A Survey on Contextual Multi-armed Bandits

Li Zhou LIZHOU@CS.CMU.EDU
Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue Pittsburgh, PA 15213, US

Editor:

Contents

1 Introduction **2**

2 Unbiased Reward Estimator **5**

3 Reduce to K-Armed Bandits **5**

4 Stochastic Contextual Bandits **6**
4.1 Stochastic Contextual Bandits with Linear Realizability Assumption 6
4.1.1 LinUCB/SupLinUCB 6
4.1.2 LinREL/SupLinREL 9
4.1.3 CofineUCB 11
4.1.4 Thompson Sampling with Linear Payoffs 11
4.1.5 SpectralUCB 14
4.2 Kernelized Stochastic Contextual Bandits 14
4.2.1 GP-UCB/CGP-UCB 16
4.2.2 KernelUCB 18
4.3 Stochastic Contextual Bandits with Arbitrary Set of Policies 19
4.3.1 Epoch-Greedy 19
4.3.2 RandomizedUCB 23
4.3.3 ILOVETOCONBANDITS 28

5 Adversarial Contextual Bandits **30**
5.1 EXP4 30
5.2 EXP4.P 34
5.3 Infinite Many Experts 39

6 Conclusion **41**

	Actions don’t change state of the world	Actions change state of the world
Learn model of outcomes	Multi-armed bandits	Reinforcement Learning
Given model of stochastic outcomes	Decision theory	Markov Decision Process

Table 1: Four scenarios when reasoning under uncertainty.[1]

1. Introduction

In a decision making process, agents make decisions based on observations of the world. Table 1 describes four scenarios when making decisions under uncertainty. In a multi-armed bandits problem, the model of outcomes is unknown, and the outcomes can be stochastic or adversarial; Besides, actions taken won’t change the state of the world.

In this survey we focus on multi-armed bandits. In this problem the agent needs to make a sequence of decisions in time $1, 2, ..., T$. At each time t the agent is given a set of K arms, and it has to decide which arm to pull. After pulling an arm, it receives a reward of that arm, and the rewards of other arms are unknown. In a stochastic setting the reward of an arm is sampled from some unknown distribution, and in an adversarial setting the reward of an arm is chosen by an adversary and is not necessarily sampled from any distribution. Particularly, in this survey we are interested in the situation where we observe side information at each time t. We call this side information the context. The arm that has the highest expected reward may be different given different contexts. This variant of multi-armed bandits is called contextual bandits.

Usually in a contextual bandits problem there is a set of policies, and each policy maps a context to an arm. There can be infinite number of policies, especially when reducing bandits to classification problems. We define the regret of the agent as the gap between the highest expected cumulative reward any policy can achieve and the cumulative reward the agent actually get. The goal of the agent is to minimize the regret. Contextual bandits can naturally model many problems. For example, in a news personalization system, we can treat each news articles as an arm, and the features of both articles and users as contexts. The agent then picks articles for each user to maximize click-through rate or dwell time.

There are a lot of bandits algorithms, and it is always important to know what they are competing with. For example, in K-armed bandits, the agents are competing with the arm that has the highest expected reward; and in contextual bandits with expert advice, the agents are competing with the expert that has the highest expected reward; and when we reduce contextual bandits to classification/regression problems, the agents are competing with the best policy in a pre-defined policy set.

[1] Table from CMU Graduate AI course slides. http://www.cs.cmu.edu/~15780/lec/10-Prob-start-mdp.pdf

As a overview, we summarize all the algorithms we will talk about in Table 2. In this table, C is the number of distinct contexts, N is the number of policies, K is the number of arms, and d is the dimension of contexts. Note that the second last column shows if the algorithm requires the knowledge of T, and it doesn't necessary mean that the algorithm requires the knowledge of T to run, but means that to achieve the proposed regret the knowledge of T is required.

Table 2: A comparison between all the contextual bandits algorithm we will talk about

Algorithm	Regret	With Hight Probability	Can Have Infinite Policies	Need to know T	adversarial reward
Reduce to MAB	$O\left(\sqrt{TCK\ln K}\right)$ or $O\left(\sqrt{TN\ln N}\right)$	no	no	yes	yes
EXP4	$O\left(\sqrt{TK\ln N}\right)$	no	no	yes	yes
EXP4.P	$O\left(\sqrt{TK\ln(N/\delta)}\right)$	yes	no	yes	yes
LinUCB	$O\left(d\sqrt{T\ln((1+T)/\delta)}\right)$	yes	yes	yes	no
SupLinUCB	$O\left(\sqrt{Td\ln^3(KT\ln T/\delta)}\right)$	yes	yes	yes	no
SupLinREL	$O\left(\sqrt{Td}(1+\ln(2KT\ln T/\delta))^{3/2}\right)$	yes	yes	yes	no
GP-UCB	$\tilde{O}\left(\sqrt{T}\left(B\sqrt{\gamma_T}+\gamma_T\right)\right)$	yes	yes	yes	no
KernelUCB	$\tilde{O}(\sqrt{B\tilde{d}T})$	yes	yes	yes	no
Epoch-Greedy	$O\left((K\ln(N/\delta))^{1/3}T^{2/3}\right)$	yes	yes	no	no
Randomized UCB	$O\left(\sqrt{TK\ln(N/\delta)}\right)$	yes	yes	no	no
ILOVETOCONBANDITS	$O\left(\sqrt{TK\ln(N/\delta)}\right)$	yes	yes	no	no
Thompson Sampling with Linear Regression	$O\left(\frac{d^2}{\epsilon}\sqrt{T^{1+\epsilon}}(\ln(Td)\ln\frac{1}{\delta})\right)$	yes	yes	no	no

2. Unbiased Reward Estimator

One challenge of bandits problems is that we only observe partial feedback. Suppose at time t the algorithm randomly selects an arm a_t based on a probability vector p_t. Denote the true reward vector by $r_t \in [0,1]^K$ and the reward vector we observed by $r'_t \in [0,1]^K$, then all the elements in r'_t are zero except r'_{t,a_t} which is equal to r_{t,a_t}. Then r'_t is certainly not a unbiased estimator of r_t because $\mathrm{E}(r'_{t,a_t}) = p_{a_t} \cdot r_{t,a_t} \neq r_{t,a_t}$. A common trick to this is to use $\hat{r}_{t,a_t} = r'_{t,a_t}/p_{a_t}$ instead of r'_{a_t}. In this way we get a unbiased estimator of the true reward vector r_t: for any arm a

$$\begin{aligned}\mathrm{E}(\hat{r}_{t,a}) &= p_a \cdot r_{t,a}/p_a + (1-p_a) * 0 \\ &= r_{t,a}\end{aligned}$$

The expectation is with respect to the random choice of arms at time t. This trick is used by many algorithms described later.

3. Reduce to K-Armed Bandits

If it is possible to enumerate all the contexts, then one naive way is to apply a K-armed bandits algorithm to each context. However, in this way we ignore all the relationships between contexts since we treat them independently.

Suppose there are C distinct contexts in the context set $\mathcal{X}$, and the context at time t is $x_t \in \{1,2,...,C\}$. Also assume there are K arms in the arm set $\mathcal{A}$ and the arm selected at time t is $a_t \in \{1,2,...,K\}$. Define the policy set to be all the possible mappings from contexts to arms as $\Pi = \{f : \mathcal{X} \to \mathcal{A}\}$, then the regret of the agent is defined as:

$$R_T = \sup_{f\in\Pi} \mathrm{E}\left[\sum_{t=1}^{T}(r_{t,f(x_t)} - r_{t,a_t})\right] \tag{1}$$

Theorem 3.1 *Apply EXP3 (Auer et al., 2002b), a non-contextual multi-armed bandits algorithm, on each context, then the regret is*

$$R_T \leq 2.63\sqrt{TCK\ln K}$$

Proof Define $n_i = \sum_{t=1}^{T} \mathbb{I}(x_t = i)$, then $\sum_{i=1}^{C} n_i = T$. We know that the regret bound of EXP3 algorithm is $2.63\sqrt{TK\ln K}$, so

$$\begin{aligned}R_T &= \sup_{f\in\Pi} \mathrm{E}\left[\sum_{t=1}^{T}(r_{t,f(x_t)} - r_{t,a_t})\right] \\ &= \sum_{i=1}^{C} \sup_{f\in\Pi} \mathrm{E}\left[\sum_{t=1}^{T} \mathbb{I}(x_t = i)(r_{t,f(x_t)} - r_{t,a_t})\right] \\ &\leq \sum_{i=1}^{C} 2.63\sqrt{n_i K \ln K} \\ &\leq 2.63\sqrt{TCK\ln K} \quad \text{(Cauchy-Schwarz inequality)}\end{aligned}$$

■

One problem with this method is that it assumes the contexts can be enumerated, which is not true when contexts are continuous. Also this algorithm treats each context independently, so learning one of them does not help learning the other ones.

If there exists is a set of pre-defined policies and we want to compete with the best one, then another way to reduce to K-armed bandits is to treat each policy as an arm and then apply EXP3 algorithm. The regret is still defined as Equation (1), but Π is now a pre-defined policy set instead of all possible mappings from contexts to arms. Let N be the number of polices in the policy set, then by applying EXP3 algorithm we get the regret bound $O(\sqrt{TN \ln N})$. This algorithm works if we have small number of policies and large number of arms; however, if we have a huge number of policies, then this regret bound is weak.

4. Stochastic Contextual Bandits

Stochastic contextual bandits algorithms assume that the reward of each arm follows an unknown probability distribution. Some algorithms further assume such distribution is sub-Gaussian with unknown parameters. In this section, we first talk about stochastic contextual bandits algorithms with linear realizability assumption; In this case, the expectation of the reward of each arm is linear with respect to the arm's features. Then we talk about algorithms that work for arbitrary set of policies without such assumption.

4.1 Stochastic Contextual Bandits with Linear Realizability Assumption

4.1.1 LinUCB/SupLinUCB

LinUCB (Li et al., 2010; Chu et al., 2011) extends UCB algorithm to contextual cases. Suppose each arm is associated with a feature vector $x_{t,a} \in \mathrm{R}^d$. In news recommendation, $x_{t,a}$ could be user-article pairwise feature vectors. LinUCB assumes the expected reward of an arm a is linear with respect to its feature vector $x_{t,a} \in \mathrm{R}^d$:

$$\mathrm{E}[r_{t,a}|x_{t,a}] = x_{t,a}^\top \theta^*$$

where θ^* is the true coefficient vector. The noise $\epsilon_{t,a}$ is assumed to be R-sub-Gaussian for any t. Without loss of generality, we assume $||\theta^*|| \leq S$ and $||x_{t,a}|| \leq L$, where $|| \cdot ||$ denotes the ℓ_2-norm. We also assume the reward $r_{t,a} \leq 1$. Denote the best arm at time t by $a_t^* = \arg\max_a x_{t,a}^\top \theta^*$, and the arm selected by the algorithm at time t by a_t, then the T-trial regret of LinUCB is defined as

$$\begin{aligned} R_T &= \mathrm{E}\left[\sum_{t=1}^{T} r_{t,a_t^*} - \sum_{t=1}^{T} r_{t,a_t}\right] \\ &= \sum_{t=1}^{T} x_{t,a_t^*}^\top \theta^* - \sum_{t=1}^{T} x_{t,a_t}^\top \theta^* \end{aligned}$$

Let $D_t \in \mathrm{R}^{t \times d}$ and $c_t \in \mathrm{R}^t$ be the historical data up to time t, where the i^{th} row of D_t represents the feature vector of the arm pulled at time i, and the i^{th} row of c_t represents the corresponding reward. If samples $(x_{t,a}, r_{t,a_t})$ are independent, then we can get a closed-form estimator of θ^* by ridge regression:

$$\hat{\theta}_t = (D_t^\top D_t + \lambda \mathrm{I}_d)^{-1} D_t^\top c_t$$

The accuracy of the estimator, of course, depends on the amount of data. Chu et al. (2011) derived a upper confidence bound for the prediction $x_{t,a}^\top \hat{\theta}_t$:

Theorem 4.1 *Suppose the rewards $r_{t,a}$ are independent random variables with means $\mathrm{E}[r_{t,a}] = x_{t,a}^\top \theta^*$, let $\epsilon = \sqrt{\frac{1}{2} \ln \frac{2TK}{\delta}}$ and $A_t = D_t^\top D_t + \mathrm{I}_d$ then with probability $1 - \delta/T$, we have*

$$|x_{t,a}^\top \hat{\theta}_t - x_{t,a}^\top \theta^*| \leq (\epsilon + 1)\sqrt{x_{t,a}^\top A_t^{-1} x_{t,a}}$$

LinUCB always selects the arm with the highest upper confidence bound. The algorithm is described in Algorithm 1.

Algorithm 1 LinUCB

```
Require: $\alpha > 0, \lambda > 0$
  $A = \lambda \mathrm{I}_d$
  $b = 0_d$
  for t=1, 2, ..., T do
      $\theta_t = A^{-1} b$
      Observe features of all $K$ arms $a \in \mathcal{A}_t : x_{t,a} \in \mathrm{R}^d$
      for a=1, 2, ... K do
          $s_{t,a} = x_{t,a}^\top \theta_t + \alpha \sqrt{x_{t,a}^\top A_t^{-1} x_{t,a}}$
      end for
      Choose arm $a_t = \arg\max_a s_{t,a}$, break ties arbitrarily
      Receive reward $r_t \in [0, 1]$
      $A = A + x_{t,a} x_{t,a}^\top$
      $b = b + x_{t,a} r_t$
  end for
```

However, LinUCB algorithm use samples from previous rounds to estimate θ^* and then pick a sample for current round. So the samples are not independent. In Abbasi-Yadkori et al. (2011) it was shown through martingale techniques that concentration results for the predictors can be obtained directly without requiring the assumption that they are built as linear combinations of independent random variables.

Theorem 4.2 (Abbasi-Yadkori et al. (2011)) *Let the noise term $\epsilon_{t,a}$ be R-sub-Gaussian where $R \geq 0$ is a fixed constant. With probability at least $1 - \delta$, $\forall t \geq 1$,*

$$\|\hat{\theta}_t - \theta^*\|_{A_t} \leq R \sqrt{2 \log \left(\frac{|A_t|^{1/2}}{\lambda^{1/2} \delta} \right)} + \lambda^{1/2} S.$$

We can now choose appropriate values of α_t for LinUCB as the right side of the inequality in Theorem 4.2. Note that here α depends on t, we denote so it is a little different than the original LinUCB algorithm (Algorithm 1) which has independent assumption.

Theorem 4.3 *Let $\lambda \geq max(1, \mathrm{L}^2)$. The cumulative regret of LinUCB is with probability at least $1-\delta$ bounded as:*

$$\begin{aligned} R_T \leq & \sqrt{Td\log(1+TL^2/(d\lambda))}\times \\ & \times \left(R\sqrt{d\log(1+TL^2/(\lambda d))+2\log(1/\delta)}+\lambda^{1/2}S\right) \end{aligned}$$

To proof Theorem 4.3, We first state two technical lemmas from Abbasi-Yadkori et al. (2011):

Lemma 4.4 (Abbasi-Yadkori et al. (2011)) *We have the following bound:*

$$\sum_{t=1}^{T} \|x_t\|^2_{A_t^{-1}} \leq 2\log\frac{|A_t|}{\lambda}.$$

Lemma 4.5 (Abbasi-Yadkori et al. (2011)) *The determinant $|A_t|$ can be bounded as:*

$$|A_t| \leq (\lambda + tL^2/d)^d.$$

We can now simplify α_t as

$$\begin{aligned} \alpha_t &\leq R\sqrt{2\log\left(|A_t|^{1/2}\lambda^{-1/2}\delta^{-1}\right)}+\lambda^{1/2}S \\ &\leq R\sqrt{d\log(1+TL^2/(\lambda d))+2\log(1/\delta)}+\lambda^{1/2}S \end{aligned}$$

where $d \geq 1$ and $\lambda \geq \max(1, L^2)$ to have $\lambda^{1/d} \geq \lambda$.

Proof [Theorem 4.3] Let $\bar{r}_t$ denote the instantaneous regret at time t. With probability at least $1-\delta$, for all t:

$$\begin{aligned} \bar{r}_t &= x_{t,*}^\top\theta^* - x_t^\top\theta^* \\ &\leq x_t^\top\hat{\theta}_t + \alpha_t\|x_t\|_{A_t^{-1}} - x_t^T\theta^* && (2) \\ &\leq x_t^\top\hat{\theta}_t + \alpha_t\|x_t\|_{A_t^{-1}} - x_t^\top\hat{\theta}_t + \alpha_t\|x_t\|_{A_t^{-1}} && (3) \\ &= 2\alpha_t\|x_t\|_{A_t^{-1}} \end{aligned}$$

The inequality (2) is by the algorithm design and reflects the optimistic principle of LinUCB. Specifically, $x_*^\top\hat{\theta}_t + \alpha_t\|x_*\|_{A_t^{-1}} \leq x_t^\top\hat{\theta}_t + \alpha_t\|x_t\|_{A_t^{-1}}$, from which:

$$x_*^\top\theta^* \leq x_*^\top\hat{\theta}_t + \alpha_t\|x_*\|_{A_t^{-1}} \leq x_t^\top\hat{\theta}_t + \alpha_t\|x_t\|_{A_t^{-1}}$$

In (3), we applied Theorem 4.2 to get:

$$x_t^\top\hat{\theta}_t \leq x_{t,*}^\top\theta^* + \alpha_t\|x_t\|_{A_t^{-1}}$$

Finally by Lemmas 4.4 and 4.5:

$$\begin{aligned}
R_T = \sum_{t=1}^{T} \bar{r}_t &\leq \sqrt{T \sum_{t=1}^{T} \bar{r}_t^2} \\
&\leq 2\alpha_T \sqrt{T \sum_{t=1}^{T} \|x_t\|^2_{A_t^{-1}}} \\
&\leq 2\alpha_T \sqrt{T \log \frac{|A_t|}{\lambda}} \\
&\leq 2\alpha_T \sqrt{T(d \log(\lambda + TL^2/d) - \log \lambda)} \\
&\leq 2\alpha_T \sqrt{Td \log(1 + TL^2/(d\lambda))}
\end{aligned}$$

Above we used that $\alpha_t \leq \alpha_T$ because α_t is not decreasing t. Next we used that $\lambda \geq \max(1, L^2)$ to have $\lambda^{1/d} \geq \lambda$. By plugging α_t, we get:

$$\begin{aligned}
R_T &\leq \sqrt{Td \log(1 + TL^2/(d\lambda))} \times \\
&\times \left(R\sqrt{d \log(1 + TL^2/(\lambda d)) + 2 \log(1/\delta)} + \lambda^{1/2} S \right) \\
&= O(d\sqrt{T \log((1+T)/\delta)})
\end{aligned}$$

■

Inspired by Auer (2003), Chu et al. (2011) proposed SupLinUCB algorithm, which is a variant of LinUCB. It is mainly used for theoretical analysis, but not a practical algorithm. SupLinUCB constructs S sets to store previously pulled arms and rewards. The algorithm are designed so that within the same set the sequence of feature vectors are fixed and the rewards are independent. As a results, an arm's predicted reward in the current round is a linear combination of rewards that are independent random variables, and so Azuma's inequality can be used to get the regret bound of the algorithm.

Chu et al. (2011) proved that with probability at least $1 - \delta$, the regret bound of SupLinUCB is $O\left(\sqrt{Td \ln^3(KT \ln(T)/\delta)}\right)$.

4.1.2 LinREL/SupLinREL

The problem setting of LinREL (Auer, 2003) is the same as LinUCB, so we use the same notations here. LinREL and LinUCB both assume that for each arm there is an associated feature vector $x_{t,a}$ and the expected reward of arm a is linear with respect to its feature vector: $\mathrm{E}[r_{t,a}|x_{t,a}] = x_{t,a}^\top \theta^*$, where θ^* is the true coefficient vector. However, these two algorithms take two different forms of regularization. LinUCB takes a ℓ_2 regularization term similar to ridge regression; that is, it adds a diagonal matrix $\lambda \mathrm{I}_d$ to matrix $D_t^\top D_t$. LinREL, on the other hand, do regularization by setting $D_t^\top D_t$ matrix's small eigenvalues to zero. LinREL algorithm is described in Algorithm 2. We have the following theorem to show that Equation (5) is the upper confidence bound of the true reward of arm a at time t. Note that the following theorem assumes the rewards observed at each time t are independent random variables. However, similar to LinUCB, this assumption is not true. We will deal with this problem later.

Algorithm 2 LinREL

```
Require: $\delta \in [0,1]$, number of trials $T$.
  Let $D_t \in \mathrm{R}^{t\times d}$ and $c_t \in \mathrm{R}^t$ be the matrix and vector to store previously pulled arm feature vectors and rewards.
  for t=1, 2, ..., T do
      Calculate eigendecomposition
          $D_t^\top D_t = U_t^\top \mathrm{diag}(\lambda_t^1, \lambda_t^2, ..., \lambda_t^d) U_t$
      where $\lambda_t^1, ..., \lambda_t^k \geq 1$, $\lambda_t^{k+1}, ..., \lambda_t^d < 1$, and $U_t^\top \cdot U_t = \mathrm{I}_d$
      Observe features of all $K$ arms $a \in \mathcal{A}_t : x_{t,a} \in \mathrm{R}^d$
      for a=1, 2, ... K do
          $\tilde{x}_{t,a} = (\tilde{x}_{t,a}^1, ..., \tilde{x}_{t,a}^d) = U_t x_{t,a}$
          $\tilde{u}_{t,a} = (\tilde{x}_{t,a}^1, ..., \tilde{x}_{t,a}^k, 0, ..., 0)^\top$
          $\tilde{v}_{t,a} = (0, ..., 0, \tilde{x}_{t,a}^{k+1}, ..., \tilde{x}_{t,a}^d)^\top$
          $w_{t,a} = \left(\tilde{u}_{t,a}^\top \cdot \mathrm{diag}\left(\frac{1}{\lambda_t^1}, ..., \frac{1}{\lambda_t^k}, 0, ..., 0\right) \cdot U_t \cdot D_t^\top\right)^\top$    (4)
          $s_{t,a} = w_{t,a}^\top c_t + ||w_{t,a}|| \left(\sqrt{\ln(2TK/\delta)}\right) + ||\tilde{v}_{t,a}||$    (5)
      end for
      Choose arm $a_t = \arg\max_a s_{t,a}$, break ties arbitrarily
      Receive reward $r_t \in [0,1]$, append $x_{t,a}$ and $r_{t,a}$ to $D_t$ and $c_t$.
  end for
```

Theorem 4.6 *Suppose the rewards $r_{\tau,a}, \tau \in 1, ..., t-1$ are independent random variables with mean $\mathrm{E}[x_{\tau,a}] = x_{\tau,a}^\top \theta^*$. Then at time t, with probability $1 - \delta/T$ all arm $a \in \mathcal{A}_t$ satisfy*

$$|w_{t,a}^\top c_t - x_{t,a}^\top \theta^*| \leq ||w_{t,a}|| \left(\sqrt{2\ln(2TK/\delta)}\right) + ||\tilde{v}_{t,a}||$$

Suppose $D_t^\top D_t$ is invertible, then we can estimate the model parameter $\hat{\theta} = (D_t^\top D_t)^{-1} D^\top c_t$. Given a feature vector $x_{t,a}$, the predicted reward is

$$r_{t,a} = x_{t_a}^\top \hat{\theta} = (x_{t,a}^\top (D_t^\top D_t)^{-1} D^\top) c_t$$

So we can view $r_{t,a}$ as a linear combination of previous rewards. In Equation (4), $w_{t,a}$ is essentially the weights for each previous reward (after regularization). We use $w_{t,a}^\tau$ to denote the weight of reward $r_{\tau,a}$.

Proof [Theorem 4.6] Let $z_\tau = r_{t,a} \cdot w_{t,a}^\tau$, then $|z_\tau| \leq w_{t,a}^\tau$, and

$$w_{t,a}^\top c_t = \sum_{\tau=1}^{t-1} z_\tau = \sum_{\tau=1}^{t-1} r_{t,a} \cdot w_{t,a}^\tau$$
$$\sum_{\tau=1}^{t-1} \mathrm{E}[z_\tau | z_1, ..., z_{\tau-1}] = \sum_{\tau=1}^{t-1} \mathrm{E}[z_\tau] = \sum_{\tau=1}^{t-1} x_{\tau,a}^\top \theta^* \cdot w_{t,a}^\tau$$

Apply Azuma's inequality we have

$$\begin{aligned}
&\mathrm{P}\left(w_{t,a}^\top c_t - \sum_{\tau=1}^{t-1} x_{\tau,a}^\top \theta^* \cdot w_{t,a}^\tau \geq ||w_{t,a}|| \left(\sqrt{2\ln(2TK/\delta)}\right)\right) \\
&= \mathrm{P}\left(w_{t,a}^\top c_t - w_{t,a}^\top D_t \theta^* \geq ||w_{t,a}|| \left(\sqrt{2\ln(2TK/\delta)}\right)\right) \\
&\leq \frac{\delta}{TK}
\end{aligned}$$

Now what we really need is the inequality between $w_{t,a}^\top c_t$ and $x_{t,a}^\top \theta^*$. Note that

$$\begin{aligned}
x_{t,a} &= U_t^\top \tilde{x}_{t,a} \\
&= U_t^\top \tilde{u}_{t,a} + U_t^\top \tilde{v}_{t,a} \\
&= D_t^\top D_t (D^\top D_t)^{-1} D_t^\top w_{t,a} + U_t^\top \tilde{v}_{t,a} \\
&= D_t^\top w_{t,a} + U_t^\top \tilde{v}_{t,a}
\end{aligned}$$

Assuming $||\theta^*|| \leq 1$, we have

$$\mathrm{P}\left(w_{t,a}^\top c_t - x_{t,a}^\top \theta^* \geq ||w_{t,a}|| \left(\sqrt{2\ln(2TK/\delta)}\right) + ||\tilde{v}_{t,a}||\right) \leq \frac{\delta}{TK}$$

Take the union bound over all arms, we prove the theorem. ■

The above proof uses the assumption that all the rewards observed are independent random variables. However in LinREL, the actions taken in previous rounds will influence the estimated $\hat{\theta}$, and thus influence the decision in current round. To deal with this problem, Auer (2003) proposed SupLinREL algorithm. SupLinREL construct S sets $\Psi_t^1, ..., \Psi_t^S$, each set Ψ_t^s contains arm pulled at stage s. It is designed so that the rewards of arms inside one stage is independent, and within one stage they apply LinREL algorithm. They proved that the regret bound of SupLinREL is $O\left(\sqrt{Td}(1 + \ln(2KT\ln T))^{3/2}\right)$.

4.1.3 CofineUCB

4.1.4 Thompson Sampling with Linear Payoffs

Thompson sampling is a heuristic to balance exploration and exploitation, and it achieves good empirical results on display ads and news recommendation (Chapelle and Li, 2011). Thompson sampling can be applied to both contextual and non-contextual multi-armed bandits problems. For example Agrawal and Goyal (2013b) provides a $O(\sqrt{NT \ln T})$ regret bound for non-contextual case. Here we focus on the contextual case.

Let $\mathcal{D}$ be the set of past observations (x_t, a_t, r_t), where x_t is the context, a_t is the arm pulled, and r_t is the reward of that arm. Thompson sampling assumes a parametric likelihood function $P(r|a, x, \theta)$ for the reward, where θ is the model parameter. We denote the true parameters by θ^*. Ideally, we would choose an arm that maximize the expected reward $\max_a \mathrm{E}(r|a, x, \theta^*)$, but of course we don't know the true parameters. Instead Thompson sampling apply a prior believe $P(\theta)$ on parameter θ, and then based on the data observed, it update the posterior distribution of θ by $P(\theta|\mathcal{D}) \propto P(\theta) \prod_{t=1}^{T} P(r_t|x_t, a_t, \theta)$. Now if we just want to maximize the immediate reward, then we would choose an arm that maximize $\mathrm{E}(r|a, x) = \int \mathrm{E}(a, x, \theta) P(\theta|\mathcal{D}) d\theta$, but in an exploration/exploitation setting, we want to choose an arm according to its probability of being optimal. So Thompson sampling randomly selects an action a according to

$$\int \mathbb{I}\left[E(r|a, \theta) = \max_{a'} E(r|a', \theta)\right] P(\theta|\mathcal{D}) d\theta$$

In the actual algorithm, we don't need to calculate the integral, it suffices to draw a random parameter θ from posterior distribution and then select the arm with highest reward under that θ. The general framework of Thompson sampling is described in Algorithm 3.

Algorithm 3 General Framework of Thompson Sampling

```
Define $\mathcal{D} = \{\}$
for $t = 1, ..., T$ do
    Receive context $x_t$
    Draw $\theta_t$ from posterior distribution $P(\theta|\mathcal{D})$
    Select arm $a_t = \arg\max_a \mathrm{E}(r|x_t, a, \theta_t)$
    Receive reward $r_t$
    $\mathcal{D} = \mathcal{D} \cup \{x_t, a_t, r_t\}$
end for
```

According to the prior we choose or the likelihood function we use, we can have different variants of Thompson sampling. In the following section we introduce two of them.

Agrawal and Goyal (2013a) proposed a Thompson sampling algorithm with linear payoffs. Suppose there are a total of K arms, each arm a is associated with a d-dimensional feature vector $x_{t,a}$ at time t. Note that $x_{t,a} \neq x_{t',a}$. There is no assumption on the distribution of x, so the context can be chosen by an adversary. A linear predictor is defined by a d-dimensional parameter $\mu \in \mathrm{R}^d$, and predicts the mean reward of arm a by $\mu \cdot x_{t,a}$. Agrawal and Goyal (2013a) assumes an unknown underlying parameter $\mu^* \in \mathrm{R}^d$ such that the expected reward for arm a at time t is $\bar{r}_{t,a} = \mu^* \cdot x_{t,a}$. The real reward $r_{t,a}$ of arm a at time t is generated from an unknown distribution with mean $\bar{r}_{t,a}$. At each time $t \in \{1, ..., T\}$ the algorithm chooses an arm a_t and receives reward r_t. Let a^* be the optimal arm at time t:

$$a_t^* = \arg\max_a \bar{r}_{t,a}$$

and $\Delta_{t,a}$ be the difference of the expected reward between the optimal arm and arm a:

$$\Delta_{t,a} = \bar{r}_{t,a^*} - \bar{r}_{t,a}$$

Then the regret of the algorithm is defined as:

$$R_T = \sum_{t=1}^{T} \Delta_{t,a_t}$$

In the paper they assume $\delta_{t,a} = r_{t,a} - \bar{r}_{t,a}$ is conditionally R-sub-Gaussian, which means for a constant $R \geq 0$, $r_{t,a} \in [\bar{r}_{t,a} - R, \bar{r}_{r,t} + R]$. There are many likelihood distributions that satisfy this R-sub-Gaussian condition. But to make the algorithm simple, they use Gaussian likelihood and Gaussian prior. The likelihood of reward $\bar{r}_{t,a}$ given the context $x_{t,a}$ is given by the pdf of Gaussian distribution $\mathcal{N}(x_{t,a}^{\top}\mu^*, v^2)$. v is defined as $v = R\sqrt{\frac{24}{\epsilon} d \ln(\frac{t}{\delta})}$, where $\epsilon \in (0, 1)$ is the algorithm parameter and δ controls the high probability regret bound. Similar to the closed-form of linear regression, we define

$$\begin{aligned}
B_t &= I_d + \sum_{\tau=1}^{t-1} x_{\tau,a} x_{\tau,a}^{\top} \\
\hat{\mu}_t &= B_t^{-1} \left(\sum_{\tau=1}^{t-1} x_{\tau,a} r_{\tau,a} \right)
\end{aligned}$$

Then we have the following theorem:

Theorem 4.7 *if the prior of μ^* at time t is defined as $\mathcal{N}(\hat{\mu}_t, v^2 B_t^{-1})$, then the posterior of μ^* is $\mathcal{N}(\hat{\mu}_{t+1}, v^2 B_{t+1}^{-1})$.*

Proof

$$\begin{aligned}
P(\mu | r_{t,a}) &\propto P(r_{t,a}|\mu) P(\mu) \\
&\propto \exp\left(-\frac{1}{2v^2} ((r_{t,a} - \mu^{\top} x_{t,a} + (\mu - \hat{\mu}_t)^{\top} B_t (\mu - \hat{\mu}_t) \right) \\
&\propto \exp\left(-\frac{1}{2v^2} (\mu^{\top} B_{t+1} \mu - 2\mu^{\top} B_{t+1} \hat{\mu}_{t+1}) \right) \\
&\propto \exp\left(-\frac{1}{2v^2} (u - \hat{\mu}_{t+1})^{\top} B_{t+1} (u - \hat{\mu}_{t+1}) \right) \\
&\propto \mathcal{N}(\hat{\mu}_{t+1}, v^2 B_{t+1}^{-1})
\end{aligned}$$

■

Theorem 4.7 gives us a way to update our believe about the parameter after observing new data. The algorithm is described in Algorithm 4.

Theorem 4.8 *With probability $1 - \delta$, the regret is bounded by:*

$$R_T = O\left(\frac{d^2}{\epsilon} \sqrt{T^{1+\epsilon}} (\ln(Td) \ln \frac{1}{\delta}) \right)$$

Algorithm 4 Thompson Sampling with Linear Payoff

```
Require: $\delta \in (0, 1]$
  Define $v = R\sqrt{\frac{24}{\epsilon} d \ln(\frac{t}{\delta})}, B = I_d, \hat{\mu} = 0_d, f = 0_d$
  for $t = 1, 2..., T$ do
      Sample $u_t$ from distribution $\mathcal{N}(\hat{\mu}, v^2 B^{-1})$
      Pull arm $a_t = \arg\max_a x_{t,a}^\top u_t$
      Receive reward $r_t$
      Update:
          $B = B + x_{t,a} x_{t,a}^\top$
          $f = f + x_{t,a} r_t$
          $\hat{\mu} = B^{-1} f$
  end for
```

Chapelle and Li (2011) described a way of doing Thompson sampling with logistic regression. Let w be the weight vector of logistic regression and w_i be the i^{th} element. Each w_i follows a Gaussian distribution $w_i \sim \mathcal{N}(m_i, q_i^{-1})$. They apply Laplace approximation to get the posterior distribution of the weight vector, which is a Gaussian distribution with diagonal covariance matrix. The algorithm is described in Algorithm 5. Chapelle and Li (2011) didn't give a regret bound for this algorithm, but showed that it achieve good empirical results on display advertising.

4.1.5 SpectralUCB

4.2 Kernelized Stochastic Contextual Bandits

Recall that in section 4.1 we assume a linear relationship between the arm's features and the expected reward: $\mathrm{E}(r) = x^\top \theta^*$; however, linearity assumption is not always true. Instead, in this section we assume the expected reward of an arm is given by an unknown (possibly non-linear) reward function $f : \mathrm{R}^d \to \mathrm{R}$:

$$r = f(x) + \epsilon \tag{6}$$

where ϵ is a noise term with mean zero. We further assume that f is from a Reproducing Kernel Hilbert Spaces (RKHS) corresponding to some kernel $k(\cdot, \cdot)$. We define $\phi : \mathrm{R}^d \to \mathcal{H}$ as the mapping from the domain of x to the RKHS $\mathcal{H}$, so that $f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}}$. In the following we talk about GP-UCB/CGP-UCB and KernelUCB. GP-UCB/CGP-UCB is a Bayesian approach that puts a Gaussian Process prior on f to encode the assumption of smoothness, and KernelUCB is a Frequentist approach that builds estimators from linear regression in RKHS $\mathcal{H}$ and choose an appropriate regularizer to encode the assumption of smoothness.

Algorithm 5 Thompson Sampling with Logistic Regression

```
Require $\lambda \geq 0$, batch size $S \geq 0$
Define $\mathcal{D} = \{\}$, $m_i = 0, q_i = \lambda$ for all elements in the weight vector $w \in \mathrm{R}^d$.
for each batch $b = 1, ..., B$ do    ▷ Process in mini-batch style
    Draw $w$ from posterior distribution $\mathcal{N}(m, \mathrm{diag}(q)^{-1})$
    for $t = 1, ..., S$ do
        Receive context $x_{b,t,j}$ for each article $j$.
        Select arm $a_t = \arg\max_j 1/(1 + \exp(-x_{b,t,j} \cdot w))$
        Receive reward $r_t \in \{0, 1\}$
        $\mathcal{D} = \mathcal{D} \cup \{x_{b,t,a_t}, a_t, r_t\}$
    end for
    Solve the following optimization problem to get $\bar{w}$

        $\frac{1}{2}\sum_{i=1}^{d} q_i(\bar{w}_i - m_i)^2 + \sum_{(x,r)\in\mathcal{D}} \ln(1 + \exp(-r\bar{w} \cdot x))$

    Set prior for next block

        $m_i = \bar{w}_i$
        $q_i = q_i + \sum_{(x,r)\in\mathcal{D}} x_i^2 p_j(1 - p_j), p_j = (1 + \exp(-\bar{w} \cdot x))^{-1}$

end for
```

4.2.1 GP-UCB/CGP-UCB

The Gaussian Process can be viewed as a prior over a regression function.

$$f(x) \sim GP(\mu(x), k(x, x'))$$

where $\mu(x)$ is the mean function and $k(x, x')$ is the covariance function:

$$\begin{aligned} \mu(x) &= \mathrm{E}(f(x)) \\ k(x, x') &= \mathrm{E}\left((f(x) - \mu(x))(f(x') - \mu(x'))\right) \end{aligned}$$

Assume the noise term ϵ in Equation (6) follows Gaussian distribution $\mathcal{N}(0, \sigma^2)$ with some variance σ^2. Then, given any finite points $\{x_1, ..., x_N\}$, their response $\boldsymbol{r}_N = [r_1, ..., r_N]^\top$ follows multivariate Gaussian distribution:

$$\boldsymbol{r}_N \sim \mathcal{N}([\mu(x_1), ..., \mu(x_N)]^\top, K_N + \sigma^2 \mathrm{I}_N)$$

where $(K_N)_{ij} = k(x_i, x_j)$. It turns out that the posterior distribution of f given $\{x_1, ..., x_N\}$ is also a Gaussian Process distribution $GP(\mu_N(x), k_N(x, x'))$ with

$$\begin{aligned} \mu_N(x) &= \boldsymbol{k}_N(x)^\top (K_N + \sigma^2 \mathrm{I})^{-1} \boldsymbol{r}_N \\ k_N(x, x') &= k(x, x') - \boldsymbol{k}_N(x)^\top (K_N + \sigma^2 \mathrm{I})^{-1} \boldsymbol{k}_N(x') \end{aligned}$$

where $\boldsymbol{k}_N(x) = [k(x_1, x), ..., k(x_N, x)]^\top$.

GP-UCB (Srinivas et al., 2010) is a Bayesian approach to infer the unknown reward function f. The domain of f is denoted by $\mathcal{D}$. $\mathcal{D}$ could be a finite set containing $|\mathcal{D}|$ $d-$dimensional vectors, or a infinite set such as R^d. GP-UCB puts a Gaussian process prior on f : $f \sim GP(\mu(x), k(x, x'))$, and it updates the posterior distribution of f after each observation. Inspired by the UCB-style algorithm (Auer et al., 2002a), it selects an point x_t at time t with the following strategy:

$$x_t = \arg\max_{x \in \mathcal{D}} \mu_{t-1}(x) + \sqrt{\beta_t} \sigma_{t-1}(x) \tag{7}$$

where $\mu_{t-1}(x)$ is the posterior mean of x, $\sigma^2_{t-1}(x) = k_{t-1}(x, x)$, and β_t is appropriately chosen constant. (7) shows the exploration-exploitation tradeoff of GP-UCB: large $\mu_{t-1}(x)$ represents high estimated reward, and large $\sigma_{t-1}(x)$ represents high uncertainty. GP-UCB is described in Algorithm 6.

Algorithm 6 GP-UCB

```
Require: $\mu_0 = 0$, $\sigma_0$, kernel $k$
  for $t = 1, 2, ...$ do
      select arm $a_t = \arg\max_{a \in \mathcal{A}} \mu_{t-1}(x_{t,a}) + \sqrt{\beta_t} \sigma_{t-1}(x_{t,a})$
      receive reward $r_t$
      Update posterior distribution of $f$; obtain $\mu_t$ and $\sigma_t$
  end for
```

The regret of GP-UCB is defined as follow:

$$R_T = \sum_{t=1}^{T} f(x^*) - f(x_t) \tag{8}$$

where $x^* = \arg\max_{x \in \mathcal{D}} f(x)$. From a bandits algorithm's perspective, we can view each data point x in GP-UCB as an arm; however, in this case the features of an arm won't change based on the contexts observed, and the best arm is always the same. We can also view each data point x as a feature vector that encodes both the arm and context information, however, in that case x^* in Equation (8) becomes $x^* = \arg\max_{x \in \mathcal{D}_t} f(x)$ where $\mathcal{D}_t$ is the domain of f under current context.

Define $\mathrm{I}(\boldsymbol{r}_A; f) = \mathrm{H}(\boldsymbol{r}_A) - \mathrm{H}(\boldsymbol{r}_A | f)$ as the mutual information between f and rewards of a set of arms $A \in \mathcal{D}$. Define the maximum information gain γ_T after T rounds as

$$\gamma_T = \max_{A:|A|=T} \mathrm{I}(\boldsymbol{r}_A; f)$$

Note that γ_T depends on the kernel we choose. Srinivas et al. (2010) showed that if $\beta_t = 2\ln(Kt^2\pi^2/6\delta)$, then GP-UCB achieves a regret bound of $\tilde{O}\left(\sqrt{T\gamma_T \ln K}\right)$ with high probability. Srinivas et al. (2010) also analyzed the agnostic setting, that is, the true function f is not sampled from a Gaussian Process prior, but has bounded norm in RKHS:

Theorem 4.9 *Suppose the true f is in the RKHS $\mathcal{H}$ corresponding to kernel $k(x, x')$. Assume $\langle f, f \rangle_{\mathcal{H}} \leq B$. Let $\beta_t = 2B + 300\gamma_t \ln^3(t/\delta)$, let the prior be $GP(0, k(x, x'))$, and the noise model be $\mathcal{N}(0, \sigma^2)$. Assume the true noise ϵ has zero mean and is bounded by σ almost surely. Then the regret bound of GP-UCB is*

$$R_T = \tilde{O}\left(\sqrt{T}\left(B\sqrt{\gamma_T} + \gamma_T\right)\right)$$

with high probability.

Srinivas et al. (2010) also showed the bound of γ_T for some common kernels. For finite dimensional linear kernel $\gamma_T = \tilde{O}(d \ln T)$; for squared exponential kernel $\gamma_T = \tilde{O}((\ln T)^{d+1})$.

CGP-UCB (Krause and Ong, 2011) extends GP-UCB and explicitly model the contexts. It defines a context space $\mathcal{Z}$ and an arm space $\mathcal{D}$; Both $\mathcal{Z}$ and $\mathcal{D}$ can be infinite sets. CGP-UCB assumes the unknown reward function f is defined over the join space of contexts and arms:

$$r = f(z, x) + \epsilon$$

where $z \in \mathcal{Z}$ and $x \in \mathcal{D}$. The algorithm framework is the same as GP-UCB except that now we need to choose a kernel k over the joint space of $\mathcal{Z}$ and $\mathcal{D}$. Krause and Ong (2011) proposed one possible kernel $k(\{z, x\}, \{z', x'\}) = k_Z(z, z')k_{\mathcal{D}}(x, x')$. We can use different kernels for the context spaces and arm spaces.

4.2.2 KernelUCB

KernelUCB (Valko et al., 2013) is a Frequentist approach to learn the unknown reward function f. It estimates f using regularized linear regression in RKHS corresponding to some kernel $k(\cdot, \cdot)$. We can also view KernelUCB as a Kernelized version of LinUCB.

Assume there are K arms in the arm set $\mathcal{A}$, and the best arm at time t is $a^* = \arg\max_{a \in \mathcal{A}} f(x_{t,a})$, then the regret is defined as

$$R_T = \sum_{t=1}^{T} f(x_{t,a_t^*}) - f(x_{t,a_t})$$

We apply kernelized ridge regression to estimate f. Given the arms pulled $\{x_1, ..., x_{t-1}\}$ and their rewards $\boldsymbol{r}_t = [r_1, ..., r_{t-1}]$ up to time $t-1$, define the dual variable

$$\alpha_t = (K_t + \gamma \mathrm{I}_t)^{-1} \boldsymbol{r}_t$$

where $(K_t)_{ij} = k(x_i, x_j)$. Then the predictive value of a given arm $x_{t,a}$ has the following closed form

$$\hat{f}(x_{t,a}) = k_t(x_{t,a})^\top \alpha_t$$

where $k_t(x_{t,a}) = [k(x_1, x_{t,a}), ..., k(x_{t-1}, x_{t,a})]^\top$. Now we have the predicted reward, we need to compute the half width of the confidence interval of the predicted reward. Recall that in LinUCB such half width is defined as $\sqrt{x_{t,a}(D_t^\top D_t + \gamma \mathrm{I}_d)^{-1} x_{t,a}}$, similarly in kernelized ridge regression we define the half width as

$$\hat{\sigma}_{t,a} = \sqrt{\phi(x_{t,a})^\top (\Phi_t^T \Phi_t + \gamma \mathrm{I})^{-1} \phi(x_{t,a})} \tag{9}$$

where $\phi(\cdot)$ is the mapping from the domain of x to the RKHS, and $\Phi_t = [\phi(x_1)^\top, ..., \phi(x_{t-1})^\top]^\top$. In order to compute (9), Valko et al. (2013) derived a dual representation of (9):

$$\hat{\sigma}_{t,a} = \gamma^{-1/2} \sqrt{k(x_{t,a}, x_{t,a}) - k_t(x_{t,a})^\top (K_t + \gamma \mathrm{I})^{-1} k_t(x_{t,a})}$$

KernelUCB chooses the action a_t at time t with the following strategy

$$a_t = \arg\max_{a \in \mathcal{A}} \left(k_t(x_{t,a})^\top \alpha_t + \eta \hat{\sigma}_{t,a} \right)$$

where η is the scaling parameter.

To derive regret bound, Valko et al. (2013) proposed SupKernelUCB based on KernelUCB, which is similar to the relationship between SupLinUCB and LinUCB. Since the dimension of $\phi(x)$ may be infinite, we cannot directly apply LinUCB or SupLinUCB's regret bound. Instead, Valko et al. (2013) defined a data dependent quantity $\tilde{d}$ called effective dimension: Let $(\lambda_{i,t})_{i \geq 1}$ denote the eigenvalues of $\Phi_t^\top \Phi_t + \gamma \mathrm{I}$ in decreasing order, define $\tilde{d}$ as

$$\tilde{d} = \min\{j : j\gamma \ln T \geq \Lambda_{T,j}\} \text{ where } \Lambda_{T,j} = \sum_{i>j} \lambda_{i,T} - \gamma$$

$\tilde{d}$ measures how quickly the eigenvalues of $\Phi_t^\top \Phi_t$ are decreasing. Valko et al. (2013) showed that if $\sqrt{\langle f, f\rangle_{\mathcal{H}}} \leq B$ for some B and if we set regularization parameter $\gamma = 1/B$ and scaling parameter $\eta = \sqrt{2\ln 2TN/\eta}$, then the regret bound of SupKernelUCB is $\tilde{O}(\sqrt{B\tilde{d}T})$. They showed that for linear kernel $\tilde{d} \leq d$; Also, compared with GP-UCB, $\mathrm{I}(\boldsymbol{r}_A; f) \geq \Omega(\tilde{d}\ln\ln T)$, which means KernelUCB achieves better regret bound than GP-UCB in agnostic case.

4.3 Stochastic Contextual Bandits with Arbitrary Set of Policies

4.3.1 Epoch-Greedy

Epoch-Greedy (Langford and Zhang, 2008) treats contextual bandits as a classification problem, and it solves an empirical risk minimization (ERM) problem to find the currently best policy. One advantage of Epoch-Greedy is that the hypothesis space can be finite or even infinite with finite VC-dimension, without an assumption of linear payoff.

There are two key problems Epoch-Greedy need to solve in order to achieve low regret: 1. how to get unbiased estimator from ERM; 2. how to balance exploration and exploitation when we don't know the time horizon T. To solve the first problem, Epoch-Greedy makes explicit distinctions between exploration and exploitation steps. In an exploration step, it selects an arm uniformly at random, and the goal is to form unbiased samples for learning. In an exploitation step, it selects the arm based on the best policy learned from the exploration samples. Of course, Epoch-Greedy adopts the trick we described in Section 2 to get unbiased estimator. For the second problem, note that since Epoch-Greedy strictly separate exploration and exploitation steps, so if it already know T in advance then it should always explore for the first T' steps, and then exploit for the following $T - T'$ steps. The reason is that there is no advantage to take an exploitation step before the last exploration step. However generally T is unknown, so Epoch-Greedy algorithm runs in a mini-batch style: it runs one epoch at a time, and within that epoch, it first performs one step of exploration, and followed by several steps of exploitation. The algorithm is shown in Algorithm 7.

Algorithm 7 Epoch-Greedy

```
Require: $s(W_\ell)$: exploitation steps given samples $W_\ell$
  Init exploration samples $W_0 = \{\}, t_1 = 1$
  for $\ell = 1, 2, ...$ do
      $t = t_\ell$    ▷ One step of exploration
      Draw an arm $a_t \in \{1, ..., K\}$ uniformly at random
      Receive reward $r_{a_t} \in [0, 1]$
      $W_\ell = W_{\ell-1} \cup (x_t, a_t, r_{a_t})$
      Solve $\hat{h}_\ell = \max_{h\in\mathcal{H}} \sum_{(x,a,r_a)\in W_\ell} \frac{r_a \mathbb{I}(h(x)=a)}{1/K}$
      $t_{\ell+1} = t_\ell + s(W_\ell) + 1$
      for $t = t_\ell + 1, ..., t_\ell - 1$ do    ▷ $s(W_\ell)$ steps of exploration
          Select arm $a_t = \hat{h}_\ell(x_t)$
          Receive reward $r_{a_t} \in [0, 1]$
      end for
  end for
```

Different from the EXP4 setting, we do not assume an adversary environment here. Instead, we assume there is a distribution P over (x, r), where $x \in \mathcal{X}$ is the context and $r = [r_1, ..., r_K] \in [0,1]^K$ is the reward vector. At time t, the world reveals context x_t, and the algorithm selects arm $a_t \in \{1, ...K\}$ based on the context, and then the world reveals the reward r_{a_t} of arm a_t. The algorithm makes its decision based on a policy/hypothesis $h \in \mathcal{H} : \mathcal{X} \rightarrow \{1, ..., K\}$. $\mathcal{H}$ is the policy/hypothesis space, and it can be an infinite space such as all linear hypothesis in dimension d, or it can be a finite space consists of $N = |\mathcal{H}|$ hypothesis. In this survey we mainly focus on finite space, but it is easy to extend to infinite space.

Let $Z_t = (x_t, a_t, r_{a_t})$ be the t^{th} exploration sample, and $Z_1^n = \{Z_1, ..., Z_n\}$. The expected reward of a hypothesis h is

$$R(h) = \mathrm{E}_{(x,r)\sim P}[r_{h(x)}]$$

so the regret of the algorithm is

$$R_T = \sup_{h\in\mathcal{H}} TR(h) - \mathrm{E}\sum_{t=1}^{T} r_{a_t}$$

The expectation is with respect to Z_1^n and any random variable in the algorithm.

Denote the data-dependent exploitation step count by $s(Z_1^n)$, so $s(Z_1^n)$ means that based on all samples Z_1^n from exploration steps, the algorithm should do $s(Z_1^n)$ steps exploitation. The hypothesis that maximizing the empirical reward is

$$\hat{h}(Z_1^n) = \arg\max_{h\in\mathcal{H}} \sum_{t=1}^{n} \frac{r_{a_t}\mathbb{I}(h(x_t) = a_t)}{1/K}$$

The per-epoch exploitation cost is defined as

$$\mu_n(\mathcal{H}, s) = \mathrm{E}_{Z_1^n}\left(\sup_{h\in\mathcal{H}} R(h) - R(\hat{h}(Z_1^n))\right) s(Z_1^n)$$

When $S(Z_1^n) = 1$

$$\mu_n(\mathcal{H}, 1) = \mathrm{E}_{Z_1^n}\left(\sup_{h\in\mathcal{H}} R(h) - R(\hat{h}(Z_1^n))\right)$$

The per-epoch exploration regret is less or equal to 1 since we only do one step exploration, so we would want to select a $s(Z_1^n)$ such that the per-epoch exploitation regret $\mu_n(\mathcal{H}, s) = 1$. Later we will show how to choose $s(Z_1^n)$.

Theorem 4.10 *For all T, n_ℓ, L such that: $T \le L + \sum_{\ell=1}^{L} n_\ell$, the regret of Epoch-Greedy is bounded by*

$$R_T \le L + \sum_{\ell=1}^{L} \mu_\ell(\mathcal{H}, s) + T\sum_{\ell=1}^{L} P[s(Z_1^\ell) < n_\ell]$$

The above theorem means that suppose we only consider the first L epochs, and for each epoch ℓ, we use a sample independent variable n_ℓ to bound $S(Z_1^\ell)$, then the regret up to time T is bounded by the above.

Proof based on the relationship between $s(Z_1^n)$ and n_ℓ, one of the following two events will occur:

1. $s(Z_1^\ell) < n_\ell$ for some $\ell = 1, ..., L$
2. $s(Z_1^\ell) \geq n_\ell$ for all $\ell = 1, ..., L$

In the second event, n_ℓ is the lower bound for $s(Z_1^\ell)$, so $T \leq L + \sum_{\ell=1}^{L} n_\ell \leq \text{Ł} + \sum_{\ell=1}^{L} s(Z_1^\ell)$, so the epoch that contains T must be less or equal to epoch L, hence the regret is less or equal to the sum of the regret in the first L epochs. Also within each epoch, the algorithm do one step exploration and then $s_{Z_1^\ell}$ exploitation, so the regret bound when event 2 occurs is

$$R_{T,2} \leq L + \sum_{\ell=1}^{L} \mu_\ell(\mathcal{H}, s)$$

The regret bound when event 1 occurs is $R_{T,1} \leq T$ because the reward $r \in [0, 1]$. Together we get the regret bound

$$\begin{aligned} R_T &\leq T \sum_{\ell=1}^{L} P[s(Z_1^\ell) < n_\ell] + \prod_{\ell=1}^{L} P[s(Z_1^\ell) \geq n_\ell] \sum_{\ell=1}^{L} (1 + \mu_\ell(\mathcal{H}, s)) \\ &\leq T \sum_{\ell=1}^{L} P[s(Z_1^\ell) < n_\ell] + L + \sum_{\ell=1}^{L} \mu_\ell(\mathcal{H}, s) \end{aligned}$$

■

Theorem 4.10 gives us a general bound, we now derive a specific problem-independent bound based on that.

One essential thing we need to do is to bound $\sup_{h \in \mathcal{H}} R(h) - R(\hat{h}(Z_1^n))$. If hypothesis space $\mathcal{H}$ is finite, we can use finite class uniform bound, and if $\mathcal{H}$ is infinite, we can use VC-dimension or other infinite uniform bound techniques. The two proofs are similar, and here to consistent with the original paper, we assume $\mathcal{H}$ is a finite space.

Theorem 4.11 (Bernstein) *If $P(|Y_i| \leq c) = 1$ and $\mathrm{E}(Y_i) = 0$, then for any $t > 0$,*

$$P(|\overline{Y}_n| > \epsilon) \leq 2 \exp\left\{ -\frac{n\epsilon^2}{2\sigma^2 + 2c\epsilon/3} \right\}$$

where $\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} Var(Y_i)$.

Theorem 4.12 *With probability $1 - \delta$, the problem-independent regret of Epoch-Greedy is*

$$R_T \leq cT^{2/3}(K \ln(|\mathcal{H}|/\delta))^{1/3}$$

Proof Follow Section 2, define $\hat{R}(h) = \frac{1}{n}\sum_i \frac{\mathbb{I}(h(x_i)=a_i)r_{a_i}}{1/K}$, the empirical sample reward of a hypothesis h. Also define $\hat{R}_i = \frac{\mathbb{I}(h(x_i)=a_i)r_{a_i}}{1/K}$, then $\mathrm{E}\hat{R}(h) = R(h)$, and

$$\begin{aligned}
\mathrm{var}(\hat{R}_i) &\leq \mathrm{E}(\hat{R}_i^2) \\
&= \mathrm{E}K^2\mathbb{I}(h(x_i) = a_i)r_{a_i}^2 \\
&\leq \mathrm{E}K^2\mathbb{I}(h(x_i) = a_i) \\
&= \mathrm{E}K^2 1/K \\
&= K
\end{aligned}$$

So the variance is bounded by K and we can apply Bernstein inequality to get:

$$P(|\hat{R}(h) - R(h)| > \epsilon) \leq 2\exp\left\{-\frac{n\epsilon^2}{2K + 2c\epsilon/3}\right\}$$

From union bound we have

$$P\left(\sup_{h\in\mathcal{H}} |\hat{R}(h) - R(h)| > \epsilon\right) \leq 2N\exp\left\{-\frac{n\epsilon^2}{2K + 2c\epsilon/3}\right\}$$

Set the right-hand side to δ and solve for ϵ we have,

$$\epsilon = c\sqrt{\frac{K\ln(N/\delta)}{n}}$$

So, with probability $1 - \delta$,

$$\sup_{h\in\mathcal{H}} |\hat{R}(h) - R(h)| \leq c\sqrt{\frac{K\ln(N/\delta)}{n}}$$

Let $\hat{h}$ be the estimated hypothesis, and h_* be the best hypothesis, then with probability $1 - \delta$,

$$R(\hat{h}) \leq \hat{R}(\hat{h}) + c\sqrt{\frac{K\ln(N/\delta)}{n}} \leq \hat{R}(h_*) + c\sqrt{\frac{K\ln(N/\delta)}{n}} \leq R(h_*) + 2c\sqrt{\frac{K\ln(N/\delta)}{n}}$$

So

$$\mu_\ell(\mathcal{H}, 1) \leq 2c\sqrt{\frac{K\ln(N/\delta)}{\ell}}$$

To make $\mu_\ell(\mathcal{H}, s) \leq 1$, we can choose

$$s(Z_1^\ell) = \lfloor c'\sqrt{\ell/(K\ln(N/\delta))}\rfloor$$

Take $n_\ell = \lfloor c'\sqrt{\ell/(K\ln(N/\delta))}\rfloor$, then $P[s(Z_1^\ell) < n_\ell] = 0$. So the regret

$$\begin{aligned}
R_T &\leq L + \sum_{\ell=1}^{L} \mu_\ell(\mathcal{H}, s) \\
&\leq 2L
\end{aligned}$$

Now the only job is to find the L. We can pick a L such that $T \leq \sum_{\ell=1}^{L} n_\ell$, so T will also satisfy $T \leq L + \sum_{\ell=1}^{L} n_\ell$.

$$\begin{aligned} T &= \sum_{\ell=1}^{L} n_\ell \\ &= \sum_{\ell=1}^{L} \lfloor c' \sqrt{\ell/K \ln(N/\delta)} \rfloor \\ &= c' \lfloor \sqrt{1/(K \ln(N/\delta))} (\sum_{\ell=1}^{L} \sqrt{\ell}) \rfloor \\ &= c'' \lfloor \sqrt{1/(K \ln(N/\delta))} L^{3/2} \rfloor \end{aligned}$$

So

$$\begin{aligned} L &= c'' \lfloor (K \ln(N/\delta))^{1/3} T^{2/3} \rfloor \\ R_T &\leq c''' (K \ln(N/\delta))^{1/3} T^{2/3} \end{aligned}$$

Hence, with probability $1 - \delta$, the regret of Epoch-Greedy is $O((K \ln(N/\delta))^{1/3} T^{2/3})$. ■

Compared to EXP4, Epoch-Greedy has a weaker bound but it converge with probability instead of expectation; Compared to EXP4.P, Epoch-Greedy has a weaker bound but it does not require the knowledge of T.

4.3.2 RandomizedUCB

Recall that in EXP4.P and Epoch-Greedy we are always competing with the best policy/expert, and the optimal regret bound $O(\sqrt{KT \ln N})$ scales only logarithmically in the number of policies, so we could boost the model performance by adding more and more potential policies to the policy set $\mathcal{H}$. With high probability EXP4.P achieves the optimal regret, however the running time scales linearly instead of logarithmically in the number of experts. As a results, we are constrained by the computational bottleneck. Epoch-Greedy could achieve sub-linear running time depending on what assumptions we make about the $\mathcal{H}$ and ERM, however the regret bound is $O((K \ln N)^{(1/3)} T^{(2/3)})$, which is sub-optimal. RandomizedUCB (Dudik et al., 2011), on the other hand, could achieve optimal regret while having a polylog(N) running time. One key difference compared to Epoch-Greedy is that it assigns a non-uniform distribution over policies, while Epoch-Greedy assigns uniform distribution when doing exploration. Also RandomizedUCB does not make explicit distinctions between exploration and exploitation.

Similar to Epoch-Greedy, let $\mathcal{A}$ be a set of K arms $\{1, ..., K\}$, and D be an arbitrary distribution over (x, r), where $x \in \mathcal{X}$ is the context and $r \in [0,1]^K$ is the reward vector. Let D_X be the marginal distribution of D over x. At time t, the world samples a (x_t, r_t) pair and reveals x_t to the algorithm, the algorithm then picks an arm $a_t \in \mathcal{A}$ and then receives reward r_{a_t} from the world. Denote a set of policies $h : \mathcal{X} \rightarrow \mathcal{A}$ by $\mathcal{H}$. The algorithm has access to $\mathcal{H}$ and makes decisions based on x_t and $\mathcal{H}$. The expected reward of a policy $h \in \mathcal{H}$ is

$$R(h) = \mathrm{E}_{(x,r) \sim D}[r_{h(x)}]$$

and the regret is defined as

$$R_T = \sup_{h \in \mathcal{H}} TR(h) - \mathrm{E}\sum_{t=1}^{T} r_{a_t}$$

Denote the sample at time t by $Z_t = (x_t, a_t, r_{a_t}, p_{a_t})$, where p_{a_t} is the probability of choosing a_t at time t. Denote all the samples up to time t by $Z_1^t = \{Z_1, ..., Z_t\}$. Then the unbiased reward estimator of policy h is

$$\hat{R}(h) = \frac{1}{t} \sum_{(x,a,r,p) \in Z_1^t} \frac{r\mathbb{I}(h(x) = a)}{p}$$

The unbiased empirical reward maximization estimator at time t is

$$\hat{h}_t = \arg\max_{h \in \mathcal{H}} \sum_{(x,a,r,p) \in Z_1^t} \frac{r\mathbb{I}(h(x) = a)}{p}$$

RandomizedUCB chooses a distribution P over policies $\mathcal{H}$ which in turn induce distributions over arms. Define

$$W_P(x,a) = \sum_{h(x)=a} P(h)$$

be the induced distribution over arms, and

$$W'_{P,\mu}(x,a) = (1 - K\mu)W_P(x,a) + \mu$$

be the smoothed version of W_P with a minimum probability of μ. Define

$$R(W) = \mathrm{E}_{(x,r)\sim D}[r \cdot W(x)]$$
$$\hat{R}(W) = \frac{1}{t} \sum_{(x,a,r,p) \in Z_1^t} \frac{rW(x,a)}{p}$$

To introduce RandomizedUCB, let's introduce POLICYELIMINATION algorithm first. POLICYELIMINATION is not practical but it captures the basic ideas behind RandomizedUCB. The general idea is to find the best policy by empirical risk. However empirical risk suffers from variance (no bias since we again adopt the trick in Section 2), so POLICYELIMINATION chooses a distribution P_t over all policies to control the variance of $\hat{R}(h)$ for all policies, and then eliminate policies that are not likely to be optimal.

By Minimax theorem Dudik et al. (2011) proved that there always exists a distribution P_t satisfy the constrain in Algorithm 8.

Theorem 4.13 (Freedman-style Inequality) *Let $y_1, ..., y_T$ be a sequence of real-valued random variables. Let $V, R \in \mathrm{R}$ such that $\sum_{t=1}^{T} \mathrm{var}[y_t] \le V$, and for all t, $y_t - \mathrm{E}_t[y_t] \le R$. Then for any $\delta > 0$ such that $R \le \sqrt{V/\ln(2/\delta)}$, with probability at least $1 - \delta$,*

$$\left| \sum_{t=1}^{T} y_t - \sum_{t=1}^{T} \mathrm{E}_t[y_t] \right| \le 2\sqrt{V \ln(2/\delta)}$$

Algorithm 8 POLICYELIMINATION

```
Require: $\delta \in (0, 1]$
  Define $\delta_t = \delta/4Nt^2$, $b_t = 2\sqrt{\frac{2K\ln(1/\delta_t)}{t}}$, $\mu_t = \min\left\{\frac{1}{2K}, \sqrt{\frac{\ln(1/\delta_t)}{2Kt}}\right\}$
  for t=1,..., T do
      Choose a distribution $P_t$ over $\mathcal{H}_{t-1}$ s.t. $\forall\ h \in \mathcal{H}_{t-1}$
          $$\mathrm{E}_{x \in D_X}\left[\frac{1}{W'_{P_t,\mu_t}(x, h(x))}\right] \le 2K \qquad (10)$$
      Sample $a_t$ from $W'_t = W'_{P_t,\mu_t}(x_t, \cdot)$
      Receive reward $r_{a_t}$
      Let
          $$\mathcal{H}_t = \left\{h \in \mathcal{H}_{t-1} : \delta_t(h) \ge \left(\max_{h' \in \mathcal{H}_{t-1}} \delta_t(h')\right) - 2b_t\right\} \qquad (11)$$
  end for
```

Theorem 4.14 *With probability at least* $1 - \delta$*, the regret of POLICYELIMINATION is bounded by:*

$$R_T = O(16\sqrt{2TK\ln\frac{4T^2N}{\delta}})$$

Proof Let

$$\hat{R}_i(h) = \frac{r_t \mathbb{I}(h(x_t) = a_t)}{W'_t(h(x_t))}$$

the estimated reward of policy h at time t. To make use of Freedman's inequality, we need to bound the variance of $\hat{R}_i(h)$

$$\begin{aligned} \mathrm{var}(\hat{R}_i(h)) &\le \mathrm{E}\hat{R}_i(h)^2 \\ &= \mathrm{E}\frac{r_t^2 \mathbb{I}(h(x_t) = a_t)}{W'_t(h(x_t))^2} \\ &\le \mathrm{E}\frac{\mathbb{I}(h(x_t) = a_t)}{W'_t(h(x_t))^2} \\ &= \mathrm{E}\frac{1}{W'_t(h(x_t))} \\ &\le 2K \end{aligned}$$

The last inequality is from the constrain in Equation (10). So

$$\sum_{i=1}^{t} \mathrm{var}[\hat{R}_i(h)^2] \le 2Kt = V_t$$

Now we need to check if R_t satisfy the constrain in Theorem 4.13. Let t_0 be the first t such that $\mu_t < 1/2K$. when $t \geq t_0$, then for all $t' \leq t$,

$$\hat{R}_{t'}(h) \leq 1/\mu_{t'} \leq 1/\mu_t = \sqrt{\frac{2Kt}{\ln(1/\delta_t)}} = \sqrt{\frac{V_t}{\ln(1/\delta_t)}}$$

So now we can apply Freedman's inequality and get

$$P(|\hat{R}(h) - R(h)| \geq b_t) \leq 2\delta_t$$

Take the union bound over all policies and t

$$\begin{aligned}\sup_{t' \in t} \sup_{h \in \mathcal{H}} P(|\hat{R}(h) - R(h)| \geq b_t) &\leq 2N \sum_{t'=1}^{t} \delta_{t'} \\ &\leq \sum_{t'=1}^{t} \frac{\delta}{2t'^2} \\ &\leq \delta\end{aligned}$$

So with probability $1 - \delta$, we have

$$|\hat{R}(h) - R(h)| \leq b_t$$

When $t < t_0$, then $\mu_t < 1/2K$ and $b_t \geq 4K\mu_t \geq 2$, then the above bound still holds since reward is bounded by 1.

To sum up, we make use of the convergence of $\sum_t \frac{1}{t^2}$ to construct δ_t so that the union bound is less than δ, and we use R_t's constrain in Freedman's inequality to construct u_t and Freedman's inequality to construct b_t.

Lemma 4.15 *With probability at least* $1 - \delta$*,*

$$|\hat{R}(h) - R(h)| \leq b_t$$

From Lemma 4.15 we have

$$\begin{gathered}\hat{R}(h) - b_t \leq R(h) \leq R(h^*) \leq \hat{R}(h^*) + b_t \\ \hat{R}(h) \leq \hat{R}(h^*) + 2b_t\end{gathered}$$

where $h^* = \max_{h \in \mathcal{H}} R(h^*)$. So we can see that h^* is always in $\mathcal{H}_t$ after the policy elimination step (Equation 11) in Algorithm 8. Also, if $R(h) \leq R(h^*) - 4b_t$, then

$$\begin{gathered}\hat{R}(h) - b_t \leq R(h) \leq R(h^*) - 4b_t \leq \hat{R}(h^*) + b_t - 4b_t \\ \hat{R}(h) \leq \hat{R}(h^*) - 2b_t\end{gathered}$$

However, as we can see from the elimination step, all the policies which satisfy $\hat{R}(h) \leq \hat{R}(h^*) - 2b_t$ is eliminated. So for all the remaining policies $h \in \mathcal{H}_t$, we have $R(h^*) - R(h) \leq 4b_t$, so the regret

$$\begin{aligned}
R_T &\leq \sum_{t=1}^{T} R(h^*) - R(h) \\
&\leq 4\sum_{t=1}^{T} b_t \\
&\leq 8\sqrt{2K \ln \frac{4Nt^2}{\delta}} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \\
&\leq 8\sqrt{2K \ln \frac{4Nt^2}{\delta}} 2\sqrt{T} \\
&\leq 16\sqrt{2TK \ln \frac{4NT^2}{\delta}}
\end{aligned}$$

■

POLICYELIMINATION describes the basic idea of RandomizedUCB, however POLICYELIMINATION is not practical because it does not actually show how to find the distribution P_t, also it requires the knowledge of D_x. To solve these problems, RandomzedUCB always considers the full set of policies and use an argmax oracle to find the distribution P_t over all policies, and instead of using D_x, the algorithm uses history samples. Define

$$\begin{aligned}
\Delta_D(W) &= R(h^*) - R(W) \\
\Delta_t(W) &= \hat{R}(\hat{h}_t) - \hat{R}(W)
\end{aligned}$$

RandomizedUCB is described in Algorithm 9. Similar to POLICYELIMINATION, P_t in RandomizedUCB algorithm is to control the variance. However, instead of controlling each policy separately, it controls the expectation of the variance with respect to the distribution Q. The right-hand side of Equation (12) is upper bounded by $c\Delta_{t-1}(W_Q)^2$, which measures the empirical performance of distribution Q. So the general idea of this optimization problem is to bound the expected variance of empirical reward with respect to all possible distribution Q, whereas if Q achieves high empirical reward then the bound is tight hence the variance is tight, and if Q has low empirical reward, the bound is loose. This makes sure that P_t puts more weight on policies with low regret. Dudik et al. (2011) showed that the regret of RandomizedUCB is $O(\sqrt{TK \ln(TN/\delta)})$.

To solve the optimization problem in the algorithm, RandomizedUCB uses an argmax oracle($\mathcal{AMO}$) and relies on the ellipsoid method. The main contribution is the following theorem:

Theorem 4.16 *In each time t RandomizedUCB makes $O(t^5K^4\ln^2(\frac{tK}{\delta}))$ calls to $\mathcal{AMO}$, and requires additional $O(t^2K^2)$ processing time. The total running time at each time t is $O(t^5K^4\ln^2(\frac{tK}{\delta})\ln N)$, which is sub-linear.*

Algorithm 9 RandomizedUCB

```
Define $W_0 = \{\}$, $C_t = 2\ln(\frac{Nt}{\delta})$, $\mu_t = \min\left\{\frac{1}{2K}, \sqrt{\frac{C_t}{2Kt}}\right\}$
for t=1,..., T do
    Solve the following optimization problem to get distribution $P_t$ over $\mathcal{H}$
        $$\min_P \sum_{h\in\mathcal{H}} P(h)\Delta_{t-1}(h)$$
      s.t. for all distribution $Q$ over $\mathcal{H}$:
        $$\mathrm{E}_{h\sim Q}\left[\frac{1}{t-1}\sum_{i=1}^{t-1}\frac{1}{W'_{P_t,\mu_t}(x,h(x))}\right] \le \max\left\{4K, \frac{(t-1)\Delta_{t-1}(W_Q)^2}{180C_{t-1}}\right\} \qquad (12)$$
    Sample $a_t$ from $W'_t = W'_{P_t,\mu_t}(x_t,\cdot)$
    Receive reward $r_{a_t}$
    $W_t = W_{t-1} \cup (x_t, a_t, r_{a_t}, W'_t(a_t))$
end for
```

4.3.3 ILOVETOCONBANDITS

(need more details)
Similar to RandomizedUCB, Importance-weighted LOw-Variance Epoch-Timed Oracleized CONtextual BANDITS algorithm (ILOVETOCONBANDITS) proposed by Agarwal et al. (2014) aims to run in time sub-linear with respect to N (total number of policies) and achieves optimal regret bound $O(\sqrt{KT\ln N})$. RandomizedUCB makes $O(T^6)$ calls to $\mathcal{AMO}$ over all T steps, and ILOVETOCONBANDITS tries to further reduce this time complexity.

Theorem 4.17 *ILOVETOCONBANDITS achieves optimal regret bound, requiring* $\tilde{O}(\sqrt{\frac{KT}{\ln(N/\delta)}})$ *calls to* $\mathcal{AMO}$ *over* T *rounds, with probability at least* $1-\delta$.

Let $\mathcal{A}$ be a finite set of K actions, $x \in \mathcal{X}$ be a possible contexts, and $r \in [0,1]^K$ be the reward vector of arms in $\mathcal{A}$. We assume (x, r) follows a distribution $\mathcal{D}$. Let Π be a finite set of policies that map contexts x to actions $a \in \mathcal{A}$, let Q be a distribution over all policies Π, and Δ^{Π} be the set of all possible Q. ILOVETOCONBANDITS is described in Algorithm 10. The $Sample(x_t, Q_{m-1}, \pi_{\tau_m-1}, \mu_{m-1})$ function is described in Algorithm 11, it samples an action from a sparse distribution over policies.

As we can see, the main procedure of ILOVETOCONBANDITS is simple. It solves an optimization problem on pre-specified rounds $\tau_1, \tau_2, ...$ to get a sparse distribution Q over all policies, then it samples an action based on this distribution. The main problem now is to choose an sparse distribution Q that achieves low regret and requires calls to $\mathcal{AMO}$ as little as possible.

Let $\hat{R}_t(\pi)$ be the unbiased reward estimator of policy π over the first t rounds (see section 2), and let $\pi_t = \arg\max_\pi \hat{R}_t(\pi)$, then the estimated empirical regret of π is $\widehat{Reg}_t(\pi) = \hat{R}_t(\pi_t) - \hat{R}(\pi)$. Given a history H_t and minimum probability μ_m, and define $b_\pi = \frac{\widehat{Reg}_t(\pi)}{\psi\mu_m}$

Algorithm 10 ILOVETOCONBANDITS

```
Require: Epoch schedule $0 = \tau_0 < \tau_1 < \tau_2 < ...$, $\delta \in (0, 1)$
  Initial weights $Q_0 = 0$, $m = 1$, $\mu_m = \min\{\frac{1}{2K}, \sqrt{\ln(16\tau_m^2|\Pi|/\delta)/(K\tau_m)}\}$
  for $t = 1, 2, ..., T$ do
      $(a_t, p_t(a_t)) = Sample(x_t, Q_{m-1}, \pi_{\tau_m - 1}, \mu_{m-1})$
      Pull arm $a_t$ and receive reward $r_t \in [0, 1]$
      if $t = \tau_m$ then
          Let $Q_m$ be a solution to (OP) with history $H_t$ and minimum probability $\mu_m$
          $m = m + 1$
      end if
  end for
```

Algorithm 11 Sample

```
Require: $x, Q, \mu$
  for $\pi \in \Pi$ and $Q(\pi) > 0$ do
      $p_{\pi(x)} = (1 - K\mu)Q(\pi) + \mu$
  end for
  Randomly draw action $a$ from $p$
  return $(a, p_a)$
```

for $\psi = 100$, then the optimization problem is to find a distribution $Q \in \Delta^{\Pi}$ such that

$$\sum_{\pi \in \Pi} Q(\pi) b_\pi \leq 2K \tag{13}$$

$$\forall \pi \in \Pi : \mathrm{E}_{x \sim H_t} \left[\frac{1}{Q^{\mu_m}(\pi(x)|x)} \leq 2K + b_\pi \right] \tag{14}$$

where Q^{μ_m} is the smoothed version of Q with minimum probability μ_m.

Note that b_π is a scaled version of empirical regret of π, so Equation (13) is actually a bound on the expected empirical regret with respect to Q. This equation can be treated as the exploitation since we want to choose a distribution that has low empirical regret. Equation (14), similar to RandomizedUCB, is a bound on the variance of the reward estimator of each policy $\pi \in \Pi$. If the policy has low empirical regret, we want it to have smaller variance so that the reward estimator is more accurate, on the other hand, if the policy has high empirical regret, then we allow it to have a larger variance.

Agarwal et al. (2014) showed that this optimization problem can be solved via coordinate descent with at most $\tilde{O}(\sqrt{Kt/\ln(N/\delta)})$ calls to $\mathcal{AMO}$ in round t, moreover, the support (non-zeros) of the resulting distribution Q at time t is at most $\tilde{O}(\sqrt{Kt/\ln(N/\delta)})$ policies, which is the same as the number of calls to $\mathcal{AMO}$. This results sparse Q and hence sub-linear time complexity for *Sample* procedure.

Agarwal et al. (2014) also showed that the requirement of τ is that $\tau_{m+1} - \tau_m = O(\tau_m)$. So we can set $\tau_m = 2^{m-1}$, then the total number of calls to $\mathcal{AMO}$ over all T round is only $\tilde{O}(\sqrt{Kt/\ln(N/\delta)})$, which is a vast improvement over RandomizedUCB.

Theorem 4.18 *With probability at least $1-\delta$, the regret of ILOVETOCONBANDITS is*

$$O(\sqrt{KT\ln(TN/\delta)} + K\ln(TN/\delta))$$

5. Adversarial Contextual Bandits

In adversarial contextual bandits, the reward of each arm does not necessarily follow a fixed probability distribution, and it can be picked by an adversary against the agent. One way to solve adversarial contextual bandits problem is to model it with expert advice. In this method, there are N experts, and at each time t each expert gives advice about which arm to pull based on the contexts. The agent has its own strategy to pick an arm based on all the advice it gets. Upon receiving the reward of that arm, the agent may adjust its strategy such as changing the weight or believe of each expert.

5.1 EXP4

Exponential-weight Algorithm for Exploration and Exploitation using Expert advice (EXP4) Auer et al. (2002b) assumes each expert generates an advice vector based on the current context x_t at time t. Advice vectors are distributions over arms, and are denoted by $\xi_t^1, \xi_t^2, ..., \xi_t^N \in [0,1]^K$. $\xi_{t,j}^i$ indicates expert i's recommended probability of playing arm j at time t. The algorithm pulls an arm based on these advice vectors. Let $r_t \in [0,1]^K$ be the true reward vector at time t, then the expected reward of expert i is $\xi_t^i \cdot r_t$. The algorithm competes with the best expert, which achieves the highest expected cumulative reward

$$G_{max} = \max_i \sum_{t=1}^{T} \xi_t^i \cdot r_t$$

The regret is defined as:

$$R_T = \max_i \sum_{t=1}^{T} \xi_t^i \cdot r_t - \mathrm{E}\sum_{t=1}^{T} r_{t,a_t}$$

The expectation is with respect to the algorithm's random choice of the arm and any other random variable in the algorithm. Note that we don't have any assumption on the distribution of the reward, so EXP4 is a adversarial bandits algorithm.

EXP4 algorithm is described in Algorithm 12. Note that the context x_t does not appear in the algorithm, since it is only used by experts to generate advice.

If an expert assigns uniform weight to all actions in each time t, then we call the expert a uniform expert.

Theorem 5.1 *For any family of experts which includes a uniform expert, EXP4's regret is bounded by* $O(\sqrt{TK\ln N})$.

Proof The general idea of the proof is to bound the expected cumulative reward $\mathrm{E}\sum_{t=1}^{T} r_{t,a_t}$, then since G_{max} is bounded by the time horizon T, we can get a bound on $G_{max} - \mathrm{E}\sum_{t=1}^{T} r_{t,a_t}$.

Algorithm 12 EXP4

```
Require: $\gamma \in (0, 1]$
  Set $w_{t,i} = 1$ for $i = 1, ..., N$
  for $t = 1, 2, ..., T$ do
      Get expert advice vectors $\{\xi_t^1, ..., \xi_t^N\}$, each vector is a distribution over arms.
      for $j = 1, ..., K$ do
          $$p_{t,j} = (1 - \gamma) \sum_{i=1}^{N} \frac{w_{t,i}\xi_{t,j}^i}{\sum_{i=1}^{N} w_{t,i}} + \frac{\gamma}{K}$$
      end for
      Draw action $a_t$ according to $p_t$, and receive reward $r_{a_t}$.
      for $j = 1, ..., K$ do    ▷ Calculate unbiased estimator of $r_t$
          $$\hat{r}_{t,j} = \frac{r_{t,j}}{p_{t,j}} \mathbb{I}(j = a_t)$$
      end for
      for $i = 1, ..., N$ do    ▷ Calculate estimated expected reward and update weight
          $$\hat{y}_{t,i} = \xi_t^i \cdot \hat{r}_t$$
          $$w_{t+1,i} = w_t \exp(\gamma \hat{y}_{t,i}/K)$$
      end for
  end for
```

Let $W_t = \sum_{i=1}^{N} w_{t,i}$, and $q_{t,i} = \frac{w_{t,i}}{W_t}$, then

$$
\begin{aligned}
\frac{W_{t+1}}{W_t} &= \sum_{i=1}^{N} \frac{w_{t+1,i}}{W_t} \\
&= \sum_{i=1}^{N} q_{t,i} \exp(\gamma \hat{y}_{t,i}/K) \\
&\leq \sum_{i=1}^{N} q_{t,i} \left[1 + \frac{\gamma}{K}\hat{y}_{t,i} + (e-2)(\frac{\gamma}{K}\hat{y}_{t,i})^2\right] && (15) \\
&\leq 1 + \frac{\gamma}{K}\sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i} + (e-2)\left(\frac{\gamma}{K}\right)^2 \sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i}^2 \\
&\leq \exp\left(\frac{\gamma}{K}\sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i} + (e-2)\left(\frac{\gamma}{K}\right)^2 \sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i}^2\right) && (16)
\end{aligned}
$$

Equation 15 is due to $e^x \leq 1 + x + (e-2)x^2$ for $x \leq 1$, Equation 16 is due to $1 + x \leq e^x$. Taking logarithms and summing over t

$$
\ln \frac{W_{T+1}}{W_1} \leq \frac{\gamma}{K} \sum_{t=1}^{T}\sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i} + (e-2)\left(\frac{\gamma}{K}\right)^2 \sum_{t=1}^{T}\sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i}^2 \qquad (17)
$$

For any expert k

$$
\begin{aligned}
\ln \frac{W_{T+1}}{W_1} &\geq \ln \frac{w_{T+1,k}}{W_1} \\
&= \ln w_{1,k} + \sum_{t=1}^{T} (\frac{\gamma}{K}\hat{y}_{t,i}) - \ln W_1 \\
&= \frac{\gamma}{K} \sum_{t=1}^{T} \hat{y}_{t,i} - \ln N
\end{aligned}
$$

Together with Equation 17 we get

$$
\sum_{t=1}^{T}\sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i} \geq \sum_{t=1}^{T} \hat{y}_{t,k} - \frac{K \ln N}{\gamma} - (e-2)\frac{r}{K} \sum_{t=1}^{T}\sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i}^2 \qquad (18)
$$

Now we need to bound $\sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i}$ and $\sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i}^2$. From the definition of $p_{t,i}$ we have $\sum_{i=1}^{N} q_{t,i}\xi_{t,j}^i = \frac{p_{t,j}-\gamma/K}{1-\gamma}$, so

$$\begin{aligned}
\sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i} &= \sum_{i=1}^{N} q_{t,i}\left(\sum_{j=1}^{K} \xi_{t,j}^i \hat{r}_{t,j}\right) \\
&= \sum_{j=1}^{K}\left(\sum_{i=1}^{N} q_{t,i}\xi_{t,j}^i\right)\hat{r}_{t,j} \\
&= \sum_{j=1}^{K}\left(\frac{p_{t,j}-\gamma/K}{1-\gamma}\right)\hat{r}_{t,j} \\
&\leq \frac{r_{t,a_t}}{1-\gamma}
\end{aligned}$$

where a_t is the arm pulled at time t.

$$\begin{aligned}
\sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i}^2 &= \sum_{i=1}^{N} q_{t,i}(\xi_{t,a_t}^i \hat{r}_{t,a_t})^2 \\
&\leq \sum_{i=1}^{N} q_{t,i}(\xi_{t,a_t}^i)^2 \hat{r}_{t,a_t}^2 \\
&\leq \sum_{i=1}^{N} q_{t,i}\xi_{t,a_t}^i \hat{r}_{t,a_t}^2 \\
&\leq \hat{r}_{t,a_t}^2 \frac{p_{t,a_t}}{1-\gamma} \\
&\leq \frac{\hat{r}_{t,a_t}}{1-\gamma}
\end{aligned}$$

Together with Equation 18 we have

$$\begin{aligned}
\sum_{t=1}^{T} r_{t,a_t} &\geq (1-\gamma)\sum_{t=1}^{T}\hat{y}_{t,k} - \frac{K\ln N}{\gamma}(1-\gamma) - (e-2)\frac{\gamma}{K}\sum_{t=1}^{T}\sum_{j=1}^{K}\hat{r}_{t,j} \\
&\geq (1-\gamma)\sum_{t=1}^{T}\hat{y}_{t,k} - \frac{K\ln N}{\gamma} - (e-2)\frac{\gamma}{K}\sum_{t=1}^{T}\sum_{j=1}^{K}\hat{r}_{t,j}
\end{aligned}$$

Taking expectation of both sides of the inequality we get

$$\begin{aligned}
\mathrm{E}\sum_{t=1}^{T} r_{t,a_t} &\geq (1-\gamma)\sum_{t=1}^{T} y_{t,k} - \frac{K\ln N}{\gamma} - (e-2)\frac{\gamma}{K}\sum_{t=1}^{T}\sum_{j=1}^{K} r_{t,j} \\
&\geq (1-\gamma)\sum_{t=1}^{T} y_{t,k} - \frac{K\ln N}{\gamma} - (e-2)\gamma\sum_{t=1}^{T}\frac{1}{K}\sum_{j=1}^{K} r_{t,j} \qquad (19)
\end{aligned}$$

Since there is a uniform expert in the expert set, so $G_{max} \geq \sum_{t=1}^{T} \frac{1}{K} \sum_{j=1}^{K} r_{t,j}$. Let $G_{exp4} = \sum_{t=1}^{T} r_{t,a_t}$, then Equation 19 can be rewritten as

$$\mathrm{E}G_{exp4} \geq (1-\gamma)\sum_{t=1}^{T} y_{t,k} - \frac{K \ln N}{\gamma} - (e-2)\gamma G_{max}$$

For any k. Let k be the arm with the highest expected reward, then we have

$$\begin{aligned} \mathrm{E}G_{exp4} &\geq (1-\gamma)G_{max} - \frac{K \ln N}{\gamma} - (e-2)\gamma G_{max} \\ G_{max} - \mathrm{E}G_{exp4} &\leq \frac{K \ln N}{\gamma} + (e-1)\gamma G_{max} \end{aligned}$$

We need to select a γ such that the right-hand side of the above inequality is minimized so that the regret bound is minimized. An additional constrain is that $\gamma \leq 1$. Taking the derivative with respect to γ and setting to 0, we get

$$\begin{aligned} \gamma^* &= \min\left\{1, \sqrt{\frac{K \ln N}{(e-1)G_{max}}}\right\} \\ G_{max} - \mathrm{E}G_{exp4} &\leq 2.63\sqrt{G_{max} K \ln N} \end{aligned}$$

Since $G_{max} \leq T$, we have $R_T = O(\sqrt{TK \ln N})$. One important thing to notice is that to get such regret bound it requires the knowledge of T, the time horizon. Later we will introduce algorithms that does not require such knowledge. ■

5.2 EXP4.P

The unbiased estimator of the reward vector used by EXP4 has high variance due to the increased range of the random variable r_{a_t}/p_{a_t} (Dudík et al., 2014), and the regret bound of EXP4, $O(\sqrt{TK \ln N})$, is hold only with expectation. EXP4.P (Beygelzimer et al., 2011) improves this result and achieves the same regret with high probability. To do this, EXP4.P combines the idea of both UCB (Auer et al., 2002a) and EXP4. It computes the confidence interval of the reward vector estimator and hence bound the cumulative reward of each expert with high probability, then it designs an strategy to weight each expert.

Similar to the EXP4 algorithm setting, there are K arms $\{1, 2, ..., K\}$ and N experts $\{1, 2, ..., N\}$. At time $t \in \{1, ..., T\}$, the world reveals context x_t, and each expert i outputs an advice vector ξ_t^i representing its recommendations on each arm. The agent then selects an arm a_t based on the advice, and an adversary chooses a reward vector r_t. Finally the world reveals the reward of the chosen arm r_{t,a_t}. Let G_i be the expected cumulative reward of expert i:

$$G_i = \sum_{t=1}^{T} \xi_t^i \cdot r$$

let p_j be the algorithm's probability of pulling arm j, and let $\hat{r}$ be the estimated reward vector, where

$$\hat{r}_j = \begin{cases} r_j/p_j & \text{if} \quad j = a_t \\ 0 & \text{if} \quad j \neq a_t \end{cases}$$

let $\hat{G}_i$ be the estimated expected cumulative reward of expert i:

$$\hat{G}_i = \sum_{t=1}^{T} \xi_t^i \cdot \hat{r}$$

let $G_{exp4.p}$ be the estimated cumulative reward of the algorithm:

$$G_{exp4.p} = \sum_{t=1}^{T} r_{a_t}$$

then the expected regret of the algorithm is

$$R_T = \max_i G_i - \mathrm{E} G_{exp4.p}$$

However, we are interested in regret bound which hold with arbitrarily high probability. The regret is bounded by ϵ with probability $1 - \delta$ if

$$P\left(\left(\max_i G_i - G_{exp4.p}\right) > \epsilon\right) \leq \delta$$

We need to bound $G_i - \hat{G}_i$ with high probability so that we can bound the regret with high probability. To do that, we need to use the following theorem:

Theorem 5.2 *Let $X_1, ..., X_T$ be a sequence of real-valued random variables. Suppose that $X_t \leq R$ and $\mathrm{E}(X_t) = 0$. Define the random variables*

$$S = \sum_{t=1}^{T} X_t, \quad V = \sum_{t=1}^{T} \mathrm{E}(X_t^2)$$

then for any δ, with probability $1 - \delta$, we have

$$S \leq \begin{cases} \sqrt{(e-2)\ln(1/\delta)}\left(\frac{V}{\sqrt{V'}} + \sqrt{V'}\right) & \text{if} \quad V' \in \left[\frac{R^2 \ln(1/\delta)}{e-2}, \infty\right) \\ R\ln(1/\delta) + (e-2)\frac{V}{R} & \text{if} \quad V' \in \left[0, \frac{R^2 \ln(1/\delta)}{e-2}\right] \end{cases}$$

To bound $\hat{G}_i$, let $X_t = \xi_t^i \cdot r_t - \xi_t^i \cdot \hat{r}_t$, so $\mathrm{E}(X_t) = 0$, $R = 1$ and

$$\begin{aligned} \mathrm{E}(X_t^2) &\leq \mathrm{E}(\xi_t^i \cdot \hat{r}_t)^2 \\ &= \sum_{j=1}^{K} p_{t,j}\left(\xi_{t,j}^i \cdot \frac{r_{t,j}}{p_{t,j}}\right)^2 \\ &\leq \sum_{j=1}^{K} \frac{\xi_{t,j}^i}{p_{t,j}} \\ &\overset{\text{def}}{=} \hat{v}_{t,i} \end{aligned}$$

The above proof used the fact that $r_{t,j} \leq 1$. Let $V' = KT$, assume $\ln(N/\delta) \leq KT$, and use δ/N instead of δ, we can apply Theorem 5.2 to get

$$P\left(G_i - \hat{G}_i \geq \sqrt{(e-2)\ln\frac{N}{\delta}}\left(\frac{\sum_{t=1}^{T}\hat{v}_{t,i}}{\sqrt{KT}} + \sqrt{KT}\right)\right) \leq \frac{\delta}{N}$$

Apply union bound we get:

Theorem 5.3 *Assume* $\ln(N/\delta) \leq KT$, *and define* $\hat{\sigma}_i = \sqrt{KT} + \frac{1}{\sqrt{KT}}\sum_{t=1}^{T}\hat{v}_{t,i}$, *we have that with probability* $1-\delta$

$$\sup_i(G_i - \hat{G}_i) \leq \sqrt{\ln\frac{N}{\delta}}\hat{\sigma}_i$$

The confidence interval we get from Theorem 5.3 is used to construct EXP4.P algorithm. The detail of the algorithm is described in Algorithm 13. We can see that EXP4.P is very similar to EXP4 algorithm, except that when updating $w_{t,i}$, instead of using estimated reward, we use the upper confidence bound of the estimated reward.

Theorem 5.4 *Assume that* $\ln(N/\delta) \leq KT$, *and the set of experts includes a uniform expert which selects an arm uniformly at randomly at each time. Then with probability* $1-\delta$

$$R_T = \max_i G_i - G_{exp4.p} \leq 6\sqrt{KT\ln(N/\delta)}$$

Proof The proof is similar to the proof of regret bound of EXP4. Basically, we want to bound $G_{exp4.p} = \sum_{t=1}^{T} r_{a_t}$, and since we can bound $\max_i G_i$ with high probability, we then get the regret of EXP4.P with high probability.
Let $q_{t,i} = \frac{w_{t,i}}{\sum_i w_{t,i}}$, $\gamma = \sqrt{\frac{K\ln N}{T}}$, and $\hat{U} = \max_i(\hat{G}_i + \hat{\sigma}_i\sqrt{\ln(N/\delta)})$. We need the following inequalities

$$\hat{v}_{t,i} \leq 1/p_{min}$$
$$\sum_{i=1}^{N} q_{t,i}\hat{v}_{t,i} \leq \frac{K}{1-\gamma}$$

To see why this is true:

$$\begin{aligned}\sum_{i=1}^{N} q_{t,i}\hat{v}_{t,i} &= \sum_{i=1}^{N} q_{t,i}\sum_{j=1}^{K}\frac{\xi_{t,j}^{i}}{p_{t,j}}\\ &= \sum_{j=1}^{K}\frac{1}{p_{t,j}}\sum_{i=1}^{N} q_{t,i}\xi_{t,j}^{i}\\ &\leq \sum_{j=1}^{K}\frac{1}{1-\gamma}\\ &= \frac{K}{1-\gamma}\end{aligned}$$

Algorithm 13 EXP4.P

```
Require: $\delta > 0$
  Define $p_{min} = \sqrt{\frac{\ln N}{KT}}$, set $w_{1,i} = 1$ for $i = 1, ..., N$.
  for $t = 1, 2, ...T$ do
      Get expert advice vectors $\{\xi_t^1, \xi_t^2, ..., \xi_t^N\}$.
      for $j = 1, 2, ..., K$ do
          $$p_{t,j} = (1 - Kp_{min}) \sum_{i=1}^{N} \frac{w_{t,i}\xi_{t,j}^i}{\sum_{i=1}^{N} w_{t,i}} + p_{min}$$
      end for
      Draw action $a_t$ according to $p_t$ and receive reward $r_{a_t}$.
      for $j = 1, ..., K$ do
          $$\hat{r}_{t,j} = \frac{r_{t,j}}{p_{t,j}} \mathbb{I}(j = a_t)$$
      end for
      for $i = 1, ..., N$ do
          $$\hat{y}_{t,i} = \xi_t^i \cdot \hat{r}_t$$
          $$\hat{v}_{t,i} = \sum_{j=1}^{K} \xi_{t,j}^i / p_{t,j}$$
          $$w_{t+1,i} = w_{t,i} \exp\left(\frac{p_{min}}{2}\left(\hat{y}_{t,i} + \hat{v}_{t,i}\sqrt{\frac{\ln(N/\delta)}{KT}}\right)\right)$$
      end for
  end for
```

We also need the following two inequalities, which has been proved in Section 5.1.

$$\sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i} \leq \frac{r_{t,a_t}}{1-\gamma}$$
$$\sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i}^2 \leq \frac{\hat{r}_{t,a_t}}{1-\gamma}$$

Let $b = \frac{p_{min}}{2}$ and $c = \frac{p_{min}\sqrt{\ln(N/\delta)}}{2\sqrt{KT}}$, then

$$\begin{aligned}\frac{W_{t+1}}{W_t} &= \sum_{i=1}^{N} \frac{w_{t+1,i}}{W_t} \\ &= \sum_{i=1}^{N} q_{t,i}\exp(b\hat{y}_{t,i} + c\hat{v}_{t,i})\end{aligned}$$

Since $e^a \leq 1 + a + (e-2)a^2$ for $a \leq 1$ and $e - 2 \leq 1$, we have

$$\begin{aligned}\frac{W_{t+1}}{W_t} &\leq \sum_{i=1}^{N} q_{t,i}(1 + b\hat{y}_{t,i} + c\hat{v}_{t,i}) + \sum_{i=1}^{N} q_{t,i}(2b^2\hat{y}_{t,i}^2 + 2c^2\hat{v}_{t,i}^2) \\ &= 1 + b\sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i} + c\sum_{i=1}^{N} q_{t,i}\hat{v}_{t,i} + 2b^2\sum_{i=1}^{N} q_{t,i}\hat{y}_{t,i}^2 + 2c^2\sum_{i=1}^{N} q_{t,i}\hat{v}_{t,i}^2 \\ &\leq 1 + b\frac{r_{t,a_t}}{1-\gamma} + c\frac{K}{1-\gamma} + 2b^2\frac{\hat{r}_{t,a_t}}{1-\gamma} + 2c^2\sqrt{\frac{KT}{\ln N}}\frac{K}{1-\gamma}\end{aligned}$$

Take logarithms on both side, sum over T and make use of the fact that $\ln(1+x) \leq x$ we have

$$\ln\left(\frac{W_{T+1}}{W_1}\right) \leq \frac{b}{1-\gamma}\sum_{t=1}^{T} r_{t,a_t} + c\frac{KT}{1-\gamma} + \frac{2b^2}{1-\gamma}\sum_{t=1}^{T}\hat{r}_{t,a_t} + 2c^2\sqrt{\frac{KT}{\ln N}}\frac{KT}{1-\gamma}$$

Let $\hat{G}_{uniform}$ be the estimated cumulative reward of the uniform expert, then

$$\begin{aligned}\hat{G}_{uniform} &= \sum_{t=1}^{T}\sum_{j=1}^{K}\frac{1}{K}\hat{r}_j \\ &= \sum_{t=1}^{T}\frac{1}{K}\hat{r}_{t,a_t}\end{aligned}$$

So

$$\begin{aligned}\ln\left(\frac{W_{T+1}}{W_1}\right) &\leq \frac{b}{1-\gamma}\sum_{t=1}^{T} r_{t,a_t} + c\frac{KT}{1-\gamma} + \frac{2b^2}{1-\gamma}\sum_{t=1}^{T} K\hat{G}_{uniform} + 2c^2\sqrt{\frac{KT}{\ln N}}\frac{KT}{1-\gamma} \\ &\leq \frac{b}{1-\gamma}\sum_{t=1}^{T} r_{t,a_t} + c\frac{KT}{1-\gamma} + \frac{2b^2}{1-\gamma}\sum_{t=1}^{T} K\hat{U} + 2c^2\sqrt{\frac{KT}{\ln N}}\frac{KT}{1-\gamma}\end{aligned}$$

Also

$$\begin{aligned}\ln(W_{T+1}) &\geq \max_i(\ln w_{T+1,i}) \\ &= \max_i \left(b\hat{G}_i + c\sum_{t=1}^{T} \hat{v}_{t,i} \right) \\ &= b\hat{U} - b\sqrt{KT\ln(N/\delta)}\end{aligned}$$

So

$$b\hat{U} - b\sqrt{KT\ln(N/\delta)} - \ln N \leq \frac{b}{1-\gamma}G_{exp4.p} + c\frac{KT}{1-\gamma} + \frac{2b^2}{1-\gamma}\sum_{t=1}^{T} K\hat{U} + 2c^2\sqrt{\frac{KT}{\ln N}}\frac{KT}{1-\gamma}$$

$$G_{exp4.p} \geq \left(1 - 2\sqrt{\frac{K\ln N}{T}}\right)\hat{U} - \ln(N/\delta) - 2\sqrt{KT\ln N} - \sqrt{KT\ln(N/\delta)}$$

We already know from Theorem 5.2 that $\max_i G_i \leq \hat{U}$ with probability $1-\delta$, and also $\max_i G_i \leq T$, so with probability $1-\delta$

$$\begin{aligned}G_{exp4.p} &\geq \max_i G_i - 2\sqrt{\frac{K\ln N}{T}}T - \ln(N/\delta) - \sqrt{KT\ln N} - 2\sqrt{KT\ln(N/\delta)} \\ &\geq \max_i G_i - 6\sqrt{KT\ln(N/\delta)}\end{aligned}$$

■

5.3 Infinite Many Experts

Sometimes we have infinite number of experts in the expert set Π. For example, an expert could be a d-dimensional vector $\beta \in \mathrm{R}^d$, and the predictive reward could be $\beta^\top x$ for some context x. Neither EXP4 nor EXP4.P are able to handle infinite experts.

A possible solution is to construct a finite approximation $\hat{\Pi}$ to Π, and then use EXP4 or EXP4.P on $\hat{\Pi}$ (Bartlett, 2014; Beygelzimer et al., 2011). Suppose for every expert $\pi \in \Pi$ there is a $\hat{\pi} \in \hat{\Pi}$ with

$$P(\pi(x_t) \neq \hat{\pi}(x_t)) \leq \epsilon$$

where x_t is the context and $\pi(x_t)$ is the chosen arm. Then the reward $r \in [0,1]$ satisfy

$$\mathrm{E}\left|r_{\pi(x_t)} - r_{\hat{\pi}(x_t)}\right| \leq \epsilon$$

We compete with the best expert in Π, the regret is

$$R_T(\Pi) = \sup_{\pi\in\Pi} \mathrm{E}\sum_{t=1}^{T} r_{\pi(x_t)} - \mathrm{E}\sum_{t=1}^{T} r_{a_t}$$

And we can bound $R_T(\Pi)$ with $R_T(\hat{\Pi})$:

$$\begin{aligned}
R_T(\Pi) &= \sup_{\pi \in \Pi} \mathrm{E} \sum_{t=1}^{T} r_{\pi(x_t)} - \sup_{\hat{\pi} \in \hat{\Pi}} \mathrm{E} \sum_{t=1}^{T} r_{\hat{\pi}(x_t)} + \sup_{\hat{\pi} \in \hat{\Pi}} \mathrm{E} \sum_{t=1}^{T} r_{\hat{\pi}(x_t)} - \mathrm{E} \sum_{t=1}^{T} r_{a_t} \\
&= \sup_{\pi \in \Pi} \inf_{\hat{\pi} \in \hat{\Pi}} \mathrm{E} \sum_{t=1}^{T} \left(r_{\pi(x_t)} - r_{\hat{\pi}(x_t)} \right) + \sup_{\hat{\pi} \in \hat{\Pi}} \mathrm{E} \sum_{t=1}^{T} r_{\hat{\pi}(x_t)} - \mathrm{E} \sum_{t=1}^{T} r_{a_t} \\
&\leq T\epsilon + R_T(\hat{\Pi})
\end{aligned}$$

There are many ways to construct such $\hat{\Pi}$. Here we talk about an algorithm called VE (Beygelzimer et al., 2011). The idea is to choose an arm uniformly at random for the first τ rounds, then we get τ contexts $x_1, ..., x_\tau$. Given an expert $\pi \in \Pi$, we can get a sequence of prediction $\{\pi(x_1), ..., \pi(x_\tau)\}$. Such sequence is enumerable, so we can construct $\hat{\Pi}$ containing one representative $\hat{\pi}$ for each sequence $\{\hat{\pi}(x_1), ..., \hat{\pi}(x_\tau)\}$. Then we apply EXP4/EXP4.P on $\hat{\Pi}$. VE is shown in Algorithm 14.

Algorithm 14 VE

```
Require: $\tau$
  for $t = 1, 2, ...\tau$ do
      Receive context $x_t$
      Choose arm uniformly at random
  end for
  Construct $\hat{\Pi}$ based on $x_1, ..., x_\tau$
  for $t = \tau + 1, ..., T$ do
      Apply EXP4/EXP4.P
  end for
```

Theorem 5.5 *For all policy sets Π with VC dimension d, $\tau = \sqrt{T\left(2d \ln \frac{eT}{d} + \ln \frac{2}{\delta}\right)}$, with probability $1 - \delta$*

$$R_T \leq 9\sqrt{2T\left(d \ln \frac{eT}{d} + \ln \frac{2}{\delta}\right)}$$

Proof Given $\pi \in \Pi$ and corresponding $\hat{\pi} \in \hat{\Pi}$

$$G_\pi = G_{\hat{\pi}} + \sum_{t=\tau+1}^{T} \mathbb{I}(\pi(x_t) \neq \hat{\pi}(x_t)) \tag{20}$$

We need to measure the expected disagreements of π and $\hat{\pi}$ after time τ. Suppose the total disagreements within time T is n, then if we randomly pick τ contexts, the probability that π and $\hat{\pi}$ produce the same sequence is

$$\begin{aligned}
P\left(\forall t \in [1, \tau], \pi(x_t) = \hat{\pi}(x_t)\right) &= \left(1 - \frac{n}{T}\right)\left(1 - \frac{n}{T-1}\right) ... \left(1 - \frac{n}{T - \tau + 1}\right) \\
&\leq \left(1 - \frac{n}{T}\right)^\tau \\
&\leq e^{-\frac{n\tau}{T}}
\end{aligned}$$

From Sauer's lemma we have that $|\hat{\Pi}| \leq (\frac{e\tau}{d})^d$ for all $\tau > d$ and the number of unique sequences produced by all $\pi \in \Pi$ is less than $(\frac{e\tau}{d})^d$ for all $\tau > d$. For a $\pi \in \Pi$ and corresponding $\hat{\pi} \in \hat{\Pi}$, we have

$$
\begin{aligned}
& P\left(\sum_{t=\tau+1}^{T} \mathbb{I}(\pi(x_t) \neq \hat{\pi}(x_t)) > n\right) \\
& \leq P\left(\exists \pi', \pi'' : \sum_{t=\tau+1}^{T} \mathbb{I}(\pi'(x_t) \neq \pi''(x_t)) > n \text{ and } \forall t \in [1, \tau], \pi'(x_t) = \pi''(x_t)\right) \\
& \leq |\Pi|^2 e^{-\frac{n\tau}{T}} \\
& \leq \left(\frac{e\tau}{d}\right)^{2d} e^{-\frac{n\tau}{T}}
\end{aligned}
$$

Set the right-hand side to $\frac{\delta}{2}$ and we get:

$$
n \geq \frac{T}{\tau}\left(2d \ln \frac{eT}{d} + \ln \frac{2}{\delta}\right)
$$

Together with Equation (20), we get with probability $1 - \frac{\delta}{2}$

$$
G_{\max(\hat{\Pi})} \geq G_{\max(\Pi)} - \frac{T}{\tau}\left(2d \ln \frac{eT}{d} + \ln \frac{2}{\delta}\right)
$$

Now we need to bound $G_{\max(\hat{\Pi})}$. From Sauer's lemma we have that $|\hat{\Pi}| \leq (\frac{e\tau}{d})^d$ for all $\tau > d$, so we can directly apply EXP4.P's bound. With probability $1 - \frac{\delta}{2}$

$$
G_{exp4.p}(\hat{\Pi}, T - \tau) \geq G_{\max(\hat{\Pi})} - 6\sqrt{2(T-\tau)(d \ln(\frac{e\tau}{d}) + \ln(\frac{2}{\delta}))}
$$

Finally, we get the bound on G_{VE}

$$
G_{VE} \geq G_{\max(\Pi)} - \tau - \frac{T}{\tau}\left(2d \ln \frac{eT}{d} + \ln \frac{2}{\delta}\right) - 6\sqrt{2(T-\tau)(d \ln(\frac{e\tau}{d}) + \ln(\frac{2}{\delta}))}
$$

Setting $\tau = \sqrt{T(2d \ln \frac{eT}{d} + \ln \frac{2}{\delta})}$ we get

$$
G_{VE} \geq G_{\max(\Pi)} - 9\sqrt{2T(d \ln \frac{eT}{d} + \ln \frac{2}{\delta})}
$$

■

6. Conclusion

The nature of contextual bandits makes it suitable for many machine learning applications such as user modeling, Internet advertising, search engine, experiments optimization etc., and there has been a growing interests in this area. One topic we haven't covered is the offline evaluation in contextual bandits. This is tricky since the policy evaluated is different from the policy that generating the data, so the arm proposed offline does not necessary match the one pulled online. Li et al. (2011) proposed an unbiased offline evaluation method assuming that the logging policy selects arm uniformly at random. Strehl et al. (2010) proposed an methods that will estimate the probability of the logging policy selecting each arm, and then adopt inverse propensity score(IPS) to evaluation new policy, Langford et al. (2011) proposed an method that combines the direct method and IPS to improve accuracy and reduce variance.

Finally, note that regret bound is not the only criteria for bandits algorithm. First of all, the bounds we talked about in this survey are problem-independent bounds, and there are problem-dependent bounds. For example, Langford and Zhang (2008) proved that although the Epoch-Greedy's problem-independent bound is not optimal, it can achieve a $O(\ln T)$ problem-dependent bound; Second, different bandits algorithms have their own different assumptions (stochastic/adversarial, linearity, number of policies, Bayesian etc.), so when choosing which one to use, we need to choose the one matches our assumptions.

References

Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24*, pages 2312–2320, 2011.

Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1638–1646, 2014.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning*, pages 127–135, 2013a.

Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 99–107, 2013b.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *The Journal of Machine Learning Research*, 3:397–422, 2003.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002a.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002b.

Peter Bartlett. Learning in sequential decision problems. `http://www.stat.berkeley.edu/~bartlett/courses/2014fall-cs294stat260/`, 2014. Contextual bandits: Infinite comparison classes.

Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 19–26, 2011.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems 24*, pages 2249–2257, 2011.

Wei Chu, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, 2011.

Miroslav Dudík, Dumitru Erhan, John Langford, Lihong Li, et al. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.

Andreas Krause and Cheng S Ong. Contextual gaussian process bandit optimization. In *Advances in Neural Information Processing Systems*, pages 2447–2455, 2011.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems 20*, pages 817–824, 2008.

John Langford, Lihong Li, and Miroslav Dudík. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1097–1104, 2011.

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. ACM, 2010.

Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306. ACM, 2011.

Niranjan Srinivas, Andreas Krause, Matthias Seeger, and Sham M Kakade. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1015–1022, 2010.

Alex Strehl, John Langford, Lihong Li, and Sham M Kakade. Learning from logged implicit exploration data. In *Advances in Neural Information Processing Systems*, pages 2217–2225, 2010.

Michal Valko, Nathan Korda, Rémi Munos, Ilias Flaounas, and Nello Cristianini. Finite-time analysis of kernelised contextual bandits. In *The 29th Conference on Uncertainty in Artificial Intelligence*, 2013.