Knowledge Infused Policy Gradients with Upper Confidence Bound for Relational Bandits

Kaushik Roy,[1] Qi Zhang,[1] Manas Gaur, [1] Amit Sheth [1]
[1]Artificial Intelligence Institute, University of South Carolina, Columbia, SC, USA
kaushikr@email.sc.edu, qz5@cse.sc.edu, mgaur@email.sc.edu, amit@sc.edu

Abstract

Contextual Bandits find important use cases in various real life scenarios such as recommendation systems. However, most of the algorithms use flat feature vectors to represent context where as in the real world, there are varying number of objects and relations among them to model in the context . For example, in a music recommendation system the user context contains what music they listen, which artists create this music, the artist albums, etc. Adding richer relational context representations also introduces a much larger context space making exploration-exploitation harder. To improve the efficiency of exploration-exploitation *Knowledge* about the context can be infused to guide the exploration-exploitation strategy. Relational context representations allow a natural way for humans to specify *Knowledge* owing to their descriptive nature. We propose Knowledge Infused Policy Gradients Upper Confidence Bound algorithm and perform experimental analysis of various simulated settings where *Knowledge* can drastically reduce the total regret and where it cannot.

1 Introduction

Contextual Bandits (CB) are an extension of the classical Multi-Armed-Bandits (MAB) setting where there arm pull depends also on a specific context (Langford and Zhang 2007). As an example, in a music recommendation system, the choice of song recommendation depends on the user context (what type of user they are). In the real world, this context is multi-relational but most CB algorithms do not model multi-relational context and instead use flat feature vectors that contain attribute-value pairs (Zhou 2015). While relational modeling allows us to enrich user context, it further complicates the exploration-exploitation problem due to introduction of a much larger context space. Initially when much of the space of context-arm configurations are unexplored, aggressive exploitation may yield sub-optimal total regret. Hence, a principled exploration-exploitation strategy that encodes high uncertainty initially that tapers off with more information is required to effectively achieve near optimal total regret. The Upper-Confidence-Bound (UCB) algorithm uses an additional term to model initial uncertainty that tapers off during each arm pull (Lai and Robbins 1985). However, though the UCB provides a reasonable generalized heuristic, the exploration strategy can further be improved with more information about the distribution. For example, if it is known that the expected reward follows a Gaussian distribution. This is what Thompson Sampling does - incorporates a prior distribution over the expected rewards for each arm and updates a Bayesian posterior (Thompson 1933). If external *Knowledge* is available the posterior can be reshaped with *Knowledge Infusion* (Chapelle and Li 2011). A couple of issues arise with reshaping a) The choice of reshaping function is difficult to determine in a principled manner b) The form of the prior and posterior is usually chosen to exploit a likelihood-conjugate prior to analytically compute posterior estimates as sampling is typically inefficient. Similarly, the choice of reshaping function needs to either be amenable to efficient sampling for exploration or analytically computed. Thus, we observe that we can instead directly optimize for the optimal arm choice through policy gradient methods (Peters and Bagnell 2010). Using a Bayesian formulation for optimization of policy in functional space, we can see that the *Knowledge Infused* reshape function can be automatically learned by the *Knowledge Infused Policy Gradients* (KIPG) algorithm in functional space.

The CB setting presents a unique challenge for *Knowledge Infusion*. Since, arm pulling happens in online fashion, the human knowledge about the user is uncertain until the human observes some arm choices. Thus KIPG algorithm needs to be less aggressive in its *Knowledge Infusion* strategy when the human is still uncertain in their understanding of the user. For this reason, we develop a UCB style uncertainty measure that considers the initial uncertainty as the human gathers more information about the user context, before providing knowledge. Thus, we develop a *Knowledge Infused Policy Gradient Upper Confidence Bound* (KIPG-UCB) algorithm to incorporate human uncertainty in providing knowledge in the *Knowledge Infusion* strategy. Our contributions are as follows:

- We develop a novel CB algorithm KIPG-UCB that reduces regret through *Knowledge Infusion* with both high-quality and noisy knowledge.
- We observe that KIPG is fundamentally a gradient ascent method and derive a regret bound that depends on the knowledge. We also use Markov's inequality to de-

rive a confidence bound (UCB) for when the knowledge is noisy.

- Through simulated experiments we perform analysis of settings where KIPG-UCB achieves drastic reduction in total regret. We compare KIPG-UCB to KIPG without a confidence bound and RB2 (Kakadiya, Natarajan, and Ravindran 2020).

2 Problem Setting

We consider the problem setting of Bernoulli Contextual Bandits with relational features. Formally, at each step k, when an arm $i \in [N] := 1, 2, ..., N$ is pulled from among N arms, the reward $r_k(i) \in \{0, 1\}$ is Bernoulli. Also, pulling an arm i depends on a relational context $c_k(i)$. Since $\pi_k(i)$, which represents the probability of choosing arm i given context $c_k(i)$, is expected to be high if $P(r_k(i) = 1 | c(i))$ is high, we directly maximize the total reward over K arm choices, $\sum_{k=1}^{K} \pi_k(i) r_k(i)$. Here $\pi_k(i) = \sigma(\Psi_k(i))$, and σ is the sigmoid function. $\Psi_k(i)$ is a relational function that includes the relational context $c_k(i)$.

3 Knowledge Infused Policy Gradients

We now develop the KIPG formulation. We first describe policy gradients for CB, extend it to functional spaces and then use Bayes rule to derive the KIPG formulation. Then, we show the connection to Thompson Sampling with posterior reshaping. Finally, we show how the Exponential Weight (Exp3) algorithm which is also derived from a gradient ascent procedure (mirror ascent) can be seen as an instance of KIPG (Auer et al. 2002).

Policy Gradients for Contextual Bandits In policy gradient methods the probability of picking an arm i given context $c(i)$, is parameterized as $\pi(i) = \sigma(\theta(i)^T c(i))$. We want to maximize the expected reward over K arm pulls $\sum_{k=1}^{K} \pi_k(i) r_k(i)$. We update the parameters for arm i, at each $\hat{k} \in K$, using gradient ascent as $\theta_{k+1}(i) = \theta_k(i) + \eta \nabla_{\theta_k(i)} (\sum_{k=1}^{\hat{k}} \pi_k(i) r_k(i))$. Here we note that the gradient $\nabla_{\theta_k(i)} \pi_k(i) = \pi_k(i) \nabla_{\theta_k(i)} \log(\pi_k(i))$ and thus we optimize:

$$\theta_{k+1}(i) = \theta_k(i) + \eta (\sum_{k=1}^{\hat{k}} \pi_k(i) \nabla_{\theta_k(i)} \log(\pi_k(i)) r_k(i))$$

Policy Gradients for Bandits in Functional Space In functional space the $\theta(i)^T c(i)$ is replaced by a function $\Psi(i)$ i.e. $\pi(i) = \sigma(\Psi(i))$, where $\Psi(i)$ is a relational function that includes context $c(i)$. Thus, the policy gradient update becomes:

$$\Psi_k(i) = \Psi_k(i) + \eta (\sum_{k=1}^{\hat{k}} \pi_k(i) \nabla_{\Psi_k(i)} \log(\pi_k(i)) r_k(i))$$

Here, $\Psi_k(i)$ at each iteration of policy gradients is grown stage wise. We start with a $\Psi_0(i)$ and update $\Psi_K(i) = \Psi_0(i) + \sum_{k=1}^{K} \eta \delta_k(i)$, where each $\delta_k(i)$ fits a function to $\pi_k(i) \nabla_{\Psi_k(i)} \log(\pi_k(i)) r_k(i)$ (Kersting and Driessens 2008). In our experiments this function is a TILDE regression tree (Blockeel and De Raedt 1998). However, we derive a Bayesian formulation for $\pi_k(i)$ for knowledge infusion. Thus, After pulling an arm i at step k, and observing rewards $r_k(i)$, and context $c_k(i)$, using Bayes rule we can write

$$P(\Psi_k(i) | r_k(i)) = \frac{P(r_k(i) | \Psi_k(i)) P(\Psi_k(i))}{\int_{\Psi_k(i)} P(r_k(i) | \Psi_k(i)) P(\Psi_k(i))}$$

Using the sigmoid function we can set $P(r_k(i) | \Psi_k(i)) = \sigma(\Psi_k(i)) = \frac{e^{\Psi_k(i)}}{(1+e^{\Psi_k(i)})}$ and use the Bayesian posterior to obtain a prior informed policy as:

$$\pi_k(i) = \frac{\sigma(\Psi_k(i)) P(\Psi_k(i))}{\int_{\Psi_k(i)} \sigma(\Psi_k(i)) P(\Psi_k(i))}$$

To optimize using policy gradients, again we note that $\nabla_{\Psi_k(i)}(\pi_k(i)) = \pi_k(i) \nabla_{\Psi_k(i)} \log(\pi_k(i))$ If we use a form for $P(\Psi_k(i))$, for which the normalization doesn't depend on $\Psi_k(i)$ such as a Laplace or a Gaussian distribution, we can take the log on both sides without loss of generality to derive the gradient $\nabla_{\Psi_k(i)} \log(\pi_k(i))$:

$$\log(\pi_k(i)) \propto \log(\sigma(\Psi_k(i))) + \log(P(\Psi_k(i))),$$

taking the gradietn gives us:

$$(I_k(i) \quad \sigma(\Psi_k(i))) + \nabla_{\Psi_k(i)} \log(P(\Psi_k(i))),$$

where I is the indicator function representing if arm i was chosen at step k. Now we can employ functional gradient ascent by fitting a weak learner (such as a TILDE tree for relational context, or linear function for propostional context) to the gradient $\pi_k(i) \nabla_{\Psi_k(i)} \log(\pi_k(i))$. Note here that $\log(P(\Psi_k(i)))$ will determine the nature of knowledge infused into the policy gradient learning setup at each k. We call this approach Knowledge Infused Policy Gradients (KIPG).

Connection with Thompson Sampling We now formalize the connection between Thompson Sampling with posterior reshaping and KIPG. For arm $i \in [N]$, at every step of arm pulling $k \in [K]$, a reward $r_k(i)$ and a context $c_k(i)$ is emitted. In Thompson Sampling, The posterior $P(\Theta_k(i) | r_k(i), c_k(i))$ for parameter $\Theta_k(i)$ representing $P(r_k(i) | c_k(i))$ is updated at each step k as

$$\frac{P(r_k(i) | \Theta_k(i), c_k(i)) \Pr(\Theta_k(i) | c_k(i))}{\int_{\Theta_k(i)} P(r_k(i) | \Theta_k(i), c_k(i)) \Pr(\Theta_k(i) | c_k(i))}$$

Finally, the optimal arm choice corresponds to the arm that has the max among the sampled $\Theta_k(i)$ $P(\Theta_k(i) | r_k(i), c_k(i))$ for each arm i. The posterior $P(\Theta_k(i) | r_k(i), c_k(i))$, can be reshaped for example by using $P(\Theta_k(i) = \mathbf{F}(\Theta_k(i) | r_k(i), c_k(i))$. The reshaping changes the sufficient statistics such as mean, variance, etc. This $\mathbf{F}$ can be informed by some knowledge of the domain. We encounter a couple of issues with Posterior Reshaping for Knowledge Infusion. First, that the choice of $\mathbf{F}$

is difficult to determine in a principled manner. Second, the choice of $\mathbf{F}$ must be determined such that it is amenable to sampling for exploration. Sampling itself is very inefficient for problems of appreciable size. Thus, we observe that we can instead directly optimize for the optimal arm choice through policy gradient methods. Using a Bayesian formulation for optimization of policy in functional space, we can see that the reshaped posterior after K iterations of arm pulling (where K is sufficiently high), corresponds to learning an optimal function $\Psi(i)$ since $\Psi(i)$ is high if $\mathbf{F}(\Theta_k(i)|r_k(i), c_k(i))$, representing $P(r(i) = 1|c(i))$, is high.

Connection with Exp3 Exp3 maximizes the total expected reward over K arm pulls $f = \sum_{k=1}^{K} \pi_k(i) r_k(i)$. Using the proximal definition of gradient descent and deriving the mirror descent objective after each arm pull, we have:

$$\pi_k(i) = \arg\max_{\pi(i)}((\gamma\ \pi(i)\ \ r_{\pi_k\ \ 1(i)}(f)) + D(\pi_k\ \ 1(i), \pi(i))).$$

Choosing $D(\pi(i), \pi_k\ \ 1(i)) = \Phi(\pi_k\ \ 1(i))\quad (\Phi(\pi(i)) + r\,\Phi(\pi_k\ \ 1(i))(\pi_k\ \ 1(i)\quad \pi(i)))$, where Φ is a convex function, we get:

$$r\,\Phi(\pi_k(i)) = r\,\Phi(\pi_k(i)) + \gamma\quad r_{\pi_k\ \ 1(i)}(f)$$

Since π is a probability we need to choose a convex Φ such that it works with probability measures. So we will choose $\Phi(x) = \sum_i x(i) \log x(i)$ to be negative entropy and we have:

$$\log(\pi_k(i)) = \log(\pi_k\ \ 1(i)) + \gamma\quad r_{\pi_k\ \ 1(i)}(f),$$

setting $\pi_k\ \ 1(i) = \sigma(\Psi_k(i))$this can be seen as,

$$\log(\pi_k(i))\ /\ \log(\sigma(\Psi_k(i))) + \log(e^{\gamma\ r_{\pi_k\ \ 1(i)}(f)}),$$

where $\log P(\Psi_k(i)) = \log(e^{\gamma\ r_{\pi_k\ \ 1(i)}(f)})$. Thus we see that *Exp3 can be seen as a case of applying a specific prior probability in KIPG*.

4 Formulation of Knowledge Infusion

At each k, the prior over functions $\Psi_k(i)$ for each arm $P(\Psi_k(i))$ determines the *Knowledge Infusion* process. We now show the formulation for infusing arm preferences as knowledge as we use this in our experiments.

$P(\Psi_k(i)) = Normal(\mu, \Sigma)$: Given a context included in $\Psi_k(i)$, if we want to prefer the arm choice i, we can specify this knowledge using a two step procedure. First we set $\Psi_k(i)_{knowledge} = \alpha$, where $\alpha\quad 1$. Then we set $P(\Psi_k(i)) = Normal(\mu = \Psi_k(i)_{knowledge}\ \ \sigma(\Psi_k(i)), \Sigma = I)$. Similarly if the arm choice i is not preferred, $\Psi_k(i)_{knowledge} = \quad \alpha$. Here α controls how quickly *Knowledge Infusion* takes place.

$P(\Psi_k(i)) = Laplace(x, b)$: Specifying α is a tricky thing to do for a human and we would like them to able to just simply specify preference over arm choice given a context instead, if they are an expert. To model an expert

- First we set $\Psi_k(i)_{knowledge} = \text{LUB}\{\alpha g$, where $\text{LUB}\{\alpha g$ stands for the least upper bound from among a set of $\alpha\ \ 2\ \ \{\alpha g$. The interpretation is that α has to be at least that high to qualify as expert knowledge. We set $\text{LUB}\{\alpha g = K\quad \max \pi_k(i)\, r_{\Psi_k(i)} \log(\pi_k(i))\quad r_k(i) = K\quad 1\quad K = K^2$ as the maximum value of $\pi_k(i) = 1$ and the maximum value of $r_{\Psi_k(i)} \log(\pi_k(i))\quad r_k(i)$ is $1\quad K$ as the maximum value of $\sum_{k=1}^{K} r_k(i) = K$. Thus we set $\Psi_k(i)_{knowledge} = \text{LUB}\{\alpha g = K^2$. The interpretation is the human has to be at least as sure as the correction required to the error in arm choice i.e. the max gradient to qualify as an expert. Therefore to prefer arm i, $\alpha = K^2$ and if arm i is not preferred, $\alpha = \quad K^2$.
- Next, we replace the $Normal(\mu, \Sigma)$ distribution with the $Laplace(x = |\Psi_k(i)_{knowledge}\quad \Psi_k|, b = 1)$ distribution. Thus, we obtain that $r_{\Psi_k(i)} \log(P(\Psi_k(i)) = \text{sign}(\Psi_k(i)_{knowledge}\quad \sigma(\Psi_k(i))) = \quad 1$. If the expert prefers the arm i, $\delta_k(i) = \delta_k(i) + 1$ and if the expert does not prefer the arm i, $\delta_k(i) = \delta_k(i)\quad 1$. This is very intuitive as it means that the $\Psi_k(i)$, representing chance of arm i being pulled is simply increased or decreased by an additive factor depending on preference, thus preventing the need to carefully specify α.
- With this insight, if suffices for the human expert to specify knowledge as a tuple

  $$\mathbf{knowledge} : (\mathbf{c_k(i)}, \mathbf{prefer(i)} = \{0, 1g),$$

  which simply means that at step k, given the context $c_k(i)$, arm i is either preferred ($prefer(i) = 1$)) or not preferred ($prefer(i) = 0$). This is much more natural and easy for the expert human to specify. Note that if the human had a reason to specify α quantifying how quickly they want the *Knowledge Infusion* to take place depending on how sure they are (expert level), we can use the $Normal$ or $Laplace$ distribution form to specify without the use of $\text{LUB}\{\alpha g$. Algorithm 1 shows the psuedocode for KIPG with expert *Knowledge Infusion*. Also, we add 1 to $r_k(i)$ so that the gradient doesn't vanish when $r(i) = 0$.

Examples of knowledge in music recommendation An example of knowledge for music recommendation at a step k, can be to define a behavior over the user set $\mathbf{U} = \{user1, user2, user3g$ with respect to the artist set $\mathbf{A} = \{artist1, artist2g$ as, $(\mathbf{sungby}(\mathbf{S}, \mathbf{A}), \mathbf{prefer}(\mathbf{listens}(\mathbf{U}, \mathbf{S})) = \mathbf{1})$. This means that *The set of users* $\mathbf{U}$ *prefer listening to songs from artists in the set* $\mathbf{A}$.

Previous work on Relational Preferences: Odom et al., have previously specified relational preference knowledge in supervised learning and imitation learning settings (Odom et al. 2015). Using their approach, at step k, the knowledge would be incorporated by an additive term to the gradient term $(I_k(i)\quad \sigma(\Psi_k(i)))$. This term is $n_k(i)_t\quad n_k(i)_f$, where $n_k(i)_t$ is the number of knowledge sources that prefer arm i and $n_k(i)_f$ is the number of knowledge sources that do not

Algorithm 1 Knowledge Infused Policy Gradients - KIPG

Initialize $\Psi_0(i) = 0$ $\forall$ arms i
for k 1 to K **do**
  set $\pi_k(i) = \sigma(\Psi_{k-1}(i))$
  Draw arm $i = \arg\max_i \pi_k(i)$ ▷ observe reward $r_k(i)$ and context $c_k(i)$
  Compute gradient $\nabla_{\Psi_k(i)} \log(\pi_k(i))$ as

$$(I_k(i) - \pi_k(i) - 1)$$

▷ Depending on preference
  Compute total gradient as $\pi_k(i) \nabla_{\Psi_k(i)} \log(\pi_k(i))(r_k(i)+1)$ ▷ add 1 smoothing
  Fit $\delta_k(i)$ to gradient using TILDE tree
  Set $\Psi_k(i) = \Psi_{k-1}(i) + \eta\delta_k(i)$
return $\pi_K(i)$

prefer arm i, at step k. We prove in Theorem 1 that the approach of Odom et al. (2015) is a specific instance of KIPG with multiple knowledge sources. For our experiments we specify only a single source of knowledge at all steps k.

Theorem 1. *At step k, For S multiple knowledge sources, that either prefer or don't prefer arm i, $k1, k2, ..kS$, assuming independence, let $P(\Psi_k(i)) = \prod_{s=1}^{S} Laplace(|\Psi_k(i) - \Psi_k(i)_{ks}|, b = 1)$. Here $\Psi_k(i)_{ks} = \Psi_k(i)_{knowledge}$ $\forall s \in S$. Then we have $\nabla_{\Psi_k} \log(\pi_k(i)) = n_k(i)_t - n_k(i)_f$.*

Proof. We know that with assuming a $Laplace(x, b)$ distribution and setting $\Psi_k(i)_{ks} = \Psi_k(i)_{knowledge} = \text{LUB}\{\alpha\}$ $\forall s \in S$, we get $\nabla_{\Psi_k(i)} \log(P(\Psi_k(i))) = \sum_{s=1}^{S} \text{sign}(\text{LUB}\{\alpha\} - \sigma(\Psi_k(i)))$. We know also that $\text{sign}(\text{LUB}\{\alpha\} - \sigma(\Psi_k(i))) = 1$ depending on if the expert prefers the arm i or not. Thus we get, $\sum_{s=1}^{S} \text{sign}(\text{LUB}\{\alpha\} - \sigma(\Psi_k(i))) = n_k(i)_t - n_k(i)_f$. □

5 Regret Bound for KIPG

We now derive a bound for the total regret after K steps of KIPG to understand the convergence of KIPG towards the optimal arm choice. Since KIPG is fundamentally a gradient ascent approach, we can use analysis similar to the regret analysis for online gradient ascent to derive the regret bound (Hazan, Rakhlin, and Bartlett 2008). Using $a^2 - (a - b)^2 = 2ab - b^2$ and letting $a = (\Psi_k(i) - \Psi^*(i))$ and $b = \nabla_{\Psi(i)_k} \sum_{k=1}^{K} \pi_k(i) r_k(i)$, We know that for a sequence over K gradient ascent iterations, $\{\Psi_k(i) | k \in [K]\}$, we have

$$(\Psi_k(i) - \Psi^*(i))^2 - (\Psi_{k-1}(i) - \Psi^*(i))^2 \\ 2\gamma(\pi_k(i) r_k(i) - \pi^*(i) r(i^*)) + \gamma^2 L$$

where $L - \nabla_{\Psi_k(i)} \sum_{k=1}^{K} \pi_k(i) r_k(i)$ is an upper bound on the gradient (Lipschitz constant) and γ is the learning rate. Using a telescoping sum over K iterations we have

$$(\Psi_K(i) - \Psi^*(i))^2 - (\Psi_0(i) - \Psi^*(i))^2 \\ 2\sum_{k=1}^{K} (\gamma(\pi_k(i) r_k(i) - \pi^*(i) r(i^*))) + \sum_{k=1}^{K} \gamma^2 L$$

and therefore

$$\sum_{k=1}^{K} (\gamma(\pi_k(i) r_k(i) - \pi^*(i) r(i^*))) \\ \frac{(\max_{\Psi_k(i)} (\Psi_k(i) - \Psi^*(i)))^2 + L^2 \sum_{k=1}^{K} \gamma^2}{2\sum_{k=1}^{K} \gamma}.$$

Solving for γ by setting $\nabla_{\gamma}(R.H.S) = 0$, we finally have our total regret bound over K steps as:

$$\sum_{k=1}^{K} (\gamma(\pi_k(i) r_k(i) - \pi^*(i) r(i^*))) \\ \frac{(\max_{\Psi_k(i)} (\Psi_k(i) - \Psi^*(i)))^2 L}{p\sqrt{K}}$$

This regret bound has a very intuitive form. It shows that the regret is bounded by how far off the learned $\Psi(i)$ from the true $\Psi^*(i)$ for each arm i. Thus we expect that in the experiments, with quality knowledge infusion this gap is drastically reduced over K steps to result in a low total regret.

6 KIPG-Upper Confidence Bound

So far we have developed KIPG for the Bandit Setting and derived a regret bound. Since KIPG estimates $\pi(i)$ after each arm pull, we can sample from $\pi(i)$ and choose the max like in Thompson Sampling. However, since the arm to pull is being learned online, the uncertainty in the arm choice even with knowledge needs to be modeled. The human providing knowledge needs to observe a few user-arm pulls to gradually improve their confidence in the knowledge provided. As the data is not available offline to study by the human, it is unlikely that the knowledge provided is perfect initially. Thus, we now derive a confidence bound to quantify the uncertainty in the arm choice. At step k, let the arm choice is be denoted by i^*. First we notice that $Z = |\pi_k(i^*) - \pi^*(i^*)|$, is binomial distributed at step k. Also, $\pi_k(i^*)$ is binomial distributed. However, for both we will use a Gaussian approximation and note that for this Gaussian, $\mu(Z) = 0$ and $\sigma(Z) - \mathbf{E}[(\pi_k(i^*) - \pi^*(i^*))^2]$, thus making this a *sub-Gaussian*(Peizer and Pratt 1968; Buldygin and Kozachenko 1980). Using Markov's inequality we have (Cohen 2015):

$$P(Z > \epsilon) - e^{-k\epsilon} \mathbf{E}[kZ] \\ \Rightarrow P(e^{kZ} > e^{k\epsilon}) - \mathbf{E}[e^{kZ}] \, e^{-k\epsilon}$$

where e^{kZ} is the moment-generating-function for Z. We know that e^{kZ} is convex and thus $e^{kZ} - \gamma(e^{kb}) + (1 - \gamma)e^{ka}$ for $Z \in [a, b]$ and $\gamma \in [0, 1]$. Thus we obtain $Z - \gamma b + (1 - \gamma)a$, which gives us $\gamma = \frac{Z - a}{b - a}$, therefore we know

$$e^{kZ} - \frac{ae^{kb} + be^{ka}}{b - a} + \frac{Z(e^{kb} - e^{ka})}{b - a}.$$

Taking Expectation on both sides we get

$$\mathbf{E}[e^{kZ}] \quad \frac{ae^{kb} + be^{ka}}{b \quad a}.$$

Let $e^{g(k)} = \frac{ae^{kb}+be^{ka}}{b \quad a}$, we get $g(k) = ka + \log(b \quad ae^{k(b \quad a)}) \quad \log(b \quad a)$. Using Taylor series expansion for $g(k)$ upto the second order term as $g(0) + r(g(k))k + \frac{r^2(g(k))k^2}{2}$, we get

$$r^2(g(k)) = \frac{ab(b \quad a)^2(\quad e^{k(b \quad a)})}{(ae^{k(b \quad a)} \quad b)^2},$$

We note that:

$$\begin{aligned} ae^{t(b \quad a)} \quad a &=) \quad ae^{t(b \quad a)} \quad b \quad a \quad b \\ &=) \quad (ae^{t(b \quad a)} \quad b)^2 \quad (b \quad a)^2 \\ &=) \quad (ae^{t(b \quad a)} \quad b)^{\ 2} \quad (b \quad a)^{\ 2} \end{aligned}$$

We know $e^{k(b \quad a)} \quad 1$, therefore we obtain

$$\begin{aligned} r^2(g(k)) \quad \frac{ab(b \quad a)^2}{(b \quad a)^2} &= \quad ab \\ =) \quad r^2(g(k)) \quad \frac{(a \quad b)^2 + (a+b)^2}{4} \\ =) \quad r^2(g(k)) \quad \frac{(b \quad a)^2}{4} \quad &=) \quad g(k) \quad \frac{(b \quad a)^2}{4}\frac{k^2}{2} \end{aligned}$$

We know that

$$\mathbf{E}[e^{kZ}] \quad e^{g(k)} \quad =) \quad \mathbf{E}[e^{kZ}] \quad e^{\frac{k^2(b \quad a)^2}{8}}$$

Once again from the Markov inequality, we have

$$P(Z > \epsilon) \quad e^{\ k\epsilon}\mathbf{E}[kZ]$$

$$=) \quad P(j\pi_k(i\ ) \quad \pi\ (i\ )j > \epsilon) \quad e^{\ k\epsilon + \frac{k^2(b \quad a)^2}{8}},$$

using $k = \frac{4\epsilon}{(b \quad a)^2}$, we get

$$P(j\pi_k(i\ ) \quad \pi\ (i\ )j > \epsilon) \quad e^{\frac{2\epsilon^2}{(b \quad a)^2}},$$

as $0 \quad (b \quad a) \quad 1$

$$P(j\pi_k(i\ ) \quad \pi\ (i\ )j > \epsilon) \quad e^{\ 2\epsilon^2}$$

$$P(j\pi_k(i\ ) \quad \pi\ (i\ )j > \epsilon) \quad e^{\ 2K\epsilon^2}.$$

Solving for ϵ we get, $\epsilon \quad \frac{\log(P(j\pi_k(i\ ) \quad \pi\ (i\ )j))}{2K}$. Thus, we draw the next optimal arm choice i at k+1 as follows:

$$\arg\max_i \left\{ i \quad \pi_{k+1}(i) = \sigma(\Psi_k(i) + \frac{\log(P(Z))}{2K}) \right\},$$

where $Z = j\pi_k(i\ ) \quad \pi\ (i\ )j$. This confidence bound also has an intuitive form as it is reasonable that the expectation $\mathbf{E}(I(j\pi_k(i\ ) \quad \pi\ (i\ )j))$ gets closer to the truth as more arms are pulled, where I is the indicator function. Since we never actually know $\pi\ (i\ )$, we set to the current best estimate. We expect that *Knowledge Infusion* will allow the error between the current best estimate and $\pi\ (i\ )$ to be small. Algorithm 2 shows how a simple modification to the psuedocode in Algorithm 1 can incorporate the bound derived.

Algorithm 2 KIPG Upper Confidence Bound - KIPGUCB

1: Initialize $\Psi_0(i) = 0$ 8 arms i
2: **for** $k \quad 1$ to K **do**
3: set $\pi_k(i) = \sigma(\Psi_{k\ \ 1}(i))$
4: Draw arm $i \ = \arg\max_i i \quad \pi_k(i)$ ▷ observe reward $r_k(i\ )$ and context $c_k(i\ )$
5: Set $\pi\ (i\ ) = \pi_k(i\ )$
6: Compute gradient $r_{\Psi_k(i\ )} \log(\pi_k(i\ ))$ as

$$\left(I_k(i\ ) \quad \pi_k(i\ ) \quad 1 \quad \frac{\log(\mathbf{E}(I(j\pi_k(i\ ) \quad \pi\ (i\ )j)))}{2k} \right)$$

▷ Depending on preference
7: Compute total gradient as $\pi_k(i\ ) r_{\Psi_k(i\ )} \log(\pi_k(i\ ))(r_k(i\ ) + 1)$ ▷ add 1 smoothing
8: Fit $\delta_k(i\ )$ to gradient using TILDE tree
9: Set $\Psi_k(i\ ) = \Psi_{k\ \ 1}(i\ ) + \eta\delta_k(i\ )$
10: return $\pi_K(i)$

7 Simulation Model and Experiments

Simulation model: We perform experiments on a simulated music recommendation dataset. The dataset simulates of songs, artists, users and albums where there are the following user behaviors:

- Behavior A: The users are fans of one of the artists in the dataset.
- Behavior B: The users follow the most popular song.
- Behavior C: They follow the most popular artist.

We will denote the set of behaviors by **Behaviors**. Figure 1(b) shows an illustration for the Schema for the simulation model depicting that M users can listen to N songs and N songs can be sung by N artists, etc. Artists and Songs have attributes "Popular" denoting if a particular artist or a song is popular among users.

Context Induction: Once the simulation model is used to generate different users based on a predefined behavior 2 **Behaviors**. We need now to generate different possible user contexts from this dataset. Since the whole dataset is not available to us offline, we construct a dataset by 50 random arm choices to induce contexts. The contexts will be represented using predicate logic clauses: antecedent (^ preconditions representing possible user context) $=)$ consequent (user song choice). For this, an inductive bias needs to be provided to induce sensible clauses. Such an inductive bias is included as background knowledge to the induction program. We use the method in Hayes et al., to automatically construct the inductive bias from the schema in Figure 1(b) (Hayes et al. 2017). The clauses are induced are kept if they satisfy minimum information criteria i.e. if they discriminate at least one user from another in their song choice, in the dataset. The clauses induced using the provided inductive bias and the are as follows:

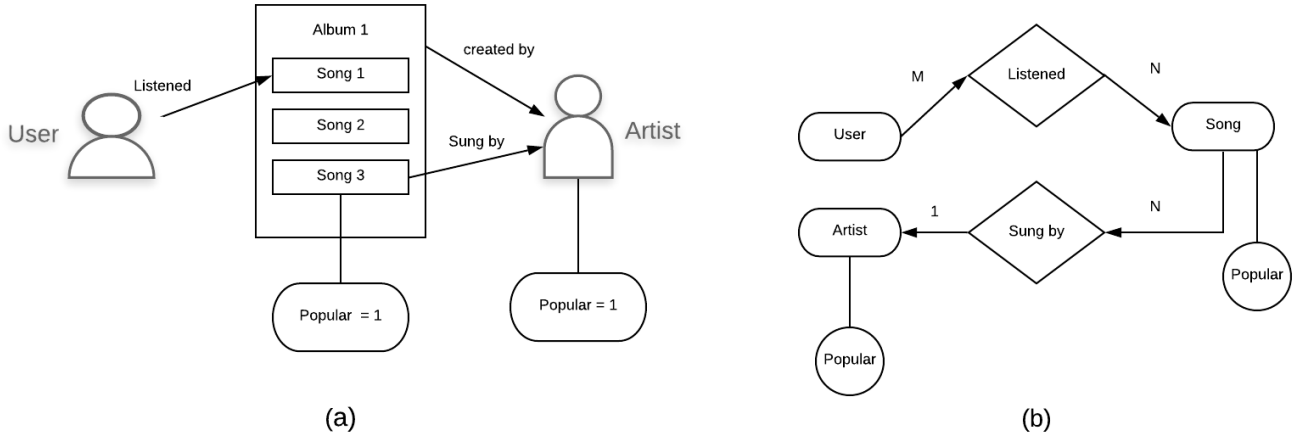


Figure 1: Illustration of the Entity Relationship Schema diagram for the Music Recommendation system being simulated (b) and a particular instantiation (a). Users listen to songs by artists. M Users can listen to N Songs, N Songs can be written by 1 Artist and Artists and Songs can be popular among Users.

- sungby(B,C) ^ : popular(C) =) listens(A,B). This context which says *User A listens to song B, if song B is sungby artist C. Also, C is not a popular artist*, describes behavior A.
- sungby(B,C) ^ popular(C) =) listens(A,B). This context which says *User A listens to song B, if song B is sunby a popular artist C*, describes behavior C.
- listened(C,B) =) listens(A,B). This context which says *user A listens to song B, if user B listened to C*, describes behavior B.

We use satisfiability of these clause antecedents as features for TILDE regression tree stumps. Figure 6 shows an example of a TILDE regression tree stump. sigmoid of the regression values represent $\pi(i) = \Pr(listens(A, B))$, where the song choices (in this case, any song instantiation of B), are the arms i.

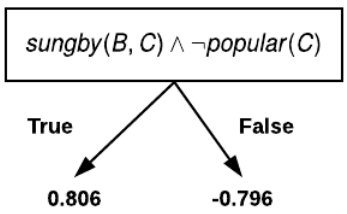


Figure 2: Example of a TILDE regression tree stump for song choice. The tree depicts that if *if song B is sungby artist C and also, C is not a popular artist*, *User A listens to B* with probability $\sigma(0.806)$. Else, *User A listens to B* with probability $\sigma(\ \ 0.796)$.

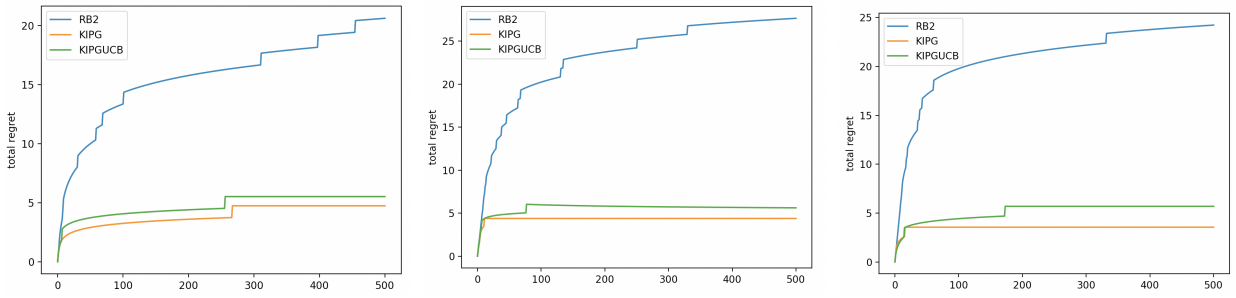
Experiments: Since this is a relational dataset we employ two baselines on recommendation of songs, 1) The RB2 algorithm by Kakadiya et al. (2020), KIPG without UCB and KIPG with UCB. For each type of users, at time step k, a recommendation is provided depending on the algorithm used, and the regret drawn from comparison to the ground truth (GT) recommendation is recorded. The regret equation for an algorithm A is:

$$R_A = \sum_{k=1}^{K} (r^{GT} \quad \pi_k(i\ )_A r_k(i\ )),$$

where i is the optimal arm drawn from $\arg\max$ over $\pi(i)$ samples at step k (See Algorithm 1,2 - line 4). r^{GT} is the reward if the ground truth optimal arm is drawn at k.

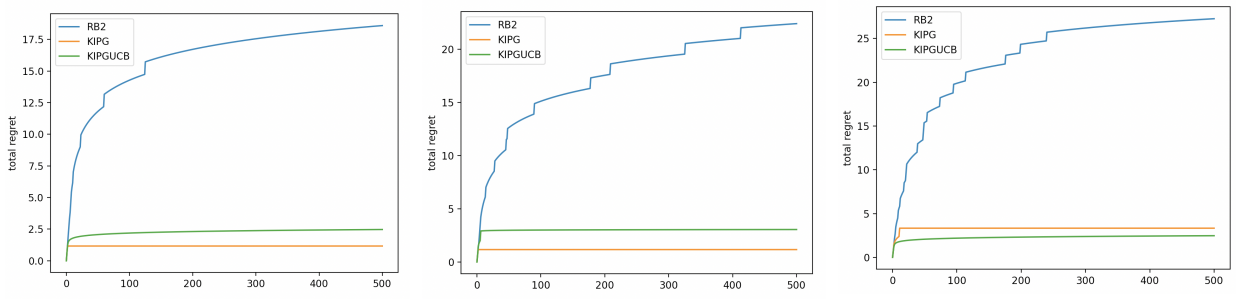
RB2: We now briefly describe the RB2 algorithm. RB2 samples from a user-arm interaction set (initially populated by random arm choices) in $k = 1\ to\ K$ batches. RB2 then fits a set of boosted trees to the resulting data at step k to obtain probability of each arm choice. The most likely arm choice at each step k given by this probability is add to the user-arm interaction set. This process repeats until K number of batches is reached.

Perfect Knowledge: It is possible that the human providing knowledge may have some previous knowledge about a user in the system. In this case, it is expected that the knowledge is pretty good from the start. In this setting, we expect the regret is ordered as $R_{KIPG} < R_{KIPGUCB} < R_{RB2}$ for most $k = 1$ to K.We expected this trend since RB2 uses no knowledge and KIPGUCB moves slower towards knowledge initially. Given that the knowledge is perfect, we expect KIPG to perform the best. We set $K = 500$. Figure 3 shows that the experiments coroborate this.



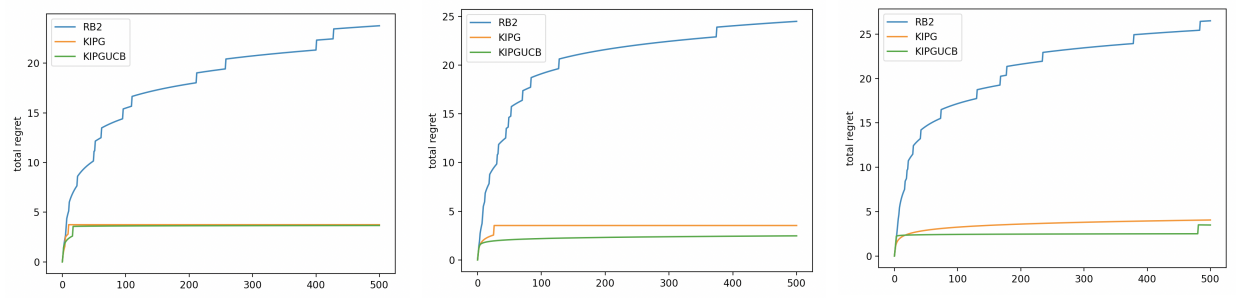
(a) Behavior A (b) Behavior B (c) Behavior C

Figure 3: Shows comparison of R_{RB2}, R_{KIPG}, $R_{KIPGUCB}$ for the perfect knowledge setting for all their behaviors. As expected we see that $R_{KIPG} < R_{KIPGUCB} < R_{RB2}$ for most $k = 1\ to\ K$.



(a) Behavior A (b) Behavior B (c) Behavior C

Figure 4: Shows comparison of R_{RB2}, R_{KIPG}, $R_{KIPGUCB}$ for the nearly perfect knowledge setting for all three behaviors. As expected we see that $R_{KIPG} < R_{KIPGUCB} < R_{RB2}$ for most $k = 1\ to\ K$.

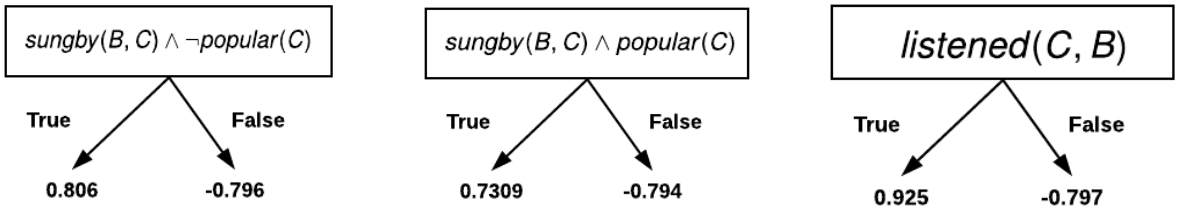


(a) Behavior A (b) Behavior B (c) Behavior C

Figure 5: Shows comparison of R_{RB2}, R_{KIPG}, $R_{KIPGUCB}$ for the noisy knowledge setting for all three behaviors. As expected we see that $R_{KIPGUCB} < R_{KIPG} < R_{RB2}$ for most $k = 1\ to\ K$.

Nearly Perfect Knowledge: Unless the human providing knowledge has some previous knowledge about some users, the human must observe some user arm interactions to improve the knowledge that they provide. We simulate this scenario by using noisy knowledge for $k = 1\ to\ 50$, where perfect knowledge is provided 80% of the time. In this setting we still expect that for most $k = 1toK$, where $K = 500$, $R_{KIPG} < R_{KIPGUCB} < R_{RB2}$. We expect this as a perfection rate of 80% is still very near perfect. Figure 4 shows this result.

Noisy Knowledge: In this setting the human again observes some user arm interactions to improve the knowledge that they provide. In this case however, the humans observation skills are less sharp. We simulate this scenario by using noisy knowledge for $k = 1\ to\ 50$, where perfect knowledge is provided 60% of the time instead of 80%. Here, we expect that for most $k = 1toK$, where $K = 500$, $R_{KIPGUCB} < R_{KIPG} < R_{RB2}$. We expect this as a perfection rate of 60% means that the tempering of *Knowledge Infusion* by KIPGUCB initially leads to better total regret



(a) Behavior A (b) Behavior B (c) Behavior C

Figure 6: Shows the combined tree stumps after K steps of KIPGUCB. We can see that the trees are easy to interpret by a human.

for KIPGUCB. Figure 5 shows this result.

8 Interpretability of Learned Policy

A set of regression trees are not usually interpretable to a human. In propositional trees, the tree paths can be combined as the final output is fundamentally a sum. In relational trees as different variables are used for the same object across different tree paths, unifying them during combination is tricky. To interpret them, a final approximation tree would need to be extracted from the sum (Craven and Shavlik 1995). To avoid this approximation, we induce possible context clauses before the tree learning. Thus, our trees are stumps as shown in the Figure 6. This makes combination simple and the regression values can safely be added without any unification, making the policy learned easily interpretable by a human. If we write out our policy tree (sum of regression stumps) as clauses, our learned policy would look as follows:

$$sungby(B, C) \text{ ^ } : popular(C) => listens(A, B), \pi(B)$$
$$sungby(B, C) \text{ ^ } popular(C) => listens(A, B), \pi(B)$$
$$listened(C, B) => listens(A, B), \pi(B)$$

, where $not(popular(C))$ is represented as $: popular(C)$. This can be easily interpreted as given each relational context, what is the probability of recommending a song instance of B. Figure 6 shows the combined tree representations from which the clauses are derived.

9 Conclusion and Future Directions

In this study, we develop a novel algorithm KIPG-UCB to perform *Knowledge Infusion* in CB settings. We show that the regret bound depends on the knowledge and hence the total regret can be reduced if the right knowledge is available. Furthermore, we develop a confidence bound to account for initial uncertainty in provided knowledge in online settings. Though we have developed a general framework for *Knowledge Infusion*, we have yet to explore knowledge forms beyond preference knowledge. With respect to enhancing the user model in the simulation, users in music recommendation systems frequently change their behavior (for example, transition from Behavior A to Behavior B). Furthermore, the user's choice of music may depend on latent behaviors that cannot be modeled such as a bias towards recycling songs heard in the same family or music that relates to a extremely recent life circumstance. This type of non-stationarity and partial observability in user context will be interesting to model. Also, if knowledge fails to lower total regret, identifying the right descriptive question to ask the human to elicit new knowledge is an interesting future direction. Relational descriptions make tackling this issue plausible. Finally, it will be interesting to mathematically evaluate if the knowledge should be incorporated at all. We aim to tackle these issues in future work.

References

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 2002. The nonstochastic multiarmed bandit problem. *SIAM journal on computing* 32(1): 48–77.

Blockeel, H.; and De Raedt, L. 1998. Top-down induction of first-order logical decision trees. *Artificial intelligence* 101(1-2): 285–297.

Buldygin, V. V.; and Kozachenko, Y. V. 1980. Sub-Gaussian random variables. *Ukrainian Mathematical Journal* 32(6): 483–489.

Chapelle, O.; and Li, L. 2011. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, 2249–2257.

Cohen, J. E. 2015. Markov's inequality and Chebyshev's inequality for tail probabilities: a sharper image. *The American Statistician* 69(1): 5–7.

Craven, M.; and Shavlik, J. 1995. Extracting tree-structured representations of trained networks. *Advances in neural information processing systems* 8: 24–30.

Hayes, A. L.; Das, M.; Odom, P.; and Natarajan, S. 2017. User friendly automatic construction of background knowledge: Mode construction from er diagrams. In *Proceedings of the Knowledge Capture Conference*, 1–8.

Hazan, E.; Rakhlin, A.; and Bartlett, P. L. 2008. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems*, 65–72.

Kakadiya, A.; Natarajan, S.; and Ravindran, B. 2020. Relational Boosted Bandits.

Kersting, K.; and Driessens, K. 2008. Non-parametric policy gradients: A unified treatment of propositional and relational domains. In *Proceedings of the 25th international conference on Machine learning*, 456–463.

Lai, T. L.; and Robbins, H. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6(1): 4–22.

Langford, J.; and Zhang, T. 2007. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in neural information processing systems* 20: 817–824.

Odom, P.; Khot, T.; Porter, R.; and Natarajan, S. 2015. Knowledge-based probabilistic logic learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Peizer, D. B.; and Pratt, J. W. 1968. A normal approximation for binomial, F, beta, and other common, related tail probabilities, I. *Journal of the American Statistical Association* 63(324): 1416–1456.

Peters, J.; and Bagnell, J. A. 2010. Policy Gradient Methods. *Scholarpedia* 5(11): 3698.

Thompson, W. R. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25(3/4): 285–294.

Zhou, L. 2015. A survey on contextual multi-armed bandits. *arXiv preprint arXiv:1508.03326* .