TEXAS
The University of Texas at Austin

Application of online learning in clinical decision support system

Haoqi Wang, Jiaxun Cui

December 2021

Outline

- Background
- Problem Setup
- Methods
- Results
- Discussion
- Next step

Background

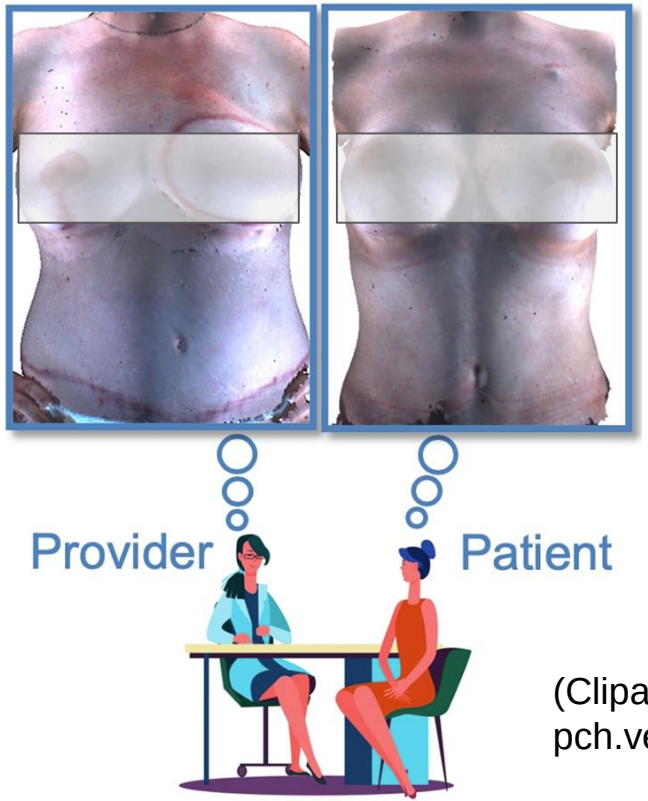
The goal of reconstructive surgery is to restore the body.

Breast reconstruction is the most common reconstructive procedure, especially for women who had mastectomy to treat breast cancer.

However, women may don’t know what to expect and can have unrealistic expectations. This can lead to bad decision and regret.

Current tools contain no visual information to directly assess patients’ expectations.

Unrealistic expectation



(Clipart designed by pch.vector / Freepik.)

Goal

- Build a tool to help woman express her mental image
- Help doctor-patient communication in plastic surgery
- Help patient form realistic expectation and reduce regret after surgery

- More specifically, the goal is to identify a clinical photograph of a prior patient that the current patient perceives as being similar to her mental image of what she will look like post-operatively.

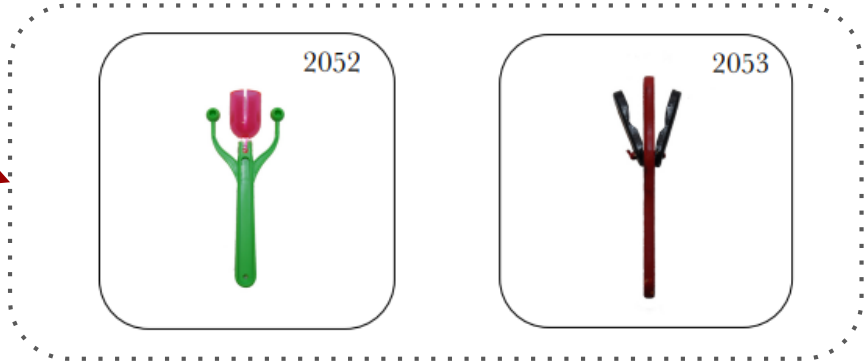
Problem Setting

1. Player has a mental image of outcome
2. Recommender system recommends an image (***n*** arms)
3. Player gives the similarity (reward with error) to the recommended image
4. Based on the reward, recommender system recommends a new image
5. Repeat for ***T*** rounds (considering limited time in clinic)
6. Finally, recommender system gives the best recommendation

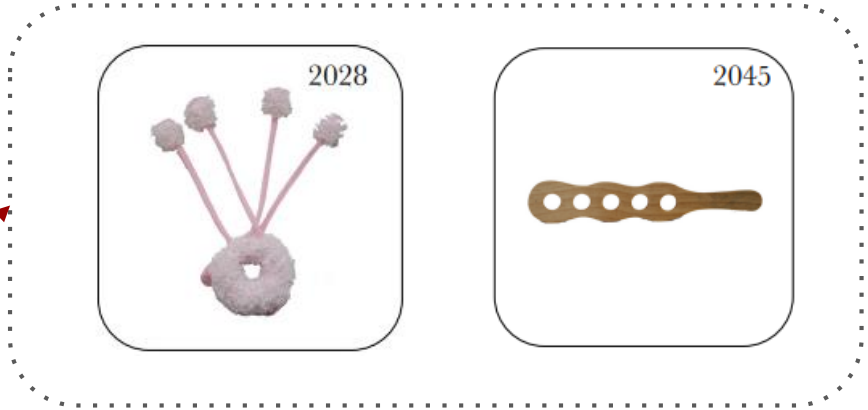
Our **goal** is to build a recommender system that finds the best arm in ***T*** rounds, so that ***T*** **<<** ***n***, leveraging the graphical relationship among the arms

Perceptual Distances Dataset (graphical relationship)

Stim_1	Stim_2	Distance
I_2052	I_2053	0.1654
I_2016	I_2048	0.1887
I_2022	I_2030	0.2075
I_2034	I_2040	0.2115
I_2029	I_2055	0.2154

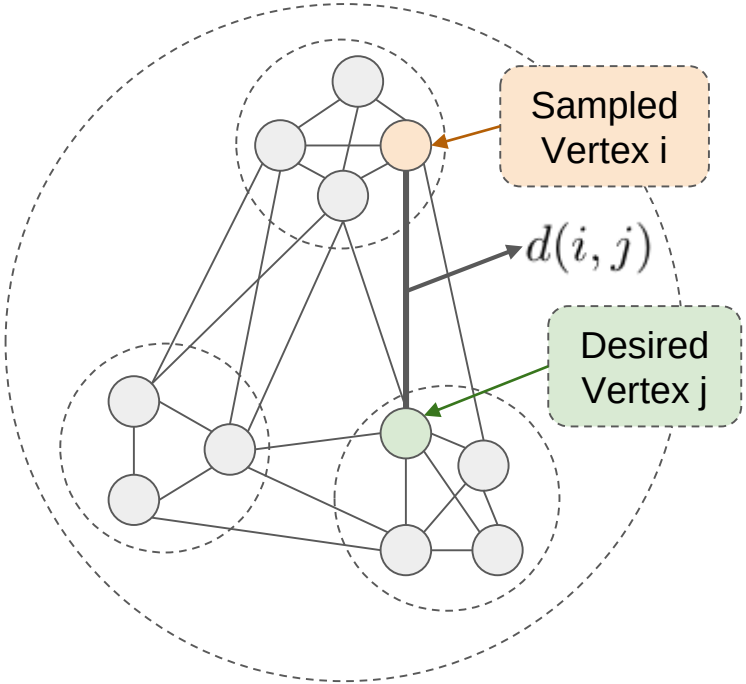


I_2018	I_2053	0.9836
I_2007	I_2064	0.9837
I_2009	I_2014	0.9841
I_2009	I_2021	0.9845
I_2027	I_2059	0.9845
I_2019	I_2049	0.9845
I_2024	I_2029	0.9854
I_2028	I_2045	0.9855



https://link.springer.com/article/10.3758/s13428-015-0647-3

Reward Design



Given **pairwise Perceptual Distances** between images of possible outcomes (generated from observe study)

- Player has a ground truth **i** in mind
- Recommender system recommends an image **j**
- Perceptual Distance between i and j is ***d(i, j)***
- Reward = 1 - ***d(i, j)*** + Noise

$$r_{i,t} = 1 - d(i,j) + \mathcal{N}(0, \sigma^2)$$

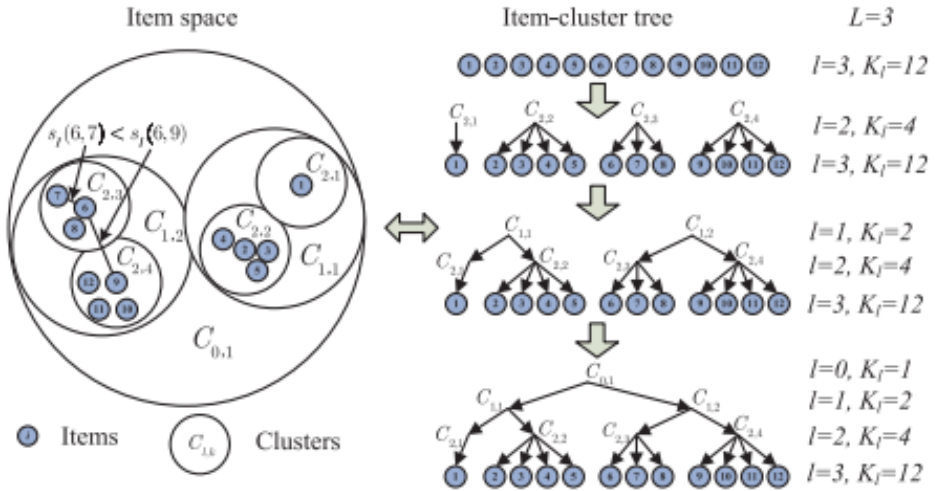
Method

1. Baseline: UCB
2. Adaptive Recommender (Song) [1]
3. Adaptive Recommender Improve (Ours)
4. GraphUCB [2]
5. NearNeighborUCB (Ours)

[1] Song, L., Tekin, C., & Van Der Schaar, M. (2014). **Online learning in large-scale contextual recommender systems.** *IEEE Transactions on Services Computing*, *9*(3), 433-445.
[2] Thaker, P. K., Rao, N., Malu, M., & Dasarathy, G. (2021). **Pure Exploration in Multi-armed Bandits with Graph Side Information.** *arXiv preprint arXiv:2108.01152*.

Adaptive recommender [1]

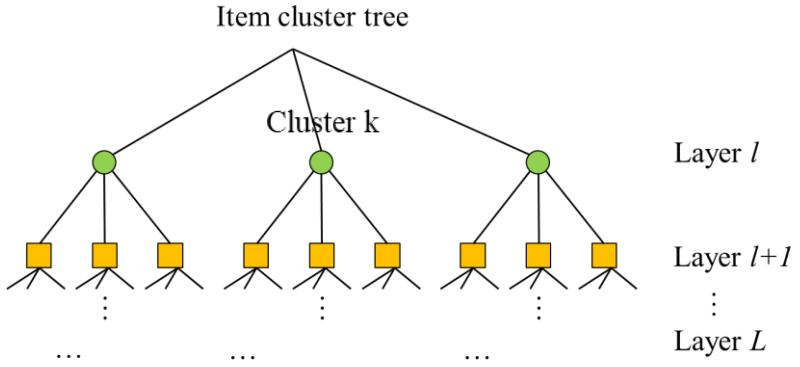
- Built item-cluster tree that fulfils the exponential tree metric
- The algorithm first sets each leaf to be a single item cluster, and then operates from leaves to root by combining clusters that are “similar”
- Depth of tree and cluster size are controlled



[1] Song, L., Tekin, C., & Van Der Schaar, M. (2014). **Online learning in large-scale contextual recommender systems.** *IEEE Transactions on Services Computing*, *9*(3), 433-445.

E epochs in total

for $l = 0 : E - 1$ **do**
  Set partition $\mathcal{K} := \mathcal{K}_l = \{C_{l,k} : 1 \le k \le K_l\}$.
  for $t = 2^l : 2^{l+1} - 1$ **do**
    Select the cluster with maximum index
    $k = \arg\max_{k' \in \mathcal{K}} I^s_{l,k}(t)$, with ties broken arbitrarily.
    Randomly select a child node $C_{l+1,\hat{k}}$ of $C_{l,k}$.
    Randomly recommend an item in cluster $C_{l+1,\hat{k}}$.
    Receive the reward: r_t.
    Record $\{t, x_t, C_{l,k}, C_{l+1,\hat{k}}, r_t\}$.



Set $l := E$, and set partition $\mathcal{K} := \mathcal{K}_E = \{C_{E,k} : 1 \le k \le K_E\}$
for $t = 2^l : T$ **do**
  Select the cluster with maximum index
  $k = \arg\max_{k' \in \mathcal{K}} I^s_{l,k}(t)$, with ties broken arbitrarily.
  Randomly recommend an item in cluster $C_{l,k}$.
  Receive the reward: r_t.
  Record $\{t, x_t, C_{l,k}, r_t\}$.

$$I^s_{l,k}(t) = \overline{r}_{l,k,t} + \sqrt{A_s \ln t / N_{l,k,t}}$$

As is the exploration-exploitation trade-off factor

Adaptive recommender Improve

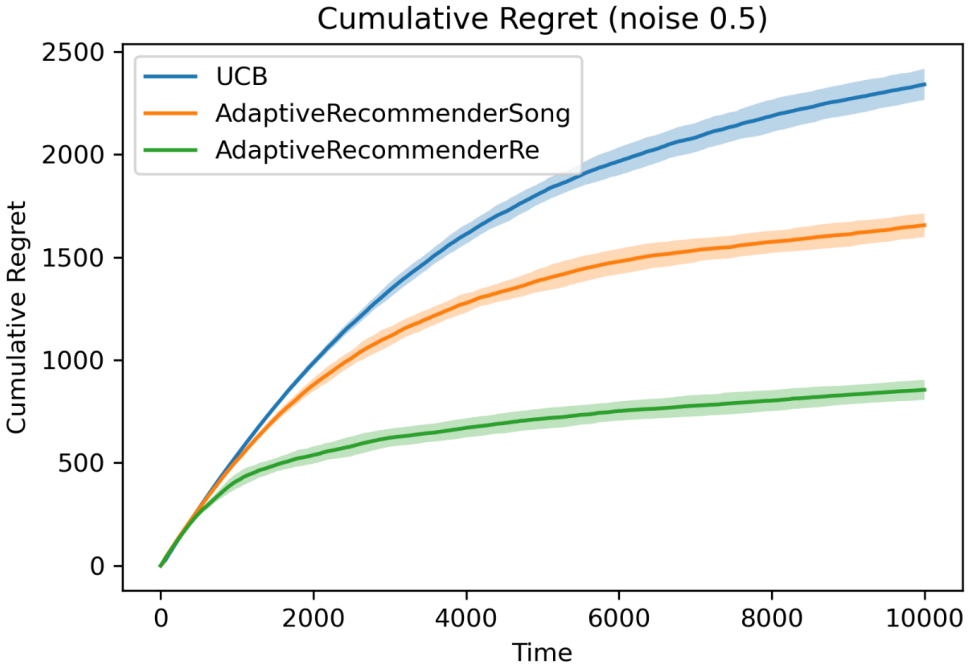
for $l = 0 : E - 1$ **do**
  Set partition $\mathcal{K} := \mathcal{K}_l = \{C_{l,k} : 1 \leq k \leq K_l\}$.
  for $t = 2^l : 2^{l+1} - 1$ **do**
    Select the cluster with maximum index
    $k = \arg\max_{k' \in \mathcal{K}} I^s_{l,k}(t)$, with ties broken arbitrarily.
    Recommend the representative item of the cluster
    Received the reward: r_t
    Update statistics of its parents and siblings based on similarities between them

Float number of plays — sibling.n_plays += 1 - distance

Scale reward based on distance — reward = (1 - distance) * reward

The rest stays the same

UCB vs Adaptive recommender (Song) vs Adaptive recommender (improve)



Experiments on Gaussian Noise with std = 0.5

- UCB is O(log n)
- Adaptive recommender (Song) is better than UCB by a constant factor
- Adaptive recommender (improve) is better than UCB by a constant factor

UCB vs Graph UCB

1: **Input** k

2: Choose each arm once

3: Subsequently choose

$$A_t = \text{argmax}_i \left(\hat{\mu}_i(t-1) + \sqrt{\frac{2 \log f(t)}{T_i(t-1)}} \right)$$

where $f(t) = 1 + t \log^2(t)$

Asymptotically Optimal UCB

input : Regularization parameter ρ,
Smoothness parameter ϵ, Error bound δ,
Total arms n, Laplacian L_G,
Sub-gaussianity parameter σ

output : A

$$V_t \leftarrow V_{t-1} + \mathbf{e}_k \mathbf{e}_k^T, \text{ and } \mathbf{x}_t \leftarrow \mathbf{x}_{t-1} + r_{t,k} \mathbf{e}_k;$$

$$\hat{\boldsymbol{\mu}}_t \leftarrow V_t^{-1} \mathbf{x}_t;$$

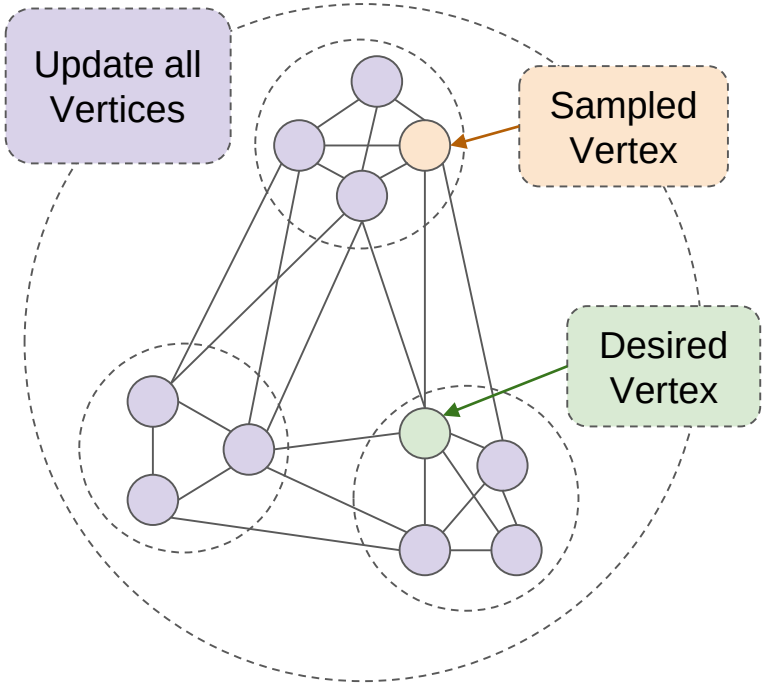
$$a_{\max} \leftarrow \arg\max_{i \in A} \left[\hat{\mu}_t^i - \beta(t_i) \sqrt{[V_t^{-1}]_{ii}} \right];$$

$$A \leftarrow \left\{ \mathbf{a} \in A \mid \hat{\mu}_t^{a_{\max}} - \hat{\mu}_t^a \leq \beta(t_a) \sqrt{[V_t^{-1}]_{aa}} + \beta(t_{a_{\max}}) \sqrt{[V_t^{-1}]_{a_{\max} a_{\max}}} \right\};$$

Graph UCB [1]

[2] Thaker, P. K., Rao, N., Malu, M., & Dasarathy, G. (2021). Pure Exploration in Multi-armed Bandits with Graph Side Information. *arXiv preprint arXiv:2108.01152*.

Graph UCB: Leveraging Graph Side Information



Complete Graph - Every two vertices are connected, Not showing all edges

Assumptions

1. **Undirected, Weighted, Complete Graph**. We know distance (similarity) between every two vertices
2. **Rewards are smooth**. Vertices with higher similarity gets more similar rewards.

Want to leverage the graph information to infer about arms that has never sampled

Laplacian-regularized least-squares optimization program

$$\hat{\boldsymbol{\mu}}_T = \arg\min_{\boldsymbol{\mu}\in\mathbb{R}^n} \left\{ \left[\sum_{t=1}^{T} (r_{t,\pi_t} - \mu_{\pi_t})^2 \right] + \rho \langle \boldsymbol{\mu}, L_G \boldsymbol{\mu} \rangle \right\}$$

Graph UCB: Leveraging Graph Side Information

Laplacian-regularized least-squares optimization program

$$\hat{\boldsymbol{\mu}}_T = \underset{\boldsymbol{\mu}\in\mathbb{R}^n}{\arg\min} \left\{ \left[\sum_{t=1}^{T} (r_{t,\pi_t} - \mu_{\pi_t})^2 \right] + \rho \langle \boldsymbol{\mu}, L_G \boldsymbol{\mu} \rangle \right\}$$

$\hat{\boldsymbol{\mu}}_T$ Graph-regularized empirical estimate of the reward of all arms

The above problem has a Closed-form solution:

$$\hat{\boldsymbol{\mu}}_T = V(\boldsymbol{\pi}_T, G)^{-1} \left(\sum_{t=1}^{T} \mathbf{e}_{\pi_t} r_{t,\pi_t} \right)$$

$$V(\boldsymbol{\pi}_T, G) \triangleq \sum_{t=1}^{T} \mathbf{e}_{\pi_t} \mathbf{e}_{\pi_t}^{\top} + \rho L_G$$

The **Laplacian matrix** is a matrix that captures characteristics of a Graph

$$L_H = U^T W U = \sum_{e\in\tilde{E}} w_e u_e u_e^T,$$

$W \in \mathbb{R}^{|\tilde{E}|\times|\tilde{E}|}$ is a diagonal matrix with edge weights.

Graph UCB: Leveraging Graph Side Information

Sampling-Policy

Cyclic among clusters

Suboptimal Arm Elimination

1. Identify arm with highest lower bound on its mean estimate

$$a_{\max} = \arg\max_{i \in A} \left[\hat{\mu}_T^i - \beta(t_i) \sqrt{[V_T^{-1}]_{ii}} \right]$$

$$\beta(t) = \left(2R \sqrt{14 \log \left(\frac{2n(t+1)^2}{\delta} \right)} + \rho\epsilon \right)$$

1. Eliminate bad arms

$$A \leftarrow \left\{ \mathbf{a} \in A \mid \hat{\mu}_t^{a_{\max}} - \hat{\mu}_t^a \leq \beta(t_{a_{\max}}) \sqrt{[V_t^{-1}]_{a_{\max} a_{\max}}} + \beta(t_a) \sqrt{[V_t^{-1}]_{aa}} \right\}$$

Sample Complexity

Theorem 1. *Consider n-armed bandit problem with mean vector $\boldsymbol{\mu} \in \mathbb{R}^n$. Let G be a given similarity graph on the vertex set $[n]$, and further suppose that $\boldsymbol{\mu}$ is ϵ-smooth. Let $\mathcal{C}$ be the set of connected components of G. Define*

$$T \triangleq \sum_{C \in \mathcal{C}} \left[\sum_{j \in \mathcal{H} \cap C} \frac{1}{\Delta_j^2} \left[112\sigma^2 \log \left(\frac{112\sigma^2 \sqrt{2} n^{\frac{1}{2}}}{\delta^{\frac{1}{2}} \Delta_j^2} \right) + \frac{\rho\epsilon}{2} \right] - \frac{\mathfrak{I}(j, G)}{2} \right]$$
$$+ \sum_{C \in \mathcal{C}} \max \left\{ \max_{l \in \mathcal{W} \cap C} \mathfrak{I}(l, G), |\mathcal{W} \cap C| \right\} + k(G), \qquad (12)$$

where $k(G) = |\mathcal{C}|$. Then, with probability at least $1-\delta$, GRUB: (a) terminates in no more than T rounds, and (b) returns the best arm $a^* = \arg\max_i \mu_i$.

$$\sum_{i \in B^\star} O\left(\frac{1}{\Delta_i^2}\right) + O(|B^\star|)$$

Delta_i is the gap between the expected rewards of the best arm and arm i, and B* is the set of arms that are “close" in terms of rewards to the best arm.

Graph UCB: Overview

Unfortunately, this algorithm does not work with our reward landscape and graph geometry

We got linear regret compared to O(log n) UCB regret

Why?

Because this algorithm introduce **bias** to the empirical mean estimate using neighbor reward under our reward setting.

Clustering

Update empirical Mean

Sub-optimal arm elimination

Algorithm 1: GRUB

input : Regularization parameter ρ,
Smoothness parameter ϵ, Error bound δ,
Total arms n, Laplacian L_G,
Sub-gaussianity parameter σ

output : A

$t \leftarrow 0$;

$A = \{1, 2, \ldots, n\}$;

$t = 0$;

$V_0 \leftarrow \rho L_G$;

$\mathcal{C}(G) \leftarrow$ `Cluster-Identification`(L_G);

for $C \in \mathcal{C}(G)$ **do**

  $t \leftarrow t + 1$;

  Pick random arm $k \in C$;

  $V_t \leftarrow V_{t-1} + \mathbf{e}_k \mathbf{e}_k^T$, and $\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} + r_{t,k}\mathbf{e}_k$;

end

while $|A| > 1$ **do**

  $t \leftarrow t + 1$;

  $\beta(t) \leftarrow 2\sigma\sqrt{14 \log\left(\frac{2n(t+1)^2}{\delta}\right)} + \rho\epsilon$;

  $k \leftarrow$ `Sampling-Policy`$(t, V_t, A, \mathcal{C}(G))$;

  Sample arm k to observe reward $r_{t,k}$;

  $V_t \leftarrow V_{t-1} + \mathbf{e}_k \mathbf{e}_k^T$, and $\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} + r_{t,k}\mathbf{e}_k$;

  $\hat{\boldsymbol{\mu}}_t \leftarrow V_t^{-1}\mathbf{x}_t$;

  $a_{\max} \leftarrow \arg\max_{i \in A} \left[\hat{\mu}_t^i - \beta(t_i)\sqrt{[V_t^{-1}]_{ii}}\right]$;

  $A \leftarrow \left\{\mathbf{a} \in A \mid \hat{\mu}_t^{a_{\max}} - \hat{\mu}_t^a \leq \beta(t_a)\sqrt{[V_t^{-1}]_{aa}}; \right.$

  $\left. +\beta(t_{a_{\max}})\sqrt{[V_t^{-1}]_{a_{\max}a_{\max}}}\right\}$;

end

return A

NearNeighbor UCB: Proposed Algorithm

Inspired by GraphUCB, we want to leverage such a bias and graph information by introducing **constant graph bias** at the beginning of UCB to help the exploring. This bias will diminish with time.

The proposed algorithm follows general UCB scheme With the following modifications:

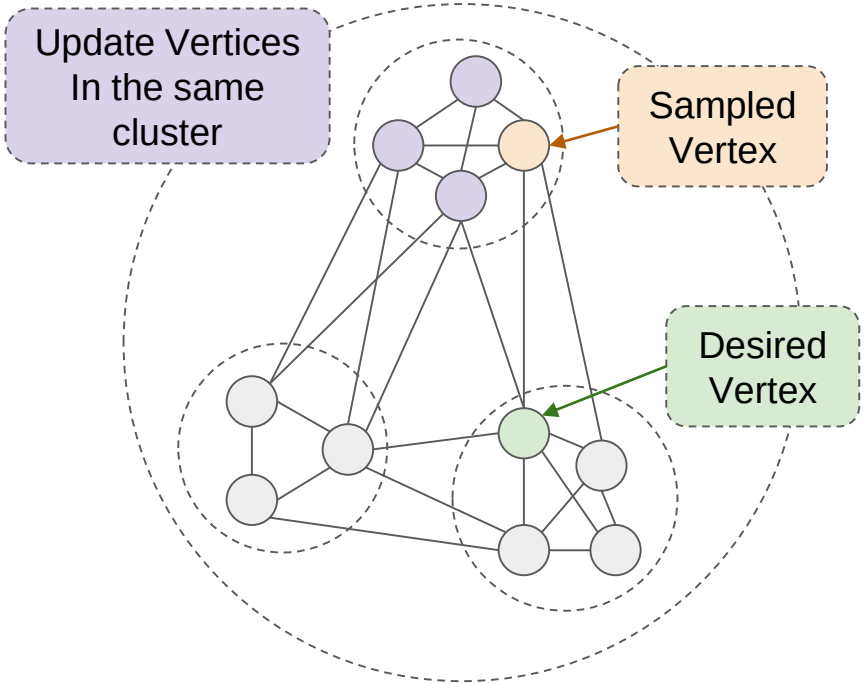
1. When an node is sampled, update all the neighbors **in the same cluster**
2. The update weights is **scaled by similarity** (**1-distance** between the nodes) (j is sampled)

$$\alpha_{i,j} = \text{Similarity between i and j}$$

$$N_{i,T} = N_{i,T-1} + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j}$$

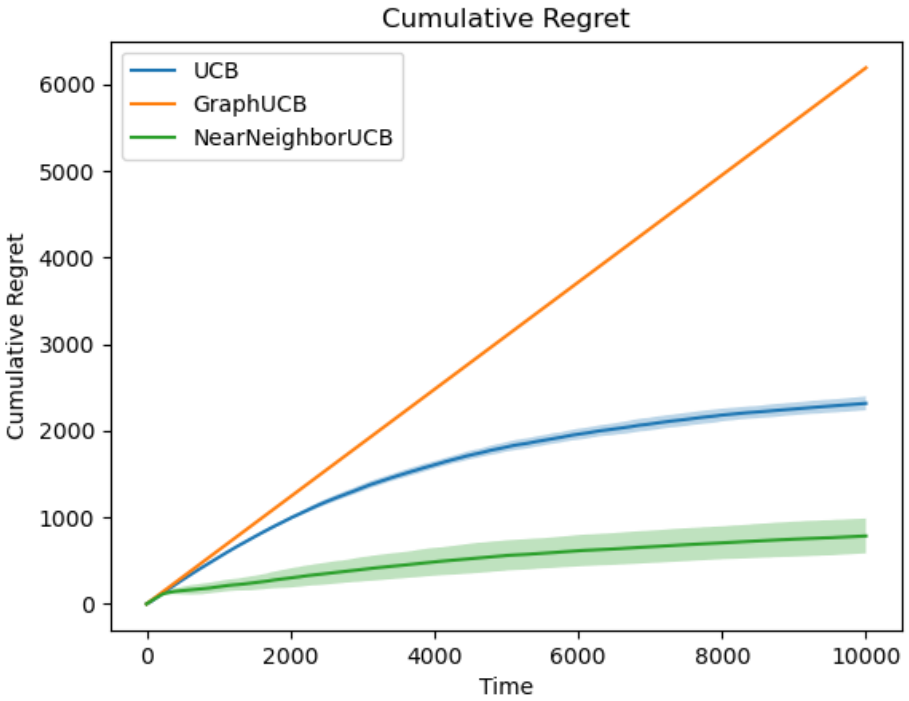
$$R_{i,T} = R_{i,T-1} + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} r_{j,T}$$

1. Only perform 1 and 2 for the **first 100 steps**



Complete Graph - Every two vertices are connected, Not showing all edges

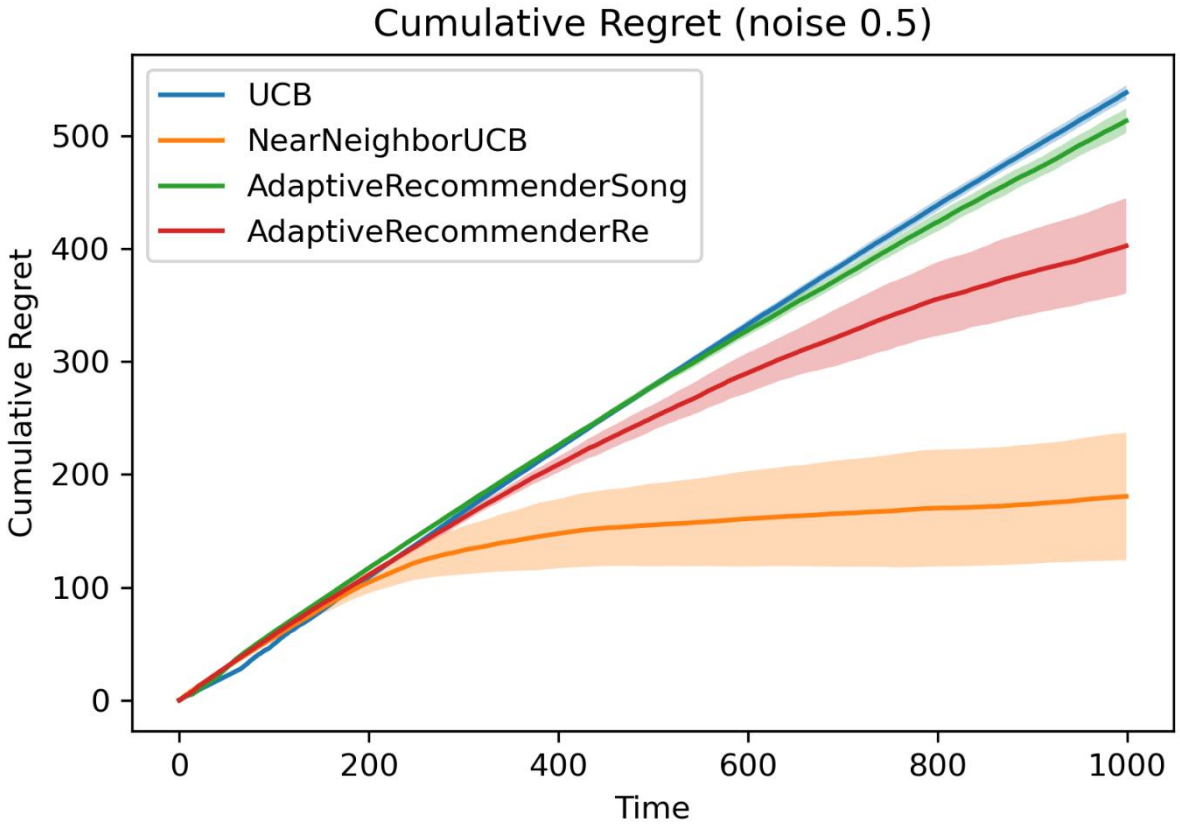
UCB vs GraphUCB vs NearNeighborUCB



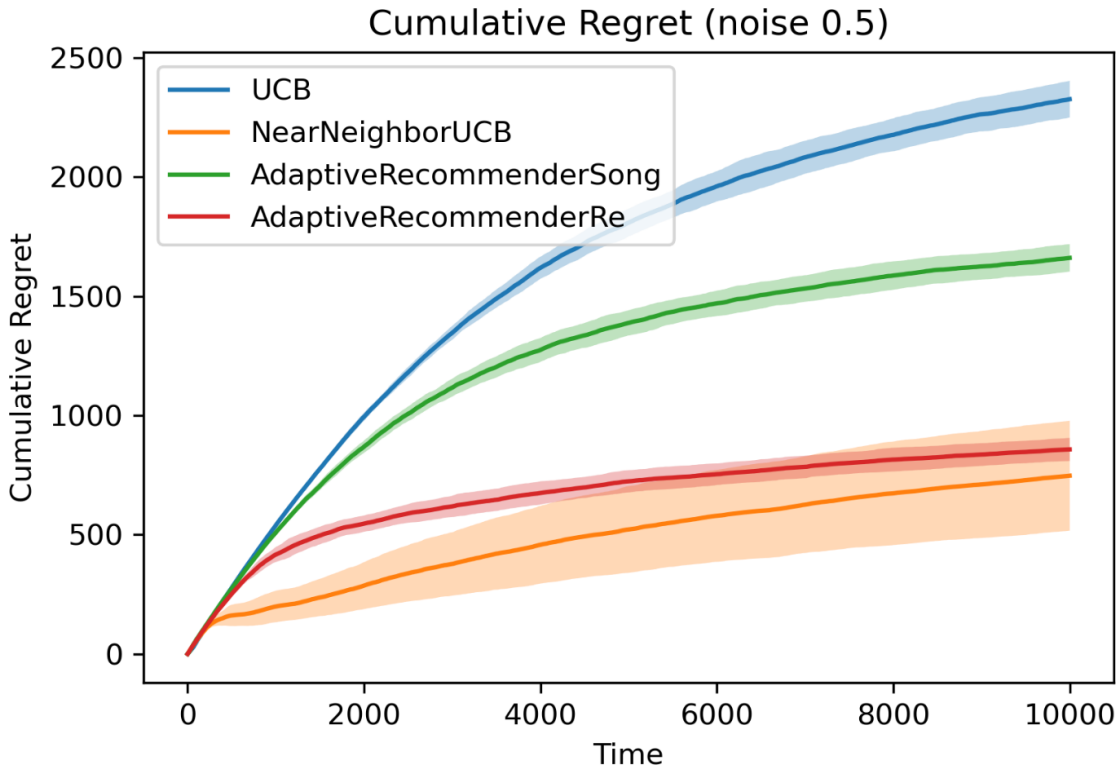
Experiments on Gaussian Noise with std = 0.5

- Graph UCB results in linear regret
- UCB is O(log n)
- NearNeighborUCB(ours) is better than UCB by a constant factor

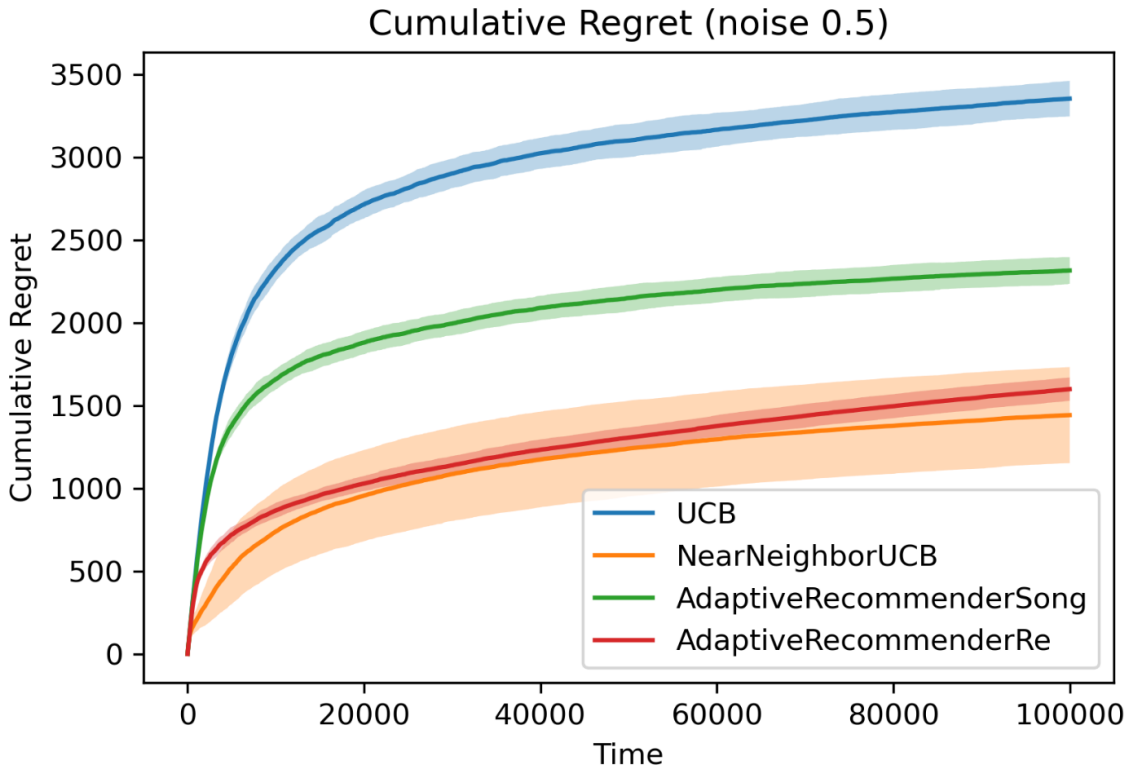
Result 1000 steps



Result 10000 steps



Result 100000 steps



Regre of NearNeighborUCB(ours) is still in the log order!

Discussion

- Though the current trial of leveraging graph side information arms are better than vanilla UCB method, it is still far from satisfactory. The expected performance is to have extremely small sample complexity to find a best arm. For example, find the best arm from 200 arms in 20-50 rounds. We are open to ideas about how to efficiently leverage the graph information among the arms.
- It is also meaningful to explore the approximation solution instead of exact solution. For example we can search on the higher-level clusters(with semantic meaning of surgical plans) and the reward should be given based on the cluster instead of individual images.

Next step

If we find the **coordinates** of images (in high dimension) in psychological space (by manifold learning, eg, multidimensional scaling), can we use the coordinates to facilitate the process?