the class of arm $i's$ distribution, according to 2.4.

Algorithm 1 GOLD Algorithm

```
Input: $\epsilon, \rho, \Psi, \delta$
Output: $\Omega$
$\hat{\Psi} \leftarrow \emptyset, T \leftarrow 0, n \leftarrow |\Psi|$
$\forall i \in \Psi, S[i] \leftarrow 0, \hat{y}_i \leftarrow 0, m_i \leftarrow 0, \beta_i(m_i, \acute{\delta}) \leftarrow 0$
for each $i \in \Psi$ do
    pull $i$ once
    $T \leftarrow T + 1, m_i \leftarrow m_i + 1$
    updates $\hat{y}_i, \beta_i(m_i, \acute{\delta})$
$G \leftarrow (\Psi, E = \emptyset)$
for each $i, j \in \Psi$ do
    if $|\hat{y}_i - \hat{y}_j| \leq b[\beta_i(m_i, \acute{\sigma}) + \beta_j(m_j, \acute{\sigma})]$ then
        $E \leftarrow E + \{e_{ij}\}$                    ▷ create an edge between i and j
while $|\hat{\Psi}| < n(1 - \rho)$ do
    $\mathcal{N} \leftarrow (\Psi - \hat{\Psi})$
    for each $i \in \mathcal{N}$ do
        pull arm $i$
        $T \leftarrow T + 1, m_i \leftarrow m_i + 1$
        updates $\hat{y}_i, \beta_i(m_i, \acute{\delta})$
        $G \leftarrow$ updatesG($G$)
        $M \leftarrow$ ConnectedComponents($G$)          ▷ return the communities
        $S, \hat{\Psi} \leftarrow$ Update$\hat{\Psi}$ andS($M, \hat{\Psi}, S, T$)
$\Omega \leftarrow \Psi$ according to $S$
return $\Omega$

procedure updatesG($G = (\Psi, E)$)
    for each $i, j \in \Psi$ do
        if $|\hat{y}_i - \hat{y}_j| > b[\beta_i(m_i, \acute{\sigma}) + \beta_j(m_j, \acute{\sigma})]$ then
            if $e_{ij} \in E$ then
                $E \leftarrow E - \{e_{ij}\}$            ▷ delete the edge between i and j
    return $G$

procedure Update$\hat{\Psi}$ andS($M, \hat{\Psi}, S, T$)
    for $i, j \in \Psi$ do
        if $|\mathcal{M}| < n(1 - \rho)$ then
            for each $i \in \mathcal{M}$ do
                $S[i] \leftarrow T$
                $\hat{\Psi} \leftarrow \hat{\Psi} \cup \{i\}$
    return $S, \hat{\Psi}$
```

In Line 7, $G(\Psi, E = 0)$, the graph of neighbor arms is created. E represents the possible edge between any pair of arms where,

$$\forall j, i \in \Psi, E_{i,j} \in \{0, 1\}.$$

In Lines 8-10, over a loop, the distance of each pair of arms (i, j) will be evaluated, and if it was according to the definition 2.4, the corresponding edge E_{ij} will be updated to one. F

Step 1: Arm Pulling (Line 11- 19).
The next step after building the primal variables is pulling arms successively. This procedure consists of two loops; one inside the other one. Pulling an arm will continue if it has not reached the *terminal status of arm*. This loop happens as the inner circle inside the outer loop. The outer loop also will go on before the coming *terminal status of the algorithm*, which is $|\hat{\Psi}| < n \times (1 - \rho)$. Pulling an active arm (the arm that has not terminated) will lead to sequential updates and recall procedures. The very first operation after updates is neighbor arms graph renewal (Line 17). This graph will be loaded into the next step, which is forming connected components of the arms community (Line 18). The last section of this loop is renovating the rank of arms and the list of terminated arms (Line 19).

Step 2: Updating neighborhood graph (Line 17 and 23-28)
For updating the graph, the distance of each pair of arms $(i, j) \in \Psi$ will be measured, and if it violates the condition in 2.4 and if they are already connected i.e $E_{ij} = 1$, the edge between them will be deleted. Although this module happens inside the loop over the set of arms that are not terminated $\mathcal{N} = \Psi - \hat{\Psi}$, it conducts the evaluation for all arms in Ψ.

Step 3: Updating $\hat{\Psi}$ and S (Line 19 and 30-36)
Inputs of this module are updated M, $\hat{\Psi}$, S, and T. Each element of M is a list of connected components. During looping over $\mathcal{M} \in$ M if its size was less than $n < (1 - \rho)$, for each arm in $\mathcal{M}$ the related rank S[i] will be set equal to the T and arm i will be considered terminated and added to $\hat{\Psi}$.

Step 4: Returning Ω (Line 20-21) .
Finally, the list of ranked arms S will be sorted increasingly. Thus, arms with lower rank are with probability at least $1 - \delta$, the outlier arms and this makes it easier to investigate them.

2.4 Expreminetal observation of Algorithm 1

The algorithm has been implemented according the following setup: $\epsilon = 5, \rho = 0.95, \delta = 0.05$ and $n = 40$ total arms.

Terminal status of arm for a given arm i happens as soon as the difference between the empirical mean of it and any other arms becomes greater than $b[\beta_i(m_i, \acute{\delta}) + \beta_j(m_j, \acute{\delta})]$. Empirically this condition for outlier arms meets sooner. Consequently, this yields bigger $S[i]$ for outliers. However, the algorithm will not be terminated. According to theorem 2.8, the termination of the algorithm depends on several parameters that among them, $\hat{\triangle} = min_{i,j \in \Psi, i \neq j}|y_i - y_j|$, and in practice $\triangle_{ij} = min_{i,j \in \Psi, i \neq j}|\hat{y}_i - \hat{y}_j|$, plays a vital role.

For instance, take the following setting: $\epsilon = 5, \rho = 0.95$ and $n = 40$ total arms, $|N_u| > n\rho$, and $|N_l| = 0$. Based on Constraint (2) size of N_u should be equal to or greater than 39. Now, Assume that we have exactly one outlier. As long as Constraint (1) gets satisfied, the termination status of the outlier arm will happen after some iteration, and the set of terminated arms $\hat{\Psi}$ will be updated and its size becomes $|\hat{\Psi}| = 1$. Although authors empathize that the algorithm is decisive and by the time outlier arms reached the terminal status, the algorithm will be terminated subsequently, this does not take place promptly. In practice, in line with algorithm 1, conditions in Line 12 and Line 32 will prevent the immediate termination of the algorithm from arising instantly after outlier arms termination. The condition in Line 32, $|M| < n(1 - \rho)$, means if the number of arms in the connected components M is less than $n(1 - \rho)$, here 2, these arms will be terminated. Since we have assumed that we have only one outlier and this outlier forms its own set of the connected component, since the condition in Line 32 is true, and therefore as the authors claimed the algorithm should stop pulling normal arms and terminate.

Notwithstanding, there is another requirement to meet. The number of terminated arms $|\hat{\Psi}|$, should comply with quality in Line 12, $|\hat{\Psi}| < n(1 - \rho)$. This is where the number of iterations will usually explode. The algorithm will not terminate sooner than having at least $|\hat{\Psi}| = 3$. Thus, this indicates the necessity of two more terminated arms. Hence, the algorithm has to maintain. According to theorem 2.7, it is clear that $\forall i, j \in \Psi$ if $\triangle_{ij} = |y_i - y_j| > 0$, when there is no restriction, all normal arms in the same group of neighborhood, will depart from their neighbors and form their own single member connected component and in consequence reach the terminal status of the arm. The very first two arms that meet the situation in Line 25 of the algorithm 1, will be added into $\hat{\Psi}$, and eventually, the algorithm will stop. The number of required total iterations, T, in this occasion as per has been mentioned in theorem 2.8 with the probability of $1 - \sigma$ is bounded.

As a deduction, in the worst-case scenario the minimum possible distance of two arms, $\triangle_{ij} = min_{i,j\in\Psi,i\neq j}|\hat{y}_i - \hat{y}_j|$ is determinative and T actually depends on that. Particularly, if the group of normal arms was so dense and close to each other, the number of iterations for removing an arm from them would be enormous.

However, the algorithm didn't reach the *terminal status of algorithm* unless by imposing some slight modification. This modification in Line 32, resulted in the termination of the algorithm as soon as the termination of outlier arms.

3 Conclusion

Anomaly detection and Multi-Armed Bandit problems have been studied individually by many researchers. However, this is not true when it comes to monitoring circumstances where these two concepts are blended. In this work, we tried to explore the Multi-Armed Bandit problem contaminated by some kind of anomalies. We tried to review cases where the anomalies are defined as *outlier*. Although it is clear that the taken approach in [2] is new and not very well-behaved when we try to expose it in the real world by implementation, regarding the definitions and ease of application it is compelling.

References

[1] Onur Atan, Cem Tekin, and Mihaela Van der Schaar. Global multi-armed bandits with hölder continuity. In *Artificial Intelligence and Statistics*, pages 28–36. PMLR, 2015.

[2] Yikun Ban and Jingrui He. Generic outlier detection in multi-armed bandit. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 913–923, 2020.

[3] Swapna Buccapatnam, Atilla Eryilmaz, and Ness B Shroff. Stochastic bandits with side observations on networks. In *The 2014 ACM international conference on Measurement and modeling of computer systems*, pages 289–300, 2014.

[4] Loc Bui, Ramesh Johari, and Shie Mannor. Clustered bandits. *arXiv preprint arXiv:1206.4169*, 2012.

[5] Stéphane Caron, Branislav Kveton, Marc Lelarge, and Smriti Bhagat. Leveraging side observations in stochastic bandits. *arXiv preprint arXiv:1210.4839*, 2012.

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[7] Richard Combes, Stefan Magureanu, and Alexandre Proutiere. Minimal exploration in structured stochastic bandits. *Advances in Neural Information Processing Systems*, 30, 2017.

[8] Samarth Gupta, Gauri Joshi, and Osman Yagan. Exploiting correlation in finite-armed structured bandits. *arXiv preprint arXiv:1810.08164*, 2018.

[9] Jiawei Han, Micheline Kamber, and Jian Pei. Data mining: Concepts and techniques [internet]. waltham, 2011.

[10] Tor Lattimore and Rémi Munos. Bounded regret for finite-armed structured bandits. *Advances in Neural Information Processing Systems*, 27, 2014.

[11] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[12] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[13] Shie Mannor and Ohad Shamir. From bandits to experts: On the value of side-observations. *Advances in Neural Information Processing Systems*, 24, 2011.

[14] Xuying Meng, Yequan Wang, Suhang Wang, Di Yao, and Yujun Zhang. Interactive anomaly detection in dynamic communication networks. *IEEE/ACM Transactions on Networking*, 29(6):2602–2615, 2021.

[15] Sandeep Pandey, Deepayan Chakrabarti, and Deepak Agarwal. Multi-armed bandit problems with dependent arms. In *Proceedings of the 24th international conference on Machine learning*, pages 721–728, 2007.

[16] Rahul Singh, Fang Liu, Yin Sun, and Ness Shroff. Multi-armed bandits with dependent arms. *arXiv preprint arXiv:2010.09478*, 2020.

[17] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

[18] Zhiyang Wang, Ruida Zhou, and Cong Shen. Regional multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 510–518. PMLR, 2018.

[19] Zhiyang Wang, Ruida Zhou, and Cong Shen. Regional multi-armed bandits with partial informativeness. *IEEE Transactions on Signal Processing*, 66(21):5705–5717, 2018.

[20] Honglei Zhuang, Chi Wang, and Yifan Wang. Identifying outlier arms in multi-armed bandit. *Advances in Neural Information Processing Systems*, 30, 2017.