Stochastic Multi-Armed Bandits With Zero-Variance Control Variates

Candidate Number:

NKHG3

Supervisor:
Dr François-Xavier Briol

A dissertation submitted in partial fulfillment

of the requirements for the degree of

MSc Data Science

of

University College London.

14709 words

Department of Statistical Science

University College London

November 23, 2021

Abstract

Reinforcement learning is a paradigm in machine learning, where the intelligent learners interact with the environment, and learn an optimal set of actions based on experience to tackle the tasks. Many algorithms in reinforcement learning are based on simulations, and Monte Carlo methods are ways of solving the reinforcement learning problem based on averaging sample returns. Therefore, the performance of these reinforcement learning algorithms relies on the performance of the Monte Carlo estimators. However, one drawback of the Monte Carlo simulation is that it requires a large number of samples, which may give estimators with high variance. Control variates are a well-established technique to reduce the variance of the Monte Carlo estimators. This thesis focuses on a certain class of reinforcement learning problems called multi-armed bandits, and it considers ways to improve the performance of this reinforcement learning algorithm via exploiting control variates. In particular, a set of control variates that are popular in the Bayesian computation literature are introduced to tackle the multi-armed bandits problems. Two algorithms named UCB-LZVCV and UCB-QZVCV that construct the control variates directly from the original data, exploit these constructed control variates to reduce the variance in the estimators and improve the performance of the stochastic multi-armed bandits algorithms are proposed. The experiments on synthetic data are presented to compare the two newly-proposed algorithms with the existing algorithms, and the results validate a significantly improved performance via using the proposed algorithms.

Acknowledgements

I have received a great deal of support and assistance along the way on this master thesis journey.

First and foremost, I am deeply grateful to my supervisor, Dr. François-Xavier Briol, for his invaluable advice and extraordinary dedicated support. You have made me feel excited and ambitious about my work. Your immense knowledge, authoritative guidance, and valuable feedback have evaluated this thesis, encouraged and lifted my academic research and daily life. You open my mind to new possibilities, and I am extremely grateful to work with you on this journey.

Furthermore, I would like to express my sincere gratitude to Arun Verma and Zhuo Sun for their inspiring discussions and insightful comments. Both of you are warm-hearted people, and I appreciate your time and effort in supporting me.

Last but not the least, my appreciation also goes out to my family and boyfriend for their unwavering support and belief in me. You are always there for me, your love and tremendous understanding have supported me through unprecedented times. Thank you.

Contents

Abbreviations

CI confidence intervals. 11, 19, 24, 33, 34, 36, 38, 40, 48, 56, 58, 60, 81, 85, 89

CLT central limit theorem. 11, 17, 19, 40, 49, 50

CV control variate. 8–10, 15, 18, 20–28, 33–43, 45, 46, 49, 55–62, 64–68, 79, 81

DP dynamic programming. 13, 14

IID independent and identically distributed. 15, 17, 25, 28, 30, 34, 35, 77, 79, 83, 88

LLN law of large numbers. 11, 16

LZV-CV linear zero-variance control variates. 46–50, 82–85

MAB multi-armed bandits. 8–10, 22, 26, 28, 29, 31–36, 40–42, 46, 51, 55, 57–59, 64–68, 82, 86

MAB-CV multi-armed bandits with control variates. 9, 26, 27, 34–36, 40, 41, 62

MC Monte Carlo. 8–27, 31, 41–43, 57, 60, 64, 66, 67

MDP Markov decision process. 13, 14

QZV-CV quadratic zero-variance control variates. 50–54, 86–89

RL reinforcement learning. 8, 10–12, 15, 21, 26, 27, 66

RMSE root mean square error. 17, 18

UCB Upper Confidence Bounds. 26, 30–33, 39, 41, 42, 67, 68

UCB-CV Upper Confidence Bounds with Control Variates. 9, 10, 26, 36, 38–40, 46, 55, 57, 59–62, 64–68, 79, 80

UCB-LZVCV Upper Confidence Bounds with Linear Zero-Variance Control Variates. 9, 10, 46, 49, 50, 52, 55–60, 62, 64, 65, 67, 83, 84

UCB-QZVCV Upper Confidence Bounds with Quadratic Zero-Variance Control Variates. 9, 10, 50, 52, 53, 55–60, 62, 64, 65, 67, 88, 89

ZV Zero Variance. 42, 67

ZV-CV zero-variance control variates. 9, 10, 41–43, 46, 50, 56, 57, 59, 60, 64–68

Chapter 1

Introduction

In the era of Big Data and cloud computing, statistical models have become more complex and computationally expensive. This is a big challenge in fields like statistical modelling, machine learning, and reinforcement learning (RL), which are often required to solve integration problems to perform inference and make predictions. However, exact inference can be intractable for most probabilistic models of practical interest. Therefore, we must resort to some form of approximation [1].

One typical solution is the Monte Carlo (MC) method, which is often applied to solve integration problems [2]. The fundamental concept of MC strategies is to learn about a system through simulation methods with random sampling. In particular, it randomly draws independent samples from the desired probability distribution, repeats this process many times and takes their sample average to estimate the target quantity.

The MC simulation methods are flexible and easy to apply. However, one drawback is that it needs to compute lots of simulations to reduce the corresponding estimation error. A control variate (CV) is a well-established and effective variance reduction technique [3] for the MC simulation of sophisticated systems [4][5][6]. By exploiting CVs in the MC approaches, it significantly reduces the variance of the MC estimators and enhances accuracy by orders of magnitude [7][8].

The use of CVs can be found across the sciences including finance [9][10][11], statistics [12][13][14], machine learning [15][16] and RL [17][18][19]. This thesis focuses on the application of CVs in a certain class of RL problems named multi-

armed bandits (MAB) [18].

In the original MAB problem, the learner is given a slot machine with several arms, and each arm has its own probability of receiving the reward, which is unknown to the player. By pulling one of such arms, the learner observes an independent and identically distributed random reward from a probability distribution specific to that arm. The objective for the learner is to maximize the sum of rewards collected through a sequence of arm pulls. In the MAB algorithm, MC methods are often used to generate samples according to the desired probability distributions, and take their sample averages to estimate the mean reward for each arm. However, it generally requires a large sample size and leads to high variance in these MC estimators. To reduce the variance of the MC estimators and use the available side information in the form of CVs in the environment, Verma and Hanawa consider a new variant of the bandit problems called multi-armed bandits with control variates (MAB-CV). Their originated algorithm Upper Confidence Bounds with Control Variates (UCB-CV) incorporates variance estimates, uses the estimators based on the linear CVs, and successfully improves the performance of MAB algorithms.

The UCB-CV algorithm requires that the mean of the correlated CV is known. However, we often experience the case where we don't know what a good CV is. Or simply, the mean of the available correlated auxiliary information in the MAB problem is unknown. In such settings, instead of using side information, it might be interesting to construct a good CV from the original dataset [13]. Motivated by this fact, two algorithms named Upper Confidence Bounds with Linear Zero-Variance Control Variates (UCB-LZVCV) and Upper Confidence Bounds with Quadratic Zero-Variance Control Variates (UCB-QZVCV) are proposed, which construct the zero-variance control variates (ZV-CV) directly from the original data using the derivatives of the log-likelihood, adopt these constructed ZV-CV to estimate the mean rewards, and seek for improvement in the performance of the MAB algorithms.

The organization of this thesis is as follows. The literature on the MC method is reviewed in chapter 2. Next, an effective variance reduction technique for the

MC estimators called CV is examined in chapter 3. Subsequently, a recent application of CVs in the bandit problems is presented in chapter 4. In this chapter, the UCB-CV algorithm that exploits the available side information in the form of CVs to obtain mean estimates with smaller variance and improve the performance of the stochastic bandits algorithm is also introduced. Motivated by these existing literature, two newly-developed algorithms UCB-LZVCV and UCB-QZVCV are proposed in chapter 5. The results of numerical experiments spanning different algorithms are demonstrated in chapter 6. The implications and limitations are discussed in chapter 7. Finally, chapter 8 concludes and suggests directions for future research.

The main contributions of this thesis are as follows:

- The RL problem, and its connection to the MC methods and CVs technique are presented in an intuitive manner with motivating examples ranging from Blackjack to news push.

- Two algorithms named UCB-LZVCV and UCB-QZVCV are proposed. These newly-developed algorithms directly construct the ZV-CV from the original dataset, exploit these constructed ZV-CV to reduce the variance of the mean reward estimators, and successfully improve the performance of the stochastic MAB algorithms.

- The performance of UCB-LZVCV and UCB-QZVCV are validated on synthetically generated data in comparison with two existing well-performed algorithms.

- The author took the initiative to code the UCB-LZVCV and UCB-QZVCV algorithms from scratch, and the initial code for these algorithms are available on this Github repository. If it is interesting for the community, it can be written up as a package.

Chapter 2

Monte Carlo Methods

This chapter reviews literature on the MC methods. The fundamental idea of MC strategies is to learn about a system through simulation methods with random sampling. In a simple MC problem, the goal is to estimate a population expectation by the corresponding sample expectation. MC method is a powerful and flexible approach, and it is proven to be useful in many fields of study including applied statistics [20], physical sciences [21], RL [22][23]. In this chapter, we first present the MC methods in an intuitive manner with some motivating examples. Next, the mathematical properties of this method is reviewed in section 2.2 to understand when and why these MC methods work. Section 2.3 presents the methods accuracy using the law of large numbers (LLN) and central limit theorem (CLT). In the end, the error estimation (i.e. the confidence intervals (CI) for the MC estimator) is evaluated in section 2.4.

2.1 Motivations

This section presents why and how MC methods can be applied in statistics and RL with some motivating examples.

In statistics, the MC method is widely used to solve problems that are hard to be solved analytically. One popular application of the MC methods is in numerical integration [24]. Deterministic analytical integration algorithms like simple Riemann integration [25], trapezoidal rule [26] and Simpson's rule [24] work reasonably well when dealing with regular integration problems in small dimensions.

However, due to the curse of dimensionality, it encounters some issues in high dimensions as stressed by Thisted [27]. MC methods offer a solution to avoid the exponential growth in computation and calculation time. It randomly draws independent samples from the desired probability distribution many times and takes their sample average to estimate the target integral. Section 2.3 has further proved that the accuracy of the MC method stays the same regardless of the number of dimensions[28]. For this reason, the MC approach has been considered in many statistical applications.

The MC method is also widely used in RL to estimate value functions and discover optimal policies [22]. Before discuss this topic in details, we first introduce several terminologies of a RL system: *agent*, *environment*, *state*, *action*, *reward*, *policy*, *reward signal*, *value function*, and optionally *model* of the environment [22]. An *agent* is the learner or decision-maker in the RL problem. The *environment* is the thing that interacts with the agent, and it compromises everything outside the agent. The *state* describes the current environment of the task. For example, suppose a robot aims to learn how to walk, then the state is the position of its legs. *Action* is what the agent can do in each state. *Reward* subsequently describes feedback from the environment. A *policy* defines the learner's way of behaving at a given time. A *reward signal* defines the goal of a RL problem. The environment sends to the agent a single number (i.e. *reward*) in each time step. The agent then aims to maximize the total received reward over the long run. The *reward signal* shows desirable actions in an immediate sense. A *value function* then defines what is good over the future. The *value* of a state is the total aggregate reward that an agent can expect in the long run, starting from that state. Another important element of an RL system is a *model* of the environment, it is something that replicates the behaviour of the environment.

Researchers in RL often adopt MC methods to learn the state-value function under a given policy. In particular, the MC methods estimate from experience, average the observed returns after visits to that state, then use it to reflect the strategy rewards. For example, we can see the application of the MC method in the popular

casino card game Blackjack. The aim is to obtain cards with a total sum that is as great as possible without exceeding 21 (*going bust*). The rules for this game can be found in example 5.1 of this book [22], here we only discuss how MC methods can be adapted in this scenario. Playing blackjack can be formulated as an episodic finite Markov decision process (MDP), as it can be expressed by a mathematical framework that models the decision making in settings where outcomes are partially random and partially controlled by a decision-maker. Each blackjack game is like an episode. Winning, losing, and drawing one game receives a reward of +1, -1, and 0 respectively. Assume each player completes independently against the dealer, and the cards come from an infinite deck with replacements.

Consider the dealer *hits* (request for additional cards, one by one) or *sticks* (stops asking for cards) based on a fixed strategy: sticks if the dealer's sum is 17 or greater; hits otherwise. The player has two possible actions to take: *hits* or *sticks*, and he makes the decisions based on three factors: the player's current sum (12–21), the dealer's one showing card (ace–10), and whether or not he holds a usable ace (i.e. counts his ace as 11 without going *bust*, which means a sum that exceeds 21). This brings the total number of states to 200.

Suppose the policy for the player is to stick for a sum of 20 or 21, and to hit otherwise. To obtain the state-value function for this policy, one could consider a MC approach; simulating many blackjack games using the policy and averaging the returns following each state. As a result, we were able to acquire the approximations of the state-value function as illustrated in Figure 2.1 (code that used to produce Figure 2.1 can be found in this Github repository). The estimates for states containing a usable ace are less certain and less regular, as these states are often less common. In any case, the value function is well estimated after 500,000 games.

The question then comes: what is the advantage of using MC methods in this blackjack setting? As mentioned, playing blackjack is formulated as an episodic finite MDP by its nature. Therefore, it appears that we may simply apply dynamic programming (DP) methods to solve this problem, as DP methods are a set of approaches for computing optimal policies given a perfect model of the environment

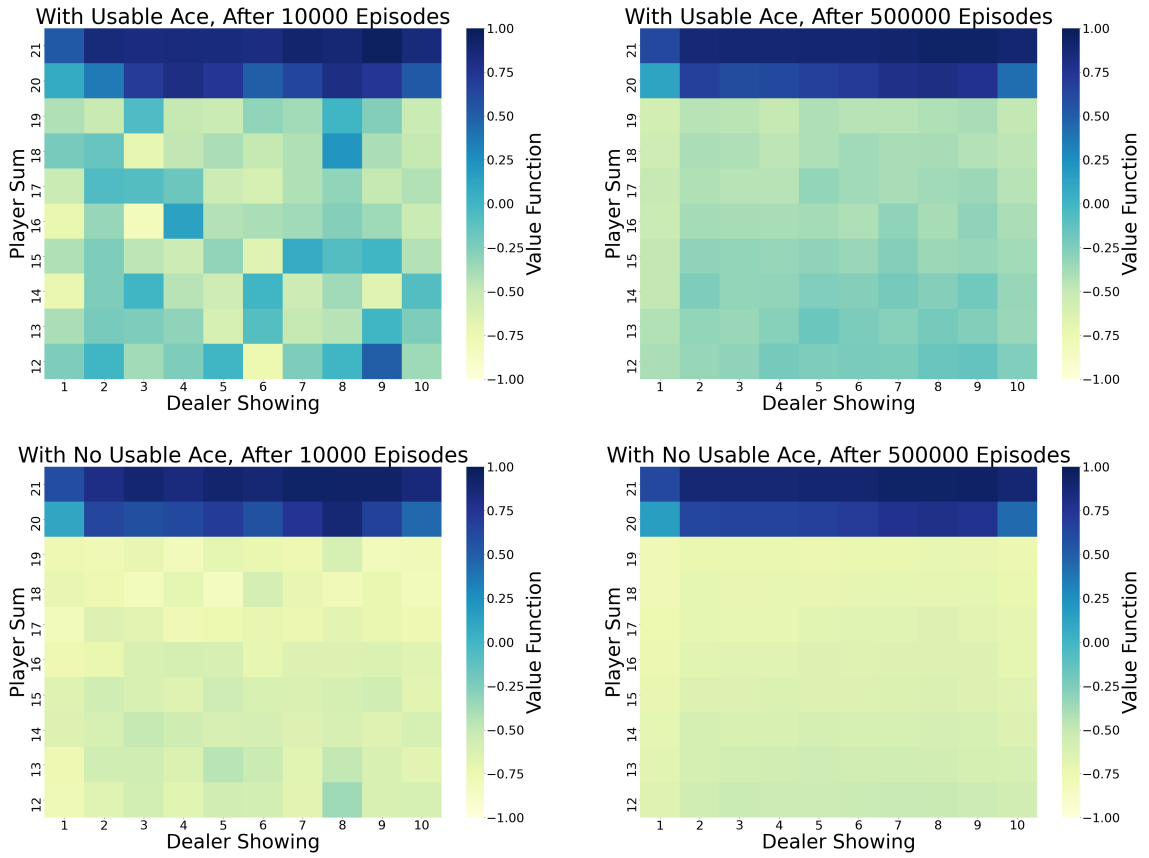

Figure 2.1: Approximate state-value functions for the blackjack policy that sticks only on 20 or 21, computed by MC policy evaluation.

as a MDP. However, computing the value function with DP methods may be hard even with the complete knowledge of the environment in this task, as it requires the distribution of next events which is hard to obtain for this blackjack example. Imagine the player has a sum of 14 and he decides to stick. What is his probability of ending this game with a reward of +1 as a function of the dealer's showing card? Before applying DP methods, one must compute all of the probabilities. Therefore, the computations can be extremely complicated and easily prone to errors. Simulating the sample games with MC methods, on the other hand, is simple. It can learn from simulated experience. Hence, one considerable advantage of the MC method is its capacity to deal with sample episodes alone even when one has comprehensive knowledge of the environment's dynamics. Furthermore, the computational cost of estimating the value of a single state is independent of the number of states. If only one or a subset of the states' value needs to be estimated, the MC approaches can be

particularly appealing. In this scenario, we can simply simulate numerous sample episodes starting from the states of interest, average returns from only these states and ignore all others.

However, one drawback of solving the blackjack problems with MC methods is that we need to compute lots of simulations, which may result in a big estimation error. To reduce the MC variance, one could consider applying the CV technique. The details of the formalization must wait until chapter 3, but the basic idea is simply to estimate the target quantity using the known correlated CVs, which requires fewer simulations and potentially leads to a more accurate result.

Another RL example that uses the MC method can be found in the news push system. The news push is a typical application of the recommender system. It is known that the recommender system has become a popular service that helps users to find interesting items. Therefore, a news company often needs to estimate the reader's click rate for the available news articles to discover the optimal recommending strategy. In such settings, the MC simulations can be used to model the probability of different outcomes and takes the sample averages as an estimator for the expected value of the click rate.

We have briefly discussed the motivations and intentions behind the MC method with practical applications, next the formal mathematical properties of the MC methods will be presented in detail.

2.2 Mathematical Properties

In this section, we present the mathematical properties of the MC methods. The presentation is inspired by that of Art Owen in this book [2], and we also use some intuitive descriptions to explain how these mathematical notations apply in the motivating RL examples.

In a MC question, the interest is in evaluating the expected value of a function $f(X)$ of the parameters $X \in D \subseteq \mathbb{R}^d$, for example, $\mu = \mathbb{E}(f(X))$. It then generates independent and identically distributed (IID) samples $\{X_1, \ldots, X_m\}$ from the distribution of X, and estimates the target quantity $\mu = \mathbb{E}(f(X))$ by their empirical

average:

$$\hat{\mu}_m = \frac{1}{m}\sum_{i=1}^{m} f(X_i). \tag{2.1}$$

With the assumption that μ exists, $\hat{\mu}_m$ almost surely converges to $\mu = \mathbb{E}(f(X))$ by the LLN. This is the primary justification for MC methods.

In the previous Blackjack example, we find the state-value function of the given policy by a MC method. In particular, many blackjack games are simulated using the policy and the averages of returns following each state are used as an estimator for the value function. Therefore, in this setting, $f(X_i)$ can be the sample return following each state under the given policy, μ is the state-value function and $\hat{\mu}_m$ is the estimates of the state-value function, which is the average sample returns in this case. In the news push example, $f(X_i)$ can be the generated click rates samples, μ can be the expected value of the click rates, and $\hat{\mu}_m$ can be the average of the click rates samples.

Commonly, $X \in D \subseteq \mathbb{R}^d$ is a random variable whose distribution has a probability density function $p(x)$, and f is a real-valued function defined over D. Therefore, $\mu = \int_D f(x)p(x)\,dx$. In this case, we can work directly with integrals to solve μ, or we may apply the MC estimator to find out this expectation as stated in equation 2.1. In settings where X is a discrete random variable whose distribution has a probability mass function $p(x)$, our parameter of interest can be written as $\mu = \sum f(x_i)p(x_i)$. For such scenarios, the input X could simply be an image. With the condition that $f(X)$ is a quantity that can be averaged (e.g. a real number or vector), we can still apply the MC methods to estimate μ. Hence, in both settings, we are able to use the MC estimator as stated in equation 2.1 to estimate the target quantity μ. This feature makes the MC methods become popular in many applications.

The LLN say $\hat{\mu}_m$ almost surely converges to $\mu = \mathbb{E}(f(X))$ as mentioned before. Although LLN suggests that MC will eventually yield a small error, it does not tell us how large m must be for this to hold. Also, this LLN does not tell if the error is likely to be sufficiently small for a given sample $X_1, \ldots, X_m$. Luckily, when $f(X)$ has a finite variance (i.e. $var(f(X)) = \sigma^2 < \infty$), the situation improves dramatically. In

IID sampling $\hat{\mu}_m$ is a random variable with its own mean and variance. In particular, the mean of $\hat{\mu}_m$ is

$$\mathbb{E}(\hat{\mu}_m) = \mathbb{E}(\frac{1}{m}\sum_{i=1}^{m} f(X_i)) = \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}(f(X_i)) = \mu. \tag{2.2}$$

This suggests the MC estimator is unbiased. By elementary manipulations, the variance of $\hat{\mu}_m$ is

$$var(\hat{\mu}_m) = E((\hat{\mu}_m - \mu)^2) = \frac{\sigma^2}{m}. \tag{2.3}$$

For large m, the CLT suggests that $\hat{\mu}_m - \mu$ is approximately normally distributed with mean 0 and variance $\frac{\sigma^2}{m}$, which yields convergence in distribution of the error:

$$\sqrt{m}(\hat{\mu}_m - \mu) \xRightarrow{m\to\infty} N(0, \sigma^2) \tag{2.4}$$

where $\Rightarrow$ denotes convergence in distribution, $\hat{\mu}_m = \frac{1}{m}\sum_{i=1}^{m} f(X_i)$ and $f(X_1), f(X_2), \ldots, f(X_m)$ are IID random variables with mean μ and finite variance $\sigma^2 > 0$ [24].

2.3 Methods Accuracy

According to the variance of $\hat{\mu}_m$, it is clear that the answer could get worse if the variance of $f(X)$ increases, and it could get better if the sample size m increases. Equation 2.3 shows the exact rate of exchange. The root mean square error (RMSE) is the standard deviation of the residuals (prediction errors). It is often used to assess the accuracy of the method by measuring how spread out the residuals are. To access the accuracy of the simple MC method, we calculate the RMSE of $\hat{\mu}_m$ and that is $\sqrt{\mathbb{E}((\hat{\mu}_m - \mu)^2)} = \frac{\sigma}{\sqrt{m}}$.

De-emphasize σ, we have RMSE= $O(m^{-1/2})$ as $m \to \infty$. Improving the estimation by one extra decimal digit of accuracy is equivalent to requesting one-tenth of the original RMSE. This would require one hundred times as much computation. The estimation could be improved with a larger sample size m, but the computing time also increases with m.

In small dimensional problems, the $m^{-1/2}$ rate may seem to be slow. For instance, when the number of dimensions = 1, the Simpson's method can integrate a function with an error rate $O(m^{-4})$ [2]. However, this error rate for MC holds regardless of the dimensions, which is considered as a striking feature of the MC method, and we will discuss this feature at the end of this section.

Another way to improve the efficiency and accuracy of the MC simulation is to decrease the variance of the estimator. Reducing σ^2 by a factor of two while keeping μ unchanged, we gain the same amount as we could via doubling m. Therefore, many variance reduction techniques have been designed for it. In particular, we will discuss a method named CV in section 3.

One highlight for the error rate of the MC estimator is that it holds regardless of the dimensions, in which case it can be fast in high dimensional settings compare to other methods like Riemann integration and Simpson's rule [24][27]. For example, increasing the number of sampled points by 4 times could halve the error in the MC estimators regardless of the number of dimensions. Therefore, in the real-world applications with high dimensional situations, the RMSE is still $\frac{\sigma}{\sqrt{m}}$ for a MC estimator. In addition, with MC algorithms, we are usually able to make idealized assumptions, for example, specific distributional forms. To this end, another advantage of the MC approach is it allows us to consider real-world complexity in our computations.

2.4 Error Estimation

One advantage of the MC method is that the sample values can be used to get an estimation of the error $\hat{\mu}_m - \mu$. In MC sampling, the average squared error is $\frac{\sigma^2}{m}$. We rarely know σ^2, but we can easily estimate it based on the sampled points. The most common estimators of σ^2 are

$$\hat{\sigma}^2 = \frac{1}{m}\sum_{i=1}^{m}(f(X_i) - \hat{\mu}_m)^2, \tag{2.5}$$

and,

$$s^2 = \frac{1}{m-1}\sum_{i=1}^{m}(f(X_i) - \hat{\mu}_m)^2. \tag{2.6}$$

where s^2 is an unbiased estimator for the variance σ^2 (i.e. $\mathbb{E}(s^2) = \sigma^2$ for $m \geq 2$). Both $\hat{\sigma}^2$ and s^2 will appear in the variance estimators that we use in this thesis.

For now, let's consider the variance estimator s^2, the MC estimation error is therefore on the order of $\frac{s}{\sqrt{m}}$. The MC estimator $\hat{\mu}_m$ has mean μ and its variance can then be estimated as $\frac{s^2}{m}$. In this way, we can evaluate the error estimation of the MC estimator $\hat{\mu}_m$ by calculating the CLT-based approximated CI for the $\hat{\mu}_m$.

With this CLT theorem (equation 2.4) and the variance estimator s^2 (equation 2.6), then for all $z \in \mathbb{R}$:

$$\mathbb{P}\left(\sqrt{m}\cdot\frac{\hat{\mu}_m - \mu}{s} \leq z\right) \to \phi(z), \tag{2.7}$$

as $m \to \infty$. Here the normal quantile function ϕ^{-1} maps (0,1) onto $\mathbb{R}$. The notations and conditions follows the same as equation 2.4. Therefore, the CLT-based approximate $100(1-\alpha)$ percent CI for the MC estimator $\hat{\mu}_m$ can be written as follows:

$$\hat{\mu}_m \pm \phi^{-1}(1-\frac{\alpha}{2})\cdot\frac{s}{\sqrt{m}}. \tag{2.8}$$

It is also common to replace the normal quantile $\phi^{-1}(1-\frac{\alpha}{2})$ by one from Student's t distribution with $m-1$ degrees of freedom. If the sampled $f(X_i)$ are normally distributed, then the intervals

$$\hat{\mu}_m \pm t_{m-1}^{1-\alpha/2}\cdot\frac{s}{\sqrt{m}} \tag{2.9}$$

have exactly $1-\alpha$ probability that containing μ. Here, $t_{m-1}^{1-\alpha/2}$ denotes the $1-\alpha/2$ quantile of the t-distribution with $m-1$ degrees of freedom. MC applications usually require large number of samples m, therefore there is no practical difference between the t-based CI from equation 2.9 and CI based on equation 2.8.

Chapter 3

Control Variates

Monte Carlo integration typically has an error variance $\frac{\sigma^2}{m}$ as introduced in chapter 2. To get a more accurate answer, one could consider increasing the sample size m. However, the computing time also increases with m as stated in section 2.3. Reducing the variance of the MC estimator by increasing m seems to be infeasible due to the cost of sampling from $p(X)$ and potentially the cost of evaluating $f(X)$ [29]. To reduce the error variance and improve the efficiency of MC simulation, one can consider ways to decrease σ instead. Methods that are designed to solve this problem are known as variance reduction techniques.

There are plenty of variance reduction methods that can be used to improve the accuracy and efficiency of MC simulation [24]. For example, strategies including antithetic sampling [30], stratification [31] and common random numbers [32], improve its efficiency via strategically sampling the input values with more care. Other techniques like conditioning [33] and CVs [24][7] get better estimates by exploiting closed-form results (closed-form results are results that you can write down the exact value of the quantity without approximation). In addition, importance sampling [34] is also considered as a variance reduction method. It modifies where we collect the sample values by purposely oversampling from some regions and then re-weighting to adjust for this distortion. In particular, this chapter focuses on a technique called CV, which significantly reduces the variance of the MC estimate of integrals and enhances accuracy by orders of magnitude [7].

This chapter is organised as follows. The intuitions behind CVs are discussed

with motivating applications in section 3.1, then the mathematical principle for the CV technique is reviewed in section 3.2. Subsequently, section 3.3 presents whether a CV successfully improves the accuracy and efficiency of the MC methods. In the end, a simple numerical example is used in section 3.4 to illustrate that the variance of the MC estimator can be significantly reduced by applying a CV.

3.1 Motivating Examples

The use of CVs can be found in various fields of study including finance [9], variational inference [14], bayesian computation [12], stochastic optimization [35], machine learning [15][16] and RL [17][18][19].

Take the Blackjack example that we discuss in section 2.1, previously, we apply MC method to find the state-value function of a given policy by simulating many Blackjack games with that given policy and averaging the returns following each state. The results in Figure 2.1 shows that the MC simulations require large number of the sample size (500,000 games) to get a well-estimated state-value function. The question then comes: are we able to apply the CV technique in our Blackjack example to reduce the variance of the MC estimators? The answer is yes. The player considers the dealer's showing card as an influencing factor when making decisions, which is related to winning or losing. Thus, the sum of dealers showing cards may correlate with the state-value function. In addition, the expected value for this sum can be available from historical data. For these reasons, the sum of dealers showing cards can be treated as a CV.

In addition, we can also find available CVs in the news push example. For instance, the number of current likes for each article can be a CV as it correlates with the click rates and the mean value for these current likes is often available in the environment. Therefore, we can exploits this number of current likes in the form of CV to estimate the mean click rate and hopefully it reduces the variance of the original MC estimators.

There are more areas in RL where using CVs are promising. For example, some applications adopt CVs for variance reduction in Monte-Carlo Tree Search

[19], which is a powerful planning approach for decision-making in single-agent and adversarial environments. In addition, CVs for variance reduction in policy gradient methods [17] and bandits problems [18] have also been examined by many researchers. In particular, chapter 4 reviews the literature on MAB problems, and studies why and how CVs can be used in the bandits settings.

We have discussed the intuitions behind CVs with some motivating applications. Next, the mathematical properties of this method are reviewed to understand when and why CVs can be used to reduce the variance of the MC estimators.

3.2 Mathematical Properties

A CV is one type of variance reduction technique that is often used in MC algorithms. As before, let X be a random variable whose distribution has a probability density function p where $X \in D \subseteq \mathbb{R}^d$. Suppose the unknown quantity of interest is $\mu = \mathbb{E}(f(X))$. In comparison with the MC estimator, an improved estimate of μ can be constructed if we have access to another statistic $h(X)$ that is correlated with $f(X)$, provided the value $\theta = \mathbb{E}(h(X))$ exists and is known. Let $\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} f(X_i)$ and $\hat{\theta} = \frac{1}{m}\sum_{i=1}^{m} h(X_i)$, then μ can be estimated by the **difference estimator**

$$\hat{\mu}_{diff} = \frac{1}{m}\sum_{i=1}^{m}(f(X_i) - h(X_i)) + \theta = \hat{\mu} - \hat{\theta} + \theta. \tag{3.1}$$

We know that $\mathbb{E}(\hat{\theta}) = \theta$, therefore, $\mathbb{E}(\hat{\mu}_{diff}) = \mu$, which says $\hat{\mu}_{diff}$ is also an unbiased estimator for μ. The variance of $\hat{\mu}_{diff}$ is

$$var(\hat{\mu}_{diff}) = \frac{1}{m} var(f(X) - h(X)). \tag{3.2}$$

If $f(X) - h(X)$ has a smaller variance than $f(X)$, we can expect decreased variance by introducing the difference estimator $\hat{\mu}_{diff}$, and this random variable $h(X)$ with known mean is the **CV**.

For the Blackjack example in section 3.1, $h(X)$ can be the sum of the dealer's showing cards, thus θ represents the expected value of this sum, $h(X_i)$ represents one sample in the sum of the generated dealer's card, m represents the total number

of the generated samples and $\hat{\theta}$ represents the average of these sum samples. The interpretations for $f(X), \mu, \hat{\mu}$ follow the same as discussed in section 2.2. As for the news push example, $h(X)$ can be the number of likes for each article, $\hat{\theta}$ can be the expected value of the likes, and $\hat{\theta}$ can be the sample average of the generated likes.

Note that the difference estimator $\hat{\mu}_{diff}$ is not the only way of using a CV, it can be used in various forms. The most common one is called the regression estimator. For any choice of the coefficient $\alpha \in \mathbb{R}$, μ can be estimated by the **regression estimator**

$$\hat{\mu}_\alpha = \frac{1}{m}\sum_{i=1}^{m}(f(X_i) - \alpha \cdot h(X_i)) + \alpha \cdot \theta = \hat{\mu} - \alpha(\hat{\theta} - \theta). \tag{3.3}$$

When $\alpha = 0$, it's just the simple MC estimator as discussed in chapter 2. When $\alpha = 1$, it is equivalent to the difference estimator (equation 3.1). $\mathbb{E}(\hat{\mu}_\alpha) = \mu$ for all α since $\mathbb{E}(\hat{\theta}) = \theta$, therefore the regression estimator is also an unbiased estimator for μ. The variance of this regression estimator is

$$var(\hat{\mu}_\alpha) = \frac{1}{m}(var(f(X)) - 2\alpha \cdot cov(f(X), h(X)) + \alpha^2 \cdot var(h(X))). \tag{3.4}$$

By differentiating, the optimal coefficient is

$$\alpha_{opt} = \frac{cov(f(X), h(X))}{var(h(X))}. \tag{3.5}$$

Substitute this optimal coefficient α_{opt} back to the variance formula (equation 3.4), the resulting minimum value of the variance is given by

$$var(\hat{\mu}_{\alpha_{opt}}) = \frac{1}{m}\left(var(f(X)) - \frac{[cov(f(X), h(X))]^2}{var(h(X))}\right) = \frac{\sigma^2}{m}(1 - \rho^2). \tag{3.6}$$

where $\rho = Corr(f(X), h(X))$ is the correlation coefficient of $f(X)$ and $h(X)$, $\sigma^2 = var(f(X))$. In regression estimator, any CV that correlates with f will be helpful in reducing the variance. Greater value of $|\rho|$ leads to a greater achievement in variance reduction by applying a CV. The variance of the regression estimator with a CV is $\frac{\sigma^2}{m}(1 - \rho^2)$. This is never worse than $\frac{\sigma^2}{m}$ and usually better.

In practice, the quantities of $cov(f(X),h(X))$ and $var(h(X))$ are often unknown. As a result, these must be approximated from the data to find the best estimation for α_{opt}. One possible solution is to estimate $cov(f(X),h(X))$ and $var(h(X))$ with the sample covariance $\hat{cov}(f(X),h(X))$ and sample variance $\hat{var}(h(X))$ respectively [36]. Thus, we have a modified estimator

$$\hat{\alpha} = \frac{\hat{cov}(f(X),h(X))}{\hat{var}(h(X))} = \frac{\sum_{i=1}^{m}(f(X_i)-\bar{f})(h(X_i)-\bar{h})}{\sum_{i=1}^{m}(h(X_i)-\bar{h})^2}, \tag{3.7}$$

where $\bar{f} = (1/m)\sum_{i=1}^{m} f(X_i)$ and $\bar{h} = (1/m)\sum_{i=1}^{m} h(X_i)$. Then the regression estimator for μ is $\hat{\mu}_{\hat{\alpha}}$. Although this brings bias, this $\hat{\alpha}$ is still consider to be a consistent estimator as its bias is usually small [23]. The estimated variance of $\hat{\mu}_{\hat{\alpha}}$ suggested by Art Owen [2] is

$$\hat{var}(\hat{\mu}_{\hat{\alpha}}) = \frac{1}{m^2}\sum_{i=1}^{m}\left(f(X_i)-\hat{\mu}_{\hat{\alpha}}-\hat{\alpha}(h(X_i)-\bar{h})\right)^2, \tag{3.8}$$

and the CLT-based approximated $100(1-\alpha)$ percent CI is

$$\hat{\mu}_{\hat{\alpha}} \pm \phi^{-1}(1-\frac{\alpha}{2})\cdot\sqrt{\hat{var}(\hat{\mu}_{\hat{\alpha}})}. \tag{3.9}$$

3.3 Method Efficiency

In terms of whether the CV improves the efficiency, we need to consider how much does it cost to use this method. Let c_f be the average total cost of sampling X_i and then computing $f(X_i)$, and let c_h be the average additional cost incurred by the CV (including cost to evaluate $h(X_i)$ but excluding the cost of sampling X_i). Assume it is cheap to compute the coefficient $\hat{\alpha}$. Then we know efficiency improves by using CVs if $(1-\rho^2)(c_f+c_h) < c_f$, which is equivalent to $|\rho| > \sqrt{\frac{c_h}{(c_f+c_h)}}$. Consider the case where $c_f = c_h$, it is beneficial to use a CV only if $|\rho| > \sqrt{\frac{1}{2}} \approx 0.71$.

3.4 Numerical Example

This section uses a numerical example to illustrate that the variance of the MC estimator can be significantly reduced by applying a CV.

Let X be a random variable that follows a uniform distribution $\mathbb{U}[0,1]$. Suppose our goal is to estimate $\mu := \mathbb{E}(f(X))$, where $f(X) = \frac{1}{1+X}$. We can then write this expected value of $f(X)$ as the integral $\mu = \int_0^1 \frac{1}{1+x}\,dx$. Solving it with the integration rule, we know the exact answer is $\mu = ln(1+1) - ln(0+1) = ln2 \approx 0.6931471805599453$.

To solve it with the MC integration, we first randomly draw m IID samples $X_1, ..., X_m$ from the desired distribution $\mathbb{U}[0,1]$. Then the MC estimator is given by $\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} f(X_i)$.

To solve it with the CV technique, we introduce a CV $h(X) = 1 + X$ with a known expected value $\mathbb{E}[h(X)] = \int_0^1 (1+x)\,dx = \frac{3}{2}$. Thus, we can get an estimation of μ using the regression estimator $\hat{\mu}_\alpha = \frac{1}{m}\sum_{i=1}^{m} f(X_i) + \alpha(\frac{1}{m}\sum_{i=1}^{m} h(X_i) - \frac{3}{2})$ as introduced in section 3.2.

In the setting with $m = 1500$ simulations and an estimated optimal coefficient $\alpha_{opt} \approx 0.4773$, we get the results as shown in Table 3.1. The exact result of μ is $ln2$ as calculated above, thus we can conclude that the variance of the MC estimator has been significantly reduced by applying a CV.

	Estimation for $\hat{I}$	**Absolute Error**	**Variance**
Monte Carlo	0.69475	0.00160	0.01947
Control Variates	0.69295	0.00020	0.00060

Table 3.1: Results For Candidate Methods (5 d.p.). The Exact Result is $\mu = ln2$

Chapter 4

Adopting Control Variates in Bandits Problems

This chapter reviews the literature on a class of RL problems called MAB and explains its connection to the MC methods and CVs technique intuitively with motivating examples including Blackjack and news push. Current literature has shown that one area where CVs can be promising is in MAB. For this reason, this chapter further examines how CVs can be adopted in the stochastic bandits algorithms. In particular, we review the paper by Verma and Hanawal, where they investigate a new variant of the bandit problems (MAB-CV), in which the agent has access to auxiliary information about the arms that is in the form of CVs. Their originated algorithm UCB-CV uses the estimators based on the linear CVs, and successfully improves the performance of MAB algorithms. The literature reviews in this chapter serve as a crucial foundation to understand the intuitions behind our proposed algorithms that are presented in chapter 5.

This chapter is organized as follows. First of all, the exploitation vs exploration dilemma in various applications is discussed in section 4.1. Next, the pivotal idea of MAB problems is reviewed in section 4.2 with motivating examples and numerical definitions. Two effective algorithms named Upper Confidence Bounds (UCB) and UCB1 that solve the MAB problem, and their connections to the MC simulations are also examined in section 4.2. A new variant of the MAB problem called MAB-CV, and an algorithm called UCB-CV that is developed by Verma and Hanawal to

solve the MAB-CV problem is then reviewed in section 4.3. In this section, we have also presented how these algorithms connect to the MC simulations, and how CV can be applied to improve the performance of the stochastic bandits algorithms.

4.1 Exploitation vs Exploration Dilemma

The exploitation vs exploration dilemma exists in various applications.

In daily life, people often need to decide where to eat. Suppose you always choose to eat in your favourite restaurant, then you will be confident and pleasant about what you will get, but miss the chance to exploring better places. If you decide to try new restaurants all the time, you may experience unpleasant food from time to time.

Another example can be found in the previously proposed news push system. The recommender has to exploit its existing knowledge about the user's previously chosen items, while exploring new items that the user might like. Exploration can help the recommender to better adapt to new situations such as cold start problems [37]. At the same time, too much exploration brings the risk of compromising the user experience. As a result, a properly balanced exploration-exploitation trade-off is crucial, and news media companies often aim to strike a balance between recommending the known most popular articles and some new articles that could be more profitable.

In addition, the trade-off between exploitation and exploration is also important in the Blackjack motivating example that we discuss in section 2.1. The player tries to balance between the existing optimal strategy and the unknown strategies that may bring more profits in the long run.

Acquiring new information while optimizing rewards based on the existing knowledge, is known as the exploitation vs. exploration trade-off in RL. If the agent has access to all the information about the environment, then it is easy to find the best strategy by always select the best action at each time step. In sequential decision-making problems, the exploitation-exploration dilemma comes from incomplete information and uncertainty: the agent needs to collect enough infor-

mation to make decisions that accumulate maximum rewards while keeping the risk under control. Exploitation allows the agent to take advantage of the known best action. Exploration takes some risk to gather information about the unknown choices. A well-balanced trade-off between exploitation and exploration can help the agent to accumulate its maximum rewards in the long run.

4.2 Multi-Armed Bandits

The *multi-armed bandit* (MAB) problem is a classic problem that well demonstrates the exploitation vs exploration dilemma. This section first presents an introduction to the classical MAB approaches with some motivating examples and mathematical definitions. The underlying principles and intuitions of the MAB approach serve as a crucial foundation to understand how bandit ideas can be applied to various applications. In addition, two effective algorithms that solve the MAB problems are also included. The discussions in this section help to understand why CVs can be used to improve the performance of the stochastic MAB problems, and we will discuss the details in section 4.3.

4.2.1 Motivating Examples

The MAB problem originally depicts a gambler in a Casino betting on slot machines. Suppose there are K possible actions (a.k.a. arms) to choose from. Each machine has a fixed but unknown reward probability [38] for hitting the jackpot. In each round, the agent chooses an arm to play, and receives the reward of the selected arm. In the classical setting, the arms rewards are assumed to be independent of each other, and by playing the arms, the learner observes IID reward samples. The agent simultaneously attempts to acquire new knowledge via exploring different machines to estimate their expected payoff (exploration) and exploit the existing knowledge to bet on the best machine (exploitation). The objective is to learn how to balance these competing tasks to maximise cumulative rewards over time.

Over the last decades, the MAB framework has attracted a lot of attention in various applications, including recommender systems [39][40], information retrieval [41], anomaly detection [42], dialogue systems [43], clinical trials [44] and

finance [45][46]. This is because it has outstanding performance on those applications and itself has certain appealing properties, such as learning from less feedback.

Let's start with our news push example. The news push is a typical application of the recommender system, where recent recommender system solutions have proposed to address the exploration and exploitation trade-off by using MAB. When a new user arrives, the recommender picks an article to display, observes whether the user clicks on this header. The goal is to maximize the total number of clicks. The actions can be the possible news articles for display. As the recommender suggests new stories, it can keep track of their click rates. In this way, click rates could be interpreted as the reward for the selected action. Moreover, the real-world applications of bandit-based recommendation system can be further found in music [47][48], online advertising [49], daily deals [50], news articles [51] and visual discovery [52].

We have introduced the MAB problem with some motivating examples, next, let's define the stochastic K-armed bandits problem in mathematical forms.

4.2.2 Mathematical Definitions

The interaction between the learner and the environment in the MAB problem can be found in Algorithm 1.

Algorithm 1 MAB problem with instance $(\boldsymbol{\mu},\boldsymbol{\sigma})$

For each round t:

1. **Environment** generates $\{f(X_{t,1}),...,f(X_{t,K})\} \in \mathbb{R}^K$, where $\mathbb{E}[f(X_{t,i})] = \mu_i$, $\text{var}(f(X_{t,i})) = \sigma_i^2$, and the sequence $\{f(X_{t,i})\}_{t\geq 1}$ are IID for all $i \in [K]$.
2. **Learner** selects an arm $I_t \in [K]$ based on past observed rewards from arms till round $t-1$.
3. **Feedback and Regret:** The learner observes the reward of the selected arm $f(X_{t,I_t})$ and incurs penalty $(\mu_{i^*} - \mu_{I_t})$.

The set of arms, or equivalently the set of actions, is represented as $[K] \doteq \{1,2,...,K\}$, where $\doteq$ means "is defined as". Each arm $i \in [K]$ has an expected reward $\mu_i \doteq \mathbb{E}(f(X_{t,i}))$ given that arm is selected, and we call this the *value* of that arm. In round t, the environment generates a vector $\{f(X_{t,i})\}_{i\in[K]}$. The random

variable $f(X_{t,i})$ here denotes the reward of arm i in round t, which are drawn from an unknown but fixed distribution with mean μ_i and variance σ_i^2 and form an IID sequence. The reward distributions are initially unknown to the learner. This means the *value* of each arm μ_i is also unknown to the learner in the initial round.

In round t, the learner selects an arm based on the past observed rewards till round $t-1$, and it observes the reward for this selected arm. Denote the selected arm in round t as I_t ($I_t \in [K]$), and the observed corresponding reward as $f(X_{t,I_t})$. In this way, although the learner does not have exact knowledge of the action values, it may have some estimations based on the past observed rewards.

If the learner maintains estimates of the action values, then there will be at least one action with the highest estimated value in any round t. Such actions are referred to as greedy actions. *Exploitation* is to recommend one of the greedy actions based on the current known user model to maximize the expected reward on the one step. On the other hand, *exploration* can help the agent gain useful information and improve our estimate of the non-greedy action's value by suggesting non-greedy actions, which may accumulate greater rewards in the long run. Whether to exploit or explore depends on various factors including the number of remaining rounds, the precise values of the estimates, and uncertainties [22]. If the trade-off between exploration and exploitation is well-balanced, the learner can accumulate more rewards in the long run. However, balancing the exploration and exploitation is a distinctive challenge in the bandits setting. Many sophisticated methods have been designed to solve this problem. One well-known algorithm named UCB will be reviewed in section 4.2.3.

Let the parameter vectors $\boldsymbol{\mu} = \{\mu_i\}_{i\in[K]}, \boldsymbol{\sigma} = \{\sigma_i^2\}_{i\in[K]}$ identify an instance of MAB problem. With mean reward vector $\boldsymbol{\mu}$, we denote the optimal arm as $i^* = \arg\max_{i\in[K]} \mu_i$, and the maximal reward mean as $\mu_{i^*} = \max_{i\in[K]} \mu_i$.

The objective of the MAB problem is to learn policies that maximize the sum of the collected rewards (i.e. the cumulative reward) in the long run. This is equivalent to minimizing the regret of the policy. Thus, one could use regret to analyse the algorithm performance. In particular, we define the cumulative regret after T rounds

as:

$$R_T \doteq T \cdot \mu_{i^*} - \mathbb{E}\left[\sum_{t=1}^{T} f(X_{t,I_t})\right]. \tag{4.1}$$

A good policy could be the *zero-regret strategy*, it is a strategy whose average regret per round $\mathbb{E}[R_T/T] \to 0$ as the number of played rounds $T \to \infty$ [53]. Intuitively, if enough rounds are played, it is guaranteed to converge to an optimal strategy (but not necessarily unique) by applying zero-regret strategies.

4.2.3 UCB Algorithm

As mentioned in section 4.2.2, MAB problems often face the exploitation-exploration dilemma. When deciding on a given time step, whether to explore or exploit depends on the number of remaining rounds, the precise values of the estimates, and uncertainties about the accuracy of the action-value estimates [22]. They are several action selection strategies to balance its exploration-exploitation trade-off. Here, we introduce a simple one called the UCB action selection [54].

With UCB algorithm, the arm chosen at time step t is given by

$$I_t \doteq \underset{i \in [K]}{\arg\max} \left(\hat{\mu}_{N_t(i),i} + c \sqrt{\frac{ln(t)}{N_t(i)}} \right), \tag{4.2}$$

where $\hat{\mu}_{N_t(i),i}$ is the estimated value of arm i in round t (i.e. the average reward actually received from arm i in round t); $ln(t)$ denotes the natural logarithm of t; $N_t(i)$ is the number of times that action i has been selected prior to time t, and c is a confidence value that controls the level of exploration.

The UCB algorithm is based on updates which take the form as stated in equation 4.2. In such cases, the basic version of UCB uses a simple MC estimator for the action value μ_i, and then uses that as the estimated mean reward for each arm $\hat{\mu}_{N_t(i),i}$.

The formula for UCB (equation 4.2) is made up of two parts: the first part $\hat{\mu}_{N_t(i),i}$ represents the exploitation, the second part $c\sqrt{\frac{ln(t)}{N_t(i)}}$ adds exploration, and the hyper-parameter c controls the degree of exploration. With the first part, the

estimated action values point out the current "best" option. As for the second half, it provides a measure of uncertainty for the estimated action values. In particular, if one arm has not been selected frequently, then $N_t(i)$ will be small. This subsequently leads to an increase in the uncertainty term, increasing the chance for this arm to be selected. Vice versus, if one arm is being frequently selected, then the learner will be more confident about its estimated values of the arms. In such situations, $N_t(i)$ grows, the uncertainty term reduces, making it less likely to be chosen as a result of exploration. However, note that there are still chances for this arm to be selected, as it may be the action with the highest value (i.e. the exploitation term can be large). In addition, the log function in the numerator makes the uncertainty term grows slowly when an action is not being selected, but shrinks rapidly when the arm is selected due to the increase in $N_t(i)$ being linear. As a result of the uncertainty in the estimated values, the exploration term will be larger for arms that have not been chosen frequently. In other words, arms that have been less explored will receive a boost even if their mean reward estimators are modest, especially if the learner has been playing for a while. Therefore, the UCB algorithm has a natural ability to well-balance the trade-off between exploration and exploitation in MAB problems.

A simple variant of the UCB algorithm is called the UCB1 policy [54]. With UCB1, the hyper-parameter c that controls the degree of exploration in the original UCB algorithm is set to $\sqrt{2}$, and the selected arm at round t is given by

$$I_t \doteq \underset{i\in[K]}{\arg\max} \left(\hat{\mu}_{N_t(i),i} + \sqrt{\frac{2 \cdot ln(t)}{N_t(i)}} \right), \tag{4.3}$$

where the notation keeps the same meanings as the ones used in equation 4.2. The process of UCB1 algorithm is given as follows: it first plays each arm once for the initialization. Then in round t, the learner selects an arm I_t based on equation 4.3. After playing arm I_t, the reward $f(X_{t,I_t})$ is observed. The learner then updates the value of $N_t(I_t)$ and re-estimates the mean reward estimators $\hat{\mu}_{N_t(I_t),I_t}$. The same procedure is repeated for the succeeding rounds.

Please kindly find the pseudo-code for policy UCB1 in Algorithm 2.

Algorithm 2 UCB1 Algorithm for MAB problem

Input: K

Play each arm $i \in [K]$ once

for $t = K+1, K+2, K+3, \ldots$ **do**

1. Select arm I_t as given in equation 4.3
2. Play arm I_t and observe its corresponding reward $f(X_{t,I_t})$
3. Increment the value of $N_t(I_t)$ by 1
4. Re-estimate the expected reward $\hat{\mu}_{N_t(I_t),I_t}$

end for

4.3 Stochastic Multi-Armed Bandits with Control Variates

The objective of the MAB problem is to learn policies that well-balanced the exploration and exploitation trade-off, and maximize the cumulative reward in the long run (section 4.2). We have seen that UCB and UCB1 provide good solutions for the MAB problems. One possible solution for reaching a better balance is to estimate arm's mean rewards with tighter confidence bounds [54][55] and lower variance [56][57]. To be more specific, the performance of a stochastic MAB algorithm is determined by the tightness of the CI of the arms' mean reward estimators [54][55]. The width of the CI depends on the variance [56][57] of this mean reward estimators. Therefore, estimators that have smaller variance lead to a narrower CI for the same number of samples. As presented in section 3.2, applying CVs never leads to an increase in the variance of the estimators, and it frequently reduces the variance. A stronger correlation between the reward samples and the CVs leads to greater achievement in variance reduction. Hence, we may improve the regret performance of a stochastic MAB algorithm by exploiting a strongly correlated CV.

The question then comes: what random variable(s) can be treated as CV(s) in the stochastic MAB problems? Let's consider a new variant of this bandits problem,

in which the learner has auxiliary information about the arms. Suppose this auxiliary information is correlated with the arm rewards, and its mean value is known from the prior historical data. Thus, this auxiliary information can be treated as CV. Therefore, if there is any available side information in the form of CV, we can use it to build tighter CI to improve the performance of the stochastic MAB problems. This is the motivation behind the idea that Verma and Hanawal [18] bring.

4.3.1 Motivating Examples

In practical applications, the extraneous factors may have impacts on the rewards of the arms and act as CVs. Let's proceed with the news push example that we discuss in section 4.2.1. Suppose the articles are of variable length, thus the reading time spends on each news could be different. Therefore, whether or not a user is interested in a displayed article may depend on the average reading time. The news company monitors the market for a while, observes the reading time for news with different lengths, and knows its mean values based on past data and experience. Hence, the random reading time can operate as a CV in estimating the expected reward (i.e. click rate) from recommending the displayed news.

4.3.2 Mathematical Definitions

The Multi-Armed Bandits with Control Variates (MAB-CV) problem can be defined and formulated as follows.

Consider a MAB problem with K arms. To make the notations coherent as section 4.2.2, we denote the set of arms as $[K] \doteq \{1,2,...,K\}$. In round t, the environment generates a vector $(\{f(X_{t,i})\}_{i\in[K]}, \{h(X_{t,i})\}_{i\in[K]})$. The random variable $f(X_{t,i})$ represents the reward of arm i in round t, which are drawn from an unknown but fixed distribution with mean μ_i and variance σ_i^2 and form an IID sequence. The random variable $h(X_{t,i})$ denotes the side information that is correlated with the reward of arm i in time step t. These random variables form an IID sequence drawn from an unknown but fixed distribution with mean θ_i and variance $\sigma_{\theta,i}^2$. The agent has access to the values of $\{\theta_i\}_{i\in[K]}$ but not their variance $\{\sigma_{\theta,i}^2\}_{i\in[K]}$. This random variable $h(X_{t,i})$ can then acts as the CV, and we denote the correlation coefficient

between the ith arm's rewards and its associated CV by ρ_i.

In this setting, the agent observes not only the reward but also the associated CV from the selected arm. This new variant of MAB is then referred as MAB-CV problem. The parameter vectors $\boldsymbol{\mu} = \{\mu_i\}_{i\in[K]}, \boldsymbol{\sigma} = \{\sigma_i^2\}_{i\in[K]}$, $\boldsymbol{\theta} = \{\theta_i\}_{i\in[K]}, \boldsymbol{\sigma_\theta} = \sigma_{\theta,i}^2\}_{i\in[K]}$, and $\boldsymbol{\rho} = \{\rho_i\}_{i\in[K]}$ identify an instance of MAB-CV problem. For a MAB-CV instance, with mean reward vector $\boldsymbol{\mu}$, the optimal arm can be denoted as $i^* = \arg\max_{i\in[K]} \mu_i$. The maximal reward mean is therefore $\mu_{i^*} = \max_{i\in[K]} \mu_i$.

Algorithm 3 MAB-CV problem with instance $(\boldsymbol{\mu},\boldsymbol{\sigma},\boldsymbol{\theta},\boldsymbol{\sigma_\theta},\boldsymbol{\rho})$

For each round t:

1. **Environment** generates $\{f(X_{t,1}),...,f(X_{t,K})\} \in \mathbb{R}^K$ and $\{h(X_{t,1}),...,h(X_{t,K})\} \in \mathbb{R}^K$, where $\mathbb{E}[f(X_{t,i})] = \mu_i$, $\text{var}(f(X_{t,i})) = \sigma_i^2$, $\mathbb{E}[h(X_{t,i})] = \theta_i$, $\text{var}(h(X_{t,i})) = \sigma_{\theta,i}^2$, and the sequence $\{f(X_{t,i})\}_{t\geq 1}$ and $\{h(X_{t,i})\}_{t\geq 1}$ are IID for all $i \in [K]$.
2. **Learner** selects an arm $I_t \in [K]$ based on past observed rewards and CV samples from arms till round $t-1$.
3. **Feedback and Regret:** The learner observes the reward $f(X_{t,I_t})$ and its corresponding CV $h(X_{t,I_t})$, then incurs penalty $(\mu_{i^*} - \mu_{I_t})$.

In round t, the learner select an arm based on the observation of past reward and CV samples, and we denote this selected arm as I_t. With the MAB-CV problem, the environment and the agent interacts as demonstrated in Algorithm 3 [18]. The objective for MAB-CV problem is to learn policies that accumulate maximum reward, which is equivalent to minimizing the regret of the policy. The regret is defined as:

$$R_T \doteq T \cdot \mu_{i^*} - \mathbb{E}[\sum_{t=1}^{T} f(X_{t,I_t})]. \tag{4.4}$$

The learner aim for a policy that has sub-linear expected regret (i.e. $\mathbb{E}[R_T/T] \to 0$ as the number of played rounds $T \to \infty$). To achieve this, the learner can exploit CVs to estimate the mean rewards with a reduced variance, which leads to tighter confidence bounds. Therefore, the agent can reach a better-balanced exploration-exploitation trade-off and begin to play the optimal arm earlier and more frequently.

4.3.3 UCB-CV Algorithm

In this section, we consider two cases where the first assumes the rewards and CVs of arms in the MAB-CV problems have a multivariate normal distribution, and the second relax the distributional assumptions of the rewards and associated CVs. It is claimed in the paper that the regret of the UCB-CV is smaller by a factor $(1-\rho^2)$ in comparison with the existing algorithms when the rewards and CVs are normally distributed (ρ is the correlation coefficient of the reward and CVs) [18]. For this reason, we mainly focus on the mathematical derivation of UCB-CV for cases where the arms have normally distributed rewards and CVs (section 4.3.3.1). The UCB-CV algorithm presented in section 4.3.3.1 can be easily modified in the non-Gaussian cases. The discussions for the general case, where no distribution assumptions are made about the reward and associated CVs can be found in section 4.3.3.2.

4.3.3.1 Arms with Normally Distributed Rewards and Control Variates

Motivated by the CV theory, Verma and Hanawa obtain mean estimators with smaller variance and tighter CI and develop an improved upper confidence bound based algorithm named UCB-CV. Their proposed algorithm uses linear CVs to estimate the mean rewards, and successfully improves the performance of the stochastic MAB algorithms.

To make a clear presentation, we consider one CV for illustration purposes to bring out the core ideas of UCB-CV. The mathematical proves can be easily adapted to the multiple CVs cases, as the derivation follows similar steps, except some variables will be changed to the appropriate matrices. Since the procedure is similar, the details won't be discussed here. If the reader is interested in an estimator with multiple CVs and the relevant mathematical derivations, please refer to Verma and Hanawal's paper [18].

Consider a new sample for arm i ($i \in [K]$) in round t:

$$\tilde{f}(X_{t,i}) = f(X_{t,i}) + \alpha_i^*(\theta_i - h(X_{t,i})), \tag{4.5}$$

where $f(X_{t,i})$ is the reward of arm i at time t, $h(X_{t,i})$ is the associated CV for arm i in round t, $\theta_i = \mathbb{E}[h(X_{t,i})]$, and $\alpha_i^* = cov(f(X_{t,i}), h(X_{t,i}))/var(h(X_{t,i}))$. With m such samples, the overall mean reward estimator for arm i is given by:

$$\hat{\mu}_{m,i}^c = \frac{1}{m}\sum_{r=1}^{m} \tilde{f}(X_{r,i}). \tag{4.6}$$

Let $\hat{\mu}_{m,i} = \frac{1}{m}\sum_{r=1}^{m} f(X_{r,i})$ and $\hat{\theta}_{m,i} = \frac{1}{m}\sum_{r=1}^{m} h(X_{r,i})$ denote the estimated sample mean of reward and CV of arm i from m samples respectively. Then, the mean reward estimator with CV can be written as:

$$\hat{\mu}_{m,i}^c = \hat{\mu}_{m,i} + \alpha_{m,i}^*(\theta_i - \hat{\theta}_{m,i}). \tag{4.7}$$

The optimal coefficient $\alpha_{m,i}^*$ is needed to be estimated, as the values of $cov(f(X_i), h(X_i))$ and $var(h(X_i))$ are unknown. With m samples, we can estimate this optimal coefficient as

$$\hat{\alpha}_{m,i}^* = \frac{\hat{cov}[f(X_i), h(X_i)]}{\hat{var}[h(X_i)]} = \frac{\sum_{r=1}^{m}(f(X_{r,i}) - \hat{\mu}_{m,i})(h(X_{r,i}) - \theta_i)}{\sum_{r=1}^{m}(h(X_{r,i}) - \theta_i)^2}, \tag{4.8}$$

which can then be used to calculate the $\hat{\mu}_{m,i}^c$ in equation 4.7.

Denote the variance of the linear CV samples as $\sigma_{c,i}^2 = var(\tilde{f}(X_i))$, and denote the variance of the mean estimator that uses these CV samples as $v_{m,i} = var(\hat{\mu}_{m,i}^c)$. These estimators are computed using correlated samples $\{\tilde{f}(X_i)\}$. When the reward and CVs are normally distributed, results demonstrated by Lemma 1 instruct how to compute an unbiased variance estimator for the $\hat{\mu}_{m,i}^c$.

Lemma 1 *Let the reward and CV of each arm have a multivariate normal distribution. After observing m samples of reward and CV from arm i, define* $\hat{v}_{m,i} =$

$\frac{Z_{m,i}\hat{\sigma}^2_{c,i}(m)}{m}$, *where*

$$Z_{m,i} = \left(1 - \frac{(\sum_{r=1}^{m}(h(X_{r,i}) - \theta_i))^2}{m\sum_{r=1}^{m}(h(X_{r,i}) - \theta_i)^2}\right)^{-1}, and, \hat{\sigma}^2_{c,i}(m) = \frac{1}{m-2}\sum_{r=1}^{m}(\tilde{f}(X_r) - \hat{\mu}^c_{m,i})^2.$$

then $\hat{v}_{m,i}$ *is an unbiased variance estimator of* $\hat{\mu}^c_{m,i}$ $\left(i.e. \mathbb{E}[\hat{v}_{m,i}] = var(\hat{\mu}^c_{m,i})\right)$.

The next results given by Verma and Hanawal is the CI for the reward estimators.

Lemma 2 *Let m be the number of rewards and associated CV samples from arm i in time step t. Then*

$$\mathbb{P}\left(|\hat{\mu}^c_{m,i} - \mu_i| \geq V_{t,m,1}\sqrt{\hat{v}_{m,i}}\right) = \frac{2}{t^2}, \tag{4.9}$$

where $V_{t,m,1}$ *denotes* $100(1-1/t^2)^{th}$ *percentile value of the t-distribution with* $m-2$ *degrees of freedom and* $\hat{v}_{m,i}$ *is an unbiased estimator for variance of* $\hat{\mu}^c_{m,i}$.

Lemma 1 and 2 are results derived in Verma and Hanawal's paper [18], and they prove these lemmas using regression theory. I have also attempted to reproduce these results. In simple words, they treat it as a linear regression problem, and estimate the mean rewards using the least square estimator. Then with the relevant finite sample properties of the least square estimator and Theorem 1 and 2 in Nelson's paper [58], they prove that $\mathbb{E}[\hat{v}_{m,i}] = var(\hat{\mu}^c_{m,i})$, which implies $\hat{v}_{m,i}$ is an unbiased estimator of $var(\hat{\mu}^c_{m,i})$. As for Lemma 2, the result is derived based on Theorem 2 in Nelson's paper [58], which shows that a linear regression implies that $\hat{\mu}^c_m$ is an unbiased estimator for μ. With the condition of constant conditional variance, it ensures that the variance estimator is also unbiased. The normality of the conditional distribution of $f(X_t)$ given $\boldsymbol{h}(X_t)$ then implies we can use t-distribution to design the CI of the mean reward estimators, leading to a valid CI as stated in Lemma 2. The detailed proof can be found in Appendix A and B.

Based on the above results, the UCB-CV is developed as follows: Let $N_t(i)$ denotes the number of rounds that arm i is selected until round t, $\hat{v}_{N_t(i),i}$ denotes the sample variance of mean reward estimator $\hat{\mu}^c_{N_t(i),i}$ for arm i, and $V_{t,N_t(i),1}$ denotes

the $100(1-1/t^2)^{th}$ percentile value of the t-distribution with $N_t(i)-2$ degrees of freedom. Therefore, the optimistic upper bound for mean reward estimator of arm i is:

$$UCB_{t,i} = \hat{\mu}^{c}_{N_t(i),i} + V_{t,N_t(i),1}\sqrt{\hat{v}_{N_t(i),i}}. \tag{4.10}$$

With UCB-CV, the selected arm at round t is given by

$$I_t \doteq \underset{i\in[K]}{\arg\max}\left(\hat{\mu}^{c}_{N_t(i),i} + V_{t,N_t(i),1}\sqrt{\hat{v}_{N_t(i),i}}\right). \tag{4.11}$$

The UCB-CV algorithm works as follows: It takes the number of arms K and a constant Q = number of CV+2 as input. It plays each arm Q times for initialization to make sure the sample variance for observation $\hat{\sigma}^2_{c,i}(m)$ are computed. Then in time step t, the learner selects an arm I_t based on equation 4.11. After playing arm I_t, the reward $f(X_{t,I_t})$ and its associated CV $h(X_{t,I_t})$ are observed. The learner then updates the value of $N_t(I_t)$ and re-estimates the optimal coefficient $\hat{\alpha}^{*}_{N_t(I_t),I_t}$, the mean reward estimator $\hat{\mu}^{c}_{N_t(I_t),I_t}$ and the sample variance of the mean reward estimator $\hat{v}_{N_t(I_t),I_t}$. The same process is repeated for the succeeding rounds. Please kindly see the pseudo-code for UCB-CV in Algorithm 4.

Algorithm 4 UCB-CV Algorithm for MAB-CV problem

Input: K, Q

Play each arm $i \in [K]$ Q times

for $t = QK+1, QK+2, QK+3, \ldots$ **do**

1. Select arm I_t as given in equation 4.11
2. Play arm I_t and observe its reward $f(X_{t,I_t})$ and associated CV $h(X_{t,I_t})$
3. Increment the value of $N_t(I_t)$ by 1
4. Re-estimate $\hat{\alpha}^{*}_{N_t(I_t),I_t}$ (equation 4.8), $\hat{\mu}^{c}_{N_t(I_t),I_t}$ (equation 4.7), $\hat{v}_{N_t(I_t),I_t}$ (Lemma 1)

end for

In comparison with UCB, UCB-CV incorporates variance estimates and uses

CVs to estimate mean rewards, leading to a sharper CI, which results in a reduction in the variance. The performance of the MAB algorithms has been successfully improved.

4.3.3.2 No Distributional Assumptions on Arms' Rewards and Control Variates

Intuitively, the UCB-CV algorithm discussed in section 4.3.3.1 can be simply extended to the general case where no distributional assumptions on rewards and associated CVs are made. It is because the CLT suggests if you have enough rewards and CVs samples from any distribution, then its mean estimates approximately follow a Gaussian distribution.

For the MAB-CV problem, the objective is to learn policies that accumulate maximum reward in the long run, hence large samples are needed to construct the mean estimators. For this reason, we may assume the number of constructed samples is enough for the mean reward estimators to follow a Gaussian distribution no matter what distributional form the arm reward and associated CVs have. However, note that in the general distribution cases, the newly constructed observation samples resulting from the linear combination of rewards and CVs are independent but not necessarily be normally distributed. For this reason, the mean estimators and the variance estimates of the mean estimators may not remain to be unbiased. Although these estimators used in UCB-CV may be biased in the general distributional settings, we still consider UCB-CV to be a reasonable approach as it gives acceptable results when rewards and associated CVs are non-Gaussian as presented in the experiments section of this paper [18].

In addition, ways to adapt the UCB-CV for the general distributions using estimators and associated CI based on the splitting and batching methods are also discussed by Verma and Hanawal. The details can be found in this paper [18].

Chapter 5

A New Version of UCB with Zero-Variance Control Variates

This chapter proposes two newly-developed algorithms that successfully improve the performance of the stochastic MAB algorithms by constructing the ZV-CV directly from the original data and using these constructed ZV-CV in the mean reward estimators. This chapter starts with a review of the ZV-CV literature, then presents the mathematical derivations of the proposed algorithms. Note that some missing proofs of the lemma in these algorithms can be found in the appendix. The performance of these two algorithms are validated on synthetically generated data, and the detailed results can be found in chapter 6.

We have seen that the CVs can be adopted in the MAB problem, in which the agent has access to the auxiliary information about the arms, and this new variant of the MAB problem is defined as MAB-CV. The condition for the MAB-CV to exist is that the auxiliary information must correlate with the arm rewards, and its mean value must be known (section 4.3). However, we often experience the case where we don't know what a good CV is. In such settings, instead of using side information, it might be interesting and possible to learn a good CV directly from the original data [13]. The ZV-CV technique is a post-processing method that reduces the variance of the MC estimators of expectation by constructing ZV-CV based on the derivatives of the log-likelihood. Motivated by the ZV-CV method, this chapter examines new versions of UCB with ZV-CV, and seeks for improvement in

the regret performance of the MAB algorithms.

5.1 Zero-Variance Control Variates

This section presents an introduction to the main concepts of the Zero Variance (ZV) method and the associated ZV-CV [29][59]. The underlying principles serve as a crucial foundation to understand how ZV-CV can be adopted to form new versions of the UCB algorithm, and why these potential algorithms can improve the performance of the stochastic MAB.

5.1.1 Overview of the Zero Variance Method

The interest of the ZV method is to evaluate the expectation of a function $f(X)$ with respect to a distribution with density $p(X)$, and it is required $X \in \mathbb{R}^{d_x}$:

$$\mu = \mathbb{E}[f(X)] = \int f(X)p(X)\,dX. \tag{5.1}$$

This method dictates that the original function $f(X)$ is substituted by an auxiliary function $\tilde{f}(X)$:

$$\tilde{f}(X) = f(X) + \alpha^T h(X) \tag{5.2}$$

where $\alpha \in \mathbb{R}^{d_a}$ and $h(X)$ denotes the CVs. It is required that $\mathbb{E}[h(X)] = 0$ to acquire an unbiased estimator for μ.

5.1.2 Zero-Variance Control Variates and Optimal Coefficients

This section presents the general expressions of the ZV-CV and the corresponding optimal coefficients in the ZV method [60][61][59][62].

Assaraf and Caffarel [60] establish the ZV-CV, which significantly reduce the variance of the MC estimator. Mira et al. [61] then suggest to construct the ZV-CV based on the derivative of the log-target density.

To meet the zero-mean requirement (i.e. $\mathbb{E}[h(X)] = 0$), the auxiliary function

$\tilde{f}(X)$ constructed by Mira et al. [61] is:

$$\tilde{f}(X) = f(X) - \frac{1}{2} \triangle_X [P(X)] + \nabla_X [P(X)] \cdot z(X), \tag{5.3}$$

$$z(X) := -\frac{1}{2} \nabla_X [log(p(X))] \tag{5.4}$$

where $P(X)$ is assumed to be a polynomial, $\nabla_X := \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, ..., \frac{\partial}{\partial x_{d_x}} \right)^T$ denotes the gradient, and $\triangle_X := \sum_{i=1}^{d_x} \frac{\partial^2}{\partial x_i^2}$ is the Laplace operator. $-\frac{1}{2} \triangle_X [P(X)] + \nabla_X [P(X)] \cdot z(X)$ in equation 5.3 is equivalent to $\alpha^T h(X)$ in equation 5.2.

In conclusion, ZV-CV is a post-processing method that constructs some CVs based on the derivatives of the log-likelihood to reduce the variance of the MC estimator, $\nabla_X [log(p(X))]$ [60][61].Once the derivatives are available, the only computation that remains is to solve a linear regression problem.

5.1.2.1 First Degree Polynomial $P(X)$

Consider the case where $P(X)$ is a first degree polynomial

$$P(X) = \alpha^T X, \tag{5.5}$$

where $\alpha \in \mathbb{R}^{d_a}$, $X \in \mathbb{R}^{d_x}$, $d_a = d_x$. The associated gradient and the Laplace operators are

$$\nabla_X [P(X)] = \alpha^T, \triangle_X [P(X)] = 0. \tag{5.6}$$

Substitute these operators to equations 5.3 and 5.4, the auxiliary function for this linear polynomial $P(X) = \alpha^T X$ equals

$$\tilde{f}(X) = f(X) + \alpha^T z(X). \tag{5.7}$$

where the CVs $h(X)$ in equation 5.2 is $z(X)$ in this case.

Therefore, with m such samples, $\mu = \mathbb{E}[f(X)]$ can be estimated by the follow-

ing estimator:

$$\hat{\mu}_{\alpha}^{lzc} = \frac{1}{m}\sum_{i=1}^{m}\tilde{f}(X_i) = \frac{1}{m}\sum_{i=1}^{m}\left(f(X_i) + \alpha^T z(X_i)\right) = \hat{\mu} + \alpha^T\hat{\theta}, \tag{5.8}$$

where $\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} f(X_i)$ and $\hat{\theta} = \frac{1}{m}\sum_{i=1}^{m} z(X_i)$. This estimator is an unbiased estimator for μ as $\mathbb{E}(\hat{\mu}_{\alpha}^{lzc}) = \mathbb{E}(\hat{\mu}) + 0 = \mu$ (the theoretical mean $\mathbb{E}(\hat{\theta}) = 0$).

The optimal polynomial coefficient α_{opt} that minimizes the variance of the auxiliary function $\tilde{f}(X)$ is

$$\alpha_{opt} = -var^{-1}[z(X)]cov[f(X), z(X)]. \tag{5.9}$$

In practice, the quantities $var[z(X)]$ and $cov[f(X), z(X)]$ are often unknown, therefore these must be estimated from the data to find the best approximation for α_{opt}. In particular, it is common to estimate these two quantities with the sample variance $\hat{var}[z(X)]$ and sample cross-covariance $\hat{cov}[f(X), z(X)]$ matrices respectively. As a result, the modified coefficient estimators $\hat{\alpha}_{opt}$ are given by

$$\hat{\alpha}_{opt} = -\hat{var}^{-1}[z(X)]\hat{cov}[f(X), z(X)] \tag{5.10}$$

$$\hat{var}[z(X)] := \frac{1}{m-1}\sum_{i=1}^{m}\left(z(X_i) - \hat{\theta}\right)\left(z(X_i) - \hat{\theta}\right)^T, \tag{5.11}$$

$$\hat{cov}[f(X), z(X)] := \frac{1}{m-1}\sum_{i=1}^{m}\left(f(X_i) - \hat{\mu}\right)\left(z(X_i) - \hat{\theta}\right), \tag{5.12}$$

where $\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} f(X_i)$ and $\hat{\theta} = \frac{1}{m}\sum_{i=1}^{m} z(X_i)$.

The mean estimator with the above estimated optimal coefficients is

$$\hat{\mu}_{\hat{\alpha}_{opt}}^{lzc} = \hat{\mu} + \left(-\hat{var}^{-1}[z(X)]\hat{cov}[f(X), z(X)]\right)^T\hat{\theta}. \tag{5.13}$$

This estimator that uses the estimated optimal coefficients $\hat{\alpha}_{opt}$ is usually biased. Although estimating α_{opt} from the sample data introduces a bias, that bias is ordi-

narily negligible. Thus, the bias from using estimated CV coefficients [2] is commonly neglected. If an unbiased estimator is required, then one can consider using an estimate of α_{opt} that is independent of the X_i used in $\hat{\alpha}_{opt}$.

5.1.2.2 Second Degree Polynomial

When higher-degree polynomials are considered, the formula for the optimal coefficients is similar. Take a quadratic polynomial as an example:

$$P(X) = c^T X + \frac{1}{2} X^T B X \tag{5.14}$$

where c is a real $d_x \cdot 1$ vector and B is a real symmetric $d_x \cdot d_x$ matrix. The associated auxiliary function is:

$$\tilde{f}(X) = f(X) - \frac{1}{2} tr(B) + (c + BX)^T z(X), \tag{5.15}$$

where $tr(B)$ denotes the trace of B, $z(X)$ is given by equation 5.4.

To conform with 5.2 (i.e. express the auxiliary function $\tilde{f}(X)$ as a linear combination of the original function $f(X)$ and CVs), equation 5.15 can be written as

$$\tilde{f}(X) = f(X) + \alpha^T h(X) \tag{5.16}$$

where the coefficient vectors α and the CVs vector $h(X)$ have $\frac{1}{2} d_x (d_x + 3)$ elements each, and they are defined as follows:

- $\alpha := (c^T, d^T, b^T)^T$, where $d := diag(B)$ is the diagonal of the matrix B, b is a column vector with $\frac{1}{2} d_x (d_x - 1)$ elements, whose element in the $\frac{(2d_x - j)(j-1)}{2} + (i - j)$ position equals the lower diagonal (i,j)-th element of B.

- $h := (z^T, u^T, v^T)^T$, where $u := X \circ z - \frac{1}{2}\mathbf{1}$ ($\circ$ and $\mathbf{1}$ denote the element-wise product and the unit vector respectively), v is a column vector containing $\frac{1}{2} d_x (d_x - 1)$ elements, whose element in the $\frac{(2d_x - j)(j-1)}{2} + (i - j)$ position equals $X_i z_j + X_j z_i$, $j \in \{1, 2, ..., d_x\}$, $i \in \{2, 3, ..., d_x\}$, $j < i$.

The number of coefficients for this second-order $P(X)$ is $\frac{1}{2} d_x (d_x + 3)$ (in the linear

$P(X)$ case, $d_a = d_x$). When estimating the optimal coefficients α for quadratic $p(X)$, formulae 5.9 to 5.12 can be applied by using $h := (z^T, u^T, v^T)^T$.

5.2 Problem Setting

Let's consider the original MAB problem setting as discussed in section 4.2, the interaction between the learner and the environment in the MAB problems can be found in Algorithm 1.

For the UCB-CV algorithm discussed in the previous chapter, it observes the CVs from the environment, thus a new variant of the MAB problem, where the agent has access to the auxiliary information about the arms, is needed. However, in our cases, we do not ask for any CV in the environment. Instead of using side information that correlated with the arm, we construct ZV-CV directly from the original data using the derivatives of the log-likelihood $\nabla_X[log(p(X))]$ [60][61].

The objective is to learn a policy that maximizes the mean cumulative reward, which is equivalent to minimize the policy's regret. To achieve this, we construct the ZV-CV as discussed in 5.1 to estimate the expected rewards, and hopefully, this leads to sharper confidence bounds to improve the regret performance of the stochastic MAB algorithms. In such ways, the learner can reach a better-balanced exploration-exploitation trade-off. The algorithm details will be given in sections 5.3-5.4.

5.3 UCB-LZVCV Algorithm

The UCB-LZVCV algorithm is motivated by the first degree polynomial ZV-CV technique discussed in section 5.1.2.1. Let's focus on the case where rewards of the arms are normally distributed, and we will discuss how easy is it to modify the algorithm for the non-Gaussian distribution cases.

First of all, since there is no available side information that correlates with the arm rewards in the environment, we need to construct the linear zero-variance control variates (LZV-CV) directly from the available data. In our MAB problem setting, the reward is in one dimension. Given $f(X_{t,i}) = X_{t,i}$ and $X_{t,i}$ has mean μ_i and

variance σ_i^2. In round t, the LZV-CV can be constructed as

$$h(X_{t,i}) := -\frac{1}{2}\nabla[log(p(X_{t,i}))]. \tag{5.17}$$

Claim 1 *When we have arms with normally distributed rewards,* $h(X_{t,i}) = -\frac{1}{2} \cdot \frac{-X_{t,i}+\mu_i}{\sigma_i^2}$. *In this way, the expected value of the LZV-CV is zero (i.e.* $\mathbb{E}[h(X_{t,i})] = \theta_i = 0$*).*

The missing proof for Claim 1 is given in the Appendix C.

With this constructed LZV-CV, we can consider a new sample for the ith arm in round t as

$$\tilde{f}(X_{t,i}) = f(X_{t,i}) - \alpha_i^* h(X_{t,i}), \tag{5.18}$$

where $f(X_{t,i})$ is the reward of arm i at time t, $h(X_{t,i})$ is the learned LZV-CV for arm i in round t, $\alpha_i^* = cov[f(X_{t,i}), h(X_{t,i})]/var[h(X_{t,i})]$.

With m such samples, the mean reward estimator for arm i is as follows:

$$\hat{\mu}_{m,i}^{lzc} = \frac{1}{m}\sum_{r=1}^{m} \tilde{f}(X_{r,i}). \tag{5.19}$$

Let $\hat{\mu}_{m,i} = \frac{1}{m}\sum_{r=1}^{m} f(X_{r,i})$ and $\hat{\theta}_{m,i} = \frac{1}{m}\sum_{r=1}^{m} h(X_{r,i})$ denote the sample mean of reward and LZV-CV for arm i with m samples respectively. Then $\hat{\mu}_{m,i}^{lzc}$ can be written as:

$$\hat{\mu}_{m,i}^{lzc} = \hat{\mu}_{m,i} - \alpha_{m,i}^* \hat{\theta}_{m,i}. \tag{5.20}$$

The optimal coefficient $\alpha_{m,i}^*$ is needed to be estimated, as the value of $cov[f(X_i), h(X_i)]$ and $var[h(X_i)]$ are unknown. With m samples, we can estimate the optimal coefficients as

$$\hat{\alpha}_{m,i}^* = \frac{\hat{cov}[f(X_i), h(X_i)]}{\hat{var}[h(X_i)]} = \frac{\sum_{r=1}^{m}(f(X_{r,i}) - \hat{\mu}_{m,i})(h(X_{r,i}) - \theta_i)}{\sum_{r=1}^{m}(h(X_{r,i}) - \theta_i)^2}, \tag{5.21}$$

which can be used to calculate the $\hat{\mu}_{m,i}^{lzc}$ in equation 5.20.

Inspired by Verma and Hanawal [18], we denote the variance of the estimator

with LZV-CV as $v_{m,i} = var(\hat{\mu}^{lzc}_{m,i})$ and denote the variance of this new sample itself as $\sigma^2_{lzc,i} = var(\tilde{f}(X_i))$. These estimators are computed using correlated samples $\{\tilde{f}(X_i)\}$. When the rewards and constructed LZV-CV have a multivariate normal distribution, after observing m samples of reward and constructing m LZV-CV from arm i, an unbiased variance estimator of $\hat{\mu}^{lzc}_{m,i}$ for each arm can be defined using the result from Lemma 3.

Lemma 3 *When the rewards and constructed LZV-CV of each arm have a multivariate normal distribution. After observing m samples of reward and constructing m LZV-CV samples from arm i, define* $\hat{v}_{m,i} = \frac{Z_{m,i}\hat{\sigma}^2_{lzc,i}(m)}{m}$, *where*

$$Z_{m,i} = \left(1 - \frac{(\sum_{r=1}^{m}(h(X_{r,i})))^2}{m\sum_{r=1}^{m}(h(X_{r,i}))^2}\right)^{-1}, and, \hat{\sigma}^2_{lzc,i}(m) = \frac{1}{m-2}\sum_{r=1}^{m}(\tilde{f}(X_{r,i}) - \hat{\mu}^{lzc}_{m,i})^2.$$

then $\hat{v}_{m,i}$ *is an unbiased variance estimator of* $\hat{\mu}^{lzc}_{m,i}$ $\left(i.e. \mathbb{E}[\hat{v}_{m,i}] = var(\hat{\mu}^{lzc}_{m,i})\right)$.

Lemma 3 can be proved using regression theory. The proof is done by treating it as a linear regression problem, and estimating the mean rewards by the least square estimator. Then with the relevant finite sample properties of the least square estimator, we get $\hat{\sigma}^2_{lzc,i}(m) = \frac{1}{m-2}\sum_{r=1}^{m}(\tilde{f}(X_r) - \hat{\mu}^{lzc}_{m,i})^2$ is an unbiased estimator of $\sigma^2_{lzc,i}$. With Theorem 1 and 2 in Nelson's paper [58], we can prove $\mathbb{E}[\hat{v}_{m,i}] = var(\hat{\mu}^{lzc}_{m,i})$, which implies $\hat{v}_{m,i}$ is an unbiased estimator for $var(\hat{\mu}^{lzc}_{m,i})$. Proof details can be found in Appendix A and C.

Since $\hat{v}_{m,i}$ is an unbiased variance estimator of $\hat{\mu}^{lzc}_{m,i}$ as suggested in Lemma 3, and Theorem 2 in Nelson's paper [58] shows that a linear regression implies $\hat{\mu}^{lzv}_{m}$ is an unbiased estimator for μ. With results from Lemma 4, when the arm's rewards and the associated LZV-CV have a multivariate normal distribution, we can use t-distribution to design the CI of the mean reward estimators. (See proof in Appendix A and C).

Lemma 4 *Let m be the number of rewards and constructed LZV-CV samples from arm i in time step t. Then*

$$\mathbb{P}\left(|\hat{\mu}^{lzc}_{m,i} - \mu_i| \geq V_{t,m,1}\sqrt{\hat{v}_{m,i}}\right) = \frac{2}{t^2}, \tag{5.22}$$

where $V_{t,m,1}$ *denotes* $100(1-1/t^2)^{th}$ *percentile value of the t-distribution with* $m-2$ *degrees of freedom and* $\hat{v}_{m,i}$ *is an unbiased estimator for variance of* $\hat{\mu}^{lzc}_{m,i}$.

Based on the above results, the UCB-LZVCV is developed as follows: let $N_t(i)$ denotes the number of rounds that arm i is selected until round t, we define the optimistic upper bound for the mean reward estimator of arm i as follows:

$$UCB_{t,i} = \hat{\mu}^{lzc}_{N_t(i),i} + V_{t,N_t(i),1} \times \sqrt{\hat{v}_{N_t(i),i}}. \tag{5.23}$$

With UCB-LZVCV, the selected arm at round t is given by

$$I_t \doteq \underset{i\in[K]}{\arg\max} \left(\hat{\mu}^{lzc}_{N_t(i),i} + V_{t,N_t(i),1} \times \sqrt{\hat{v}_{N_t(i),i}} \right). \tag{5.24}$$

The UCB-LZVCV algorithm works as follows: It takes the number of arms K as the input. It plays each arm 3 times and constructs the associated LZV-CV for initialization to make sure the sample variance for observations $\hat{\sigma}^2_{lzc,i}(m)$ are computed. Then in time step t, the learner selects an arm I_t based on equation 5.24. After playing arm I_t, the reward $f(X_{t,I_t})$ are observed, and the associated LZV-CV $h(X_{t,I_t})$ is constructed. The learner then updates the value of $N_t(I_t)$ and re-estimates the optimal coefficients $\hat{\alpha}^*_{N_t(I_t),I_t}$, the mean reward estimator $\hat{\mu}^{lzc}_{N_t(I_t),I_t}$ and the estimated variance of the mean reward estimator $\hat{v}_{N_t(I_t),I_t}$. The same process is repeated for the succeeding rounds. Please kindly see the pseudo-code for UCB-LZVCV in Algorithm 5.

Intuitively, the UCB-LZVCV algorithm discussed above can be simply extended to the general case where no distributional assumptions on rewards are made. We just modify the constructed LZV-CV in equation 5.17 with the density specific to that distribution. It is because the CLT suggests if you have enough rewards and CVs samples from any distribution, then its mean estimates approximately follow a Gaussian distribution. Since large samples are used to construct the mean estimators. For this reason, we may assume the number of constructed samples is enough for the mean reward estimators to follow a Gaussian distribution no matter what distributional form the arm rewards have. However, in the general distribution

Algorithm 5 UCB-LZVCV Algorithm for MAB problem

Input: K

Play each arm $i \in [K]$ 3 times and construct the associated LZV-CV (equation 5.17)

for $t = 3K+1, 3K+2, 3K+3, ...$ **do**

1. Select arm I_t as given in equation 5.24
2. Play arm I_t, observe its reward $f(X_{t,I_t})$ and construct the associated LZV-CV $h(X_{t,I_t})$ (equation 5.17)
3. Increment the value of $N_t(I_t)$ by 1
4. Re-estimate $\hat{\alpha}^*_{N_t(I_t),I_t}$ (equation 5.21), $\hat{\mu}^{lzc}_{N_t(I_t),I_t}$(equation 5.20) and $\hat{v}_{N_t(I_t),I_t}$ (Lemma 3)

end for

cases, the newly constructed observation samples (equation 5.18) are independent but not necessarily be normally distributed. For this reason, the mean estimators $\hat{\mu}^{lzc}_{m,i}$ and the variance estimates of the mean estimators $\hat{v}_{m,i}$ as constructed before may not remain to be unbiased. Although these estimators may be biased in the general distributional settings, we still consider UCB-LZVCV to be a reasonable approach as it gives acceptable results when rewards are non-Gaussian as presented in the experiments section (section 6.1).

5.4 UCB-QZVCV Algorithm

The UCB-QZVCV is motivated by the quadratic ZV-CV technique discussed in section 5.1.2.2. Let's focus on the case where rewards of the arms are normally distributed first. For a similar reason as discussed in UCB-LZVCV, the results can be simply extended to the general case where no distributional assumptions on rewards are made according to the CLT. In particular, we modify the constructed quadratic zero-variance control variates (QZV-CV) in equations 5.25-5.27 with the density specific to that distribution. Note that in the general distributional settings, the mean estimators and the variance estimates of the mean estimators may be biased if one uses the results in the Gaussian case. However, we still consider UCB-QZVCV to be a reasonable approach even for the non-Gaussian cases as it gives acceptable

results as presented in the experiments section (section 6.1).

As before, the reward is in one dimension in the MAB settings. Given $f(X_{t,i}) = X_{t,i}$ and $X_{t,i}$ have mean μ_i and variance σ_i^2. Motivated by equation 5.16, we can construct the associated QZV-CV for arm i in round t as follows:

$$h(X_{t,i}) = (z(X_{t,i}), u(X_{t,i}))^T, \tag{5.25}$$

$$z(X_{t,i}) := -\frac{1}{2}\nabla[log(p(X_{t,i}))], \tag{5.26}$$

$$u(X_{t,i}) := X_{t,i} \cdot z(X_{t,i}) - \frac{1}{2}. \tag{5.27}$$

In this way, $\mathbb{E}[h(X_{t,i})] = \theta_i = (\theta_{i,z}, \theta_{i,u})^T = (0,0)^T$ as required.

Claim 2 *When we have arms with normally distributed rewards,* $z(X_{t,i}) = -\frac{1}{2} \cdot \frac{-X_{t,i}+\mu_i}{\sigma_i^2}$ *and* $u(X_{t,i}) = \frac{1}{2\sigma_i^2}(X_{t,i}^2 - X_{t,i}\mu_i - \sigma_i^2)$. *In this way, the expected value of the QZV-CV is zero (i.e.* $\mathbb{E}[h(X_{t,i})] = \theta_i = 0$*).*

The missing proof for Claim 2 is given in the Appendix D.

Now, consider a new sample for the ith arm in round t as

$$\tilde{f}(X_{t,i}) = f(X_{t,i}) - \alpha_i^{*T} h(X_{t,i}), \tag{5.28}$$

where $f(X_{t,i})$ is the reward of arm i at time t; $\alpha_i^* = (c_i, d_i)^T$, α_i^{*T} is the transpose of α_i^*; $c_i = cov[f(X_i), z(X_i)]\ /var[z(X_i)]$, $d_i = cov[f(X_i), u(X_i)]/var[u(X_i)]$; $h(X_{t,i})$ is the constructed QZV-CV vector as shown in equations 5.25-5.27. The coefficient vector α_i^* and the QZV-CV vector $h(X_{t,i})$ have 2 elements each (reward is in one dimension $d_x = 1$, thus the number of coefficients is $\frac{1}{2} \times 1 \times (1+3) = 2$).

With m such samples, the mean reward estimator for arm i is as follows:

$$\hat{\mu}_{m,i}^{qzc} = \frac{1}{m}\sum_{r=1}^{m} \tilde{f}(X_{r,i}). \tag{5.29}$$

Let $\hat{\mu}_{m,i} = \frac{1}{m}\sum_{r=1}^{m} f(X_{r,i})$, $\hat{\alpha}_i^* = (\hat{c}_i, \hat{d}_i)^T$, and $\hat{\theta}_{m,i} = (\hat{\theta}_{m,i,z}, \hat{\theta}_{m,i,u})^T$ where $\hat{\theta}_{m,i,z} =$

$\frac{1}{m}\sum_{r=1}^{m} z(X_{r,i})$ and $\hat{\theta}_{m,i,u} = \frac{1}{m}\sum_{r=1}^{m} u(X_{r,i})$. Then $\hat{\mu}_{m,i}^{qzc}$ can be written as:

$$\hat{\mu}_{m,i}^{qzc} = \hat{\mu}_{m,i} - \hat{\alpha}_i^{*T}\hat{\theta}_{m,i}. \tag{5.30}$$

Let m be the number of rewards and learned QZV-CV for arm i. We introduce new notation here as everything is in vector or matrix form now. In particular, let H_i be a $m \times 2$ matrix whose r-th row is $(z(X_{r,i}), u(X_{r,i}))$, $F_i = (f(X_{1,i}), ..., f(X_{m,i}))^T$. By simple numerical calculation, we get $S_{H_iH_i} = (m-1)^{-1}(H_i{}^T H_i - m\hat{\theta}_{m,i}\hat{\theta}_{m,i}^T)$, which denotes the variance matrix for the constructed QZV-CV, $S_{F_iH_i} = (m-1)^{-1}(H_i{}^T F_i - m\hat{\theta}_{m,i}\hat{\mu}_{m,i})$, which denotes the co-variance matrix between the reward $f(X_i)$ and constructed QZV-CV $h(X_i)$. Extending the arguments used in equation 5.21 to get estimated coefficients for a scalar to a vector, the estimated coefficient vector is

$$\hat{\alpha}_i^* = S_{H_iH_i}^{-1} S_{F_iH_i}. \tag{5.31}$$

With similar derivations for Lemma 3 and 4 in UCB-LZVCV, we can get the results as shown in Lemma 5 and 6 for UCB-QZVCV. The missing proofs can be found in Appendix A and D.

Lemma 5 *When the reward and constructed QZV-CV of each arm have a multivariate normal distribution. After observing m samples of reward and constructing m QZV-CV samples from arm i, define* $\hat{v}_{m,i} = \frac{Z_{m,i}\hat{\sigma}_{qzc,i}^2(m)}{m}$, *where*

$$Z_{m,i} = \left(1 + \frac{(\hat{\theta}_{m,i} - \theta_i)^T S_{H_iH_i}^{-1}(\hat{\theta}_{m,i} - \theta_i)}{1 - 1/m}\right),$$

$$\hat{\sigma}_{qzc,i}^2(m) = \frac{1}{m-3}\sum_{r=1}^{m}(\tilde{f}(X_{r,i}) - \hat{\mu}_{m,i}^{qzc})^2,$$

$\hat{\sigma}_{qzc,i}^2(m)$ *is the sample variance of the new observations constructed with QZV-CV. Then* $\hat{v}_{m,i}$ *is an unbiased variance estimator of* $\hat{\mu}_{m,i}^{qzc}$ $\left(i.e. \mathbb{E}[\hat{v}_{m,i}] = var(\hat{\mu}_{m,i}^{qzc})\right)$.

Lemma 6 *Let m be the number of rewards and constructed QZV-CV samples from*

arm i in time step t. Then

$$\mathbb{P}\left(|\hat{\mu}^{qzv}_{m,i} - \mu_i| \geq V_{t,m,2}\sqrt{\hat{v}_{m,i}}\right) = \frac{2}{t^2}, \tag{5.32}$$

where $V_{t,m,2}$ *denotes* $100(1-1/t^2)^{th}$ *percentile value of the t-distribution with* $m-3$ *degrees of freedom and* $\hat{v}_{m,i}$ *is an unbiased estimator for variance of* $\hat{\mu}^{qzv}_{m,i}$.

Let $N_t(i)$ denotes the number of rounds that arm i is selected until round t, we select the arm that maximizes the optimistic upper bound of its mean estimator:

$$I_t \doteq \underset{i\in[K]}{\arg\max}\left(\hat{\mu}^{qzc}_{N_t(i),i} + V_{t,N_t(i),2} \times \sqrt{\hat{v}_{N_t(i),i}}\right). \tag{5.33}$$

The UCB-QZVCV algorithm works as follows: It takes the number of arms K as the input. It plays each arm 4 times and constructs the associated QZV-CV for initialization to make sure the sample variance for observation $\hat{\sigma}^2_{qzc,i}(m)$ in Lemma 5 are computed. Then in round t, the learner selects an arm I_t based on equation 5.33. After playing arm I_t, the reward $f(X_{t,I_t})$ are observed, and the QZV-CV $h(X_{t,I_t})$ is constructed. The learner then updates the value of $N_t(I_t)$ and re-estimates the optimal coefficients $\hat{\alpha}^*_{I_t}$, the mean reward estimator $\hat{\mu}^{qzc}_{N_t(I_t),I_t}$ and the estimated variance of the mean reward estimator $\hat{v}_{N_t(I_t),I_t}$. The same process is repeated for the succeeding rounds. Please kindly see the pseudo-code for UCB-QZVCV in Algorithm 6.

Algorithm 6 UCB-QZVCV Algorithm for MAB problem

Input: K

Play each arm $i \in [K]$ 4 times and construct the associated QZV-CV (equation 5.25)

for $t = 4K+1, 4K+2, 4K+3, \ldots$ **do**

1. Select arm I_t as given in equation 5.33
2. Play arm I_t, observe its reward $f(X_{t,I_t})$ and construct the associated QZV-CV $h(X_{t,I_t})$ (equation 5.25)
3. Increment the value of $N_t(I_t)$ by 1
4. Re-estimate $\hat{\alpha}^*_{I_t}$ (equation 5.31), $\hat{\mu}^{qzc}_{N_t(I_t),I_t}$ (equation 5.30) and $\hat{v}_{N_t(I_t),I_t}$ (Lemma 5)

end for

Chapter 6

Experiments

This chapter validates and evaluates the performance of UCB-LZVCV and UCB-QZVCV algorithms on various synthetically generated problem instances, and compares the results with existing algorithms including UCB1 [54] and UCB-CV [18]. The results show that both newly proposed algorithms achieve significant improvements in the stochastic MAB algorithms in comparison with that when UCB1 is used. Furthermore, given that it is claimed the regret of the UCB-CV is smaller by a factor $(1-\rho^2)$ in comparison with the existing algorithms when the rewards and CVs are normally distributed (ρ is the correlation coefficient of the reward and CVs) [18]. The proposed algorithms even outperform the UCB-CV algorithms in many problem instances where the rewards are normally distributed. In addition, UCB-LZVCV and UCB-QZVCV have the advantage to construct the relevant CVs directly from the original data, which can be appealing in situations where no side information in the form of CVs is available in the environment.

I take the initiative of coding the UCB-LZVCV and UCB-QZVCV algorithms from scratch. If it is interesting for the community, it can be written up as a package. In addition, I produce several regret plots to illustrate the performance of the proposed algorithms in comparison with existing algorithms in various settings. The reproducible code is available on this Github repository. The implementation of the UCB1 algorithm is based on the pseudo-code given in Algorithm 1 and 2. UCB-CV is implemented based on Algorithm 3 and 4. The newly-developed algorithms UCB-LZVCV and UCB-QZVCV are built to improve the performance of the MAB

algorithm as described in Algorithm 1, and these two algorithms are implemented based on Algorithm 5 and 6 respectively.

All of the instances in this chapter are repeated 100 times for each algorithm, and the average regret is plotted with a 95% CI (the vertical line on each curve indicates the CI).

6.1 Regret in Varying Distribution Forms

This section evaluates the regret performance of the 4 algorithms when the sample rewards and CVs (when UCB-CV is considered) in the environment are generated from different distribution forms. In this way, we can compare how does the performance of each algorithm vary in different distributional settings. In particular, we consider 3 common distribution forms (i.e Gaussian distribution, Student's t-distribution and logistic distribution). These are continuous distributions that are supported on the real numbers, hence we can directly construct the relevant ZV-CV from the original data using the derivatives of the log-likelihood.

For consistency, we set $K = 10$ arms and consider a consistent mean setting in instances 1-3. We also compare how does the regret change with different parameter settings, the details can be found in section 6.2.

6.1.1 Gaussian Distribution

Instance 1 Consider the case where the rewards have a multivariate normal distribution, and the reward of each arm has two components. In round t, the reward of arm $i \in [K]$ is given as follows:

$$f(X_{t,i}) = X_{t,i} = V_{t,i} + H_{t,i}, \tag{6.1}$$

where $V_{t,i} \sim N(\mu_{v,i}, \sigma^2_{v,i})$ and $H_{t,i} \sim N(\mu_{h,i}, \sigma^2_{h,i})$. Therefore, $X_{t,i} \sim N(\mu_{v,i} + \mu_{h,i}, \sigma^2_{v,i} + \sigma^2_{h,i})$. Set the mean for each arm $i \in [K]$ as $\mu_{v,i} = 0.6 - (i-1) \times 0.05$ and $\mu_{h,i} = 0.8 - (i-1) \times 0.05$, and the standard deviation value $\sigma_{v,i} = \sigma_{h,i} = 0.1$ for all arms.

For UCB-LZVCV and UCB-QZVCV algorithms, we assume no CV is available in the environment, and we construct the relevant ZV-CV directly from the

data using the derivatives of the log-likelihood. For the development of UCB-CV, we simply treat the second component $H_{t,i}$ as a CV that correlates with the reward $f(X_{t,i})$ (i.e. only 1 CV for this instance). The correlation between this CV and the reward is $\rho_i = \sqrt{\sigma^2_{h,i}/(\sigma^2_{v,i} + \sigma^2_{h,i})}$.

Performance Analysis The resulting regret plot is shown in Figure 6.1. This plot indicates the objectives of UCB-LZVCV and UCB-QZVCV algorithms are successfully reached in this instance setting. Both algorithms find a well-balanced trade-off between exploration and exploitation, and identify the optimal arm quickly. The regret performance is improved over UCB1 and UCB-CV by both algorithms. In addition, the advantages of the proposed UCB-LZVCV and UCB-QZVCV algorithms have been further demonstrated in this instance, as both algorithms construct the relevant ZV-CV directly from the original data and outperform the UCB-CV algorithm which requires additional side information from the environment. Furthermore, in comparison with UCB-LZVCV, we can spot that the performance of the MAB algorithms has been further improved by using UCB-QZVCV. One explanation for this can be that UCB-QZVCV uses two constructed CVs, while UCB-LZVCV only constructs one in this setup.

Regarding the performance of UCB1 and UCB-CV, the resulting plot verifies the theoretical presentations in chapter 4. For UCB1, we can see that the vast majority of the regret occurs during the beginning rounds, and the increase rate for regret becomes slower after 10000 rounds. Without knowing the mean and variance of reward for each arm, UCB1 reaches a logarithm regret in the long run. This matches with what is presented in section 4.2.3: the UCB1 algorithm has a natural ability to balance the trade-off between exploration and exploitation, and finally reaches an ideal regret over time. As for UCB-CV, the regret is almost flat after 5000 rounds, This further proves that applying CVs to estimate the mean rewards and incorporating variance estimates, can lead to tighter confidence bounds and reduced variance in the MC estimators. In comparison with the UCB1 algorithm, the agent can reach a better-balanced exploration-exploitation trade-off and begin to play the optimal arm earlier and more frequently with UCB-CV.

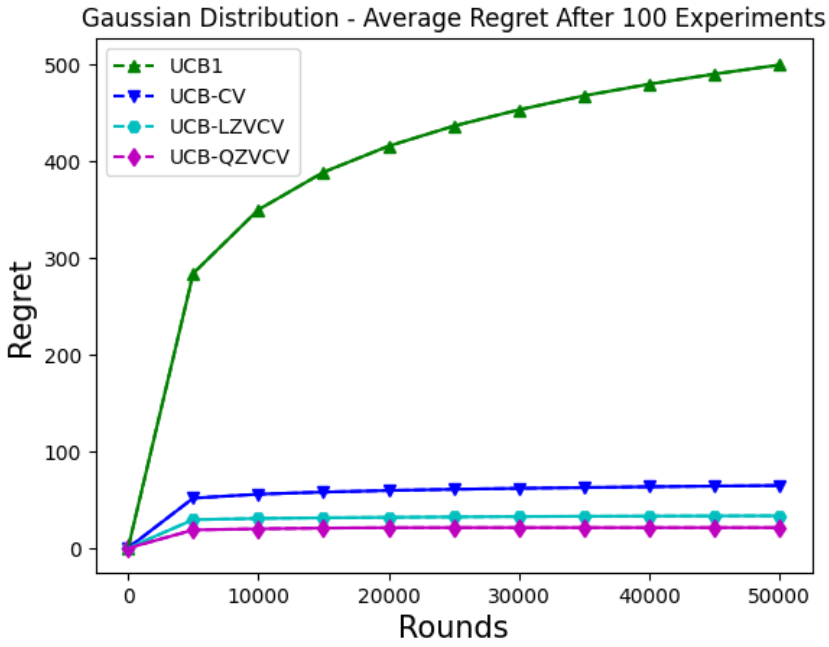

Figure 6.1: Comparing regret incurred by UCB1, UCB-CV, UCB-LZVCV and UCB-QZVCV when arms have normally distributed rewards.

Note that for illustration purposes, Figures 6.1-6.3 only show the average regret over 100 runs in certain rounds (i.e. starting from 0 to 50000 rounds, record once every 5000 rounds). Although we cannot see how the regret varies in the initialization rounds of each algorithm, it is clear that the vast majority of the regret occurs during the initialization rounds, where each arm is being tried several times for certain purposes as discussed in earlier chapters. The 95% CI indicated by the vertical line is plotted on these candidate rounds. However, the CI is too narrow to be visualized as the y-axis regret is too wide.

6.1.2 Student's t-Distribution

Instance 2 This instance follows the same parameter settings as the instance 1, expect the reward and associated CVs samples follow Student's t-distribution with 10 degrees of freedom.

Performance Analysis The resulting regret plot is shown in Figure 6.2. We can see that both UCB-LZVCV and UCB-QZVCV improve the performance of the MAB algorithm. These two proposed algorithms significantly outperform UCB1 in terms of resulting in tighter CI and achieving lower regret in the long run. Although the

UCB-CV outperforms the two proposed algorithms in this setting, UCB-CV has the limitation that requires available side information in the form of CVs from the environment. On the other hand, our proposed algorithms can directly construct the relevant ZV-CV from the data in the original MAB setting, as the logistic distribution is supported on all real numbers. For this reason, these two newly-developed algorithms seem to be appealing in the cases where no side information that correlates with the reward is available, or simply the mean of the available side information is unknown.

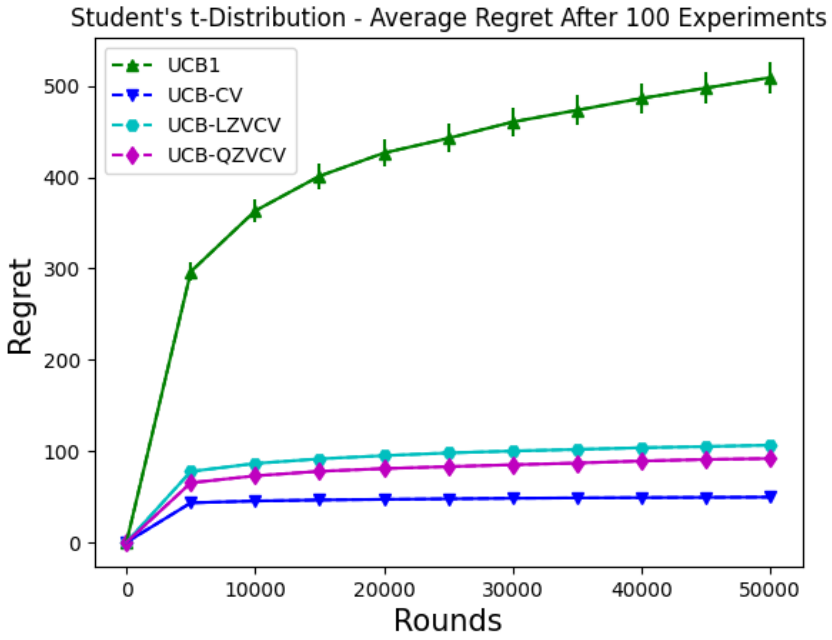

Figure 6.2: Comparing regret incurred by UCB1, UCB-CV, UCB-LZVCV and UCB-QZVCV when sample rewards follow Student's t-distribution.

6.1.3 Logistic Distribution

Instance 3 This instance follows the same parameter settings as the instance 1, expect the samples of reward and associated CVs are generated from logistic distribution, where the value of scale is set to 1 for all arms.

Performance Analysis The resulting regret plot is shown in Figure 6.3. The interpretation is similar to the performance analysis given in instance 2. The objectives of the UCB-LZVCV and UCB-QZVCV algorithms are successfully reached in this instance setting, as both algorithms significantly improve the regret performance of

the UCB1. Notice that the CI for UCB1 is wide in comparison with other listed algorithms. UCB1 uses a simple MC estimator $\hat{\mu}_i$ to estimate the mean reward for each arm μ_i. On the other hand, UCB-CV uses the available CVs to estimate the mean reward, UCB-LZVCV and UCB-QZVCV apply the self-constructed ZV-CV in the mean reward estimators. In addition, these three algorithms all incorporate the estimators of the variance for the estimated mean rewards. As a result, the mean reward estimate is improved, and the confidence bound is tighter for these three algorithms, which explains the plotting results here.

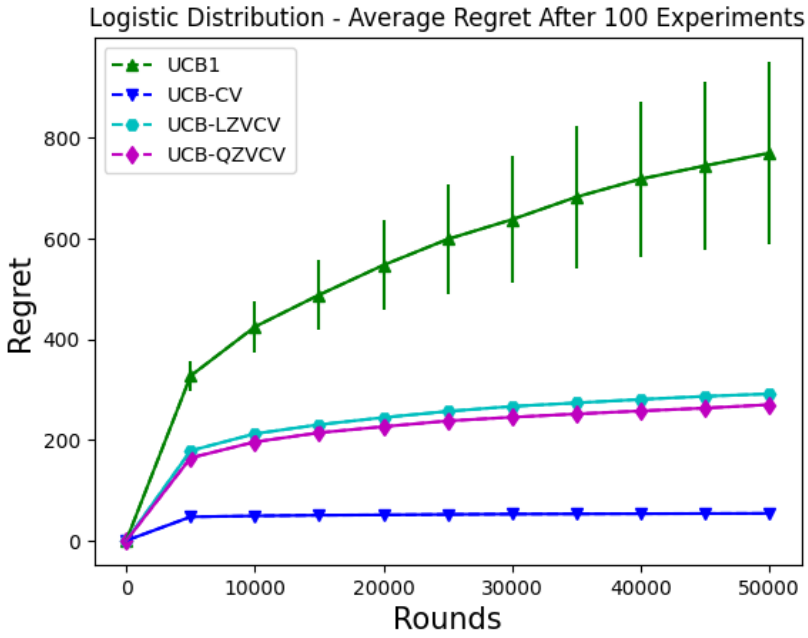

Figure 6.3: Comparing regret incurred by UCB1, UCB-CV, UCB-LZVCV and UCB-QZVCV when sample rewards follow logistic distribution.

6.2 Regret with Different Parameter Settings

This section examines how regret varies with different parameter settings to see if the experimental results are reliable.

6.2.1 Regret vs Varying Number of Arms K

For bandit algorithms, if the number of arms K increases, then the regret also increase linearly with respect to K. Intuitively, more arms mean more exploration, leading to more regret incurred in the initial rounds. To examine whether the pro-

posed algorithms follow this fact and test if we have correctly implemented the candidate algorithms, we derive problem instance with varying number of arms as follows:

Instance 4 Similar to the instance 1, the reward and associated CVs (if available) in the environment have a multivariate normal distribution, and the rewards are expressed as a sum of two components. Set the mean for each arm $i \in [K]$ as $\mu_{v,i} = 0.6 - (i-1) \times 0.05$ and $\mu_{h,i} = 0.8 - (i-1) \times 0.05$, and the standard deviation value $\sigma_{v,i} = \sigma_{h,i} = 0.1$ for all arms. We vary K over the values $\{10, 15, 20, 25, 30\}$.

Performance Analysis The resulting regret plot is shown in Figure 6.4. As expected, the results show that the regret incurred by all of the candidate algorithms increases linearly with an increase in the number of arms K.

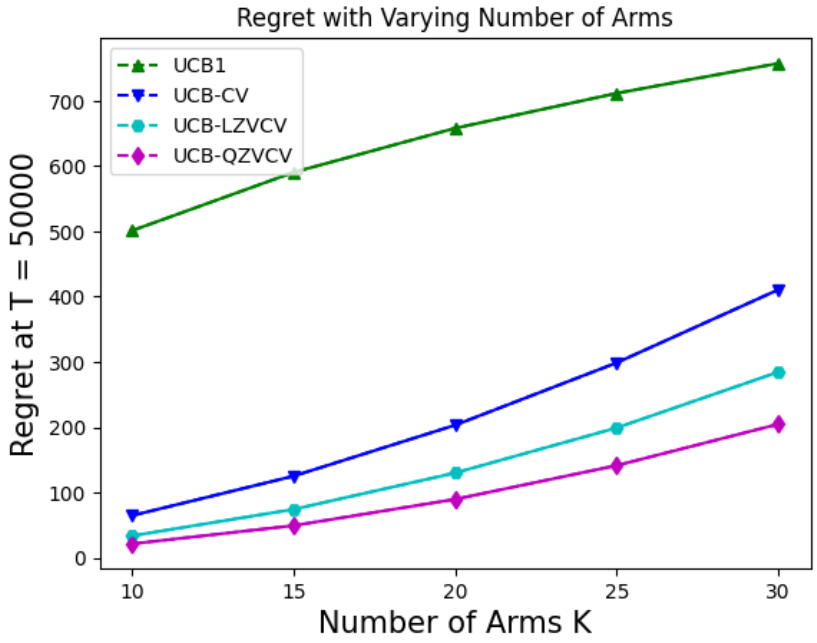

Figure 6.4: Comparing regret with different number of arms K.

6.2.2 Regret vs Varying Reward Variance

From chapter 3, we know that in regression estimator, any CV that correlates with the target of interest will be helpful in reducing the variance. Greater value of correlation coefficient leads to a greater achievement in variance reduction by applying a CV. It is also suggested that UCB-CV has better regret bounds when correlation

between arm rewards and CVs is higher [18]. To validate this and examine how regret varies with different variance in our proposed algorithms, we derive problem instance with different values of $\sigma_{v,i}$ as in instance 5. Note we are varying the standard derivation for V_i, not the standard derivation for H_i (H_i is the CV in the MAB-CV settings).

Instance 5 Similar to the instance 1, the reward and associated CVs (if available) in the environment have a multivariate normal distribution, and the rewards are expressed as sum of two components. Set $K = 10$ arms, the mean for each arm $i \in [K]$ as $\mu_{v,i} = 0.6 - (i-1) \times 0.05$ and $\mu_{h,i} = 0.8 - (i-1) \times 0.05$, and the standard deviation value $\sigma_{h,i} = 0.1$ for all arms. We vary $\sigma_{v,i}$ over the values $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. In this way, we know the variance of the reward varies over $\{0.02, 0.05, 0.1, 0.17, 0.26\}$ by simple numerical calculation. Increasing $\sigma_{v,i}$ results in an increased variance of the reward. For UCB-CV, the correlation coefficient between the CV and the reward is $\rho_i = \sqrt{\sigma_{h,i}^2/(\sigma_{v,i}^2 + \sigma_{h,i}^2)}$. Therefore, with varying $\sigma_{v,i}$, this correlation coefficient in UCB-CV varies over $\{0.7071, 0.4472, 0.3162, 0.2425, 0.1961\}$. Increasing $\sigma_{v,i}$ reduces the correlation coefficient ρ_i in the UCB-CV algorithm.

Performance Analysis The resulting regret plot is shown in Figure 6.5. As expected, we observe that the regret increases as the value of the standard derivation $\sigma_{v,i}$ goes up. It is because an increase in $\sigma_{v,i}$ leads to a decrease in the correlation coefficient between the reward and associated CVs in UCB-CV as suggested above. In addition, the result shows that the regret at a total number of rounds $T = 50000$ remains almost the same with varying values of $\sigma_{v,i}$ for UCB-LZVCV and UCB-QZVCV, it is because these algorithms construct the CVs directly from the sample data. Note that the increase in $\sigma_{v,i}$ also leads to an increase in the variance of the reward. As the variance for the reward increases, it becomes harder to distinguish the optimal arm with the highest reward. Therefore, regrets often go up with increased variance of the reward. This explains the slight increase in the UCB1 curve.

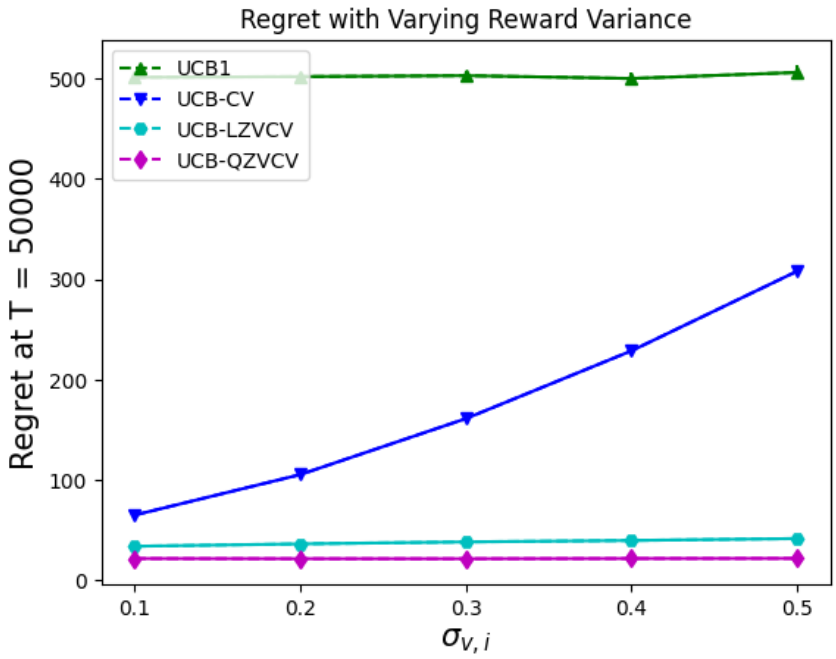

Figure 6.5: Comparing regret with different values of $\sigma_{v,i}$.

Chapter 7

Discussion

The preceding experiments demonstrated that the use of UCB-LZVCV and UCB-QZVCV algorithms improve the performance of the stochastic MAB algorithms. Both proposed algorithms achieve significantly lower regret in the long run in comparison with the UCB1 algorithm. Theoretically, it is because that the UCB1 algorithm uses the MC estimators to estimate the mean reward for each arm, which potentially leads to high variance results. On the other hand, the proposed algorithms UCB-LZVCV and UCB-QZVCV use the estimators based on the constructed ZV-CV and incorporate variance estimates to improve the variance of MC estimators. In this way, our proposed algorithms estimate the mean rewards with sharper confidence bounds, so the learner can find a better-balanced trade-off between exploration and exploitation, and start to play the optimal arm more frequently and earlier. As a result, the performance of the MAB algorithms is improved with our proposed algorithms.

Another advantage demonstrated by the theoretical presentations and the experiment results is that UCB-LZVCV and UCB-QZVCV can construct the ZV-CV directly from the original dataset by using the derivative of log-likelihood. This is particularly appealing as it is often the case where no side information is available in the form of CV in the practical MAB problems.

Furthermore, given that it is claimed that the regret of the UCB-CV is smaller by a factor $(1-\rho^2)$ in comparison with the existing algorithms when the rewards and CVs are normally distributed [18]. ρ is the correlation coefficient of the reward and

CVs. Our proposed algorithms UCB-LZVCV and UCB-QZVCV even outperform the UCB-CV algorithm in the same problem setting as presented in the UCB-CV paper where rewards are normally distributed [18] (see instance 1). In addition, we have further demonstrated that the two proposed algorithms also outperform UCB-CV in different parameter settings as presented in instances 4 and 5.

However, one limitation of the UCB-LZVCV and UCB-QZVCV algorithms is that we are constructing the relevant ZV-CV using the derivatives of the log-target density. For this to be a valid CV, we need the distribution of X to be defined on real numbers as presented in section equation 5.1. Consider a MAB setting where the rewards follow an uniform distribution, which is only defined on $[0,1]$. If one still constructs the CV using the derivatives of the log-target density, we can no longer show that such CVs integrate to zero.

Chapter 8

General Conclusions

One of the challenges that arise in RL is to find a well-balanced trade-off between exploitation and exploration. MAB problem is a classical problem that well-demonstrated the exploitation and exploration dilemma in RL. Various algorithms have been designed to balance this trade-off and improve the performance of the stochastic MAB algorithms. The goal of this thesis was to develop some algorithms that can construct some ZV-CV directly from the data, adopt these constructed ZV-CV to reduce the variance in the mean reward estimators, and improve the regret performance of the MAB algorithm.

To do so, basic concepts about the MC and CV methods were reviewed in an intuitive manner with motivating examples to introduce the general framework of the MC simulations in terms of solving the integration problems, and explain why CVs can be adopted to reduce the variance of the MC estimators. Next, an interesting class of the RL problems called MAB is introduced, and its connections to MC simulations and CVs techniques are explained with some running examples. Equipped with this background, two algorithms that improve the performance of the stochastic MAB algorithm are presented. In detail, the first algorithm UCB1 uses the MC simulations to estimate the mean rewards of the MAB problem based on averaging sample returns, and the second algorithm UCB-CV uses the available side information in the form of CVs in the environment to improve the variance of estimates. Motivated by the fact that it is often the case where no side information that correlates with the reward is available, or simply the mean of the available side

information is unknown in the environment, the literature on ZV method and the associated ZV-CV was reviewed, as it suggests ways to directly construct CV using the derivative of the log-likelihood. Combining the previously reviewed concepts, two new versions of the UCB algorithms that learn the relevant ZV-CV directly from the original data to reduce the variance of the MC estimators were proposed to improve the performance of the stochastic MAB algorithm. Finally, the results of the numerical experiments that compare the existing algorithms with the two newly-developed algorithms were presented and the implications and limitations were also discussed.

In the numerical experiments, both UCB-LZVCV and UCB-QZVCV algorithms were found to be successfully applied to improve the regret performance of the MAB algorithms when the distribution of rewards are defined on real numbers. Given that it is claimed that the regret of UCB-CV is smaller compare to the existing algorithms (including UCB-V [56], Thompson Sampling [63] and EUCBV [57] in this paper [18]) when the rewards follow normal distribution. The experimental results demonstrated that the proposed algorithms even outperform the UCB-CV algorithm in many problem instances where the rewards are normally distributed (instances 1, 4 and 5). These showed the algorithms presented in this thesis are potentially of interest. However, in the ZV methods, the relevant ZV-CV are constructed using the derivatives of the log-target density, which means we need the distribution of X to be defined on real numbers to construct the valid ZV-CV. This implies that further research should target the extensions where the rewards follow more general distributions [18][58].

Moreover, further future work can focus on the following area. First, the algorithms presented in this thesis construct the ZV-CV with first and second-order polynomial, it may be interesting to implement new variants of the UCB broadly that constructs the CVs based on more complicated or even non-parametric transformations [13], such as based on higher-order polynomials [29], kernels [64][65][66], neural networks [67][68][69]. However, one of the challenges with these more complex methods is they are more complex to fit. We may gain variance reduction from

using these more complicated methods with the cost of increased computation time in the MAB algorithm. This trade-off should be well-balanced if one decides to develop new versions of UCB based on these more advanced methods. Another interesting direction is to combine the UCB-CV with the proposed algorithms, in which we use both the side information available in the form of CVs and the self-constructed ZV-CV to reduce the variance in the mean reward estimators. In this way, it can potentially lead to better results than doing one separately. Furthermore, although the initial code for the algorithms presented in this thesis is available on this Github repository, it may be also interesting for the community to write up as a package.

Bibliography

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[2] Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.

[3] Zdravko Botev and Ad Ridder. Variance reduction. *Wiley StatsRef: Statistics Reference Online*, pages 1–6, 2017.

[4] Stephen S Lavenberg and Peter D Welch. A perspective on the use of control variables to increase the efficiency of monte carlo simulations. *Management Science*, 27(3):322–335, 1981.

[5] Stephen S Lavenberg, Thomas L Moeller, and Peter D Welch. Statistical results on control variables with application to queueing network simulation. *Operations Research*, 30(1):182–202, 1982.

[6] Barry L Nelson. Batch size effects on the efficiency of control variates in simulation. *European Journal of Operational Research*, 43(2):184–196, 1989.

[7] Stephen S Lavenberg and Peter D Welch. A perspective on the use of control variables to increase the efficiency of monte carlo simulations. *Management Science*, 27(3):322–335, 1981.

[8] Christiane Lemieux. Control variates. *Wiley StatsRef: Statistics Reference Online*, pages 1–8, 2014.

[9] B Fathi Vajargah, A Salimipour, and S Salahshour. Variance analysis of control variate technique and applications in asian option pricing. *Int. J. Industrial Mathematics*, 8(1):61–67, 2016.

[10] Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer Science & Business Media, 2013.

[11] John Hull and Alan White. The use of the control variate technique in option pricing. *Journal of Financial and Quantitative analysis*, 23(3):237–251, 1988.

[12] Adrian FM Smith and Gareth O Roberts. Bayesian computation via the gibbs sampler and related markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(1):3–23, 1993.

[13] Shijing Si, Chris Oates, Andrew B Duncan, Lawrence Carin, François-Xavier Briol, et al. Scalable control variates for monte carlo methods via stochastic optimization. *arXiv preprint arXiv:2006.07487*, 2020.

[14] John Paisley, David Blei, and Michael Jordan. Variational bayesian inference with stochastic search. *arXiv preprint arXiv:1206.6430*, 2012.

[15] Ludovic Goudenège, Andrea Molent, and Antonino Zanette. Variance reduction applied to machine learning for pricing bermudan/american options in high dimension. *arXiv preprint arXiv:1903.11275*, 2019.

[16] Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *J. Mach. Learn. Res.*, 21(132):1–62, 2020.

[17] Ching-An Cheng, Xinyan Yan, and Byron Boots. Trajectory-wise control variates for variance reduction in policy gradient methods. In *Conference on Robot Learning*, pages 1379–1394. PMLR, 2020.

[18] Arun Verma and Manjesh K Hanawal. Stochastic multi-armed bandits with control variates. *arXiv preprint arXiv:2105.03962*, 2021.

[19] Joel Veness, Marc Lanctot, and Michael Bowling. Variance reduction in monte-carlo tree search. *Advances in Neural Information Processing Systems*, 24:1836–1844, 2011.

[20] Andrew J Cassey and Ben O Smith. Simulating confidence for the ellison–glaeser index. *Journal of Urban Economics*, 81:85–103, 2014.

[21] M. Rosenbluth and A. W. Rosenbluth. Monte carlo calculation of the average extension of molecular chains. *Journal of Chemical Physics*, 23:356–359, 1955.

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

[23] Joel Veness, Marc Lanctot, and Michael Bowling. Variance reduction in monte-carlo tree search. *Advances in Neural Information Processing Systems*, 24:1836–1844, 2011.

[24] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, New York, 2004.

[25] Lawrence M Graves. Riemann integration and taylor's theorem in general analysis. *Transactions of the American Mathematical Society*, 29(1):163–177, 1927.

[26] Kendall E Atkinson. *An introduction to numerical analysis (2nd ed.* John wiley & sons, New York, 1989.

[27] RA Thisted. *Elements of Statistical Computing: Numerical Computation*. Chapman and Hall, New York, 1988.

[28] William H Press, Saul A Teukolsky, Brian P Flannery, and William T Vetterling. *Numerical recipes in Fortran 77: volume 1, volume 1 of Fortran numerical recipes: the art of scientific computing*. Cambridge university press, 1992.

[29] Leah F South, Chris J Oates, Antonietta Mira, and Christopher Drovandi. Regularised zero-variance control variates for high-dimensional variance reduction. *arXiv preprint arXiv:1811.05073*, 2018.

[30] Mike Wu, Noah Goodman, and Stefano Ermon. Differentiable antithetic sampling for variance reduction in stochastic variational inference. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2877–2886. PMLR, 2019.

[31] Aaron R Dinner, Erik H Thiede, Brian Van Koten, and Jonathan Weare. Stratification as a general variance reduction method for markov chain monte carlo. *SIAM/ASA Journal on Uncertainty Quantification*, 8(3):1139–1188, 2020.

[32] RY Rubinstein and G Samorodnitsky. Variance reduction by the use of common and antithetic random variables. *Journal of Statistical Computation and Simulation*, 22(2):161–180, 1985.

[33] Juan Miguel Montes, Valentina Prezioso, and Wolfgang J Runggaldier. Monte carlo variance reduction by conditioning for pricing with underlying a continuous-time finite state markov process. *SIAM Journal on Financial Mathematics*, 5(1):557–580, 2014.

[34] Peter W Glynn and Donald L Iglehart. Importance sampling for stochastic simulations. *Management science*, 35(11):1367–1392, 1989.

[35] Chong Wang, Xi Chen, Alexander Smola, and Eric P Xing. Variance reduction for stochastic gradient optimization. 2013.

[36] Joel Veness, Marc Lanctot, and Michael Bowling. Variance reduction in monte-carlo tree search. *Advances in Neural Information Processing Systems*, 24:1836–1844, 2011.

[37] Hastagiri P Vanchinathan, Isidor Nikolic, Fabio De Bona, and Andreas Krause. Explore-exploit in top-n recommender systems via gaussian pro-

cesses. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 225–232, 2014.

[38] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[39] Jérémie Mary, Romaric Gaudel, and Philippe Preux. Bandits and recommender systems. In *International Workshop on Machine Learning, Optimization and Big Data*, pages 325–336. Springer, 2015.

[40] Andrea Barraza-Urbina. The exploration-exploitation trade-off in interactive recommender systems. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 431–435, 2017.

[41] David E Losada, Javier Parapar, and Alvaro Barreiro. Multi-armed bandits for adjudicating documents in pooling-based evaluation of information retrieval systems. *Information Processing & Management*, 53(5):1005–1025, 2017.

[42] Kaize Ding, Jundong Li, and Huan Liu. Interactive anomaly detection on attributed networks. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 357–365, 2019.

[43] Bing Liu, Tong Yu, Ian Lane, and Ole Mengshoel. Customized nonlinear bandits for online response selection in neural conversation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[44] Audrey Durand, Charis Achilleos, Demetris Iacovides, Katerina Strati, Georgios D Mitsis, and Joelle Pineau. Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis. In *Machine learning for healthcare conference*, pages 67–82. PMLR, 2018.

[45] Weiwei Shen, Jun Wang, Yu-Gang Jiang, and Hongyuan Zha. Portfolio choices with orthogonal bandit learning. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[46] Xiaoguang Huo and Feng Fu. Risk-aware multi-armed bandit problem with application to portfolio selection. *Royal Society open science*, 4(11):171377, 2017.

[47] Himan Abdollahpouri and Steve Essinger. Towards effective exploration/exploitation in sequential music recommendation. *arXiv preprint arXiv:1812.03226*, 2018.

[48] Xinxi Wang, Yi Wang, David Hsu, and Ye Wang. Exploration in interactive personalized music recommendation: a reinforcement learning approach. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 11(1):1–22, 2014.

[49] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. *Advances in neural information processing systems*, 24:2249–2257, 2011.

[50] Anisio Lacerda, Rodrygo LT Santos, Adriano Veloso, and Nivio Ziviani. Improving daily deals recommendation using explore-then-exploit strategies. *Information Retrieval Journal*, 18(2):95–122, 2015.

[51] 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.

[52] Choon Hui Teo, Houssam Nassif, Daniel Hill, Sriram Srinivasan, Mitchell Goodman, Vijai Mohan, and SVN Vishwanathan. Adaptive, personalized diversity for visual discovery. In *Proceedings of the 10th ACM conference on recommender systems*, pages 35–38, 2016.

[53] Joannes Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In *European conference on machine learning*, pages 437–448. Springer, 2005.

[54] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

[55] Aurélien Garivier and Olivier Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory*, pages 359–376. JMLR Workshop and Conference Proceedings, 2011.

[56] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

[57] Subhojyoti Mukherjee, KP Naveen, Nandan Sudarsanam, and Balaraman Ravindran. Efficient-ucbv: An almost optimal algorithm using variance estimates. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[58] Barry L Nelson. Control variate remedies. *Operations Research*, 38(6):974–992, 1990.

[59] Theodore Papamarkou, Antonietta Mira, and Mark Girolami. Zero variance differential geometric markov chain monte carlo algorithms. *Bayesian Analysis*, 9(1):97–128, 2014.

[60] Roland Assaraf and Michel Caffarel. Zero-variance principle for monte carlo algorithms. *Physical review letters*, 83(23):4682, 1999.

[61] Antonietta Mira, Reza Solgi, and Daniele Imparato. Zero variance markov chain monte carlo for bayesian estimators. *Statistics and Computing*, 23(5):653–662, 2013.

[62] Nial Friel, Antonietta Mira, and Chris J Oates. Exploiting multi-core architectures for reduced-variance estimation with intractable likelihoods. *Bayesian Analysis*, 11(1):215–245, 2016.

[63] Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *Artificial intelligence and statistics*, pages 99–107. PMLR, 2013.

[64] Chris J Oates, Mark Girolami, and Nicolas Chopin. Control functionals for monte carlo integration. *Journal of the Royal Statistical Society B: Statistical Methodology*, 2017.

[65] Chris J Oates, Jon Cockayne, François-Xavier Briol, and Mark Girolami. Convergence rates for a class of estimators based on stein's method. *Bernoulli*, 25(2):1141–1159, 2019.

[66] Alessandro Barp, Chris Oates, Emilio Porcu, Mark Girolami, et al. A riemann-stein kernel method. *arXiv preprint arXiv:1810.04946*, 2018.

[67] Will Grathwohl, Dami Choi, Yuhuai Wu, Geoffrey Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *arXiv preprint arXiv:1711.00123*, 2017.

[68] Hao Liu, Yihao Feng, Yi Mao, Dengyong Zhou, Jian Peng, and Qiang Liu. Action-depedent control variates for policy optimization via stein's identity. *arXiv preprint arXiv:1710.11198*, 2017.

[69] Ruosi Wan, Mingjun Zhong, Haoyi Xiong, and Zhanxing Zhu. Neural control variates for monte carlo variance reduction. In *ECML/PKDD (2)*, pages 533–547, 2019.

[70] F Hayashi. *Econometrics*. Princeton University Press, 2000.

[71] Sara A Van De Geer. Least squares estimation. *Encyclopedia of Statistics in Behavioral Science*, 2005.

[72] Bruce Schmeiser. Batch size effects in the analysis of simulation output. *Operations Research*, 30(3):556–568, 1982.

Appendix A

Regression Theory

Consider the general regression instance with m samples and f features:

$$Y_i = \boldsymbol{X_i}^T \boldsymbol{\theta} + \epsilon_i, i \in 1, 2, ..., m, \tag{A.1}$$

where $Y_i \in \mathbb{R}$ is the ith response variable, $\boldsymbol{X}_i \in \mathbb{R}^f$ is the ith feature vector, $\boldsymbol{\theta} \in \mathbb{R}^f$ is the unknown regression parameters, and ϵ_i is a Gaussian noise with mean 0 and constant variance σ^2. These ϵ_i are independent of $\boldsymbol{X}_i$ and form a IID sequence. Let

$$\boldsymbol{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_m \end{pmatrix}, \boldsymbol{X} = \begin{pmatrix} X_{11} & \dots & X_{1f} \\ \vdots & \dots & \vdots \\ X_{m1} & \dots & X_{mf} \end{pmatrix}, \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_m \end{pmatrix}, \tag{A.2}$$

the least square estimator is given by

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{Y}. \tag{A.3}$$

Fact 1. The finite sample properties of $\hat{\boldsymbol{\theta}}$ are:

1. $\mathbb{E}[\hat{\boldsymbol{\theta}} | \boldsymbol{X}] = \boldsymbol{\theta}$ (unbiased estimator)
2. $var(\hat{\boldsymbol{\theta}} | \boldsymbol{X}) = \sigma^2 (\boldsymbol{X}^T \boldsymbol{X})^{-1}$ (expression for the variance)
3. $var(\hat{\boldsymbol{\theta}}_i | \boldsymbol{X}) = \sigma^2 (\boldsymbol{X}^T \boldsymbol{X})^{-1}_{ii}$ (element-wise variance)

where $(\boldsymbol{X}^T \boldsymbol{X})^{-1}_{ii}$ is the ii-th element of the matrix $(\boldsymbol{X}^T \boldsymbol{X})^{-1}$. The first two properties

are derived from [70], and the third is taken from [71].

The finite sample properties for the estimator of variance σ^2 is then reviewed.

Fact 2. (Proposition 1.2 of [70]) Let $\hat{\sigma}^2 = \frac{1}{m-f}\sum_{i=1}^{m}(Y_i - \boldsymbol{X}_i^T\hat{\boldsymbol{\theta}})^2$ be estimator of σ^2 and $m > f$ (so that $\hat{\sigma}^2$ is well defined). Then $\mathbb{E}[\hat{\sigma}^2|\boldsymbol{X}] = \sigma^2$ which implies that $\hat{\sigma}^2$ is an unbiased estimator of σ^2.

Appendix B

Missing Proofs in UCB-CV

Lemma 1 *Let the reward and CV of each arm have a multivariate normal distribution. After observing m samples of reward and CV from arm i, define* $\hat{v}_{m,i} = \frac{Z_{m,i}\hat{\sigma}^2_{c,i}(m)}{m}$, *where*

$$Z_{m,i} = \left(1 - \frac{(\sum_{r=1}^{m}(h(X_{r,i}) - \theta_i))^2}{m\sum_{r=1}^{m}(h(X_{r,i}) - \theta_i)^2}\right)^{-1}, and, \hat{\sigma}^2_{c,i}(m) = \frac{1}{m-2}\sum_{r=1}^{m}(\tilde{f}(X_r) - \hat{\mu}^c_{m,i})^2.$$

then $\hat{v}_{m,i}$ *is an unbiased variance estimator of* $\hat{\mu}^c_{m,i}$ $\left(i.e. \mathbb{E}[\hat{v}_{m,i}] = var(\hat{\mu}^c_{m,i})\right)$.

Proof In UCB-CV, we consider new samples for arm i ($i \in [K]$):

$$\tilde{f}(X_t) = f(X_t) + \alpha^*(\theta - h(X_t)), t \in \{1,2,...,m\}, \tag{B.1}$$

where the subscript for arm (indexed by i) is dropped for simplicity. Under the normality assumption, $\tilde{f}(X_t)$ can be written as follows:

$$\tilde{f}(X_t) = \mu + \alpha^*(\theta - h(X_t)) + \epsilon_t, t \in \{1,2,...,m\}, \tag{B.2}$$

where $\epsilon_1, \epsilon_2, ..., \epsilon_m$ are IID normally distributed random variables with mean 0 and variance σ^2. Thus, we have

$$\tilde{\boldsymbol{f}}(\boldsymbol{X}) = \boldsymbol{Y}\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \tag{B.3}$$

where

$$\tilde{\boldsymbol{f}}(\boldsymbol{X}) = \begin{pmatrix} \tilde{f}(X_1) \\ \vdots \\ \tilde{f}(X_m) \end{pmatrix}, \boldsymbol{Y} = \begin{pmatrix} 1 & \theta - h(X_1) \\ \vdots & \vdots \\ 1 & \theta - h(X_m) \end{pmatrix}, \boldsymbol{\alpha} = \begin{pmatrix} \mu \\ \alpha^* \end{pmatrix}, \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_m \end{pmatrix}. \tag{B.4}$$

Let the least square estimator of μ be $\hat{\mu}^c$ and α^* be $\hat{\alpha}^*$. With the finite sample properties of the least square estimator (Fact 1), we know that the variance of mean estimator for arm i in the UCB-CV algorithm is

$$var(\hat{\mu}^c_{m,i}) = \sigma^2_{c,i}(\boldsymbol{Y}^T\boldsymbol{Y})^{-1}_{11}, \tag{B.5}$$

where $var(\tilde{f}(X_i)) = \sigma^2_{c,i}$.

Let's compute the value of $(\boldsymbol{Y}^T\boldsymbol{Y})^{-1}_{11}$ first.

$$\boldsymbol{Y}^T\boldsymbol{Y} = \begin{pmatrix} 1 & \dots & 1 \\ \theta - h(X_1) & \dots & \theta - h(X_m) \end{pmatrix} \begin{pmatrix} 1 & \theta - h(X_1) \\ \vdots & \vdots \\ 1 & \theta - h(X_m) \end{pmatrix} = \begin{pmatrix} m & \sum_{r=1}^m (\theta - h(X_r)) \\ \sum_{r=1}^m (\theta - h(X_r)) & \sum_{r=1}^m (\theta - h(X_r))^2 \end{pmatrix}$$

$$(\boldsymbol{Y}^T\boldsymbol{Y})^{-1} = \frac{1}{m\sum_{r=1}^m (\theta - h(X_r))^2 - (\sum_{r=1}^m (\theta - h(X_r)))^2} \begin{pmatrix} \sum_{r=1}^m (\theta - h(X_r))^2 & -\sum_{r=1}^m (\theta - h(X_r)) \\ -\sum_{r=1}^m (\theta - h(X_r)) & m \end{pmatrix}$$

$$\begin{aligned} (\boldsymbol{Y}^T\boldsymbol{Y})^{-1}_{11} &= \frac{\sum_{r=1}^m (\theta - h(X_r))^2}{m\sum_{r=1}^m (\theta - h(X_r))^2 - (\sum_{r=1}^m (\theta - h(X_r)))^2} \\ &= \frac{1}{m} \cdot \frac{1}{1 - \frac{(\sum_{r=1}^m (\theta - h(X_r)))^2}{m\sum_{r=1}^m (\theta - h(X_r))^2}} \\ &= \frac{1}{m} \cdot \left(1 - \frac{(\sum_{r=1}^m (\theta - h(X_r)))^2}{m\sum_{r=1}^m (\theta - h(X_r))^2}\right)^{-1} \\ &= \frac{Z_m}{m} \end{aligned}$$

where $Z_m = \left(1 - \frac{(\sum_{r=1}^m (\theta - h(X_r)))^2}{m\sum_{r=1}^m (\theta - h(X_r))^2}\right)^{-1}$.

After observing m samples of rewards and associated CV from arm i, we get the estimator for $var(\hat{\mu}^c)$ as follows:

$$\hat{v}_{m,i} = \frac{Z_m \hat{\sigma}^2_{c,i}(m)}{m}. \tag{B.6}$$

With Fact 2, we know $\hat{\sigma}^2_{c,i}(m) = \frac{1}{m-2}\sum_{r=1}^{m}(\tilde{f}(X_r) - \hat{\mu}^c_{m,i})^2$ is an unbiased estimator of $\sigma^2_{c,i}$. With Theorem 1 and Theorem 2 in Nelson's paper [58], we know $\mathbb{E}[\hat{v}_{m,i}] = var(\hat{\mu}^c_{m,i})$, which implies that $\hat{v}_{m,i}$ is an unbiased estimator of $var(\hat{\mu}^c_{m,i})$.

Lemma 2 *Let m be the number of rewards and associated CV samples from arm i in time step t. Then*

$$\mathbb{P}\left(|\hat{\mu}^c_{m,i} - \mu_i| \geq V_{t,m,1}\sqrt{\hat{v}_{m,i}}\right) = \frac{2}{t^2}, \tag{B.7}$$

where $V_{t,m,1}$ denotes $100(1-1/t^2)^{th}$ percentile value of the t-distribution with $m-2$ degrees of freedom and $\hat{v}_{m,i}$ is an unbiased estimator for variance of $\hat{\mu}^c_{m,i}$.

Proof The proof follows from Theorem 1 and Theorem 2 in Nelson's paper [58]. Here, we have m observations of the arm rewards and associated CV, we replace their parameter with our arm specific parameters. We use t-distribution for CI, therefore the value of $\mathbb{P}\left(|\hat{\mu}^c_{m,i} - \mu_i| \geq V_{t,m,1}\sqrt{\hat{v}_{m,i}}\right)$ depends only on the value of $V_{t,m,1}$. Therefore we have

$$\begin{aligned}
\mathbb{P}\left(|\hat{\mu}^c_{m,i} - \mu_i| \geq V_{t,m,1}\sqrt{\hat{v}_{m,i}}\right) &= 1 - \mathbb{P}\left(|\hat{\mu}^c_{m,i} - \mu_i| \leq V_{t,m,1}\sqrt{\hat{v}_{m,i}}\right) \\
&= 1 - (1 - \frac{2}{t^2}) \\
&= \frac{2}{t^2}.
\end{aligned}$$

Appendix C

Missing Proofs in UCB-LZVCV

Claim 1 *When we have arms with normally distributed rewards,* $h(X_{t,i}) = -\frac{1}{2} \cdot \frac{-X_{t,i}+\mu_i}{\sigma_i^2}$. *In this way, the expected value of the LZV-CV is zero (i.e.* $\mathbb{E}[h(X_{t,i})] = \theta_i = 0$*).*

Proof Consider arms with normally distributed rewards in the MAB problem. It is known that in the linear $P(X) = \alpha^T X$ case, $\alpha \in \mathbb{R}^{d_a}$, $X \in \mathbb{R}^{d_x}$, $X_i \in \mathbb{R}^{d_x}$, $d_a = d_x$ (section 5.1.2.1). In our MAB problem setting, the reward is in one dimension, $d_x = d_a = 1$. Given $f(X_{t,i}) = X_{t,i}$ and $X_{t,i}$ are normally distributed with mean μ_i and variance σ_i^2. The density is

$$p(X_{t,i}) = \frac{1}{\sqrt{2\pi}\sigma_i} exp(-\frac{(X_{t,i}-\mu_i)^2}{2\sigma_i^2}). \tag{C.1}$$

Thus the LZV-CV can be constructed as

$$\begin{aligned} h(X_{t,i}) &= z(X_{t,i}) \\ &:= -\frac{1}{2}\nabla[log(p(X_{t,i}))] \\ &= -\frac{1}{2}\nabla[log(\frac{1}{\sqrt{2\pi}\sigma}) - \frac{(X_{t,i}-\mu)^2}{2\sigma^2}] \\ &= -\frac{1}{2} \cdot \frac{-2(X_{t,i}-\mu)}{2\sigma^2} \\ &= -\frac{1}{2} \cdot \frac{-X_{t,i}+\mu}{\sigma^2}. \end{aligned}$$

In this way, $\mathbb{E}[h(X_{t,i})] = \frac{-\mu_i+\mu_i}{\mathbb{E}[\sigma_i^2]} = 0$.

Lemma 3 *When the reward and constructed LZV-CV of each arm have a multivariate normal distribution. After observing m samples of reward and constructing m LZV-CV samples from arm i, define* $\hat{v}_{m,i} = \frac{Z_{m,i}\hat{\sigma}^2_{lzc,i}(m)}{m}$, *where*

$$Z_{m,i} = \left(1 - \frac{(\sum_{r=1}^{m}(h(X_{r,i})))^2}{m\sum_{r=1}^{m}(h(X_{r,i}))^2}\right)^{-1}, and, \hat{\sigma}^2_{lzc,i}(m) = \frac{1}{m-2}\sum_{r=1}^{m}(\tilde{f}(X_{r,i}) - \hat{\mu}^{lzc}_{m,i})^2.$$

then $\hat{v}_{m,i}$ *is an unbiased variance estimator of* $\hat{\mu}^{lzc}_{m,i}$ $\left(i.e. \mathbb{E}[\hat{v}_{m,i}] = var(\hat{\mu}^{lzc}_{m,i})\right)$.

Proof In the UCB-LZVCV algorithm, suppose we observe m new samples for the arm $i \in [K]$:

$$\tilde{f}(X_t) = f(X_t) - \hat{\alpha}^* h(X_t), t \in \{1,2,...,m\}, \tag{C.2}$$

where the subscript for arm (indexed by i) is dropped for simplicity. Under the normality assumption, $\tilde{f}(X_t)$ can be written as follows:

$$\tilde{f}(X_t) = \mu - \alpha^* h(X_t) + \epsilon_t, t \in \{1,2,...,m\}, \tag{C.3}$$

where $\epsilon_1, \epsilon_2, ..., \epsilon_m$ are IID normally distributed random variables with mean 0 and variance σ^2. Thus, we have

$$\tilde{\boldsymbol{f}}(\boldsymbol{X}) = \boldsymbol{Y}\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \tag{C.4}$$

where

$$\tilde{\boldsymbol{f}}(\boldsymbol{X}) = \begin{pmatrix} \tilde{f}(X_1) \\ \vdots \\ \tilde{f}(X_m) \end{pmatrix}, \boldsymbol{Y} = \begin{pmatrix} 1 & -h(X_1) \\ \vdots & \vdots \\ 1 & -h(X_m) \end{pmatrix}, \boldsymbol{\alpha} = \begin{pmatrix} \mu \\ \alpha^* \end{pmatrix}, \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_m \end{pmatrix}. \tag{C.5}$$

Let the least square estimator of μ be $\hat{\mu}^{lzc}$ and α^* be $\hat{\alpha}^*$. With the finite sample properties of the least square estimator (Fact 1), we know that the variance of mean

estimator for arm i in the UCB-LZVCV algorithm is

$$var(\hat{\mu}_{m,i}^{lzc}) = \sigma_{lzc,i}^2 (\boldsymbol{Y}^T \boldsymbol{Y})_{11}^{-1}, \tag{C.6}$$

where $var(\tilde{f}(X_i)) = \sigma_{lzc,i}^2$.

Let's compute the value of $(\boldsymbol{Y}^T \boldsymbol{Y})_{11}^{-1}$ first.

$$\boldsymbol{Y}^T \boldsymbol{Y} = \begin{pmatrix} 1 & \dots & 1 \\ -h(X_1) & \dots & -h(X_m) \end{pmatrix} \begin{pmatrix} 1 & -h(X_1) \\ \vdots & \vdots \\ 1 & -h(X_m) \end{pmatrix} = \begin{pmatrix} m & \sum_{r=1}^m -h(X_r) \\ \sum_{r=1}^m -h(X_r) & \sum_{r=1}^m (-h(X_r))^2 \end{pmatrix}$$

$$(\boldsymbol{Y}^T \boldsymbol{Y})^{-1} = \frac{1}{m\sum_{r=1}^m (-h(X_r))^2 - (\sum_{r=1}^m -h(X_r))^2} \begin{pmatrix} \sum_{r=1}^m (-h(X_r))^2 & -\sum_{r=1}^m -h(X_r) \\ -\sum_{r=1}^m -h(X_r) & m \end{pmatrix}$$

$$\begin{aligned} (\boldsymbol{Y}^T \boldsymbol{Y})_{11}^{-1} &= \frac{\sum_{r=1}^m (-h(X_r))^2}{m\sum_{r=1}^m (-h(X_r))^2 - (\sum_{r=1}^m -h(X_r))^2} \\ &= \frac{1}{m} \cdot \frac{1}{1 - \frac{(\sum_{r=1}^m -h(X_r))^2}{m\sum_{r=1}^m (-h(X_r))^2}} \\ &= \frac{1}{m} \cdot \left(1 - \frac{(\sum_{r=1}^m -h(X_r))^2}{m\sum_{r=1}^m (-h(X_r))^2}\right)^{-1} \\ &= \frac{1}{m} \cdot \left(1 - \frac{(\sum_{r=1}^m h(X_r))^2}{m\sum_{r=1}^m (h(X_r))^2}\right)^{-1} \\ &= \frac{Z_m}{m} \end{aligned}$$

where $Z_m = \left(1 - \frac{(\sum_{r=1}^m h(X_r))^2}{m\sum_{r=1}^m (h(X_r))^2}\right)^{-1}$.

After observing m samples of rewards and constructing m associated LZV-CV from arm i, we get the estimator for $\hat{\mu}^{lzc}$ as follows:

$$\hat{v}_{m,i} = \frac{Z_m \hat{\sigma}_{lzc,i}^2(m)}{m}. \tag{C.7}$$

With Fact 2, we know $\hat{\sigma}_{lzc,i}^2(m) = \frac{1}{m-2}\sum_{r=1}^m (\tilde{f}(X_r) - \hat{\mu}_{m,i}^{lzc})^2$ is an unbiased estimator

of $\sigma^2_{lzc,i}$. With Theorem 1 and Theorem 2 in Nelson's paper [58], $\mathbb{E}[\hat{v}_{m,i}] = var(\hat{\mu}^{lzc}_{m,i})$ implies that $\hat{v}_{m,i}$ is an unbiased estimator of $var(\hat{\mu}^{lzc}_{m,i})$.

Lemma 4 *Let m be the number of rewards and constructed LZV-CV samples from arm i in time step t. Then*

$$\mathbb{P}\left(|\hat{\mu}^{lzc}_{m,i} - \mu_i| \geq V_{t,m,1}\sqrt{\hat{v}_{m,i}}\right) = \frac{2}{t^2}, \tag{C.8}$$

where $V_{t,m,1}$ denotes $100(1-1/t^2)^{th}$ percentile value of the t-distribution with $m-2$ degrees of freedom and $\hat{v}_{m,i}$ is an unbiased estimator for variance of $\hat{\mu}^{lzc}_{m,i}$.

Proof The proof follows from Theorem 1 and Theorem 2 in Nelson's paper [58]. Here, we have m observations of the arm rewards and constructed LZV-CV, we replace their parameter with our arm specific parameters. Use t-distribution for CI, therefore the value of $\mathbb{P}\left(|\hat{\mu}^{lzc}_{m,i} - \mu_i| \geq V_{t,m,1}\sqrt{\hat{v}_{m,i}}\right)$ depends only on the value of $V_{t,m,1}$. Therefore we have

$$\begin{aligned}
\mathbb{P}\left(|\hat{\mu}^{lzc}_{m,i} - \mu_i| \geq V_{t,m,1}\sqrt{\hat{v}_{m,i}}\right) &= 1 - \mathbb{P}\left(|\hat{\mu}^{lzc}_{m,i} - \mu_i| \leq V_{t,m,1}\sqrt{\hat{v}_{m,i}}\right) \\
&= 1 - (1 - \frac{2}{t^2}) \\
&= \frac{2}{t^2}.
\end{aligned}$$

Appendix D

Missing Proofs in UCB-QZVCV

Claim 2 *When we have arms with normally distributed rewards,* $z(X_{t,i}) = -\frac{1}{2} \cdot \frac{-X_{t,i}+\mu_i}{\sigma_i^2}$ *and* $u(X_{t,i}) = \frac{1}{2\sigma_i^2}(X_{t,i}^2 - X_{t,i}\mu_i - \sigma_i^2)$. *In this way, the expected value of the QZV-CV is zero (i.e.* $\mathbb{E}[h(X_{t,i})] = \theta_i = 0$*).*

Proof Consider arms with normally distributed rewards in the MAB problem. In our MAB problem setting, the reward is in one dimension $d_x = 1$, The number of coefficients for this second-order $P(X)$ is $\frac{1}{2} \times 1 \times (1+3) = 2$. Given $f(X_{t,i}) = X_{t,i}$ and $X_{t,i}$ are normally distributed with mean μ_i and variance σ_i^2. The density is

$$p(X_{t,i}) = \frac{1}{\sqrt{2\pi}\sigma_i} exp(-\frac{(X_{t,i}-\mu_i)^2}{2\sigma_i^2}). \tag{D.1}$$

Motivated by equation 5.16, the QZV-CV can be constructed as

$$h(X_{t,i}) = (z(X_{t,i}), u(X_{t,i}))^T, \tag{D.2}$$

$$\begin{aligned} z(X_{t,i}) &:= -\frac{1}{2}\nabla[log(p(X_{t,i}))] \\ &= -\frac{1}{2} \cdot \frac{-X_{t,i}+\mu_i}{\sigma_i^2}, \end{aligned}$$

$$
\begin{aligned}
u(X_{t,i}) &:= X_{t,i} \circ z(X_{t,i}) - \frac{1}{2}\mathbf{1} \\
&= X_{t,i} \cdot \left(-\frac{1}{2} \cdot \frac{-X_{t,i} + \mu_i}{\sigma_i^2} \right) - \frac{1}{2} \\
&= \frac{1}{2\sigma_i^2}(X_{t,i}^2 - X_{t,i}\mu_i - \sigma_i^2)
\end{aligned}
$$

where $\circ$ and $\mathbf{1}$ denote the element-wise product and the unit vector respectively. In this way,

$$
\begin{aligned}
\theta_{i,z} &= \mathbb{E}[z(X_{t,i})] \\
&= \mathbb{E}[-\frac{1}{2} \cdot \frac{-X_{t,i} + \mu_i}{\sigma_i^2}] \\
&= \mathbb{E}[\frac{-X_{t,i} + \mu_i}{\sigma_i^2}] \\
&= \frac{\mathbb{E}[-X_{t,i}] + \mu_i}{\mathbb{E}[\sigma_i^2]} \\
&= \frac{-\mu_i + \mu_i}{\mathbb{E}[\sigma_i^2]} \\
&= 0
\end{aligned}
$$

$$
\begin{aligned}
\theta_{i,u} &= \mathbb{E}[u(X_{t,i})] \\
&= \mathbb{E}[\frac{1}{2\sigma_i^2}(X_{t,i}^2 - X_{t,i} \cdot \mu_i - \sigma_i^2)] \\
&= \frac{1}{2\sigma_i^2}\left(\mathbb{E}[X_{t,i}^2] - \mathbb{E}[X_{t,i} \cdot \mu_i] - \sigma_i^2\right) \\
&= \frac{1}{2\sigma_i^2}\left(var(X_{t,i}) + (\mathbb{E}[X_{t,i}])^2 - \mathbb{E}[X_{t,i} \cdot \mu_i] - \sigma_i^2\right) \\
&= \frac{1}{2\sigma_i^2}\left(\sigma_i^2 + (\mathbb{E}[X_{t,i}])^2 - \mathbb{E}[X_{t,i} \cdot \mu_i] - \sigma_i^2\right) \\
&= \frac{1}{2\sigma_i^2}\left(\sigma_i^2 + \mu_i^2 - \mu_i^2 - \sigma_i^2\right) \\
&= 0
\end{aligned}
$$

Therefore, we have $\mathbb{E}[h(X_{t,i})] = \theta_i = (\theta_{i,z}, \theta_{i,u})^T = (0,0)^T$ as required.

Lemma 5 *When the reward and constructed QZV-CV of each arm have a multivari-*

ate normal distribution. After observing m samples of reward and constructing m QZV-CV samples from arm i, define $\hat{v}_{m,i} = \frac{Z_{m,i}\hat{\sigma}^2_{qzc,i}(m)}{m}$*, where*

$$Z_{m,i} = \left(1 + \frac{(\hat{\theta}_{m,i} - \theta_i)^T S^{-1}_{H_i H_i}(\hat{\theta}_{m,i} - \theta_i)}{1 - 1/m}\right),$$

$$\hat{\sigma}^2_{qzc,i}(m) = \frac{1}{m-3}\sum_{r=1}^{m}(\tilde{f}(X_{r,i}) - \hat{\mu}^{qzc}_{m,i})^2,$$

$\hat{\sigma}^2_{qzc,i}(m)$ *is the sample variance of the new observations constructed with QZV-CV. Then* $\hat{v}_{m,i}$ *is an unbiased variance estimator of* $\hat{\mu}^{qzc}_{m,i}$ $\left(i.e. \mathbb{E}[\hat{v}_{m,i}] = var(\hat{\mu}^{qzc}_{m,i})\right)$.

Proof In the UCB-QZVCV algorithm, suppose we observe m new samples for the arm $i \in [K]$:

$$\tilde{f}(X_t) = f(X_t) - \hat{\alpha}^{*T} h(X_t), t \in \{1, 2, ..., m\}, \tag{D.3}$$

where the subscript for arm (indexed by i) is dropped for simplicity. Under the multivariate normality assumption, $\tilde{f}(X_t)$ can be written as follows:

$$\tilde{f}(X_t) = \mu - \alpha^{*T} h(X_t) + \epsilon_t, t \in \{1, 2, ..., m\}, \tag{D.4}$$

where $\epsilon_1, \epsilon_2, ..., \epsilon_m$ are IID normally distributed random variables with mean 0 and variance σ^2. Thus, we have

$$\tilde{\boldsymbol{f}}(\boldsymbol{X}) = \boldsymbol{Y}\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \tag{D.5}$$

where

$$\tilde{\boldsymbol{f}}(\boldsymbol{X}) = \begin{pmatrix} \tilde{f}(X_1) \\ \vdots \\ \tilde{f}(X_m) \end{pmatrix}, \boldsymbol{Y} = \begin{pmatrix} 1 & -z(X_1) & -u(X_1) \\ \vdots & \vdots & \vdots \\ 1 & -z(X_m) & -u(X_m) \end{pmatrix}, \boldsymbol{\alpha} = \begin{pmatrix} \mu \\ c \\ d \end{pmatrix}, \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_m \end{pmatrix}. \tag{D.6}$$

Let the least square estimator of μ be $\hat{\mu}^{qzc}$, c be $\hat{c}$ and d be $\hat{d}$. With the finite sample

properties of the least square estimator (Fact 1), we know that the variance of mean estimator for arm i in the UCB-QZVCV algorithm is

$$var(\hat{\mu}_{m,i}^{qzc}) = \sigma_{qzc,i}^2 (\boldsymbol{Y}^T \boldsymbol{Y})_{11}^{-1}, \tag{D.7}$$

where $var(\tilde{f}(X_i)) = \sigma_{qzc,i}^2$ is the sample variance of the new observations and $(\boldsymbol{Y}^T \boldsymbol{Y})_{11}^{-1}$ is the upper left most element of matrix $(\boldsymbol{Y}^T \boldsymbol{Y})^{-1}$[72].

With [7] and [58], we know that after m observations, the unbiased estimator of $var(\hat{\mu}_{m,i}^{qzc})$ is given by

$$\hat{v}_{m,i} = \frac{Z_{m,i} \hat{\sigma}_{qzc,i}^2(m)}{m}, \tag{D.8}$$

where $Z_{m,i} = \left(1 + \frac{(\hat{\theta}_{m,i} - \theta_i)^T S_{H_i H_i}^{-1} (\hat{\theta}_{m,i} - \theta_i)}{1 - 1/m}\right)$ and $\hat{\sigma}_{qzc,i}^2(m) = \frac{1}{m-3} \sum_{r=1}^{m} (\tilde{f}(X_{r,i}) - \hat{\mu}_{m,i}^{qzc})^2$. With Fact 2, $\hat{\sigma}_{qzc,i}^2(m)$ is the sample variance of the new observations constructed with QZV-CV, and it is the unbiased estimator of $\sigma_{qzc,i}^2$. By the above derivations, $\hat{v}_{m,i}$ is an unbiased variance estimator of $\hat{\mu}_{m,i}^{qzc}$ $\left(i.e. \mathbb{E}[\hat{v}_{m,i}] = var(\hat{\mu}_{m,i}^{qzc})\right)$.

Lemma 6 *Let m be the number of rewards and constructed QZV-CV samples from arm i in time step t. Then*

$$\mathbb{P}\left(|\hat{\mu}_{m,i}^{qzv} - \mu_i| \geq V_{t,m,2} \sqrt{\hat{v}_{m,i}}\right) = \frac{2}{t^2}, \tag{D.9}$$

where $V_{t,m,2}$ denotes $100(1 - 1/t^2)^{th}$ percentile value of the t-distribution with $m-3$ degrees of freedom and $\hat{v}_{m,i}$ is an unbiased estimator for variance of $\hat{\mu}_{m,i}^{qzv}$.

Proof The proof follows from Theorem 1 and Theorem 2 in Nelson's paper [58]. Here, we have m observations of the arm rewards and constructed QZV-CV, we replace their parameters with our arm specific parameters. Use t-distribution for CI, therefore the value of $\mathbb{P}\left(|\hat{\mu}_{m,i}^{qzc} - \mu_i| \geq V_{t,m,2} \sqrt{\hat{v}_{m,i}}\right)$ depends only on the value of

$V_{t,m,2}$. Therefore we have

$$\begin{aligned}\mathbb{P}\left(|\hat{\mu}_{m,i}^{qzc} - \mu_i| \geq V_{t,m,2}\sqrt{\hat{v}_{m,i}}\right) &= 1 - \mathbb{P}\left(|\hat{\mu}_{m,i}^{qzc} - \mu_i| \leq V_{t,m,2}\sqrt{\hat{v}_{m,i}}\right) \\ &= 1 - (1 - \frac{2}{t^2}) \\ &= \frac{2}{t^2}.\end{aligned}$$