Neural Contextual Bandits for Personalized Recommendation

Yikun Ban
Yunzhe Qi
Jingrui He

University of Illinois Urbana-Champaign
{Yikunb2, Yunzheq2, Jingrui}@illinois.edu

Time: 9:00 AM – 12:30 PM, 13 May 2024
Location: Virgo 1, Resorts World Sentosa Convention Centre, Singapore
Website: www.banyikun.com/wwwtutorial/
Interactions in Machine Learning

Teacher or Environment

Curated Data or Tasks

Human

Interactions

Results

Interactive Learning (Human)

Interactions:
- Preprocess
- Interpret

Data

Machine Learning Algorithms

Results

Machine Learning (Conventional)

---
Interactions in Machine Learning

Interactive Learning (Human)

Teacher

Environment

Interactions

Curated Data Or Tasks

Human

Results

Interactive Machine Learning

Expert or Environment

Interactions

Data

Machine Learning Algorithms

Results

Interactive Machine Learning (IML) is the core of Artificial Intelligence (AI).

Many IML scenarios can be formulated as **sequential decision-making**.

**One round of Decision-making**

1. **Presented with**
2. **Choose** $K$ Arms (Actions)
3. **Give Feedback**
4. **Receive Feedback**
5. **Improve**

**Learner (Model)**

**Expert/Environment**

**Rewards**

---

Sequential Recommendation: Bandits Formulation

Sequential Recommendation:

1. Presented with $K$ Arms (Actions)
2. Choose
3. Purchase? Yes/No
4. Receive Feedback
5. Improve

One round of Recommendation

Rec. Sys

K Arms (Actions)

Expert/Environment

Rewards

Goal: Maximize \( \sum_{t=1}^{T} \mathbb{E}[r_{t,i_t}] \) Or Minimize \( \sum_{t=1}^{T} (\mathbb{E}[r_{t,i_t}] - \mathbb{E}[r_{t,i_t}]) \), where \( i^*_t = \arg \max_{i \in [K]} \mathbb{E}[r_{t,i}] \).

Round t:

Choose Arm \( i_t \), Observe \( r_{t,i_t} \)
Dilemma of **exploitation** and **exploration** is ubiquitous in **human decision-making**.

**Exploitation:**
- Exploit past data or observations.
- E.g., estimation by a greedy model.

**Exploration:**
- Explore new knowledge for long-term benefit.
- E.g., take uncertain actions.

Dilemma of exploitation and exploration is a fundamental problem in sequential decision-making.

Can you tell me a joke?

Sure, but...

Which one to pick?

ChatGPT

User

Exploitation VS Exploration in Sequential Recommendation

Advantages of Bandit-based Methods

- No Requirement for Large Collected Data.
- Adapt over Time.
- Explicit Exploration.

Tutorial Roadmap

Linear/Neural Contextual Bandits → Collaborative Bandits → Future Trends
Neural Contextual Bandits: Roadmap

**Fundamental Exploration**
- Upper Confidence Bound
- Thompson Sampling
- Exploration Network

**Efficient Exploration**
- Neural Linear UCB
- Neural Network with Perturbed Reward
- Inverse Weight Gap Strategy
Popular existing exploration strategies.

- **$\epsilon$-greedy**: With probability $1 - \epsilon$, greedily choose one arm according to history; Otherwise, choose an arm randomly.

---

Popular existing exploration strategies.

- **ε-greedy**: With probability $1 - \epsilon$, greedily choose one arm according to history; Otherwise, choose an arm randomly.

- **Upper Confidence Bound [1]**:

  ![Diagram of Upper Confidence Bound](image)

  - **Learner (Model)**
  - **Reward estimation** $f(x_{t,i})$
  - **Confidence Interval (Uncertainty)**

Linear Bandits: Joint Problem Definition

In round $t$: A user is serving

- **Arm Pool:**
  - Arm 1
  - Arm 2
  - Arm $k$
  - $d$-dimensional Feature Vector

- **User Pool:**
  - $d$-dimensional Preference Vector (Unknown)

- **Reward:**
  - Linear reward function
  - $r_{t,i} = \theta^T x_{t,i} + \eta_{t,i}$
  - Noise, $\nu$-sub-Gaussian

Linear Bandits: Disjoint Problem Definition

In round $t$: A user is serving

- **Arm Pool**:
  - Arm 1
  - Arm 2
  - Arm $k$
  - $d$-dimensional Feature Vector

- **User Pool**:
  - User 1
  - User 2
  - User $n$
  - $d$-dimensional Preference Vector (Unknown)

- **Reward**:
  - $r_{t,1}$
  - $r_{t,2}$
  - $r_{t,k}$
  - Linear reward function

Given User $j$,

$$r_{t,i} = \theta_j^T x_{t,i} + \eta_{t,i}$$

Noise, $\nu$-sub-Gaussian

---

Linear UCB: Algorithm

Confidence Interval:
\[ |\hat{r}_{t,a} - x_{t,a}^\top \theta^*| \leq (\alpha + 1)s_{t,a}. \]

Joint Linear Models

\[ \text{for } t = 1, 2, 3, \ldots, T \text{ do } \]
\[ \theta_t \leftarrow A^{-1}b \]

Observe \( K \) features, \( x_{t,1}, x_{t,2}, \ldots, x_{t,K} \in \mathbb{R}^d \)

\[ \text{for } a = 1, 2, \ldots, K \text{ do } \]
\[ p_{t,a} \leftarrow \theta_t^\top x_{t,a} + \alpha \sqrt{x_{t,a}^\top A^{-1}x_{t,a}} \{ \text{Computes upper confidence bound} \} \]

end for

Choose action \( a_t = \arg \max_a p_{t,a} \) with ties broken arbitrarily

Observe payoff \( r_t \in \{0, 1\} \)

\[ A \leftarrow A + x_{t,a_t}x_{t,a_t}^\top \]
\[ b \leftarrow b + x_{t,a_t}r_t \]

end for

---

Linear UCB: Regret Analysis

**Confidence Interval:**

With high probability,

\[
|\hat{r}_{t,a} - x_{t,a}^\top \theta^*| \leq (\alpha + 1)s_{t,a}.
\]

where

\[
s_{t,a} = \sqrt{x_{t,a}^\top A_t^{-1} x_{t,a}} \in \mathbb{R}_+
\]

\[
\hat{r}_{t,a} - x_{t,a}^\top \theta^* = x_{t,a}^\top \theta_t - x_{t,a}^\top \theta^*
\]

\[
= x_{t,a}^\top A_t^{-1} b_t - x_{t,a}^\top A_t^{-1} (I_d + D_t^\top D_t) \theta^*
\]

\[
= x_{t,a}^\top A_t^{-1} D_t^\top y_t - x_{t,a}^\top A_t^{-1} (\theta^* + D_t^\top D_t \theta^*)
\]

\[
= x_{t,a}^\top A_t^{-1} D_t^\top (y_t - D_t \theta^*) - x_{t,a}^\top A_t^{-1} \theta^*,
\]

**Regret Upper Bound**

\[
O \left( \sqrt{Td \ln^3 (KT \ln(T)/\delta)} \right).
\]

- **The Number of Rounds**
- **The Number of Items**
- **Dimensionality of Item Context Vector**

---

Neural Bandits: Problem Formulation

In round $t$:

- **Arm Pool**: $x_{t,1}, x_{t,2}, \ldots, x_{t,k}$
  - Arm 1
  - Arm 2
  - Arm k
  - $d$-dimensional Feature Vector

- **User Pool**: $h$
  - User Preference Vector
  - Preference Function (Unknown)

- **Reward**: $r_{t,1}, r_{t,2}, r_{t,k}$
  - $r_{t,i} = h(x_{t,i}) + \eta_{t,i}$
  - General reward function (Linear/Non-Linear)
  - Sub-Gaussian Noise

A sufficiently wide neural network behaves like a linearized model governed by the derivative of network with respect to its parameters (Gradient).

With near-infinite width, Neural network behaves like a kernel predictor with Neural Tangent Kernel (NTK)

$$\Theta(x, x'; \theta) = \nabla_\theta f(x; \theta) \cdot \nabla_\theta f(x'; \theta).$$

Neural UCB: Method

\[ f(x; \theta) = \sqrt{m} W_L \sigma \left( W_{L-1} \sigma \left( \cdots \sigma (W_1 x) \right) \right) \]

\[ U_{t,a} = f(x_{t,a}; \theta_{t-1}) + \gamma_{t-1} \]

\[ \text{mean} \]

\[ \text{variance} \]

\[ \sqrt{Z_{t-1}^{-1} g(x_{t,a}; \theta_{t-1})} / m \]

Compared with LinUCB (Li et al. 2010)

\[ U_{t,a} = \langle x_{t,a}, \theta_{t-1} \rangle + \gamma_{t-1} \sqrt{x_{t,a}^\top Z_{t-1}^{-1} x_{t,a}} \]

\[ \text{mean} \]

\[ \text{variance} \]

In each round, a user is serving

\begin{equation}
\text{for } t = 1, \ldots, T \text{ do}
\end{equation}

\begin{itemize}
\item \text{Observe } \{x_{t,a}\}_{a=1}^{K}
\item \text{for } a = 1, \ldots, K \text{ do}
\begin{itemize}
\item Compute \( U_{t,a} = f(x_{t,a}; \theta_{t-1}) + \gamma_{t-1} \sqrt{g(x_{t,a}; \theta_{t-1})^T Z_{t-1}^{-1} g(x_{t,a}; \theta_{t-1})/m} \)
\item Let \( a_t = \arg\max_{a\in[K]} U_{t,a} \)
\end{itemize}
\end{itemize}

\text{end for}

\text{Play } a_t \text{ and observe reward } r_{t,a_t}

\text{Compute } Z_t = Z_{t-1} + g(x_{t,a_t}; \theta_{t-1}) g(x_{t,a_t}; \theta_{t-1})^T/m

\text{Let } \theta_t = \text{TrainNN}(\lambda, \eta, J, m, \{x_{i,a_t}\}_{i=1}^t, \{r_{i,a_t}\}_{i=1}^t, \theta_0)

\text{Train Neural Networks}

\text{Compute}

\begin{align*}
\gamma_t = & \sqrt{1 + C_1 m^{-1/6} \sqrt{\log m} L^4 t^{7/6} \lambda^{-7/6}} \\
& \left( \nu \frac{\log \det Z_t}{\det \lambda I} + C_2 m^{-1/6} \sqrt{\log m} L^4 t^{5/3} \lambda^{-1/6} - 2 \log \delta + \sqrt{\lambda} S \right) \\
& + (\lambda + C_3 t L) \left[ (1 - \eta m \lambda)^{1/2} \sqrt{t/\lambda} + m^{-1/6} \sqrt{\log m} L^{7/2} t^{5/3} \lambda^{-5/3} (1 + \sqrt{t/\lambda}) \right].
\end{align*}

\text{end for}
Neural UCB: Regret Analysis

- Definition of **NTK Matrix** on all observed contexts of T rounds.

\[
\tilde{H}^{(l)}_{i,j} = \Sigma^{(l)}_{i,j} = \langle x^i, x^j \rangle, \quad A^{(l)}_{i,j} = \begin{pmatrix} \Sigma^{(l)}_{i,i} & \Sigma^{(l)}_{i,j} \\ \Sigma^{(l)}_{j,i} & \Sigma^{(l)}_{j,j} \end{pmatrix},
\]

\[
\Sigma^{(l+1)}_{i,j} = 2E_{(u,v) \sim N(0,A^{(l)}_{i,j})} [\sigma(u)\sigma(v)],
\]

\[
\tilde{H}^{(l+1)}_{i,j} = 2\tilde{H}^{(l)}_{i,j}E_{(u,v) \sim N(0,A^{(l)}_{i,j})} [\sigma'(u)\sigma'(v)] + \Sigma^{(l+1)}_{i,j}.
\]

Then, \( H = (\tilde{H}^{(L)} + \Sigma^{(L)})/2 \) is called the **neural tangent kernel (NTK)** matrix on the context set.

- Analyze dynamics of gradient and NTK regression.

**Assumption:** \( H \preceq \lambda_0 I \).

- Satisfied if no two observed arm contexts are parallel.

**Lemma:** When neural network is wide enough,

\[
h(x^i) = \langle g(x^i; \theta_0), \theta^* - \theta_0 \rangle, \quad \sqrt{m}\|\theta^* - \theta_0\|_2 \leq \sqrt{2h^\top H^{-1}h}, \tag{5.1}
\]

for all \( i \in [TK] \).
Neural UCB: Regret Analysis

Assumption: \( H \succeq \lambda_0 I \).

Satisfied if no two contexts in \( \{x^i\}_{i=1}^{TK} \) are parallel.

\[
\begin{align*}
    h(x^i) &= \langle g(x^i; \theta_0), \theta^* - \theta_0 \rangle, \\
    S &= \sqrt{2h^\top H^{-1} h}, \\
    \tilde{d} &= \frac{\log \det(I + H/\lambda)}{\log(1 + TK/\lambda)}.
\end{align*}
\]

LinUCB:

Let \( h = [h(x^i)]_{i=1}^{TK} \in \mathbb{R}^{TK} \). Set \( J = \tilde{\Theta}(TL/\lambda) \), \( \eta = \Theta((mTL + m\lambda)^{-1}) \) and \( S = 2\sqrt{h^\top H^{-1} h} \). Under the overparameterized setting \( (m \gg 1) \), with probability at least \( 1 - \delta \),

\[
    R_T = \tilde{O}\left(\sqrt{\tilde{d}T} \sqrt{\max\{\tilde{d}, S^2\}}\right).
\]

Theorem

Upper Bound of Neural Parameters

Effective dimension in NTK Space
Neural UCB: Empirical Evaluation

- NeuralUCB uses neural networks for exploitation, and gradient to explore.
- NeuralUCB achieve $\tilde{O}(\sqrt{T})$ regret upper bound, similar to LinearUCB.
- NeuralUCB generally outperforms linear contextual bandits.

Popular existing exploration strategies.

- **$\epsilon$-greedy**: With probability $1 - \epsilon$, greedily choose one arm according to history; Otherwise, choose an arm randomly.

- **Upper Confidence Bound.**

- **Thompson Sampling**:

Linear Thompson Sampling

**Reward Distribution (Gaussian Prior)**

User Preference Parameter (Unknown)

\[ \mathcal{N}(b_i(t)^T \mu, \sigma^2) \]

Arm context

Arm

**for all** \( t = 1, 2, \ldots, \)**

Sample \( \tilde{\mu}(t) \) from distribution \( \mathcal{N}(\mu, \sigma^2 B^{-1}) \).

Play arm \( a(t) := \arg \max_i b_i(t)^T \tilde{\mu}(t) \), and observe reward \( r_t \).

Update

\[
\begin{align*}
B &= B + b_{a(t)}(t) b_{a(t)}(t)^T, \\
\hat{f} &= f + b_{a(t)}(t) r_t, \\
\hat{\mu} &= B^{-1} \hat{f}.
\end{align*}
\]

**end for**

---

Neural Thompson Sampling

Reward Distribution (Gaussian Prior)

\[ N(h(x_t, k), \nu^2) \]

Expected Reward and Variance

Arm

Estimated Distribution:

\[ N(f(x_t, k; \theta_{t-1}), \nu^2 \sigma_{t,k}^2) \]

\[ f(x; \theta) = \sqrt{m} W_L \sigma(W_{L-1} \sigma(\cdots \sigma(W_1 x))) \]

In each round, a user is serving

\[
\text{for } t = 1, \ldots, T \text{ do } \quad K \text{ arms}
\]
\[
\text{for } k = 1, \ldots, K \text{ do }
\]
\[
\sigma_{t,k}^2 = \lambda g^\top(x_{t,k}; \theta_{t-1}) U_{t-1}^{-1} g(x_{t,k}; \theta_{t-1}) / m
\]
\[
\text{Sample estimated reward } \tilde{r}_{t,k} \sim N(f(x_{t,k}; \theta_{t-1}), \nu^2 \sigma_{t,k}^2)
\]
\[
\text{end for}
\]
\[
\text{Pull arm } a_t \text{ and receive reward } r_{t,a_t}, \text{ where } a_t = \arg\max_a \tilde{r}_{t,a}
\]
\[
\text{Set } \theta_t \text{ to be the output of gradient descent for solving (2.3)}
\]
\[
U_t = U_{t-1} + g(x_{t,a_t}; \theta_t) g(x_{t,a_t}; \theta_t) / m
\]
\[
\text{end for}
\]

Compared to NeuralUCB:

\[
U_{t,a} = f(x_{t,a}; \theta_{t-1}) + \gamma_{t-1} \sqrt{g(x_{t,a}; \theta_{t-1})^\top Z_{t-1}^{-1} g(x_{t,a}; \theta_{t-1}) / m}
\]
Neural Thompson Sampling

- NeuralTS and NeuralUCB have similar performance when network is trained every iteration.
- NeuralTS is more robust than NeuralUCB when network is trained in batch.
- NeuralTS introduces more robustness in exploration.

EE-Net: Background

- UCB-based and TS-based exploration highly rely on large-deviation-based statistical confidence interval.

- Ideal scenario:

  - Expected reward
  - Estimated Reward

  50% Pr

  And

  50% Pr

  Symmetric

EE-Net: Motivation

- UCB-based and TS-based exploration highly rely on large-deviation-based statistical confidence bound.

- In practice, may be:

  - Expected reward
    - 80% Pr
  - Estimated Reward
    - And
    - 20% Pr

- Asymmetric
  - 10% Pr
  - And
  - 90% Pr

References:
EE-Net: Motivation

Why making exploration?

Because we cannot make accurate prediction on a subset of data.

Goal of exploration: Fill the gap between expected reward and estimated reward.

Two types of exploration: “Upward” exploration and “downward” Exploration.

- **Upward** Exploration: $h(x_{t,i})$ Expected reward
- **Downward** Exploration: $f_1(x_{t,i}; \theta^1)$ Estimation

**Case 1: Upward Exploration**
- Positive Potential gain: $h(x_{t,i})$
- Underestimation

**Case 2: Downward Exploration**
- Negative Potential gain: $f_1(x_{t,i}; \theta^1)$
- Overestimation

References:
Adapt to Exploration Direction is Challenging

Datasets | Upward Exploration | Downward Exploration
---|---|---
Mnist | 76.3% | 23.7%
Disin | 29.1% | 70.9%
MovieLens | 58.6% | 41.4%
Yelp | 55.3% | 44.7%

Challenge:

Not 50% vs 50%!

Motivation: Can we have an **adaptive** exploration strategy for both “upward” and “downward” exploration?

**Proposed solution:** We propose to use another neural network to learn the gap between expected reward and estimated reward (**potential gain**) incorporating exploration direction.

---

Motivation: Can we have an **adaptive** exploration strategy for both “upward” and “downward” exploration?

**Proposed solution:** We propose to use another neural network to learn the gap between expected reward and estimated reward (**potential gain**) incorporating exploration direction.

**Exploitation neural network** $f_1$ to estimate reward:
- Given an arm $x_{t,i}$,
  \[
  f_1(x_{t,i}; \theta^1) = W_L \sigma(W_{L-1} \sigma(... \sigma(W_1 \cdot )))
  \]
- $f_1(x_{t,i}; \theta^1)$ is to **estimate expected reward** represented by some unknown function $h(x_{t,i})$.
- In round $t$, $\theta^1$ is **trained on data of past $t - 1$ rounds**, using gradient descent.

---

EE-Net: Exploration Neural Networks

- **Exploration neural network** $f_2$ (novel component) to estimate potential gain:
  - Given an arm $x_{t,i}$ and its estimation $f_1(x_{t,i}; \theta^1)$, **expected potential gain** is defined as:
    \[
    h(x_{t,i}) - f_1(x_{t,i}; \theta^1),
    \]
    where $h(x_{t,i})$ is the expected reward.
  - Thus, given the received reward $r_{t,i}$, **potential gain** is defined as:
    \[
    r_{t,i} - f_1(x_{t,i}; \theta^1),
    \]
    where $\mathbb{E}[r_{t,i}] = h(x_{t,i})$.
  - Potential gain has a good property: **Indicating exploration direction**.

EE-Net: Exploration Neural Networks

- **Exploration neural network** $f_2$ (novel component) to estimate potential gain:
  - Potential gain has good property: indicating exploration direction.

$$h(x_{t,i}) - f_1(x_{t,i}; \theta^1) > 0$$

$$h(x_{t,i}) - f_1(x_{t,i}; \theta^1) < 0$$

Case 1: Upward Exploration

Case 2: Downward Exploration

Exploration neural network $f_2$ (novel component) to estimate potential gain:

- Label of $f_2$: $r_{t,i} - f_1(x_{t,i})$

$$f_2(x_{t,i}, \theta^2) = W_L \sigma(W_{L-1} \sigma(... \sigma(W_1 \cdot )))$$

- What is input of $f_2$?

---

Exploration neural network $f_2$ (novel component) to estimate potential gain:

- Input of $f_2$: Gradient of $f_1$ with respect to $\theta^1$:
  $$\nabla_{\theta^1} f(x_{t,i}; \theta^1)$$

Rational:

- Incorporate both feature of input and discriminative information of $f_1$.
- Based on [2,3], $f_1$ has the following confidence bound:
  $$|h(x_{t,i}) - f_1(x_{t,i}; \theta^1_{t-1})| \leq \Psi(\nabla_{\theta^1_{t-1}} f_1(x_{t,i}; \theta^1_{t-1}))$$

Here, instead of choosing a fixed form $\Psi$, we use $f_2$ to learn it.

- In this way, $\theta^2$ is trained on $\{\nabla_{\theta^1} f(x_{t,i}; \theta^1_{t-1})\}_{t=1}^{T}$ to store historical information.
EE-Net: Overview

**Statistical Confidence Interval**

Confidence Interval learned by neural network (Our approach)

<table>
<thead>
<tr>
<th>Methods</th>
<th>&quot;Upward&quot; Exploration</th>
<th>&quot;Downward&quot; Exploration</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε-Greedy</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>NeuralUCB</td>
<td>√</td>
<td>×</td>
</tr>
<tr>
<td>NeuralTS</td>
<td>Randomly</td>
<td>Randomly</td>
</tr>
<tr>
<td>EE-Net</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

EE-Net: Workflow

1. Receive n arms (actions)
2. Calculate exploitation and exploration scores for each arm
3. Pick the arm with maximal exploitation-exploration score
4. Receive feedback; Update exploitation and exploration networks (SGD)

References:
EE-Net: Theoretical Analysis

- Proof Workflow of NeuralUCB [1] and NeuralTS [2]:
  - Gradient Descent in Bandits
  - Gradient Descent on Ridge Regression
  - NTK Regression
  - Expected Reward

- Proof Workflow of EE-Net [3,4]:
  - Gradient Descent in Bandits
  - Online Gradient Descent
  - Expected Reward

Assumption 1: For any $t \in [T], i \in [n], \|x_{t,i}\|_2 = 1$, and $r_{t,i} \in [0, 1]$.

- Assumption 1 is standard and mild in analysis of over-parameterized neural networks.
- **No assumption** on distribution of arm contexts.
- Then, we have the following **average error bound for exploration network $f_2$**:

\[
\hat{i} = \arg\max_{i \in [k]} \left( \frac{f_1(x_{t,i}; \theta_{t-1}^1)}{\sqrt{m}} + \frac{f_2(\phi(x_{t,i}); \theta_{t-1}^2)}{\sqrt{m}} \right).
\]

Then, with probability at least 1 $- \delta$, we have
\[
\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{r_{t,i}} \left[ \min \left\{ \frac{f_2(\phi(x_{t,i}); \theta_{t-1}^2)}{\sqrt{m}} - (r_{t,i} - f_1(x_{t,i}; \theta_{t-1}^1)) / \sqrt{m} \right\}, 1 \right] \leq \sqrt{\frac{\Psi(\theta_0^2, R)}{T}} + \mathcal{O} \left( \frac{3LR}{\sqrt{2T}} \right) + \sqrt{\frac{2\log(\mathcal{O}(1)/\delta)}{T}}.
\]
EE-Net: Theoretical Analysis

**Lemma 1.** For any $\delta \in (0, 1)$, $R > 0$, suppose $m$ satisfies the conditions in Theorem 6. In round $t \in [T]$, let

$$
\hat{i} = \arg \max_{i \in [k]} \left( f_1(x_{t,i}; \theta_{t-1})/\sqrt{m} + f_2(x_{t,i}; \theta_{t-1}^2)/\sqrt{m} \right).
$$

Then, with probability at least $1 - \delta$, we have

$$
\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{r_{t,i}} \left[ \min \left\{ \left| f_2(x_{t,i}; \theta_{t-1}^2)/\sqrt{m} - (r_{t,i} - f_1(x_{t,i}; \theta_{t-1}^2)/\sqrt{m}) \right|, 1 \right\} \right]
\leq \sqrt{\frac{\Psi(\theta_0^2, R)}{T}} + O \left( \frac{3LR}{\sqrt{2T}} \right) + \sqrt{\frac{2 \log(O(1)/\delta)}{T}}.
$$

(1) **Complexity term $\Psi$: Infimum of regression error** caused by function class $B(\theta^2, R)$:

$$
B(\theta_0^2, R) = \{ \tilde{\theta}^2 \in \mathbb{R}^p : \| \tilde{\theta}^2 - \theta_0^2 \|_2 \leq O(\frac{R}{\sqrt{m}}) \}, \quad \Psi(\theta_0^2, R) = \inf_{\theta^2 \in B(\theta_0^2, R)} \sum_{t=1}^{T} (f_2^2(x_{t,i}; \tilde{\theta}^2) - r_{t,i}^2)^2.
$$

(2) **Price** of picking function class $B(\theta^2, R)$ controlled by radius $R$.

(3) **Confidence bound** for predictions of $f_2$.

EE-Net: Regret Upper Bound

Then, we have following regret upper bound $\tilde{O}(\sqrt{T})$ for EE-Net:

**Theorem.** Let $f_1, f_2$ follow the setting of $f$ (Eq. (5.1)) with the same width $m$ and depth $L$. Suppose $m \geq \Omega(\text{poly}(T, L, R, \log(1/\delta)))$, $\eta_1 = \eta_2 = \frac{\sqrt{R}}{m\sqrt{T}}$ and $\Psi(\theta_0^2, R) \& \Psi^*(\theta_0^2, R) \leq \Psi$. Then, for any $\delta \in (0, 1)$, $R > 0$, with probability at least $1 - \delta$ over the initialization, there exists a constant $\nu$, such that the pseudo regret of Algorithm 1 in $T$ rounds satisfies

$$R_T \leq \sqrt{T} \cdot \mathcal{O}(RL + \sqrt{\Psi} + 2\sqrt{2\log(\mathcal{O}(1)/\delta)}) + \mathcal{O}(1)$$

Compared to existing works NeuralUCB [3] and NeuralTS [4]:

$$R_T \leq \mathcal{O}(\sqrt{dT \log T + S^2}) \cdot \mathcal{O}(\sqrt{d \log T}), \quad \text{and} \quad d = \frac{\log \det(I + H/\lambda)}{\log(1 + Tn/\lambda)}$$

1) **[Better Interpretability]**: Have the similar complexity term but $\Psi$ easier to interpret.
2) **[Contexts]**: Allow arm contexts to be repeatedly observed.
3) **[Tighter Bound]**: EE-Net improves by a multiplicative factor $\log T$. 

---

EE-Net: Empirical Experiments

Setup:

- Classification and recommendation dataset.
- 5 state-of-the-art baselines including $\epsilon$-greedy, UCB, TS exploration strategy.
- All methods have the same exploitation network $f_1$.

EE-Net achieves substantial improvements, because all improvements purely come from exploration!

Neural Contextual Bandits: Roadmap

**Fundamental Exploration**
- Upper Confidence Bound
- Thompson Sampling
- Exploration Network

**Efficient Exploration**
- Neural Linear UCB
- Neural Network with Perturbed Reward
- Inverse Weight Gap Strategy
Neural Linear UCB

In each round, a user is serving

\[
\text{for } t = 1, \ldots, T \text{ do}
\]
- receive feature vectors \( \{x_{t,1}, \ldots, x_{t,K}\} \)
- choose arm \( a_t = \arg\max_{k \in [K]} \theta_{t-1}^T \phi(x_{t,k}; w_{t-1}) \)
- reward \( \hat{r}_t \)
- update \( A_t \) and \( b_t \) as follows:
  \[
  A_t = A_{t-1} + \phi(x_{t,a_t}; w_{t-1}) \phi(x_{t,a_t}; w_{t-1})^T, \quad b_t = b_{t-1} + \hat{r}_t \phi(x_{t,a_t}; w_{t-1})
  \]
- if \( \text{mod}(t, H) = 0 \) then
  - \( w_t \leftarrow \text{output of Algorithm 2} \)
  - \( q = q + 1 \)
- else
  - \( w_t = w_{t-1} \)
end if
end for

Output \( w_T \)

Update Neural Network Parameter:

Loss function:

\[
L_q(w) = \sum_{i=1}^{qH} (\theta_i^T \phi(x_{i,a_i}; w) - \hat{r}_i)^2.
\]

- Gradient Descent.

\[
\phi(x; w) = \sqrt{m} \sigma(W_L \sigma(W_{L-1} \cdots \sigma(W_1 x) \cdots)).
\]
In each round, a user is serving

\[
\text{for } t = 1, \ldots, T \text{ do} \\
\text{if } t > K \text{ then} \\
\quad \text{Initialization: Pull each arm once} \\
\quad \text{Pull arm } a_t \text{ and receive reward } r_{t, a_t}, \text{ where } a_t = \arg\max_{i \in [K]} f(x_i, \theta_{t-1}). \\
\quad \text{Generate } \{\gamma_{s}^{t}\}_{s \in [t]} \sim \mathcal{N}(0, \nu^{2}). \\
\quad \text{Set } \theta_t \text{ by the output of gradient descent for solving Eq (3.2).} \\
\text{else} \\
\quad \text{Pull arm } a_K. \\
\text{end if} \\
\text{end for}
\]

\[
\min_{\theta} \mathcal{L}(\theta) = \sum_{s=1}^{t} \left( f(x_{a_s}; \theta) - (r_{s, a_s} + \gamma_{a_s}^{s}) \right)^{2} / 2 + m\lambda\|\theta - \theta_{0}\|_{2}^{2} / 2
\]

Implicit Exploration:

- Back Propagation
- Perturbed Reward
- Received Reward

In each round, a user is serving

for \( t = 1, 2, \ldots, T \) do

- Receive contexts \( \mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,K} \), and compute \( \hat{y}_{t,a} = \hat{f}_t(S; \mathbf{x}_{t,a}, \mathbf{e}_{1:S}) \), \( \forall a \in [K] \)
- Let \( b = \arg\min_a \hat{y}_{t,a} \), \( p_{t,a} = \frac{1}{K + \gamma (\hat{y}_{t,b} - \hat{y}_{t,a})} \), and \( p_{t,b} = 1 - \sum_{a \neq b} p_{t,a} \)
- Sample arm \( a_t \sim p_t \) and observe output \( y_{t,a_t} \)
- Update \( \theta_{t+1} = \prod_{\mathbf{p}^{\text{prob}}_{\rho_0} (\theta_0)} \left( \theta_t - \eta_t \nabla L_{\text{Sq}}^{(S)} (y_{t,a_t}, \{ \hat{f}_t(S; \mathbf{x}_{t,a_t}, \mathbf{e}_{s}) \}_{s=1}^{S}) \right) \).

end for

\( \hat{r}_t = \arg\max f(\mathbf{x}_{t,i}; \theta_t) \)  
\( \mathbb{P}_{t,i} \propto \frac{1}{\hat{r}_t - \hat{r}_{t,i}} \)  
Selection Probability

\( \hat{r}_t \)  
\( \mathbb{P}_{t,1} \uparrow (1) \text{ Less certainty} \)  
\( \mathbb{P}_{t,2} \downarrow (1) \text{ More certainty} \)

### Neural SquareCB: Inverse Gap Strategy

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Regret</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural UCB [Zhou et al., 2020]</td>
<td>$\tilde{O}(\tilde{d}\sqrt{T})$</td>
<td>Bound depends on $\tilde{d}$ and could be $\Omega(T)$ in worst case.</td>
</tr>
<tr>
<td>Neural TS Zhang et al. [2021]</td>
<td>$\tilde{O}(\tilde{d}\sqrt{T})$</td>
<td>Bound depends on $\tilde{d}$ and could be $\Omega(T)$ in worst case.</td>
</tr>
<tr>
<td>EE-Net [Ban et al., 2022b]</td>
<td>$\tilde{O}(\sqrt{T})$</td>
<td>Assumes that the contexts at every round are drawn i.i.d and needs to store all the previous networks.</td>
</tr>
<tr>
<td>NeuSquareCB (This work)</td>
<td>$\tilde{O}(\sqrt{KT})$</td>
<td>No dependence on $\tilde{d}$ and holds even when the contexts are chosen adversarially.</td>
</tr>
</tbody>
</table>

- Remove dependence of effective dimension.
- Minimize dependence on Neural Tangent Kernel.
Takeaways

**Fundamental Exploration**

- Neural UCB [1]  -- An Extension of LinUCB to NTK Space
- Neural TS [2]   -- An Extension of LinTS to NTK Space
- EE-Net [3]      -- Another Neural Network for Exploration

**Efficient Exploration**

- Neural Linear UCB [4]  -- LinUCB with Neural Representation
- Neural Network with Perturbed Reward [5] -- Implicit Exploration by Perturbing Rewards

Tutorial Roadmap

Linear/Neural Contextual Bandits → Collaborative Bandits → Future Trends
Collaborative Bandits

**Introduction**
- Background & Motivations
- Challenges

**Online Clustering of Bandits**
- Clustering of Linear Bandits
- Clustering of Neural Bandits

**Graph Bandit Learning with Collaboration**
- User side: Graph Neural Bandits
- Arm side: Neural Bandit with Arm Group Graph
- Other Scenarios: Bandit Learning with Graph Feedback & Online Graph Classification with Neural Bandit

**Bandits for Combo Recommendation**
- Multi-facet Contextual Bandits
Collaborative Contextual Bandits: Background & Motivation

- Conventional approaches, e.g., **collaborative and content-based filtering**:

<table>
<thead>
<tr>
<th></th>
<th>📖</th>
<th>🛒</th>
<th>🎧</th>
<th>🎮</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>B</td>
<td>✗</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>C</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>D</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>✔️</td>
<td>✔️</td>
<td>?</td>
<td>✗</td>
</tr>
</tbody>
</table>

**Challenges:**

- **Cold-start** problem (Lack of history data);
- **Rapid change** of recommendation content and user interests.
- Dilemma of **Exploitation** and **Exploration**.
- Online recommendation scenario (in each round):

The dilemma of exploitation and exploration:

- Items with sufficient user interactions
- New item
- Select?

Learner

---

Collaborative Contextual Bandits: Background & Motivation

- One user’s decision is affected by other users.

- Motivations: Utilizing the mutual influence / user collaborative effects can
  - Improve recommendation quality.
  - Alleviate the interaction scarcity issue in terms of individual users.
  - Rapidly adapt to new users / items based on interactions with other users.

Collaborative Contextual Bandits: Challenges

- **Challenge #1**: How to formally model user collaborations?
  - User clusters [1, 2, 3, 4, 5, 6, 7], graphs with user nodes [10], etc.

- **Challenge #2**: How to discover user correlations?
  - Leveraging the known user correlation information from the environment [8, 9];
  - User clustering based on their past interactions [2,3,4,5,7], exploitation-exploration graph construction [10].

- **Challenge #3**: How to utilize user correlation to improve recommendation quality?
  - Combination of linear estimations [1, 2, 3, 4, 5, 6], gradient-based meta-learning [7], graph neural networks [10], etc.

2. Li et. al., Improved algorithm on online clustering of bandits. IJCAI 2019.
5. Ban et. al., Local clustering in contextual multi-armed bandits. WWW 2021.
10. Qi et. al., Graph neural bandits. KDD 2023.
Roadmap

Introduction
- Background & Motivations
- Challenges

Online Clustering of Bandits
- Clustering of Linear Bandits
- Clustering of Neural Bandits

Graph Bandit Learning with Collaboration
- User side: Graph Neural Bandits
- Arm side: Neural Bandit with Arm Group Graph
- Other Scenarios: Bandit Learning with Graph Feedback & Online Graph Classification with Neural Bandit

Bandits for Combo Recommendation
- Multi-facet Contextual Bandits
Online Clustering of Bandits

- Two problem settings in standard MAB algorithms:
  - **Ignore user heterogeneity**
  - **Ignore user correlations**

  (1) Joint Modeling

  (2) Disjoint Modeling

- For **trade-off** between user heterogeneity and user correlations:
  - Objective #1: **Identify user clusters** in MAB;
  - Objective #2: **Exploit the user clusters** to improve the recommendation.

2. Li et. al., Improved algorithm on online clustering of bandits. IJCAI 2019.
5. Ban et. al., Local clustering in contextual multi-armed bandits. WWW 2021.
Online Clustering of Linear Bandits

- **Clustering of Linear Bandits:**

  - **Under linear** stochastic contextual bandit settings: \( r = \langle \theta_u, x \rangle + \eta \).

  - **User correlation intensity** between \( u, u' \) is measured by \( \| \theta_u - \theta_{u'} \|_2 \).
    1. User clusters with **identical preferences** \([1, 2, 3, 4, 5]\) \((\forall u, u' \in \mathcal{N}: \theta_u = \theta_{u'})\).
      - Global clustering with evolving connected components

  3. **A generalized formulation**: \( \gamma \)-cluster of users \([6]\) \((\forall u, u' \in \mathcal{N}: \| \theta_u - \theta_{u'} \|_2 \leq \gamma)\).
      - Seed-based Local clustering

---

2. Li et. al., Improved algorithm on online clustering of bandits. IJCAI 2019.
5. Li et. al., Collaborative filtering bandits. SIGIR 2016.
**Challenge 1:** When to ensure a set of identified users is a true cluster?
- Cluster: A set of users with similar expected rewards.
- Expected rewards of users are unknown.

**Challenge 2:** Can we further reduce the clustering complexity?
- Previous works have clustering complexity $O(n)$.
- $n$ is the number of users.

**Challenge 3:** Can we consider and address soft clustering?
- Consider overlapping clusters.
- A user is allowed to belong to multiple clusters.
Characterizing similar users’ behaviors:

- **Definition ($\gamma$-Cluster):** Given a subset of users $\mathcal{N} \subseteq \mathcal{N}$ and a threshold $\gamma > 0$, $\mathcal{N}$ is considered a $\gamma$-Cluster if it satisfies: $\forall i, j \in \mathcal{N}$, $\|\theta_i - \theta_j\| < \gamma$.

**Objectives:**

- **Objective #1:** Identify clusters among users, such that the clusters returned by the proposed algorithm are true $\gamma$–Clusters with probability at least $1-\delta$.
- **Objective #2:** Leverage user clusters to improve the quality of recommendation, evaluated by Regret.

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T} R_t\right] = \sum_{t=1}^{T} \left(\theta_{i_t}^T x_t^* - \theta_{i_t}^T x_t\right)$$

- **Clustering Module + Pulling Module**

---

1. Yikun Ban and Jingrui He. Local Clustering in Contextual Multi-Armed Bandits. WWW 2021.
Identify $k$ clusters, given $k$ seeds in each round:

- **Seed selection**: Randomly choose $k$ users.
- **Neighbors**: Two users are neighbors if they belong to the same $\gamma$-cluster.
- **Potential neighbors**: User $i$ is considered as the potential neighbor of seed user $s$, when:

$$\|\hat{\theta}_{i,t} - \hat{\theta}_{s,t}\| \leq B_{\theta,i}(m_{i,t}, \delta') + B_{\theta,s}(m_{s,t}, \delta').$$

- **Cluster**: Seed user + Its potential neighbors.

**User specific bound**: with a high probability, $\|\hat{\theta}_{i,t} - \theta_i\| \leq B_{\theta,i}(m_{i,t}, \delta')$

$$B_{\theta,s}(m_{i,t}, \delta') = \frac{\sigma \sqrt{2d \log t + 2 \log(2/\delta') + 1}}{\sqrt{1 + h(m_{i,t}, H)}}.$$  

$$h(m_{i,t}, H) = \left(\frac{2m_{i,t}}{4} - 8 \log\left(\frac{m_{i,t} + 3}{H}\right) - 2 \sqrt{m_{i,t} \log\left(\frac{m_{i,t} + 3}{H}\right)}\right).$$

- **Seed-user parameter**
- **User-specific bound**
- **Seed-specific bound**

---

1. Yikun Ban and Jingrui He. Local Clustering in Contextual Multi-Armed Bandits. WWW 2021.
Evolution of neighbors: $\|\hat{\theta}_{i,t} - \hat{\theta}_{s,t} \| \leq B_{\theta,i}(m_{i,t}, \delta') + B_{\theta,s}(m_{s,t}, \delta')$. 

User/seed specific bound is shrinking as more rounds are played for these users.

Termination criterion
- Given cluster $N_{s,t}$, Clustering Module outputs this cluster when

$$\sup \{ B_{\theta,i}(m_{i,t}, \delta') : i \in N_{s,t} \} < \frac{\gamma}{8}$$

1. Yikun Ban and Jingrui He. Local Clustering in Contextual Multi-Armed Bandits. WWW 2021.
**LOCB: Pulling Module**

- **Individual CB vs. Cluster CB**
  - Confidence interval for each cluster
    \[
    P \left( \forall t \in [T], |\hat{\theta}_{N_s,t}^T x_{a,t} - \theta_{N_s,t}^T x_{a,t}| > CB_{r,N_s,t} \right) < \delta'
    \]
  - Confidence interval for each user
    \[
    P \left( \forall t \in [T], |\hat{\theta}_{i,t}^T x_{a,t} - \theta_i^T x_{a,t}| > CB_{r,i} \right) < \delta'
    \]

- **Pulling Module selects one arm by Cluster UCB:**
  \[
  x_t = \arg \max_{x_{a,t} \in X_t} \hat{\theta}_{N_s,t}^T x_{a,t} + CB_{r,N_s,t}
  \]
  Cluster behavior
  \[
  \hat{\theta}_{N_s,t} = \frac{1}{|N_s,t|} \sum_{i \in N_s,t} \hat{\theta}_{i,t}
  \]

- **Cluster-level exploration**

---

1. Yikun Ban and Jingrui He. Local Clustering in Contextual Multi-Armed Bandits. WWW 2021.
A user may belong to multiple overlapping clusters:

- Cluster selection

Pulling Module selects the cluster with the maximum potential:

\[ x_t = \arg \max_{x_{a,t} \in X_t} \max_{s \in S_t(i_t)} \left( \hat{\theta}^T_{N_{s,t}} x_{a,t} + CB_{r,N_{s,t}} \right) \]
LOCB: Results

- **Theoretical analysis:**
  - **Correctness ✓**
    
    **Theorem 5.1 (Correctness).** Given a threshold $\gamma$ and a set of seeds $S \subseteq N$, for each $s \in S$, let $N_s$ represent the cluster output by LOCB with respect to $s$. The terminate criterion of Clustering module is defined as:
    
    \[
    \sup \{ B_{\theta_i}(m_{i,t}, \delta') : i \in N_{x,t} \} < \frac{\gamma}{8}
    \]
    
    Then, with probability at least $1 - \delta$, after the Clustering module terminates, for each $s \in S$, it has
    
    \[\forall i, j \in N_s, \| \theta_i - \theta_j \| < \gamma.\]

  - **Efficiency ✓**
    
    **Theorem 5.2.** Suppose each user is evenly served and $m_{i,t} \geq \frac{2x2^i}{\lambda} \log \left( \frac{2n}{\delta} \right) \log \left( \frac{32^i}{3} \log \left( \frac{2n}{\delta} \right) \right)$ for any $i \in N$. Then, with probability at least $1 - \delta$, the number of rounds $\hat{T}$ needed for the Clustering module to terminate is upper bounded by
    
    \[\hat{T} < \frac{2n}{C} \log \left( \frac{32^i}{3} \log \left( \frac{2n}{\delta} \right) \right) \left( \log \left( \frac{2(d+1)N}{\delta} \right) - \frac{\gamma^2 - 256}{512\sigma^2} \right) + n.\]
    
    where $C = \frac{\lambda^2}{16\sigma^2}$.

  - **Effectiveness ✓**
    
    **Theorem 5.3.** Suppose that each user is evenly served. Given $\gamma$ and a set of seeds $S$, after $T > \hat{T}$ rounds, the accumulated regret of LOCB can be upper bounded as follows:
    
    \[R_T \leq \sqrt{nT \cdot \sqrt{2d \log (1 + T/dn)} \cdot O \left( \sqrt{d \log (T/d)} \right)} + \left( T - O(n \log nd) \right) \gamma + O(n \log nd) \cdot O \left( \sqrt{d \log (T/n)} \right).\]

- **Evaluations:**
  - **Improve performance up to 12.4%**.

<table>
<thead>
<tr>
<th></th>
<th>Synthetic</th>
<th>Yelp</th>
<th>MovieLens</th>
<th>Yahoo</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>Pre</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>N-CLUB</td>
<td>0.390</td>
<td>0.246</td>
<td>0.943</td>
<td>0.484</td>
</tr>
<tr>
<td>ST-CLUB</td>
<td>0.578</td>
<td>0.549</td>
<td>0.612</td>
<td>0.626</td>
</tr>
<tr>
<td>ST-SCLUB</td>
<td>0.714</td>
<td>0.745</td>
<td>0.687</td>
<td>0.768</td>
</tr>
<tr>
<td>N-LOCB</td>
<td>0.662</td>
<td>0.618</td>
<td>0.714</td>
<td>0.675</td>
</tr>
<tr>
<td>LOCB</td>
<td><strong>0.880</strong></td>
<td><strong>0.913</strong></td>
<td><strong>0.856</strong></td>
<td><strong>0.879</strong></td>
</tr>
</tbody>
</table>

1. Yikun Ban and Jingrui He. Local Clustering in Contextual Multi-Armed Bandits. WWW 2021.
Roadmap

Introduction
- Background & Motivations
- Challenges

Online Clustering of Bandits
- Clustering of Linear Bandits
- Clustering of Neural Bandits

Graph Bandit Learning with Collaboration
- User side: Graph Neural Bandits
- Arm side: Neural Bandit with Arm Group Graph
- Other Scenarios: Bandit Learning with Graph Feedback & Online Graph Classification with Neural Bandit

Bandits for Combo Recommendation
- Multi-facet Contextual Bandits
Online Clustering of Neural Bandits

- **Challenge 1**: How to efficiently determining a user’s relative group?
  - User relative group: A set of users with **same expected rewards on a specific item (arm)**.
  - Expected rewards of users are unknown. The mapping function $h(x)$ can be linear or non-linear.

- **Challenge 2**: Effective parametric representation of dynamic clusters?
  - Introducing **meta-learner** capable of representing and swiftly adapting to evolving user clusters.
  - Enabling the rapid acquisition of nonlinear cluster representations.

- **Challenge 3**: Balancing exploitation and exploration?
  - A novel UCB-type exploration strategy.
  - Taking both user-side and meta-side information into account.

---

1. Yikun Ban et. al., Meta Clustering of Neural Bandits. In submission.
Characterizing user clusters without linear assumptions:

**Definition 3.1** (Relative Cluster). In round \( t \), given an arm \( x_{t,i} \in X_t \), a relative cluster \( N(x_{t,i}) \subseteq N \) with respect to \( x_{t,i} \) satisfies

\[
\begin{align*}
(1) & \quad \forall u, u' \in N(x_{t,i}), \mathbb{E}[r_{t,i}|u] = \mathbb{E}[r_{t,i}|u'] \\
(2) & \quad \nexists N' \subseteq N, \text{s.t. } N' \text{ satisfies (1) and } N(x_{t,i}) \subseteq N'.
\end{align*}
\]

**Definition 3.2** (\( \gamma \)-gap). Given two different cluster \( N(x_{t,i}), N'(x_{t,i}) \), there exists a constant \( \gamma > 0 \), such that

\[
\forall u \in N(x_{t,i}), u' \in N'(x_{t,i}), |\mathbb{E}[r_{t,i}|u] - \mathbb{E}[r_{t,i}|u']| \geq \gamma.
\]

**Objectives:**

- **Objective #1**: Identify clusters among users, such that the clusters returned by the proposed algorithm are accurate user clusters.
- **Objective #2**: Leverage user correlations to improve the quality of recommendation, evaluated by Pseudo Regret.

\[
R_T = \sum_{t=1}^{T} \mathbb{E}[r_t^* - r_t | u_t, X_t],
\]

**General reward function**

\[
\mathbb{E}[r_t^* | u_t, X_t] = \max_{x_{t,i} \in X_t} h_{u_t}(x_{t,i})
\]
M-CNB: Clustering Module

- **Identify relative cluster for target user** $u_t \in N$:
  - **Arm-specific**: Different arms can induce distinct user clusters.
  - **User models**: Each user $u \in N$ is assigned with their own user models $f(\cdot; \theta^u)$.
  - **Potential neighbors**: User $u$ is the potential neighbor of target user $u_t$, when:

$$\hat{N}_{u_t}(x_{t,i}) = \{u \in N \mid |f(x_{t,i}; \theta^u_{t-1}) - f(x_{t,i}; \theta^u_t)| \leq \frac{\nu - 1}{\nu} \gamma\}.$$  

- **Meta-adaptation**: Adapting to estimated user clusters.
  - Randomly draw a few samples from the historical data of detected cluster $\{T^u_{t-1}\}_{u \in \hat{N}_{u_t}(x_{t,i})}$.
  - The meta-model $f(\cdot; \Theta)$ is adapted through a few steps of SGD.

---

1. Yikun Ban et al., Meta Clustering of Neural Bandits. In submission.
M-CNB: Pulling Module

Informative UCB for reward estimation:

\[
\sum_{t=1}^{T} \mathbb{E}_{r_t|x_t} \left[ |f(x_t; \Theta_t) - r_t| \mid u_t \right] 
\leq \sum_{t=1}^{T} \frac{O(\|\nabla \theta f(x_t; \Theta_t) - \nabla \theta f(x_t; \theta^u_0)\|_2)}{m^{1/4}} + \sum_{u \in \mathcal{N}} \mu^u_T \left( \sqrt{\frac{S + 1}{2 \mu^u_T}} + \sqrt{\frac{2 \log(1/\delta)}{\mu^u_T}} \right),
\]

Pulling Module selects one arm by Cluster UCB:

\[
x_t = \arg_{x_{t,i} \in X_t} \max U_{t,i}
\]

\[
U_{t,i} = f(x_{t,i}; \Theta_{t,i}) + \frac{\|\nabla \theta f(x_{t,i}; \Theta_{t,i}) - \nabla \theta f(x_{t,i}; \theta^u_0)\|_2}{m^{1/4}} + \sqrt{\frac{S + 1}{2 \mu^u_t}} + \sqrt{\frac{2 \log(1/\delta)}{\mu^u_t}}
\]

Gradient Discrepancy between User Model and the Meta-Model

User-side Upper Bound based on Service Frequency

1. Yikun Ban et al., Meta Clustering of Neural Bandits. In submission.
M-CNB: Theoretical and Empirical Results

Theoretical analysis from two aspects:

- **Instance-dependent Regret Bound**

  **Theorem 5.1.** Given the number of rounds $T$ and $y$, for any $\delta \in (0, 1), R > 0$, suppose $m \geq \tilde{\Omega}(\text{poly}(T, L, R) \cdot Kn \log(1/\delta))$, $\eta_1 = \eta_2 = R^\gamma$, and $\mathbb{E}(|N_{u_t}(x_t)|) = \frac{n}{q}, t \in [T]$. Then, with probability at least $1 - \delta$ over the initialization, Algorithm 1 achieves the following regret upper bound:

  $R_T \leq \sqrt{qT \cdot S^*_{TK} + O(1) + O(\sqrt{2qT \log(O(1)/\delta)})}.

  where $S^*_{TK} = \inf_{\theta \in B(\theta_0, R)} \sum_{t=1}^{TK} L_t(\theta)$.

- **NTK-regression based Regret Bound**

  **Lemma 5.3.** Suppose Assumption 5.1 and conditions in Theorem 5.1 holds where $m \geq \tilde{\Omega}(\text{poly}(T, L) \cdot Kn \lambda_0^{-1} \log(1/\delta))$. With probability at least $1 - \delta$ over the initialization, there exists $\theta' \in B(\theta_0, \tilde{\Omega}(T^{3/2}))$, such that

  $\mathbb{E}[S^*_{TK}] \leq \mathcal{O}\left[\sum_{t=1}^{TK} \mathcal{L}_t(\theta')\right] \leq \tilde{O}\left(\sqrt{d + S} \right)^2 \cdot \tilde{d}$.

Evaluations:

- M-CNB (red curve) outperforms baselines, for both recommendation and classification data sets.

---

1. Yikun Ban et al., Meta Clustering of Neural Bandits. In submission.
Online Clustering of Bandits

Motivations: We need to estimate user correlations on the fly, during online recommendation.

Clustering of Linear Bandits [1, 2, 3, 4, 5, 6]:

- Under linear stochastic contextual bandit settings: \( r = \langle \theta_u, x \rangle + \eta \).
- User correlation intensity between \( u, u' \) is measured by \( \|\theta_u - \theta_{u'}\|_2 \).
- Adopt combination of linear estimators for reward estimation & exploration.

Clustering of Neural Bandits [7]:

- Under neural stochastic contextual bandit settings: \( r = h_u(x) + \eta \).
- User clusters with identical preferences (\( \forall u, u' \in \mathcal{N}, x \in \mathbb{R}^d : h_u(x) = h_{u'}(x) \)).
- Utilizing gradient-based Meta-Learning for reward estimation & exploration.

2. Li et. al., Improved algorithm on online clustering of bandits. IJCAI 2019.
5. Ban et. al., Local clustering in contextual multi-armed bandits. WWW 2021.
Roadmap

Introduction
- Background & Motivations
- Challenges

Online Clustering of Bandits
- Clustering of Linear Bandits
- Clustering of Neural Bandits

Graph Bandit Learning with Collaboration
- User side: Graph Neural Bandits
- Arm side: Neural Bandit with Arm Group Graph
- Other Scenarios: Bandit Learning with Graph Feedback & Online Graph Classification with Neural Bandit

Application in Recommender Systems
- Multi-facet Personalized Recommendation
Collaborative Exploration: Graph Bandits Learning

Clustering of Bandits \([1,2]\)

- **Coarse-grained** user correlations:
  - Users within the same cluster share **identical preferences**.
  - Contribute **equally** to serving user.

Graph Bandits Learning \([3]\)

- **Fine-grained** user correlations:
  - Heterogeneity of users is preserved.
  - Contribute **differently** to serving user.

3. Y. Qi, Y. Ban*, and J. He. Graph neural bandits. KDD 2023.
GNB: Exploitation and Exploration Graphs

User Exploitation Graph

\[ \text{Correlation w.r.t. Exploitation} \]

User Exploration Graph

\[ \text{Correlation w.r.t. Exploration} \]

1. Yunzhe Qi*, Yikun Ban*, and Jingrui He. Graph neural bandits. KDD 2023.
For each round $t \in [T]$:
- Receive a target user $u_t \in \mathcal{U}$, and candidate arms (items) $\mathcal{X}_t$.
  - $\mathcal{X}_t = \{x_{i,t} \in \mathbb{R}^d, \ (e.g., \ (\circ \square) ) \}_{i \in [a]}$

- Reward $r_{i,t} = h\left(G^{(1)}_{i,t}, u_t, x_{i,t}\right) + \varepsilon_{i,t}$.
  - Zero-mean noise

- Learner selects arm $x_t \in \mathcal{X}_t$ as the recommendation.
For each round \( t \in [T] \):
- Receive a target user \( u_t \in \mathcal{U} \), and candidate arms (items) \( \mathcal{X}_t \).
  - \( \mathcal{X}_t = \{ \mathbf{x}_{i,t} \in \mathbb{R}^d, \text{ (e.g., )} \}_{i \in [a]} \)
- Reward \( r_{i,t} = h(G_{i,t}^{(1),*}, u_t, \mathbf{x}_{i,t}) + \epsilon_{i,t} \).
- Learner selects arm \( x_t \in \mathcal{X}_t \) as the recommendation.

Definition: User Correlation (Exploitation) Graph
- Given arm \( x_{i,t} \), unknown user exploitation graph \( G_{i,t}^{(1),*} = (\mathcal{U}, E, W_{i,t}^{(1),*}) \):
  - \( \mathcal{U} \): set of nodes (users)
  - \( E = \{ e(u, u') \}_{u, u' \in \mathcal{U}} \): set of edges
  - \( W_{i,t}^{(1),*} \): set of edge weights \( W_{i,t}^{(1),*} = \psi(1)(\mathbb{E}[r_{i,t} | \mathbf{u}_1, \mathbf{x}_{i,t}], \mathbb{E}[r_{i,t} | \mathbf{u}_2, \mathbf{x}_{i,t}]) \).

User correlations w.r.t. the expected reward (Exploitation Graph)
**Definition: User Exploration Graph**

- For arm $x_{i,t}$, unknown user exploration graph
  
  $\mathcal{G}^{(2),*}_{i,t} = (\mathcal{U}, E, W_{i,t}^{(2),*})$

  Set of edge weights

- For users $u_1, u_2 \in \mathcal{U}$, the corresponding edge weight:
  
  $w_{i,t}^{(2),*}(u_1, u_2) = \Psi^{(2)} \left( E[r_{i,t} | u_1, x_{i,t}] - f_{u_1}^{(1)}(x_{i,t}),
  
  E[r_{i,t} | u_2, x_{i,t}] - f_{u_2}^{(1)}(x_{i,t}) \right)$

**Potential Gain:**

- $\mathbb{E}[r \mid u, x] - f_u^{(1)}(x)$
- Measures the uncertainty for the reward estimation

---

1. Yunzhe Qi, Yikun Ban*, and Jingrui He. Graph neural bandits. KDD 2023.
For each round $t \in [T]$:  
- Receive a target user $u_t \in \mathcal{U}$, and candidate arms $\mathcal{X}_t$.  
  - $\mathcal{X}_t = \{x_{i,t} \in \mathbb{R}^d, \text{ (e.g., } \bigcirc \text{) } \}_{i \in [a]}$  

- Reward $r_{i,t} = h \left( g_{i,t}^{(1)\ast}, u_t, x_{i,t} \right) + \epsilon_{i,t}$.

- Learner selects arm $x_t \in \mathcal{X}_t$ as the recommendation.

**Objective: Minimizing Pseudo Regret**

$$R(T) = \sum_{t=1}^{T} \mathbb{E} [r_t^* - r_t]$$

**Definition: User Correlation (Exploitation) Graph**

- Given arm $x_{i,t}$, unknown user exploitation graph $g_{i,t}^{(1)\ast} = (\mathcal{U}, E, W_{i,t}^{(1)\ast})$
  - $W_{i,t}^{(1)\ast}$: set of edge weights

- For users $u_1, u_2 \in \mathcal{U}$, the corresponding edge weight:
  - $w_{i,t}^{(1)\ast}(u_1, u_2) = \psi^{(1)} \left( \mathbb{E}[r_{i,t} | u_1, x_{i,t}], \mathbb{E}[r_{i,t} | u_2, x_{i,t}] \right)$
GNB: Framework Overview

1. User Graph Estimation

2. Reward & Potential Gain Estimation

3. Arm Selection

4. Parameter Update via GD

Get $V^{(1)}_{g_{nn}}$ & Feed to

Exploitation GNN $f^{(1)}_{g_{nn}}$($\cdot$)

$w_{u,t}^{(1)*} = \psi^{(1)}(\mathbb{E}[r_{u,t} | u_1, x_{u,t}], \mathbb{E}[r_{u,t} | u_2, x_{u,t}])$

Exploitation User Graph Est.

User-Arm Pair $(u_t, x_{u,t})$

Exploitation User Graph Est.

$g_{u,t}^{(1)}$

Reward Est.

$\hat{r}_{u,t}$

Potential Gain Est.

$\hat{b}_{u,t}$

Select Arm $x_t = \arg\max_{x_{u,t} \in x_t} (\hat{r}_{u,t} + \hat{b}_{u,t})$

Get $V^{(2)}_{g_{nn}}$ & Feed to

Exploration GNN $f^{(2)}_{g_{nn}}$($\cdot$)

$w_{u,t}^{(2)*} = \psi^{(2)}(\mathbb{E}[r_{u,t} | u_1, x_{u,t}] - f^{(1)}_{u_1}(x_{u,t}), \mathbb{E}[r_{u,t} | u_2, x_{u,t}] - f^{(1)}_{u_2}(x_{u,t}))$

Exploration User Graph Est.

True Edge Weight to Estimate:

True Edge Weight to Estimate:

1. Yunzhe Qi, Yikun Ban*, and Jingrui He. Graph neural bandits. KDD 2023.
User Exploitation Graph Estimation

User Preference (expected reward) Estimation:
- Estimated by user exploitation networks
  \( \{ f_u^{(1)} \}_{u \in \mathcal{U}} \)
- Approximating \( \mathbb{E}[r | u, x] \)
- Input: \( x \)  Label: \( r \)

- User Exploitation Graph Estimation:
  - Given arm \( x_{i,t} \), estimated user exploitation graph
    \( \mathcal{G}_{i,t}^{(1)} = (\mathcal{U}, E, W_{i,t}^{(1)}) \)
    - \( W_{i,t}^{(1)} \): set of estimated edge weights
  - For users \( u_1, u_2 \in \mathcal{U} \), estimated edge weight
    \( w_{i,t}^{(1)}(u_1, u_2) = \Psi^{(1)}(f_{u_1}^{(1)}(x_{i,t}), f_{u_2}^{(1)}(x_{i,t})) \)

User correlations w.r.t. the expected reward (Exploitation Graph)
User Exploration Graph Estimation

Potential Gain:
- Estimated by user exploration networks
  \[ \{ f_{u}^{(2)} \}_{u \in \mathcal{U}} \]
  - Input: \( \nabla f_{u}^{(1)}(x) \) -- the gradients of \( f_{u}^{(1)} \).
  - Label: \( r_u - f_{u}^{(1)}(x) \).

User Exploration Graph Estimation:
- Given arm \( x_{i,t} \), estimated user exploration graph
  \[ G_{i,t}^{(2)} = (\mathcal{U}, E, W_{i,t}^{(2)}) \]
  Edge weight estimations

- For users \( u_1, u_2 \in \mathcal{U} \), estimated edge weight
  \[ w_{i,t}^{(2)}(u_1, u_2) = \psi^{(2)} \left( f_{u_1}^{(2)}(\nabla f_{u_1}^{(1)}(x_{i,t})), f_{u_2}^{(2)}(\nabla f_{u_2}^{(1)}(x_{i,t})) \right) \]

Estimated Potential Gain

User correlations w.r.t. the Potential Gain (Exploration Graph)

---
1. Yunzhe Qi*, Yikun Ban*, and Jingrui He. Graph neural bandits. KDD 2023.
1. User Graph Estimation

2. Reward & Potential Gain Estimation

3. Arm Selection

4. Parameter Update via GD

True Edge Weight to Estimate:
\[ w_{t}^{(1)}(\cdot) = \psi_{t}^{(1)}(E[r_{t}, u_{1}, x_{t}], E[r_{t}, u_{2}, x_{t}]) \]

True Edge Weight to Estimate:
\[ w_{t}^{(2)}(\cdot) = \psi_{t}^{(2)}(E[r_{t}, u_{1}, x_{t}] - f_{u_{1}}^{(1)}(x_{t}), E[r_{t}, u_{2}, x_{t}] - f_{u_{2}}^{(1)}(x_{t})) \]

Exploitation User Graph Est.

Exploration User Graph Est.

Reward Est.
\[ \hat{r}_{i,t} \]

Potential Gain Est.
\[ \hat{b}_{i,t} \]

Select Arm
\[ x_{t} = \arg\max_{x_{t} \in X_{t}} (\hat{r}_{i,t} + \hat{b}_{i,t}) \]

User-Arm Pair
\[ (u_{i}, x_{t}) \]

Get \( f_{g_{2}}^{(2)}(\cdot) \) & Feed to

Get \( f_{g_{1}}^{(1)}(\cdot) \) & Feed to

Exploitation GNN \( f_{g_{1}}^{(1)}(\cdot) \)

Exploration GNN \( f_{g_{2}}^{(2)}(\cdot) \)

Exploitation GNN

Yunzhe Qi*, Yikun Ban*, and Jingrui He. Graph neural bandits. KDD 2023.
For each arm $x_{i,t} \in \mathcal{X}_t$, reward estimation with estimated user exploitation graph $g_{i,t}^{(1)}$.

Given target user $u_t$, obtain User-specific Arm Representation $H_{agg}$:

- Choose the rep. for target user $u_t$
- $H_{agg}^{(1)}$
For each arm $x_{i,t} \in \mathcal{X}_t$, reward estimation with estimated user exploitation graph $g_{i,t}^{(1)}$. 

Point reward estimation for each arm $x_{i,t} \in \mathcal{X}_t$:

$\hat{r}_{i,t} = f_{gnn}^{(1)}(x_{i,t}, g_{i,t}^{(1)}; [\Theta_{gnn}^{(1)}]_{t-1})$

Choose the rep. for target user $u_t$

$H_{agg}^{(1)}$
GNB: Potential Gain Estimation

- For each arm $x_{i,t} \in X_t$, reward estimation with estimated user exploration graph $G_{i,t}^{(2)}$.
- Potential gain estimation for each arm $x_{i,t} \in X_t$:

$$\hat{b}_{i,t} = f_{gnn}^{(2)} \left( \nabla f_{gnn}^{(1)}(G_{i,t}^{(1)}, x_{i,t}; \Theta_{gnn}^{(1)}_{t-1}) \right)$$

**Input:** Gradients of $f_{gnn}^{(1)}$ in terms of $x_{i,t}$

**Fully-Connected layer for user $u_1$**

**Fully-Connected layer for user $u_2$**

**...**

**Fully-Connected layer for user $u_N$**

**Aggregation on $G_{i,t}^{(2)}$ with GNN for $k$-hops**

**Choose the rep. for target user $u_t$**

- Yunzhe Qi*, Yikun Ban*, and Jingrui He. Graph neural bandits. KDD 2023.
GNB: Framework Overview

1. User Graph Estimation
2. Reward & Potential Gain Estimation
3. Arm Selection
4. Parameter Update via GD

True Edge Weight to Estimate:

\[ w_{i,t}^{(1)} = \psi^{(1)} \left( \mathbb{E}[r_{i,t} | u_1, x_{i,t}], \mathbb{E}[r_{i,t} | u_2, x_{i,t}] \right) \]

Feed to Exploitation GNN \( f_{gNN}^{(1)}(\cdot) \)

Get \( V f_{gNN}^{(1)}(\cdot) \) & Feed to Exploitation GNN \( f_{gNN}^{(1)}(\cdot) \)

True Edge Weight to Estimate:

\[ w_{i,t}^{(2)} = \psi^{(2)} \left( \mathbb{E}[r_{i,t} | u_1, x_{i,t}] - f_{u_1}^{(1)}(x_{i,t}), \mathbb{E}[r_{i,t} | u_2, x_{i,t}] - f_{u_2}^{(1)}(x_{i,t}) \right) \]

Feed to Exploration GNN \( f_{gNN}^{(2)}(\cdot) \)

Get \( V f_{gNN}^{(2)}(\cdot) \) & Feed to Exploration GNN \( f_{gNN}^{(2)}(\cdot) \)

\[ \hat{r}_{i,t} \]

\[ \hat{b}_{i,t} \]

Select Arm \( x_t = \arg\max_{x_t \in X_t} (\hat{r}_{i,t} + \hat{b}_{i,t}) \)

1. Yunzhe Qi*, Yikun Ban*, and Jingrui He. Graph neural bandits. KDD 2023.
GNB: Arm Selection & Training

- **Arm Selection Strategy:**
  - Select arm $x_t = \text{argmax}_{x_{i,t} \in X_t} \left( \hat{r}_{i,t} + \hat{b}_{i,t} \right)$.
  - Receive the corresponding true reward $r_t$.

- Train the model parameters (User models + GNN models) with **Gradient Descent (GD)**.

- Update user graphs.

---

1. Yunzhe Qi*, Yikun Ban*, and Jingrui He. Graph neural bandits. KDD 2023.
GNB: Theoretical Analysis

- Pseudo regret for $T$ rounds:

$$R(T) = \sum_{t=1}^{T} \mathbb{E}[r_t^* - r_t]$$

- Given sufficiently large network width $m$ (over-parameterization), under mild assumptions, with the probability at least $1 - \delta$:

$$R(T) \leq \sqrt{T} \cdot O(L \xi_L) \cdot \sqrt{2 \log\left(\frac{Tn \cdot a}{\delta}\right)} + \sqrt{T} \cdot O(L) + O(\xi_L) + O(1).$$

where $n$ is the number of users, $a$ is the number of arms in each round, and $T$ is the number of rounds.

Remarks:

- Achieves the regret bound of $O(\sqrt{T \log(nT)})$.
  - Existing works with user clustering need $O(\sqrt{nT \log(T)})$ for user collaboration modeling.

- Free of the terms $d$
  - $d$ (arm context dimension, common in linear bandit works)

Experiments: Real Data Sets

- **Experiment settings:**
  - Under **online recommendation** settings, we evaluate the proposed GNB framework on **six** real data sets with different specifications.
  - We include **nine** state-of-the-art related algorithms as the baselines, including both linear and neural algorithms.

- **Summary of experiment results:**
  - **Neural algorithms** generally perform better than **linear ones**, with the representation power of neural networks.
  - GNB can generally achieve the **best performance** against the strong baselines.

---

Roadmap

**Introduction**
- Background & Motivations
- Challenges

**Online Clustering of Bandits**
- Clustering of Linear Bandits
- Clustering of Neural Bandits

**Graph Bandit Learning with Collaboration**
- User side: Graph Neural Bandits
- Arm side: Neural Bandit with Arm Group Graph
- Other Scenarios: Bandit Learning with Graph Feedback & Online Graph Classification with Neural Bandit

**Bandits for Combo Recommendation**
- Multi-facet Contextual Bandits
Online recommendation scenario (in each round):

Presented with items → Choose item & Recommend → User feedback

Leverage the available arm group information can help improve recommendation quality.

1. Yunzhe Qi et. al., Neural bandit with arm group graph. KDD 2022.
The group (category) information for arms (items) is commonly accessible:

- **Media contents:**
  - Music, Movies (grouped by genres)

- **Text contents:**
  - Articles (grouped by literary styles)

- **E-commerce:**
  - Restaurants (grouped by cuisine types)

- Etc.

No existing MAB method trying to directly leverage the available arm group information.

1. Yunzhe Qi et al., Neural bandit with arm group graph. KDD 2022.
Formal Problem Definition

- **Arm Groups:**
  - Assume a fixed pool $\mathcal{C}$ of $|\mathcal{C}| = N_c$ arm groups.
  - Each arm group $c \in \mathcal{C}$ (e.g., movie genre) relates to an arm distribution $D_c$.

- **For each round** $t \in [T]$:
  - Receive a set of arms $X_t$, and the corresponding set of arm groups $\mathcal{C}_t \subseteq \mathcal{C}$.
    - $X_t = \{x_{c,t}^{(i)} \in \mathbb{R}^{d_x}, \text{ (e.g., )} \}_c \in \mathcal{C}_t, i \in [n_c, t]$
    - $x_{c,t}^{(i)} \sim D_c$
  - Reward $r_{c,t}^{(i)} = h(W^*, x_{c,t}^{(i)}) + \epsilon_{c,t}^{(i)}$.
    - Unknown affinity matrix for arm groups: $W^* \in \mathbb{R}^{N_c \times N_c}$
  - Learner chooses arm $x_t \in X_t$.

- **Objective: Minimizing Pseudo Regret**

$$R(T) = \sum_{t=1}^{T} \mathbb{E}[(r^*_t - r_t)]$$
$$= \sum_{t=1}^{T} |h(W^*, x^*_t) - h(W^*, x_t)|$$

Optimal arm  Chosen arm

---

1. Yunzhe Qi et. al., Neural bandit with arm group graph. KDD 2022.
Apply Arm Group Graph (AGG) to model arm group correlations:

In round $t \in [T]$:

- Undirected graph $G_t = (V, E, W_t)$
  - $V$: set of nodes
    - Each node is an arm group $c \in \mathcal{C}$, $N_c$ nodes in total
  - $E = \{e(c, c')\}_{c, c' \in \mathcal{C}}$: set of edges
  - $W_t$: Set of edge weights

- Arm group correlations are modeled by the edge weights from set $W_t$.

True reward: $r_{c,t}^{(i)} = h(G^*, x_{c,t}^{(i)}) + \epsilon_{c,t}^{(i)}$.

- Unknown true graph: $G^*$
- Unknown affinity matrix: $W^* \in \mathbb{R}^{N_c \times N_c}$
Proposed Framework: AGG-UCB

(1) Reward estimation

(2) Back-prop. & calculate UCB($x_{c,t}^{(i)}$)

(3) AGG Update (to $G_{t+1}$)

(4) Parameter update via GD

1. Yunzhe Qi et al., Neural bandit with arm group graph. KDD 2022.
AGG-UCB: Arm Group Graph Estimation

- Recall for Arm Groups:
  - Assume a fix pool $\mathcal{C}$ of $|\mathcal{C}| = N_c$ arm groups.
  - Each group $c \in \mathcal{C}$ has a context distribution $\mathcal{D}_c$.

- Definition: True edge weights
  - For $c, c' \in \mathcal{C}$, true edge weight in $\mathcal{G}^*$:
    - $w^*(c, c') = \exp\left(\frac{-\left\|\mathbb{E}_{x \sim \mathcal{D}_c} [\phi(x)] - \mathbb{E}_{x' \sim \mathcal{D}_{c'}} [\phi(x')]}{\sigma_s}\right)^2\right)$
    - $\phi(\cdot)$: kernel mapping function

- Arm Group Graph estimation:
  - Estimated edge weight in round $t \in [T]$:
    - $w_t(c, c') = \exp\left(\frac{-\left\|\Psi_t(\mathcal{D}_c) - \Psi_t(\mathcal{D}_{c'})\right\|^2}{\sigma_s}\right)$
    - Kernel Mean Embedding $[1]$: $\Psi_t(\mathcal{D}_c)$
  - $w_t(c, c') \in W_t$: weight for edge $e(c, c') \in E$ in graph $\mathcal{G}_t$

AGG-UCB: Arm Reward Estimation

- Reward estimation with estimated $G_t$.
- Point reward estimation for each arm $x_{c,t}^{(i)} \in X_t$:

\[
\hat{r}_{c,t}^{(i)} = f(G_t, x_{c,t}^{(i)}; \Theta_{t-1})
\]

Choose the rep. for group $c$

Fully-Connected Network

Estimated Reward

Fully-Connected layer for group $c_1$

Fully-Connected layer for group $c_2$

\vdots

Fully-Connected layer for group $c_{N_c}$

Aggregation on $G_t$ with GNN for $k$-hops

\[H_{gnn}\]

\[H_0\]

Concatenation

1. Yunzhe Qi et. al., Neural bandit with arm group graph. KDD 2022.
AGG-UCB: Arm Selection & Training

- **Exploration** with Upper Confidence Bound (UCB):
  - The UCB(·) satisfies:
    \[
    \mathbb{P} \left( \left| f(g_t, x^{(i)}_{c,t}, \Theta_{t-1}) - h(g^*, x^{(i)}_{c,t}) \right| > \text{UCB} \left( x^{(i)}_{c,t} \right) \right) \leq \delta
    \]
    Reward Est.  Exp. Reward

- **Arm Selection Strategy**:
  - Select arm \( x_t = \arg\max_{x^{(i)}_{c,t} \in \mathcal{X}_t} \left( \hat{r}^{(i)}_{c,t} + \gamma \cdot \text{UCB} \left( x^{(i)}_{c,t} \right) \right) \)
  - Receive the corresponding true reward \( r_t \)

1. Yunzhe Qi et al., Neural bandit with arm group graph. KDD 2022.
Theoretical and Empirical Results

- **Theoretical**: Given sufficiently large network width $m$, with the probability at least $1 - \delta$:

$$R(T) \leq 2 \cdot (2B_4 \sqrt{T} + 2 - B_4) + 2\sqrt{2dT \log(1 + T/\lambda)} + 2T$$

$$\cdot (\sqrt{\lambda S} + \sqrt{1 - 2\log(\delta/2) + (\tilde{d} \log(1 + T/\lambda)))}$$

Achieves the regret bound of $O(\sqrt{T \log^2(T) \cdot \log(N_c)})$

- **Empirical**: Leveraging **arm group information with AGG-UCB** can improve good performances.

![Figure 1: Cumulative regrets on recommendation data sets](image1)

![Figure 2: Cumulative regrets on Classification data sets](image2)

1. Yunzhe Qi et. al., Neural bandit with arm group graph. KDD 2022.
Graph Bandit Learning: Other Scenarios

1. **Bandit Learning with Graph Feedback** [1]:
   - Arms are nodes on a graph $G = (V, E)$. In each round $t \in [T]$, the learner chooses one node $I_t \in V$.
   - Learner observes **reward for chosen arm** $I_t$, and **neighbor rewards** (e.g., out-neighbors in a directed graph).
   - **Objective**: minimizing the cumulative pseudo regret over $T$ rounds.

2. **Optimal Graph Search with Bandit** [2]:
   - In each round $t \in [T]$, the learner aims to choose **one graph** $G_t \in \mathcal{G}$, from a **fixed** graph domain $\mathcal{G}$.
     Reward generated by $r_t = h(G_t) + \epsilon_t$.
   - **Objective**: minimizing the cumulative pseudo regret over $T$ rounds.
   - **Application example**: material designing, drug search.

---

2. Kassraie et al., Graph Neural Network Bandits. NeurIPS 2022.
Roadmap

**Introduction**
- Background & Motivations
- Challenges

**Online Clustering of Bandits**
- Clustering of Linear Bandits
- Clustering of Neural Bandits

**Graph Bandit Learning with Collaboration**
- User side: Graph Neural Bandits
- Arm side: Neural Bandit with Arm Group Graph
- Other Scenarios: Bandit Learning with Graph Feedback & Online Graph Classification with Neural Bandit

**Bandits for Combo Recommendation**
- Multi-facet Contextual Bandits
Motivated Case: Promotion Campaign

E-commerce

Which to pick?

Snack 1
.
.
.
Snack n_1

Beverage 1
.
.
.
Beverage n_2

Toiletry 1
.
.
.
Toiletry n_3

Snacks

Beverages

Toiletries

Sub-reward can be unknown

Overall Feedback

Customer

Application: Multi-facet Recommendation with Neural Bandits

Learner

Select

Arm 1

. .

Arm n_1

Select

Arm 1

. .

Arm n_2

Select

Arm 1

. .

Arm n_3

Bandit 1

Bandit 2

Bandit 3

Sub-reward can be unknown

Arm* (Bandit 1)

Arm* (Bandit 2)

Arm* (Bandit 3)

User

Final Reward
(Overall Feedback)

Sub-reward Functions (unknown):

\[ r_t^1 = h_1(x_t^1) \] (Linear or Non-linear)
\[ r_t^2 = h_2(x_t^2) \]
\[ \vdots \]
\[ r_t^K = h_K(x_t^K) \]

Assumption 1: \( h_k(0) = 0, \forall k \)

Final Reward Function (unknown):

\[ R_t = H(r_t^1, r_t^2, ..., r_t^K) + \epsilon_t \]

Expectation: \( H(X_t) = E[R_t|X_t] = H(r_t^1, r_t^2, ..., r_t^K) \)

Assumption 2: \( H \) is \( \bar{C} \)-Lipschitz continuous.

Evaluation Measure: Regret

\[ \text{Reg} = E \left[ \sum_t (R_t^* - R_t) \right] \]
\[ = \sum_t [H(X_t^*) - H(X_t)] \]

Goal: Minimize the regret of \( T \) rounds.
MuFasa: Exploitation (Neural Network Model)

As all bandits serve the same user

\[ x_t^1 \rightarrow \text{Fully-Connected} \rightarrow \text{Shared Layers} \rightarrow F(X_t; \theta) \]

Estimated Final Reward (to learn \( H(X_t) \))

Input: K arms

\[ \vdots \]

\[ x_t^K \rightarrow \text{Fully-Connected} \]

MuFasa: Exploration (Upper Confidence Bound)

- **UCB:** \( \mathbb{P} \left( |\mathcal{F}(X_t; \theta_t) - \mathcal{H}(X_t)| > \text{UCB}(X_t) \right) \leq \delta, \)

- **K selected arms are determined by:**

  \[ X_t = \arg \max_{X_t' \in S_t} (\mathcal{F}(X'; \theta_t) + \text{UCB}(X'_t)). \]

  Where

  \[ S_t = \{(x_1^1, \ldots, x_k^k, \ldots, x^K_t) \mid x_k^k \in X_t^k, k \in [K]\}, \]

  (all possible combinations of K arms)

MuFasa: Novel Upper Confidence Bound (UCB)

- With the assembled neural framework (MuFasa):

- With probability at least $1 - \delta$, the UCB holds

$$|\mathcal{F}(X_t; \theta_t) - \mathcal{H}(X_t)| \leq \tilde{C} \sum_{k=1}^{K} \mathcal{B}^k + \mathcal{B}^F = UCB(X_t), \text{ where}$$

$$\mathcal{B}^k = y_1 \|g_k(x^k_t; \theta^k_t)/\sqrt{m_1}\|_{A_{t}^{k-1}}^{k-1} + y_2(\frac{\delta}{k+1})\|g_k(x^k_0; \theta^k_0)/\sqrt{m_1}\|_{A_{t}^{k-1}}^{k-1} + y_1 y_3 + y_4$$

$$\mathcal{B}^F = y_1 \|G(f_t; \theta^\Sigma_t)/\sqrt{m_2}\|_{A_{t}^{F-1}}^{F-1} + y_2(\frac{\delta}{k+1})\|G(f_t; \theta^\Sigma_t)/\sqrt{m_2}\|_{A_{t}^{F-1}}^{F-1} + y_1 y_3 + y_4$$

Regret Analysis

\[
\text{Reg} = E \left[ \sum_t (R^*_t - R_t) \right] = \sum_t [H(X^*_t) - H(X_t)]
\]

- After T rounds, with probability at least \(1 - \delta\),

\[
\text{Reg} \leq (\tilde{C}K + 1)\sqrt{T}2\sqrt{\tilde{P}} \log(1 + T/\lambda) + 1/\lambda + 1
\]

\[
\cdot \left( \sqrt{\tilde{P} - 2} \log \left( \frac{(\lambda + T)(1 + K)}{\lambda \delta} \right) + 1/\lambda + \lambda^{1/2}S + 2 \right) + 2(\tilde{C}K + 1),
\]

- Achieve near-optimal regret bound \(\tilde{O}\left((K + 1)\sqrt{T}\right)\), same as a single linear bandit \(\tilde{O}(\sqrt{T})\)

All Sub-rewards Available (Different Final Reward Function)

Observation:
- Superiority of MuFasa is slightly higher on $H_2$, compared to $H_1$.

Insights:
- MuFasa can select arms according to different weights of bandits (Bandit 1 has higher weight in $H_2$).

Figure: Regret comparison on Mnist+NotMnist with $H_1$.

$$H_1(\text{vec}(r_t)) = r^1_t + r^2_t$$

Figure: Regret comparison on Mnist+NotMnist with $H_2$.

$$H_2(\text{vec}(r_t)) = 2r^1_t + r^2_t$$

Partial Sub-rewards Available

Observation:

- With one sub-reward, MuFasa still outperforms all baselines.
- Without any sub-reward, MuFasa’s performance is close to the best baseline.

Figure: Regret comparison on Yelp with different reward availability.

Figure: Regret comparison on Mnist+NotMnist with different reward availability.

Tutorial Roadmap

Linear/Neural Contextual Bandits → Collaborative Bandits → Future Trends
Q: Can we have a **transparent** exploration with clear rationales and explanations?

- **Challenges:**
  - More exploration models based on neural networks (**Black Box**).

- **Future Directions:**
  - Bayesian Bandits/RL.
  - Causal Bandits/RL.

**Black Box !**
E.g. [Ban et al. ICLR 2022]
Q: How to ensure **fairness** in the context of exploration?

- **Challenges:**
  - Non-IID data.
  - Balance required between **exploration power** and **fairness**.

- **Future Directions:**
  - Derive fairness confidence interval for exploration.
  - Fairness Regularization.

---

**Trustworthy Exploration: Privacy**

**Q:** Can we have an exploration strategy preserving privacy?

- **Challenges:**
  - Privacy-preserving exploration methods.

- **Future Directions:**
  - Federated Bandits/RL.

---

Customized Exploration: Large Language Model

(1) Prompting with Exploration

Prompt 1
Prompt 2
Prompt 3

Exploit
Explore

ChatGPT

(2) Answering with User-specific Exploration

Response 1
Response 2
Response 3

Exploit
Explore

ChatGPT

(3) Fine-tuning with Exploration

1 2 3 ... 4 5 6

Large Language Model

Linear/Neural Contextual Bandits

- NeuralUCB
- NeuralTS
- EE-Net
- Neural-LinUCB
- Neural-Bandit-Perturbed Reward
- Neural-SquareCB

Collaborative Bandits

- Clustering of Linear Bandits
- Clustering of Neural Bandits
- Graph Bandit Learning
  - Graph Neural Bandits
  - Arm Group Graph
- Multi-facet Neural Bandits

Future Trends

- Trustworthy Neural Bandits with Recommendation
- LLM with Bandits for Exploration
Neural Contextual Bandits for Personalized Recommendation

University of Illinois Urbana-Champaign
{Yikunb2, Yunzheq2, Jingrui}@illinois.edu

Time: 9:00 AM – 12:30 PM, 13 May 2024
Location: Virgo 1, Resorts World Sentosa Convention Centre, Singapore
Website: www.banyikun.com/wwwtutorial/