AI INSTITUTE #AIISC
UNIVERSITY OF SOUTH CAROLINA

Knowledge Infused Policy Gradients Upper Confidence Bound for Contextual Bandits

Team

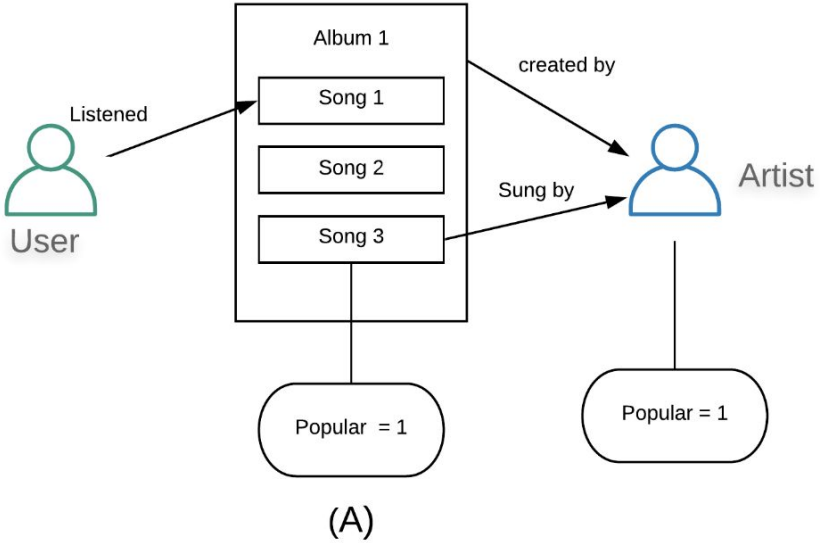
Kaushik Roy
Ph.D. Student
Artificial Intelligence Institute
University of South Carolina

Qi Zhang
Assistant Professor
Artificial Intelligence Institute
University of South Carolina

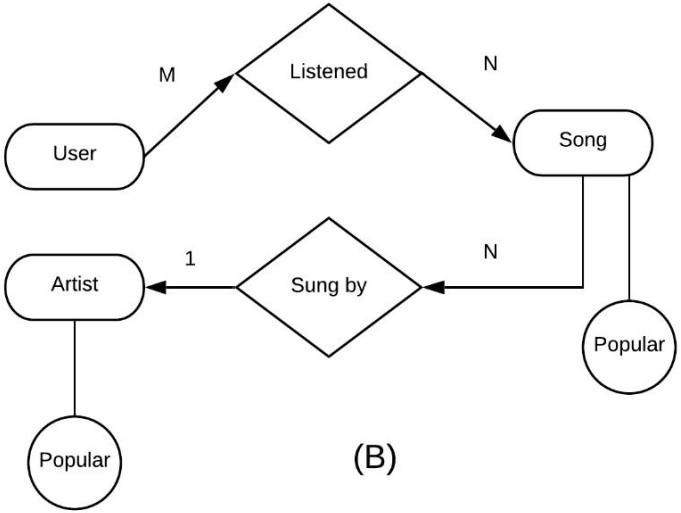
Manas Gaur
Ph.D. Student
Artificial Intelligence Institute
University of South Carolina

Amit Sheth
Director and Professor
Artificial Intelligence Institute
University of South Carolina

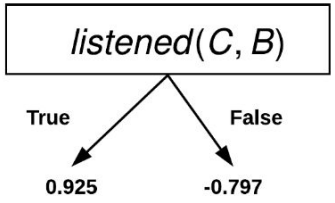
Music Recommendation System



(A)

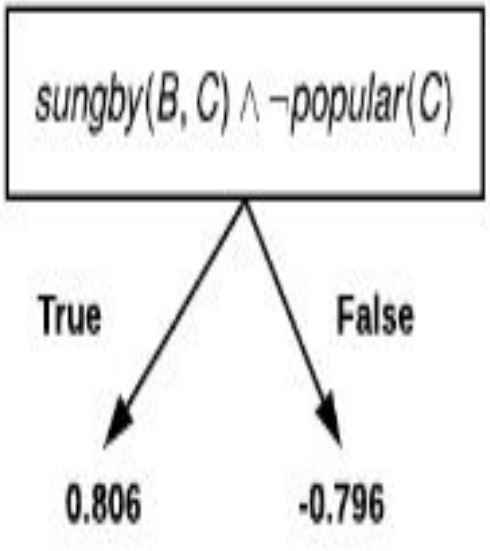


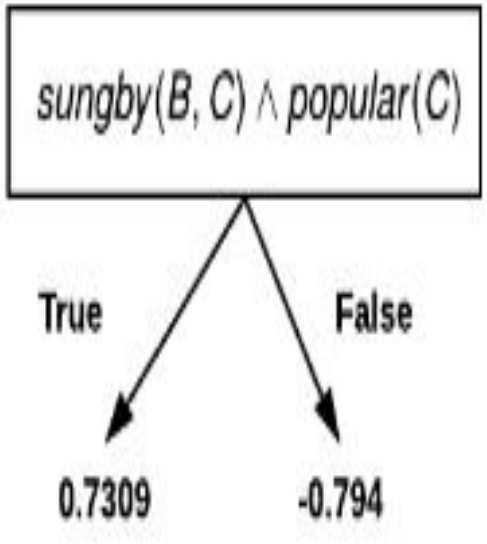
(B)

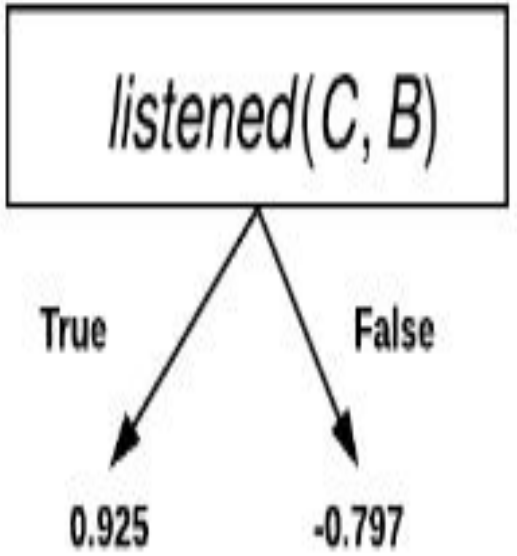


(C)

Types of Behaviors amongst Users in the System







Knowledge Infused Contextual Bandits

- In online settings, learning takes a lot of time
- Prior knowledge about user behavior can be used to accelerate learning
- This knowledge can be incorporated into the learning algorithm seamlessly

Incorporating Knowledge in the Bandit Setting

Algorithm 1 Knowledge Infused Policy Gradients - KIPG

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

$$(I_k(i^*) - \pi_k(i^*) \pm 1)$$

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

Incorporate Knowledge into the gradient when learning

$$(I(s,a) - P(\mathbf{D}|\psi(s,a))) + \sum_i \alpha_i(-\mathbf{sign}(\psi(s,a) - \omega_i))$$

Uncertainty Quantification in an Online Setting

- People's behavior unclear as data is being streamed in
- As such, knowledge about the persons behavior by an expert needs to be observed for a while causing initial uncertainty
- Therefore, a confidence bound is required to quantify the uncertainty facilitating robust exploration

Incorporating an Upper Confidence bound

Algorithm 2 KIPG Upper Confidence Bound - KIPGUCB

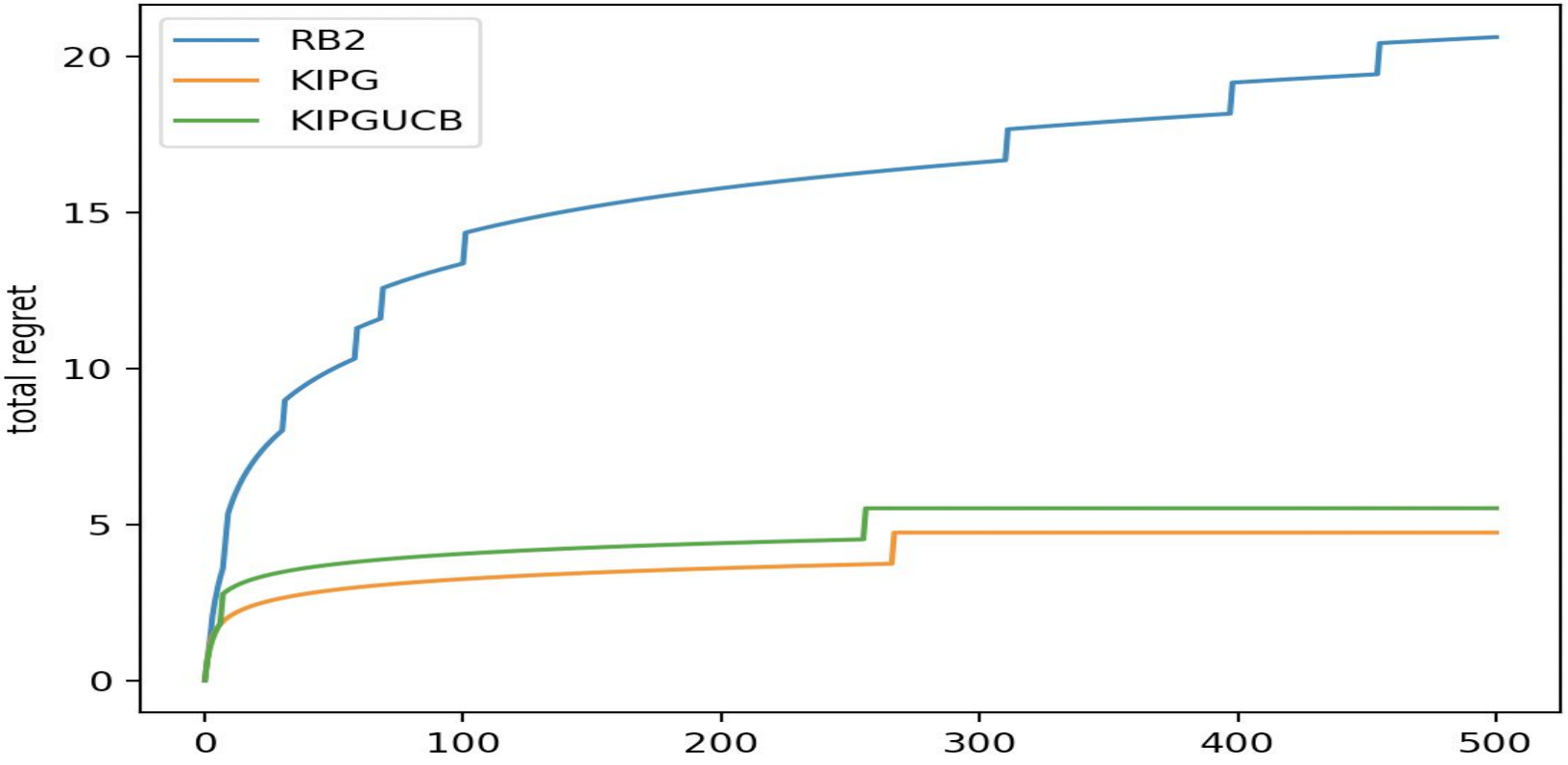
Initialize $\Psi_0(i) = 0 \; \forall$ arms i
for $k \leftarrow 1$ to K **do**
    set $\pi_k(i) = \sigma(\Psi_{k-1}(i))$
    Draw arm $i^* = \arg\max_i i \sim \pi_k(i)$ ▷ observe reward $r_k(i^*)$ and context $c_k(i^*)$
    Set $\pi^*(i^*) = I(\pi_k(i^*) = i^*)$
    Compute $\nabla_{\Psi_k(i^*)} \log(\pi_k(i^*))$ as ▷ $\pm$ Depending on preference

$$\left(I_k(i^*) - \pi_k(i^*) \pm 1 - \frac{\log(|\pi_k(i^*) - \pi^*(i^*)|)}{2k} \right)$$

    Compute 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 $\pi_K(i)$

Incorporate Knowledge into the gradient when learning with an Upper Confidence Bound

Results



Thanks!

Any questions?

You can find me at:

kaushikr@email.sc.edu