Horde of Bandits using GMRFs

Sharan Vaswani, Mark Schmidt, Laks Lakshmanan
University of British Columbia
Introduction

Input:

Protocol:
For $t = 1$ to $T$

- Recommend an item $j$ to target user $i_t$
- Observe rating $r_{i,t,j}$
- Obtain new estimate for preferences of user $i$
Introduction

- New marketer without any user meta-data or available rating information.
- Each item j can be described by its content $x_j$.
- The generative model for ratings is linear, i.e.

$$r_{i,j} = (w^*_i)^T x_j + \eta_{i,j,t}$$

Rating by user i on item j  
“True” user preference  
Zero mean sub-gaussian noise

Framework: **Contextual Bandits**\(^1\) - Sequential framework to trade-off exploration (learning about user preferences) and exploitation (making good recommendations).

**Aim:** Minimize regret across a time horizon $T$.

$$R(T) = \sum_{t=1}^{T} \left[ \max_{j \in C_t} (w_{i_t}^T x_j) - w_{i_t}^T x_{j_t} \right]$$

- **Regret**
- **Set of context vectors**
- **“True” preference for the target user**
- **Item chosen by the algorithm**
Gang of Bandits (GOB)$^1$

Motivation: Users interact with each other. Especially true for RS associated with social networks. Eg: 

Basic Idea:

- Extend the contextual bandits to use the social network to make better recommendations by sharing feedback between users.
- Assume homophily$^2$: Users connected in the network have similar preferences

Estimating user preferences in GOB

\[ w_t = \arg\min_w \left[ \sum_{i=1}^{n} \sum_{k \in M_{i,t}} (w_i^T x_k - r_{i,k})^2 + \lambda w^T (L \otimes I_d) w \right] \]

- \( w_t \): \( n \)-dimensional mean estimate of concatenated preferences
- \( M_{i,t} \): Set of items rated by user i until round t
- \( r_{i,k} \): Observed rating
- \( L \): Laplacian of the social network

Limitations of previous work:
- Algorithm in [1] has \( O(n^2d^2) \) space and time complexity. Not scalable
- Clustering based approaches\(^3\) lose personalization.

Contributions

- Propose a scalable solution for estimating the mean by making a connection to Gaussian Markov Random Fields.
- Analyze the computational complexity and prove regret bounds for Epoch-Greedy and Thompson sampling.
- Propose a heuristic to learn the graph on the fly.
Scaling up GOB

Basic Idea: Mean estimation in GOB is equivalent to MAP estimation in a GMRF

Likelihood: \( r_{i,j} \sim \mathcal{N}(w_i^T x_j, \sigma^2) \)
Prior: \( w \sim \mathcal{N}(0, (\lambda L \otimes I_d)^{-1}) \)

Posterior:
\[
\mathcal{N}(\hat{w}_t, \Sigma_t^{-1}) \quad \hat{w}_t = \frac{1}{\sigma^2} \Sigma_t^{-1} b_t
\]
\[
\Sigma_t = \frac{1}{\sigma^2} X_t^T X_t + \lambda (L \otimes I_d) \quad b_t = X_t^T r_t
\]

- Solve by conjugate gradient in time \( O(\kappa(nd^2 + d \cdot \text{nnz}(L))) \)
- Space: \( O(nd^2 + \text{nnz}(L)) \)
Contributions

- Make a connection to Gaussian Markov Random Fields and propose a scalable solution for estimating the mean
- Analyze the computational complexity and prove regret bounds for Epoch-Greedy and Thompson sampling.
- Propose a heuristic to learn the graph on the fly.
Algorithms - Upper Confidence Bound

UCB rule:

\[ \hat{j}_t = \arg \max_{j \in C_t} \left( \langle w_t, x_{i_t,j} \rangle + \alpha_t \sqrt{x_{i_t,j}^T \Sigma_t^{-1} x_{i_t,j}} \right) \]

- Requires \( O(d) \) time
- Requires \( O(\kappa(nd^2 + d \cdot \text{nnz}(L))) \) time

- Not scalable if the number of context vectors is large
Algorithms - Epoch-Greedy

- **Algorithm:** Divide T into Q epochs. In each epoch,
  - Do 1 round of random exploration (Pick an available item at random)
  - Do “some” steps of exploitation i.e. \( \hat{j}_t = \arg\max_{j \in C_t} \langle w_t, x_{i_t,j} \rangle \)

- **Regret Bound:**

\[
R(T) = \tilde{O}\left(n^{1/3} \left(\frac{\text{Tr}(L^{-1})}{\lambda n}\right)^{\frac{1}{3}} T^{\frac{2}{3}}\right)
\]

Connectivity of the graph Sub-optimal dependence on T
Algorithms - Thompson Sampling

- **Algorithm:**
  - Obtain a sample from the posterior, i.e. $\tilde{w}_t \sim \mathcal{N}(w_t, \Sigma_t^{-1})$
  - Pick greedily using this sample i.e. $j_t = \arg\max_{j \in C_t} \langle \tilde{w}_t, x_{i_t,j} \rangle$

- **Naive Sampling:**
  - Compute sparse Cholesky factor from prior covariance = $L \otimes I_d$
  - Make rank-1 updates to it to obtain Cholesky factor for covariance at round $t$

**Problem:** Cholesky factor gets dense, leading to $O(d^2n^2)$ cost for sampling
Algorithms - Thompson Sampling

- **Proposed Sampling**\(^1\):

\[ \sum_t \tilde{\mathbf{w}}_t = (L \otimes I_d)\tilde{\mathbf{w}}_0 + X_t^T\tilde{\mathbf{r}}_t \]

- **Regret Bound:**

\[ R(T) = \tilde{\mathcal{O}} \left( \frac{dn\sqrt{T}}{\sqrt{\lambda}} \right) \sqrt{ \log \left( \frac{3 \text{Tr}(L^{-1})}{n} + \frac{\text{Tr}(L^{-1})T}{\lambda dn^2\sigma^2} \right) } \]

\[ \text{Near-optimal dependence on } T \]

---

1. Papandreou et al. "Gaussian sampling by local perturbations"
Experiments - Scalability

**Graph Based:** G-EG, GOBLIN (Cesa-Bianchi’13), GOBLIN++ (scalable GOBLIN), G-TS

**Baselines: No sharing:** EG-IND, LINUCB-IND, TS-IND; **Clustering:** CLUB

**Scaling with Dimension**

**Scaling with Network Size**
Experiments - Regret

Delicious

Last FM
Contributions

- Make a connection to Gaussian Markov Random Fields and propose a scalable solution for estimating the mean
- Analyze the computational complexity and prove regret bounds for Epoch-Greedy and Thompson sampling.
- Propose a heuristic to learn the graph on the fly.
Learning the Graph

**L-EG**: Learning starting from empty graph; **U-EG**: Updating starting from given graph
Future Work

- Tighten the regret bound for Thompson Sampling
- Prove regret bounds for “Learning the graph” variant
Contributions

- Make a connection to Gaussian Markov Random Fields and propose a scalable solution for estimating the mean
- Analyze the computational complexity and prove regret bounds for Epoch-Greedy and Thompson sampling.
- Propose a heuristic to learn the graph on the fly.
Backup slides
Learning the Graph

\[
[w_t, V_t] = \arg\min_{w, V} \|r_t - \Phi_t w\|^2_2 + \text{Tr} \left( V (\lambda W^T W + V_{t-1}^{-1}) \right) + \lambda_2 \|V\|_1 - (dn + 1) \ln |V|
\]

- Estimate of the inverse covariance matrix
- Data fitting term
- Smoothness of preferences wrt to learnt graph
- Smooth change in graph structure
- Encourage a sparse graph
- Penalize complexity of graph
- Solve by alternating minimization:
  - w-subproblem: Same as MAP estimation
  - V-subproblem: Same as Graphical Lasso

\[
V_t = \arg\min_V \text{Tr} \left( (V [\lambda W_t^T W_t + V_{t-1}^{-1}]) + \lambda_2 \|V\|_1 - (dn + 1) \ln |V| \right)
\]

Empirical Covariance