Laplacian-regularized Graph Bandits

Laura Toni
University College London

25 September 2019
Laplacian-regularized Graph Bandits

Laura Toni

Kaige Yang
UCL

Xiaowen Dong
University of Oxford
Outline

• Graphs and Bandit
• Importance of Graphs in Decision-Making
• A Laplacian Perspective
• Output and Intuitions
• Conclusions
Main Challenges in DMS

Training data

observation

updated knowledge

action

Theoretically addressed by

- Multi-arm bandit problem
- Reinforcement Learning
Main Challenges in DMS

- Training data
  
  - observation
  - updated knowledge
  
  - action

Theoretically addressed by:
- Multi-arm bandit problem
- Reinforcement Learning

- Find the optimal trade-off between exploration and exploitation: bandit and RL problems

- Sampling-efficiency: the learning performance does not scale with the ambient dimension (number of arms, states, etc): structured learning
Structured DMS - Main Challenges

- In DMSs, context or action payoffs (data) have semantically reach information

Structured problems obviate the curse of dimensionality by exploiting the data structure
Structured DMS - Main Challenges

- In DMSs, context or action payoffs (data) have semantically reach information

Structured problems obviate the curse of dimensionality by exploiting the data structure
Structured DMS - Main Challenges

- In DMSs, context or action payoffs (data) have semantically reach information

Structured problems obviate the curse of dimensionality by exploiting the data structure

Graph Clustering
- reducing the curse of dimensionality
- degradation in real-world data

C. Gentile et al., “Online Clustering of Bandits”, ICML 2014
Structured DMS - Main Challenges

- In DMSs, context or action payoffs (data) have semantically reach information

Structured problems obviate the curse of dimensionality by exploiting the data structure

Graph Clustering
- reducing the curse of dimensionality
- degradation in real-world data

Need for more sophisticated frameworks (than clustering) to handle high-dimensional and structured data

C. Gentile et al., “Online Clustering of Bandits”, ICML 2014
Structured DMS - Main Challenges

- In DMSs, context or action payoffs (data) have **semantically reach information**

- It is important to identify and **leverage the structure** underneath these data

Many works on Bandit are graph based, see overview [1]

- **data-structure in bandits:**
  - Gentile, C., Li, S., and Zappella, G. “Online clustering of bandits”, ICML 2014
  - Yang, K. and Toni, L., “Graph-based recommendation system”, IEEE GlobalSIP, 2018

In DMSs, context or action payoffs (data) have semantically reach information.

It is important to identify and leverage the structure underneaths these data.

Many works on Bandit are graph based, see overview [1]

- **data-structure in bandits:**
  - Gentile, C., Li, S., and Zappella, G. “Online clustering of bandits”, ICML 2014
  - Yang, K. and Toni, L., “Graph-based recommendation system”, IEEE GlobalSIP, 2018

Can we capture the external information beyond data-structure?

---

In DMSs, context or action payoffs (data) have semantically reach information.

It is important to identify and leverage the structure underneath these data.

Many works on Bandit are graph based, see overview [1]

- spectral bandits:
  - other recent works on asynchronous and decentralized network bandits

Structured DMS - Main Challenges

- In DMSs, context or action payoffs (data) have **semantically reach information**

- It is important to identify and **leverage the structure** underneath these data

Many works on Bandit are graph based, see overview [1]

- **spectral bandits:**
  - other recent works on asynchronous and decentralized network bandits
    - single user bandit
    - no per-user error bound → coarse regret upper bounds scaling linearly with the number of users
    - high computational complexity

Structured DMS - Main Challenges

- In DMSs, context or action payoffs (data) have **semantically reach information**
- It is important to identify and **leverage the structure** underneath these data
- Highly interesting studies on graph-bandit already published, but most of them work in the **graph spatial (vertex) domain**
In DMSs, context or action payoffs (data) have semantically reach information.

It is important to identify and leverage the structure underneath these data.

Highly interesting studies on graph-bandit already published, but most of them work in the graph spatial (vertex) domain.

Data can be high-dimensional, time-varying, and composition of superimposed phenomena.

Need proper framework to capture both data-structure and external-geometry information (graphs).
Structured DMS - Main Challenges

- In DMSs, context or action payoffs (data) have semantically reach information.
- It is important to identify and leverage the structure underneath these data.
- Highly interesting studies on graph-bandit already published, but most of them work in the graph spatial (vertex) domain.
- Data can be high-dimensional, time-varying, and composition of superimposed phenomena.
- Need proper framework to capture both data-structure and external-geometry information (graphs).

Graph signal processing (GSP) can be applied to DMSs to address the above challenges and needs.
Structured but irregular data can be represented by graph signals

Goal: to capture both structure (edges) and data (values at vertices)
For example, [31]–[34] examine the analogous operators in the continuous setting. In some probability domains: the vertex domain and the graph spectral domain. While we often start with the classical analog case, the graph signal to a vertex with a negative signal: the blue (positive) and black (negative) bars coming out of the vertices. (b) is then determined by taking an inverse graph Fourier transform (4) of the signal plotted in (a) is then determined by taking an inverse graph Fourier transform (4) of the signal plotted in (a).

\[ \hat{f}(l) = \langle f, \chi_l \rangle = \sum_{n=1}^{N} f(n) \chi^*_l(n) \]

\[ f(n) = \sum_{l=0}^{N-1} \hat{f}(l) \chi_l(n), \ \forall n \in \mathbb{R} \]

The graph Fourier transform (3) and its inverse (4) can be used to decompose a signal into its frequency components on a graph. The forward transform is given by:

\[ \hat{f}(l) = \langle f, \chi_l \rangle = \sum_{n=1}^{N} f(n) \chi^*_l(n) \]

The inverse transform is given by:

\[ f(n) = \sum_{l=0}^{N-1} \hat{f}(l) \chi_l(n), \quad \forall n \in \mathbb{N} \]

Where \( \chi_l \) are the eigenfunctions of the graph Laplacian, \( \hat{f}(l) \) are the Fourier coefficients of a smooth signal, and \( f(n) \) is the signal in the graph spectral domain. In this case, the signal is a compressible vector, represented by the blue (positive) and black (negative) bars coming out of the vertices. The physical interpretation of Laplacian eigenfunctions is that they indicate local variation of the signal. The Laplacian eigendecomposition can be carried out via an orthonormal basis on the vertices of the graph, introducing a change of variables to incorporate the geometric structure of the underlying manifold.

For example, in a network analysis, the low-frequency eigenfunctions represent smooth signals, while the high-frequency eigenfunctions represent rapid changes or jumps. This is analogous to the classical analog case, where the signal is a real-valued function on a manifold.

**References**

Filtering and Smoothness

Filtering of graph signals

\[ \hat{f}(l) = \hat{g}(\lambda_l) \hat{f}(l) \]

IGFT

\[ y = \sum_{l=0}^{N-1} \hat{g}(\lambda_l) \hat{f}(l) x_l(n) \]

Example

\[ y^T L y = 10.75 \]

\[ f^T L f = 61.93 \]

M. Defferrard, Deep Learning on Graphs: a journey from continuous manifolds to discrete networks (KCL/UCL Junior Geometry Seminar)
Filtering and Smoothness

**Filtering**

- Input signal $f$
- Graph Fourier Transform (GFT) $\hat{f}(l)$
- Filtered signal $\hat{g}(\lambda)\hat{f}(l)$
- Inverse Graph Fourier Transform (IGFT) $y = \sum_{l=0}^{N-1} \hat{g}(\lambda_l)\hat{f}(l)\chi_l(n)$

**Denoising problem**

$$y^* = \arg\min_y \left\{ ||y - f||^2 + \gamma y^T Ly \right\}$$

$$y^* = (I + \gamma L)^{-1} f = \chi(I + \gamma \Lambda)^{-1} \chi^T f$$

remove noise by low-pass filtering in the graph spectral domain

M. Defferrard, Deep Learning on Graphs: a journey from continuous manifolds to discrete networks (KCL/UCL Junior Geometry Seminar)
Filtering and Smoothness

Filtering of graph signals

\[ \hat{f}(l) = \sum_{\lambda} \hat{g}(\lambda) \tilde{f}(l) \chi_l(n) \]

\[ y = y^T Ly = 10.75 \]

\[ f^T L f = 61.93 \]

Denoising problem

\[ y^* = \arg \min_y \left\{ \| y - f \|_2^2 + \gamma y^T L y \right\} \]

\[ y^* = (I + \gamma L)^{-1} f = \chi (I + \gamma \Lambda)^{-1} \chi^T f \]

\[ g(L) \]

Observation: the low-pass filtered signal is much smoother than the input signal!

M. Defferrard, Deep Learning on Graphs: a journey from continuous manifolds to discrete networks (KCL/UCL Junior Geometry Seminar)
Filtering and Smoothness

---

**Filtering**

- Low-pass
- High-pass
- Band-pass

---

**Example**

- **Input signal** $x$
- **Filtering problem**
  - $y^T Ly = 10.75$
  - $f^T L f = 61.93$

---

**Denoising problem**

$$y^* = \arg \min_y \left\{ \| y - f \|^2_2 + \gamma y^T Ly \right\}$$

$$y^* = \left( I + \gamma L \right)^{-1} g(L) \\ \hat{f}(l)$$

---

M. Defferrard, Deep Learning on Graphs: a journey from continuous manifolds to discrete networks (KCL/UCL Junior Geometry Seminar)
GSP for Online DMS

GSP to exploit spectral properties

Training data

Exploration exploitation trade-off

GSP-Based MAB

- **Data-efficiency**: learn in a sparse domain
- **Accuracy**: learning representation that preserves the geometry of the problem
- **Mathematical framework** is missing
- Not many works beyond smoothness
Outline

- Graphs and Bandit
- Importance of Graphs in Decision-Making
- A Laplacian Perspective
- Output and Intuitions
- Conclusions
**Aim**: Infer the best item by running a sequence of trials

\[ y = x^T \theta + \eta \]

\( x \in \mathbb{R}^d \): item feature vector

\( \theta \in \mathbb{R}^d \): user parameter vector

\( y \): linear payoff

\( \eta \): \( \sigma \)-sub-Gaussian noise
Aim: Infer the best item by running a sequence of trials

Well known bandit problem with assumptions:
(i) stochasticity, (ii) i.i.d., (iii) stationarity

\[ y = x^T \theta + \eta \]
**Recommendation Model**

**Aim:** Infer the best item by running a sequence of trials

\[ y = x^T \theta + \eta \]

Well known **bandit problem** with assumptions:
(i) stochasticity, (ii) i.i.d., (iii) stationarity

**Our interest:** multi-user (high-dimensional) case

Today’s talk
Settings

- centralized agent
- $m$ arms and $n$ users
- users appearing uniformly at random
- At round $t$, user $i_t$ appears, and an agent
  - chooses an arm $a_t$
  - receives a reward $y_t = x^{T}_{a_t} \theta_{i_t} + \eta_t$
Settings

- centralized agent
- $m$ arms and $n$ users
- users appearing uniformly at random
- At round $t$, user $i_t$ appears, and an agent
  - chooses an arm $a_t$
  - receives a reward $y_t = x^T_{a_t} \theta_{i_t} + \eta_t$
- Sequential sampling strategy (bandit algorithm)

$$a_{t+1} = F_t(i_1, a_1, y_1, \ldots, i_t, a_t, y_t \mid i_{t+1})$$

- Goal: Maximize sum of rewards $\mathbb{E} \left[ \sum_{t=1}^{T} y_t \right]$
• \( G = (V, E, W) \): undirected-weighted graph
• \( W_{i,j} = W_{j,i} \): captures similarity between users \( i \) and \( j \) (i.e., \( \theta_{i,j} = \theta_{j,i} \))
• \( L = D - W \): combinatorial Laplacian of \( G \)
• $\mathcal{G} = (V, E, W)$: undirected-weighted graph
• $W_{i,j} = W_{j,i}$: captures similarity between users $i$ and $j$ (i.e., $\theta_{i,j} = \theta_{j,i}$)
• $L = D - W$: combinatorial Laplacian of $\mathcal{G}$

Similarity captured in the latent space
Assumptions

- User preferences mapped into a graph of similarities
  \[ \Theta = [\theta_1, \theta_2, \ldots, \theta_n]^T \in \mathbb{R}^{n \times d} : \text{signal on graph} \]

- Exploitation of smoothness prior

\[
tr(\Theta^T \mathcal{L} \Theta) = \frac{1}{4} \sum_{k=1}^{d} \sum_{i \sim j} \left( \frac{W_{ij}}{D_{ii}} + \frac{W_{ji}}{D_{jj}} \right) (\Theta_{ik} - \Theta_{jk})^2
\]
Assumptions

• User preferences mapped into a graph of similarities
  \( \Theta = [\theta_1, \theta_2, \ldots, \theta_n]^T \in \mathbb{R}^{n \times d} \): signal on graph

• Exploitation of smoothness prior

\[
\text{smoothness measure } \quad tr(\Theta^T \mathcal{L} \Theta) = \frac{1}{4} \sum_{k=1}^{d} \sum_{i \sim j} \left( \frac{W_{ij}}{D_{ii}} + \frac{W_{ji}}{D_{jj}} \right) (\Theta_{ik} - \Theta_{jk})^2
\]
Assumptions

- User preferences mapped into a graph of similarities
  \[ \Theta = [\theta_1, \theta_2, \ldots, \theta_n]^T \in \mathbb{R}^{n \times d} : \text{signal on graph} \]

- Exploitation of smoothness prior

  \[
  \text{smoothness measure} \quad tr(\Theta^T L \Theta) = \frac{1}{4} \sum_{k=1}^{d} \sum_{i \sim j} \left( \frac{W_{ij}}{D_{ii}} + \frac{W_{ji}}{D_{jj}} \right) (\Theta_{ik} - \Theta_{jk})^2
  \]

- Smoothness of \( \Theta \) over graph \( \mathcal{G} \) can be quantified using the Laplacian quadratic form

- We express smoothness as a function of the random-walk Laplacian

  \[ L = D^{-1}L \quad \text{with} \quad L_{ii} = 1 \quad \text{and} \quad \sum_{j \neq i} L_{ji} = -1 \]
Assumptions

- User preferences mapped into a graph of similarities
  \[ \Theta = [\theta_1, \theta_2, \ldots, \theta_n]^T \in \mathbb{R}^{n \times d} : \text{signal on graph} \]

- Exploitation of smoothness prior

\[
tr(\Theta^T \mathcal{L} \Theta) = \frac{1}{4} \sum_{k=1}^{d} \sum_{i \sim j} \left( \frac{W_{ij}}{D_{ii}} + \frac{W_{ji}}{D_{jj}} \right) (\Theta_{ik} - \Theta_{jk})^2
\]

- Smoothness of \( \Theta \) over graph \( G \) can be quantified using the Laplacian quadratic form

- We express smoothness as a function of the random-walk Laplacian

\[
\mathcal{L} = D^{-1}L \quad \text{with} \quad \mathcal{L}_{ii} = 1 \text{ and } \sum_{j \neq i} \mathcal{L}_{ji} = -1
\]

- avoiding a regret scaling with \( D_{ii} \)
- achieving convexity property needed to bound the estimation error
Problem Formulation

Given
- the users graph $\mathcal{G}$
- arm feature vector $x_a, a \in \{1,2,\ldots,m\}$
- no information about the user $\theta_i, i \in \{1,2,\ldots,n\}$?

The agent seeks the optimal selection strategy that minimizes the cumulative (pseudo) regret

$$R_T = \sum_{t=1}^{T} \left( (x_i^*)^T \theta_i - x_i^T \theta_i \right)$$
Given

- the users graph $\mathcal{G}$
- arm feature vector $x_a, a \in \{1, 2, \ldots, m\}$
- no information about the user $\theta_i, i \in \{1, 2, \ldots, n\}$?

The agent seeks the optimal selection strategy that minimizes the cumulative (pseudo) regret

$$R_T = \sum_{t=1}^{T} \left( (x_i^*)^T \theta_i - x_i^T \theta_i \right)$$

Under smoothness prior, the users parameter vector is estimated as

$$\hat{\Theta}_t = \arg\min_{\Theta \in \mathbb{R}^{n \times d}} \sum_{i=1}^{n} \sum_{\tau \in t_i} (x_{i\tau}^T \theta_i - y_{i\tau})^2 + \alpha \text{tr}(\Theta^T \mathcal{L} \Theta)$$
Problem Formulation

Given
- the users graph $\mathcal{G}$
- arm feature vector $x_a, a \in \{1,2,\ldots,m\}$
- no information about the user $\theta_i, i \in \{1,2,\ldots,n\}$

The agent seeks the optimal selection strategy that minimizes the cumulative (pseudo) regret

$$R_T = \sum_{t=1}^{T} \left( (x_i^*)^T \theta_i - x_i^T \theta_i \right)$$

Under smoothness prior, the users parameter vector is estimated as

$$\hat{\Theta}_t = \arg\min_{\Theta \in \mathbb{R}^{n \times d}} \sum_{i=1}^{n} \sum_{\tau \in t_i} (x_i^T \theta_i - y_{i,\tau})^2 + \alpha \text{tr}(\Theta^T \mathcal{L} \Theta)$$

fidelity term
Given
- the users graph $G$
- arm feature vector $x_a, a \in \{1,2,\ldots,m\}$
- no information about the user $\theta_i, i \in \{1,2,\ldots,n\}$

The agent seeks the optimal selection strategy that minimizes the cumulative (pseudo) regret

$$R_T = \sum_{t=1}^{T} \left( (x_t^*)^T \theta_t - x_t^T \theta_t \right)$$

Under smoothness prior, the users parameter vector is estimated as

$$\hat{\Theta}_t = \arg \min_{\Theta \in \mathbb{R}^{n \times d}} \sum_{i=1}^{n} \sum_{\tau \in t_i} (x_T^T \theta_i - y_{i,\tau})^2 + \alpha \text{tr}(\Theta^T L \Theta)$$

fidelity term  smoothness regularizer
Problem Formulation

Given

- the users graph $\mathcal{G}$
- arm feature vector $x_a, a \in \{1,2,\ldots,m\}$
- no information about the user $\theta_i, i \in \{1,2,\ldots,n\}$ ?

The agent seeks the optimal selection strategy that minimizes the cumulative (pseudo) regret

$$R_T = \sum_{t=1}^{T} \left( (x^*_t)^T \theta_t - x^T_t \theta_t \right)$$

Under smoothness prior, the users parameter vector is estimated as

$$\hat{\Theta}_t = \arg\min_{\Theta \in \mathbb{R}^{n \times d}} \sum_{i=1}^{n} \sum_{\tau \in t_i} (x^T_\tau \theta_i - y_{i,\tau})^2 + \alpha \text{tr}(\Theta^T \mathcal{L} \Theta)$$

The agent selects sequential actions as follows

$$x_{i,t} = \arg\max_{(x,\theta) \in \mathcal{D}, \mathcal{C}_{i,t}} x^T \theta$$
Problem Formulation

Given

- the users graph $\mathcal{G}$
- arm feature vector $x_a, a \in \{1,2,\ldots,m\}$
- no information about the user $\theta_i, i \in \{1,2,\ldots,n\}$

The agent seeks the optimal selection strategy that minimizes the cumulative (pseudo) regret

$$R_T = \sum_{t=1}^{T} \left( (x_i^*)^T \theta_i - x_t^T \theta_i \right)$$

Under smoothness prior, the users parameter vector is estimated as

$$\hat{\Theta}_t = \arg\min_{\Theta \in \mathbb{R}^{n \times d}} \sum_{i=1}^{n} \sum_{\tau \in t_i} (x_{i,\tau}^T \theta_i - y_{i,\tau})^2 + \alpha \text{tr}(\Theta^T \mathcal{L} \Theta)$$

The agent selects sequential actions as follows

$$x_{i,t} = \arg\max_{(x,\theta) \in \mathcal{D} \times \mathcal{Q}_i} x^T \theta$$
Problem Formulation

Main Challenges
• smoothness not imposed in the observation domain but in the representation one
• no theoretical error bound for Laplacian regularized estimate
• computational complexity

Main Novelties
• derivation single-user estimation error bound
• proposed single-user UCB in bandit problem
• low-complexity (local) algorithm
• cumulative regret bound as a function of graph properties
Laplacian-regularized Estimator

Closed form solution

$$\hat{\Theta}_t = \arg \min_{\Theta \in \mathbb{R}^{n \times d}} \sum_{i=1}^{n} \sum_{t \in t_i} (x_{i,t}^T \theta_i - y_{i,t})^2 + \alpha \, tr(\Theta^T L \Theta)$$

$$vec(\hat{\Theta}_t) = (\Phi_t \Phi_t^T + \alpha \mathcal{L} \otimes I)^{-1} \Phi_t Y_t \quad \Phi_t = [\phi_1, \phi_2, ..., \phi_t] \in \mathbb{R}^{nd \times t}$$

where $\otimes$ is the Kronecker product, and $vec(\hat{\Theta}_t)$ is a concatenation of column of $\hat{\Theta}_t$
Closed form solution

\[
\hat{\Theta}_t = \arg \min_{\Theta \in \mathbb{R}^{nxd}} \sum_{i=1}^{n} \sum_{\tau \in t_i} (x^T_\tau \theta_i - y_{i,\tau})^2 + \alpha \text{ tr}(\Theta^T \mathcal{L} \Theta)
\]

\[\text{vec}(\hat{\Theta}_t) = (\Phi_t \Phi_t^T + \alpha \mathcal{L} \otimes I)^{-1} \Phi_t Y_t, \quad \Phi_t = [\phi_1, \phi_2, ..., \phi_t] \in \mathbb{R}^{nd \times t}\]

where \(\otimes\) is the Kronecker product, and \(\text{vec}(\hat{\Theta}_t)\) is a concatenation of column of \(\hat{\Theta}_t\)

**Lemma 1.** \(\hat{\Theta}_t\) is obtained from Eq. 5, let \(\hat{\theta}_{i,t}\) be the \(i\)-th row of \(\hat{\Theta}_t\) which is the estimate of \(\theta_i\). \(\hat{\theta}_{i,t}\) can be approximated by:

\[\hat{\theta}_{i,t} \approx A_{i,t}^{-1} X_{i,t} Y_{i,t} - \alpha A_{i,t}^{-1} \sum_{j=1}^{n} L_{ij} A_{j,t}^{-1} X_{j,t} Y_{j,t} \quad (7)\]

where \(A_{i,t} = \sum_{\tau \in t_i} x_\tau x_\tau^T \in \mathbb{R}^{d \times d}\) is the Gram matrix of user \(i\), \(L_{ij}\) is the \((i,j)\)-th element in \(\mathcal{L}\), \(Y_{i,t} = [y_{i,1}, ..., y_{i,t_i}]\) are the collection of payoffs associated with user \(i\) up to time \(t\).
\( C_{i,t} = \{ \theta_{i,t} : ||\hat{\theta}_{i,t} - \theta_{i,t}||_{\Lambda_{i,t}} \leq \beta_{i,t} \} \) \hfill (8)

\[ \Lambda_{i,t} = \Lambda_{i,t} + 2\alpha L_{ii}I + \alpha^2 \sum_{j=1}^{n} L_{ij}^2 A_{j,t}^{-1} \] \hfill (10)

precision matrix of \( vec(\hat{\Theta}_t) \)
\[ C_{i,t} = \{ \theta_{i,t} : \| \hat{\theta}_{i,t} - \theta_{i,t} \|_{\Lambda_{i,t}} \leq \beta_{i,t} \} \] (8)

\[ \Lambda_{i,t} = A_{i,t} + 2\alpha L_{i,i}I + \alpha^2 \sum_{j=1}^{n} L_{i,j}^2 A_{j,t}^{-1} \] (10)
Laplacian-regularized Estimator

\[ C_{i,t} = \{ \theta_{i,t} : \| \hat{\theta}_{i,t} - \theta_{i,t} \| \Lambda_{i,t} \leq \beta_{i,t} \} \]  \quad (8)

\[ \Lambda_{i,t} = A_{i,t} + 2\alpha L_{i,t}I + \alpha^2 \sum_{j=1}^{n} L_{ij}^2 A_{j,t}^{-1} \]  \quad (10)

**Lemma 2.** \( t_i \) is the set of time at which user \( i \) is served up to time \( t \). \( A_{i,t} = \sum_{\tau \in t_i} x_{\tau}x_{\tau}^T \), \( V_{i,t} = A_{i,t} + \alpha L_{ii}I \), \( \xi_{i,t} = \sum_{\tau \in t_i} x_{\tau}x_{\tau}^T \), \( \eta_{i,\tau} = \sum_{\tau \in t_i} x_{\tau} x_{\tau}^T \), \( I \in \mathbb{R}^{d \times d} \) is the identity matrix. \( \Lambda_{i,t} \) is defined in Eq. 10. Denote \( \Delta_i = \sum_{j=1}^{n} L_{ij} \theta_j \), the size of the confidence set defined in Eq. 8 satisfies the following upper bound with probability 1 - \( \delta \) with \( \delta \in [0, 1] \).

\[ \beta_{i,t} = \sigma \sqrt{2 \log \frac{|V_{i,t}|^{1/2}}{\delta |A_{i,t}|^{1/2}}} + \sqrt{\alpha} \| \Delta_i \|_2 \]
\[
C_{i,t} = \{\theta_{i,t} : \|\hat{\theta}_{i,t} - \theta_{i,t}\|_{\Lambda_{i,t}} \leq \beta_{i,t}\} \quad (8)
\]

\[
\Lambda_{i,t} = A_{i,t} + 2\alpha \mathcal{L}_{ii} I + \alpha^2 \sum_{j=1}^{n} \mathcal{L}_{ij}^2 A_{j,t}^{-1} \quad (10)
\]

**Lemma 2.** \(t_i\) is the set of time at which user \(i\) is served up to time \(t\). \(A_{i,t} = \sum_{\tau \in t_i} x_\tau x_\tau^T\), \(V_{i,t} = A_{i,t} + \alpha \mathcal{L}_{ii} I\), \(\xi_{i,t} = \sum_{\tau \in t_i} x_\tau \eta_{i,\tau}\), \(I \in \mathbb{R}^{d \times d}\) is the identity matrix. \(\Lambda_{i,t}\) is defined in Eq. 10. Denote \(\Delta_i = \sum_{j=1}^{n} \mathcal{L}_{ij} \theta_j\), the size of the confidence set defined in Eq. 8 satisfies the following upper bound with probability \(1 - \delta\) with \(\delta \in [0,1]\).

\[
\beta_{i,t} = \sigma \sqrt{2 \log \frac{|V_{i,t}|^{1/2}}{\delta |\alpha I|^{1/2}}} + \sqrt{\alpha \|\Delta_i\|_2}
\]

**Graph Information**

\[
\Delta_i = \sum_{i=1}^{n} \mathcal{L}_{ij} \theta_j = \theta_i - \sum_{j \neq i} (-\mathcal{L}_{ij} \theta_j)
\]
Laplacian-regularized Estimator

\[ C_{i,t} = \{ \theta_{i,t} : ||\hat{\theta}_{i,t} - \theta_{i,t}||_{\Lambda_{i,t}} \leq \beta_{i,t} \} \quad (8) \]

\[ \Lambda_{i,t} = A_{i,t} + 2\alpha \mathcal{L}_{ii} I + \alpha^2 \sum_{j=1}^{n} \mathcal{L}_{ij}^2 A_{j,t}^{-1} \quad (10) \]

**Lemma 2.** \( t_i \) is the set of time at which user \( i \) is served up to time \( t \). \( A_{i,t} = \sum_{\tau \in t_i} x_{\tau} x_{\tau}^T \), \( V_{i,t} = A_{i,t} + \alpha \mathcal{L}_{ii} I \), \( \xi_{i,t} = \sum_{\tau \in t_i} x_{\tau} \eta_{i,\tau} \), \( I \in \mathbb{R}^{d \times d} \) is the identity matrix. \( \Lambda_{i,t} \) is defined in Eq. 10. Denote \( \Delta_i = \sum_{j=1}^{n} \mathcal{L}_{ij} \theta_j \), the size of the confidence set defined in Eq. 8 satisfies the following upper bound with probability \( 1 - \delta \) with \( \delta \in [0, 1] \).

\[
\beta_{i,t} = \sigma \sqrt{2 \log \frac{|V_{i,t}|^{1/2}}{\delta |\alpha I|^{1/2}}} + \sqrt{\alpha ||\Delta_i||_2}
\]
The basic idea is selecting arm taking into account the optimism in the face of uncertainty. The selection principle which is based on the principle called Laplacian regret. In contrast, the two unique properties in user. In the other case, if the symmetric normalized user will be combinatorially, if the combinatorial Laplacian instead of other commonly used Laplacians. Specifically, if the combinatorial Laplacian is used, the term of each user depends the similarity (smoothness) between user which is quantified. Clearly, the computation complexity of Lemma 4. In this case, the graph structure is included in the confidence set which is not homogeneous across all levels of smoothness in Fig. 5. We provide empirical evidence on used in b), empty graph, i.e., which means instead of other commonly used Laplacians. Specifically, if the combinatorial Laplacian is used in the initialization in computation complexity.

Algorithm 1: GraphUCB

| Input : $\alpha, T, \mathcal{L}, \delta$ |
| Initialization : For any $i \in \{1, 2, ..., n\}$ |
| $\hat{\theta}_0, i = 0 \in \mathbb{R}^d$, $\Lambda_0, i = 0 \in \mathbb{R}^{d \times d}$, $A_0, i = 0 \in \mathbb{R}^{d \times d}$, $\beta_{i,t} = 0$ |
| for $t \in [1, T]$ do |
| User index $i_t$ is selected |
| 1. $A_{i,t} \leftarrow A_{i,t-1} + x_{i,t-1}x_{i,t-1}^T$ if $i = i_t$. |
| 2. $A_{j,t} \leftarrow A_{j,t-1}$, $\forall j \neq i_t$. |
| 3. Update $\Lambda_{i,t}$ |
| 4. Select $x_{i,t}$ $\arg\max_{x \in \Omega} x^T \hat{\theta}_{i,t} + \beta_{i,t}||x||_{\Lambda_{i,t}^{-1}}$ |
| 5. Receive the payoff $y_{i,t}$. |
| 6. Update $\hat{\Theta}_t$ |
| end |
Lemma 3. Define $\Psi_{i,t_i} = \frac{\sum_{t=1}^{t_i} ||x_{i,t}||^2_{\Lambda_{i,t_i}^{-1}}}{\sum_{t=1}^{t_i} ||x_{i,t}||^2_{V_{i,t}^{-1}}}$, where $V_{i,t_i} = A_{i,t_i} + \alpha L_{ii}I$ and $\Lambda_{i,t_i}$ defined\(^1\) in Eq. 10. Without loss of generality, assume $||x_{i,t}||_2 \leq 1$ for any $t, t_i$ and $i$, then

$$\Psi_{i,t_i} \in (0, 1]$$  \hspace{1cm} (14)

Furthermore, denser connected graph leads to smaller $\Psi_{i,t_i}$. Empirical evidence is provided in Fig. 1-b.

$$\Lambda_{i,t} = A_{i,t} + 2\alpha L_{ii}I + \alpha^2 \sum_{j=1}^{n} L_{ij}^2 A_{j,t}^{-1}$$  \hspace{1cm} (10)
Lemma 3. Define $\Psi_{i,t_i} = \frac{\sum_{t=1}^{t_{i}} ||x_{i,t}||^2_2 \Lambda^{-1}_{i,t_i}}{\sum_{t=1}^{t_{i}} ||x_{i,t}||^2_2 V^{-1}_{i,t_i}}$, where

$V_{i,t_i} = A_{i,t_i} + \alpha L_{ii} I$ and $\Lambda_{i,t_i}$ defined in Eq. 10. Without loss of generality, assume $||x_{i,t}||_2 \leq 1$ for any $t$, $t_i$ and $i$, then

$\Psi_{i,t_i} \in (0, 1]$ \hspace{1cm} (14)

Furthermore, denser connected graph leads to smaller $\Psi_{i,t_i}$. Empirical evidence is provided in Fig. 1-b.

$\Lambda_{i,t} = A_{i,t} + 2\alpha L_{ii} I + \alpha^2 \sum_{j=1}^{n} L_{ij}^2 A_{j,t}^{-1}$ \hspace{1cm} (10)

it provides a comparison with no-graph UCB
Regret Analysis

Single User Regret
The cumulative regret over $t_i$ of user $i$ satisfies the following upper bound with probability $1 - \delta$

$$\Theta\left(\left(\sqrt{d \log(t_i)} + \sqrt{\alpha \| \Delta_i \|_2}\right)\Psi_{i,t_i} \sqrt{dt_i \log(t_i)}\right) = \Theta\left(d \sqrt{t_i \Psi_{i,t_i}}\right)$$

Network Regret
Assuming users are served uniformly, then, over the time horizon $T$, the total cumulative regret $R_T = \sum_{i=1}^{n} R_{i,t_i}$ experienced by all users satisfies the following upper bound with probability $1 - \delta$

$$\Theta\left(d \sqrt{Tn \max_i \Psi_{i,t_i}}\right)$$
Single User Comparison

Single user

LinUCB

\[ O\left( \left( \sqrt{d \log(t_i)} + \sqrt{\alpha \| \theta_i \|_2} \right) \sqrt{dt_i \log(t_i)} \right) \]

GraphUCB

\[ O\left( \left( \sqrt{d \log(t_i)} + \sqrt{\alpha \| \Delta_i \|_2} \right) \Psi_{i,t_i} \sqrt{dt_i \log(t_i)} \right) \]

\[ \| \Delta_i \|_2 \in [0, \| \theta_i \|_2] \quad \Psi_{i,t_i} \in [0,1] \]

Single user comparison

\[ \Theta_{u \neq D} \otimes A_A \]

LinUCB

\[ \Theta \left( (\sqrt{d \log(t_i)} + \sqrt{\alpha \|\theta_i\|_2}) \sqrt{dt_i \log(t_i)} \right) \]

GraphUCB

\[ \Theta \left( (\sqrt{d \log(t_i)} + \sqrt{\alpha \|\Delta_i\|_2}) \Psi_{i,t_i} \sqrt{dt_i \log(t_i)} \right) \]

\[ \|\Delta_i\|_2 \in [0, \|\theta_i\|_2] \quad \Psi_{i,t_i} \in [0,1] \]

Smoothness and connectivity reduce the regret

Single User Comparison

Single user


LinUCB \[ O\left( \left( \sqrt{d \log(t_i)} + \sqrt{\alpha \| \theta_i \|_2} \right) \sqrt{dt_i \log(t_i)} \right) \]

GraphUCB \[ O\left( \left( \sqrt{d \log(t_i)} + \sqrt{\alpha \| \Delta_i \|_2} \right) \Psi_{i,t_i} \sqrt{dt_i \log(t_i)} \right) \]

All users

GOB.Lin \[ O\left( nd \sqrt{T} \right) \]

GraphUCB \[ O\left( d \sqrt{Tn} \max_i \Psi_{i,t_i} \right) \]
Single User Comparison

**Single user**

- **LinUCB**
  \[ \mathcal{O}\left(\sqrt{d \log(t_i)} + \sqrt{\alpha \|\theta_i\|_2} \sqrt{dt_i \log(t_i)}\right) \]

- **GraphUCB**
  \[ \mathcal{O}\left(\sqrt{d \log(t_i)} + \sqrt{\alpha \|\Delta_i\|_2} \Psi_{i,t_i} \sqrt{dt_i \log(t_i)}\right) \]

**All users**

- **GOB.Lin**
  \[ \mathcal{O}\left(nd \sqrt{T}\right) \]

- **GraphUCB**
  \[ \mathcal{O}\left(d \sqrt{Tn \max_i \Psi_{i,t_i}}\right) \]

---

Results - Synthetic

Figure 3: Performance on graph types: ER (a), BA (b), WS (c) and RBF (d).

Terms of the kernel set as 0.5. In addition, we generate another RBF kernel graph the same as the previous one except that edge with weight less than 0.5 are removed. This graph is used to test the effect of sparsity on algorithms.

The results are shown in Figure 2. Under all graph models, GraphUCB outperforms its competitors consistently with a large margin. GraphUCB-Local shows a slight worse performance than GraphUCB. This is due to the approximation introduced by Eq. 7. This is a tight approximation when T is large, so at the initial phase when T is small, GraphUCB-Local experiences more regret. However, it is a worth price since GraphUCB-Local reduces the computation complexity from $\tilde{O}(n^2d^2)$ to $\tilde{O}(nd^2)$. At early phase, CLUB performs better than LinUCB due to clustering users into groups. However, its regret does not converge fast. This is because CLUB approximate each user feature by that of clustering which eventually limits its ability to model individual users accurately.

7.2 Performance on Graph Structure

Graphs used in our experiments have properties affecting the performance of proposed algorithms. We examine these graph properties.

Sparsity of RBF-graph: We test the effect of sparsity on performance of algorithms. We first generate a fully connected graph and generated the edge weights randomly. Then, $\epsilon$ is generated following Eq. 21 with $\epsilon = 4$. To control the sparsity, we set a threshold $\epsilon_2 [0,1)$ on edge weights such that edges with weights less than $\epsilon_2$ are removed.

7.3 Experiments on Real-World Data

We then carry out experiments on two real-world datasets that are commonly used in bandit problems: Movielens [Lam and Herlocker, 2006] and Netflix [Bennett et al., 2007]. We follow the data pre-processing steps in [Valko et al., 2014]. We sample 50 users and test algorithms over $T = 1000$.

In Figure 3, we see that the proposed GraphUCB and...
**Results - Real World Data**

Figure 3: Performance on graph types: ER (a), BA (b), WS (c) and RBF (d).

- Sparsity of RBF-graph: We test the effect of sparsity on performance of algorithms. We first generate a fully connected graph and generated the edge weights randomly. Then, $\tau$ is generated following Eq. 21 with $\tau = 4$. To control the sparsity, we set a threshold $\tau$ on edge weights such that edges with weights less than $\tau$ are removed.

7.3 Experiments on Real-World Data

We then carry out experiments on two real-world datasets that are commonly used in bandit problems: MovieLens [Lam and Herlocker, 2006] and Netflix [Bennett et al., 2007]. We follow the data preprocessing steps in [Valko et al., 2014]. We sample 50 users and test algorithms over $T = 1000$.

Figure 5: Performance on MovieLens (a) and Netflix (b).
Results - Graph Features

(a) Smoothness: $\gamma$ in Eq. 21

(b) RBF (Sparsity)
Conclusions

• Proposed GraphUCB to solve the stochastic linear bandit problem with multiple users - known user graph

• Single-user UCB

• GraphUCB leads to lower cumulative regret as compared to algorithms which ignore user graph

• Proposed local-GraphUCB - need further investigation
Conclusions

- Proposed GraphUCB to solve the stochastic linear bandit problem with multiple users - known user graph
- Single-user UCB
- GraphUCB leads to lower cumulative regret as compared to algorithms which ignore user graph
- Proposed local-GraphUCB - need further investigation
- Next?
  - better understanding of the effect of the graph
  - bandit optimality as function of graph features
- graph learning and other GSP properties applied to MABs?
References

- Yang, K. and Toni, L., Graph-based recommendation system, IEEE GlobalSIP, 2018
- Gentile, C., Li, S., and Zappella, G. Online clustering of bandits, ICML 2014
- M. Valko et al., “Spectral Bandits for Smooth Graph Functions”, JMLR 2014
- Q. Gu and J. Han, “Online spectral learning on a graph with bandit feedback”, in Proc. IEEE Int. Conf. on Data Mining, 2014
- Vaswani, S., Schmidt, M., and Lakshmanan, L. V., Horde of bandits using gaussian markov random fields, AISTATS 2017
Thank You! Questions?

Learning and Signal Processing Lab
UCL

https://laspucl2016.com