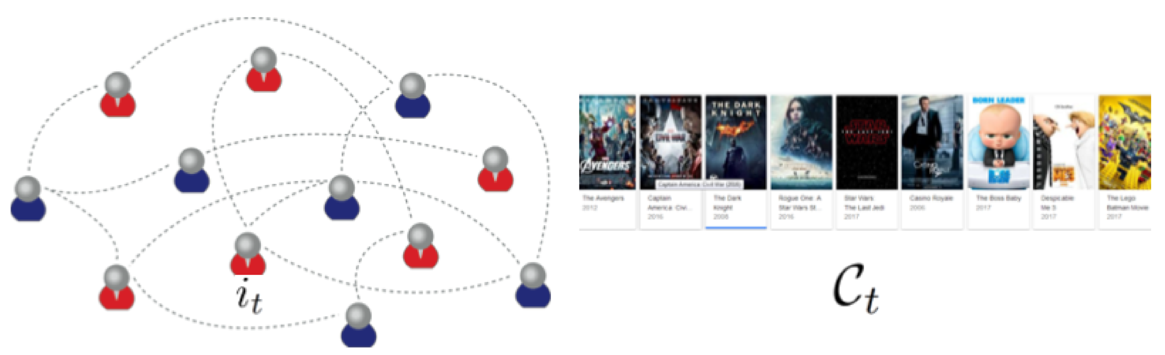
Horde of Bandits using Gaussian Markov Random Fields

Sharan Vaswani, Mark Schmidt, Laks.V.S.Lakshmanan
Department of Computer Science, University of British Columbia

Summary

- We address a recommender system setting without prior rating data and constantly-changing items (E.g. news articles)
- Previous work (Cesa-Bianchi'13) show that, in addition to features, sharing information across users improves performance. But previous algorithms are not scalable. Other previous approaches cluster nodes (Gentile'14), but lose personalization.
- We show how to scale to large graphs by making a connection to Gaussian Markov Random Fields (GMRFs).
- We also prove regret bounds and give a heuristic to learn the graph on the fly.

Gang of Bandits model (Cesa-Bianchi'13)



- **Input:**
  - Recommender system (RS) with no past rating data or user meta-data.
  - Each item j can be described by a set of features $\mathbf{x}_j$.
  - RS has an associated network (with Laplacian L).
- **Aim:** Use **contextual bandits** to trade-off exploration (learn users' preferences) and exploitation (recommend items which the user likes) and the **associated network** to share information between users and improve recommendations.
- **Assumptions:**
  - **Linear generative model:** $r_{i,j} = \langle \mathbf{w}_i^*, \mathbf{x}_j \rangle + \eta_{i,j,t}$.
  - **Homophily:** Users connected in the network have similar preferences.
- **Objective:** Minimize regret $R(T) = \sum_{t=1}^{T} \left[\max_{j \in \mathcal{C}_t} \left(\langle \mathbf{w}_{i_t}^*, \mathbf{x}_j \rangle \right) - \langle \mathbf{w}_{i_t}^*, \mathbf{x}_{j_t} \rangle \right]$
- **Mean estimation:** $\mathbf{w}_t = \operatorname{argmin}_{\mathbf{w}} \left[\sum_{i=1}^{n} \sum_{k \in \mathcal{M}_{i,t}} (\mathbf{w}_i^T \mathbf{x}_k - r_{i,k})^2 + \lambda \mathbf{w}^T (L \otimes I_d) \mathbf{w} \right]$

Scaling up Gang of Bandits

- **Basic Idea:** Mean estimation is equivalent to MAP estimation in a GMRF.
- Likelihood: $r_{i,j} \sim \mathcal{N}(\langle \mathbf{w}_i, \mathbf{x}_j \rangle, \sigma^2)$, Prior: $\mathbf{w} \sim \mathcal{N}(0, (\lambda L \otimes I_d)^{-1})$
- Posterior: $\mathcal{N}(\hat{\mathbf{w}}_t, \Sigma_t^{-1})$ such that $\Sigma_t = \frac{1}{\sigma^2} X_t^T X_t + \lambda (L \otimes I_d)$ and $\hat{\mathbf{w}}_t = \frac{1}{\sigma^2} \Sigma_t^{-1} \mathbf{b}_t$ with $\mathbf{b}_t = X_t^T \mathbf{r}_t$.
- Solve linear system using conjugate gradient in $O(\kappa(nd^2 + d \cdot \text{nnz}(L)))$ time and $O(nd^2 + \text{nnz}(L))$ space.

Algorithms

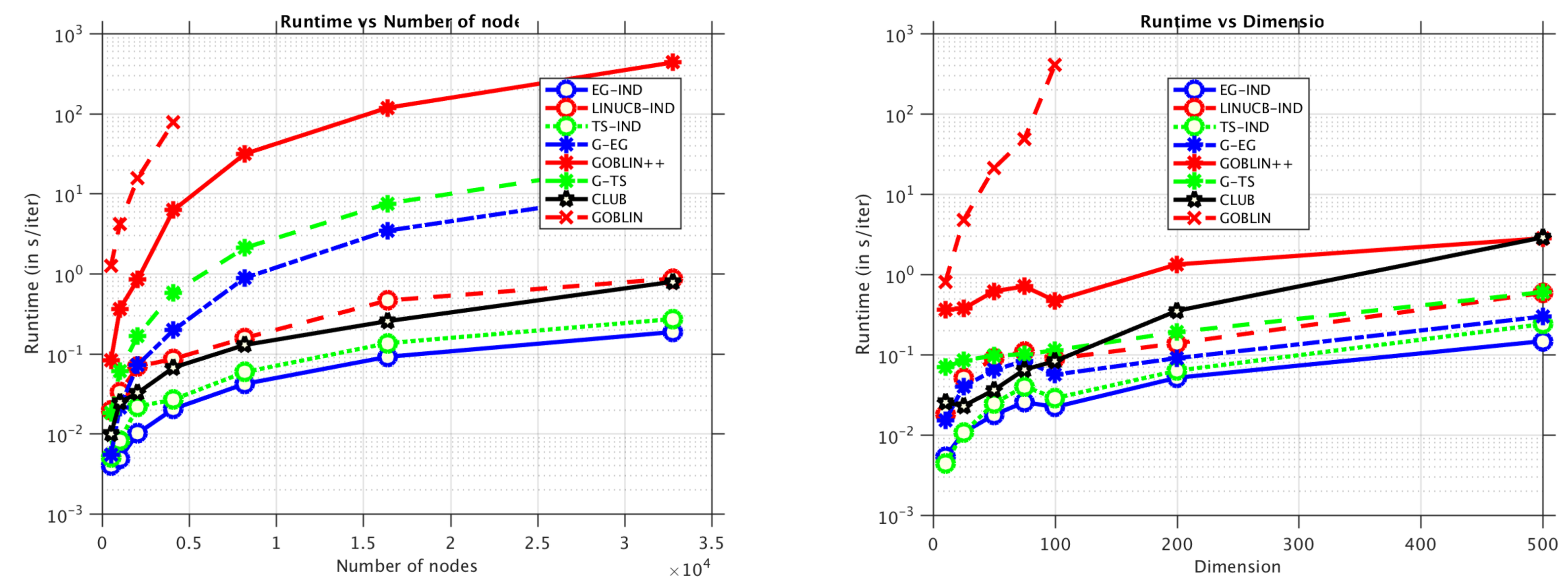
- **UCB**
  - **Algorithm:** Pick $j_t = \operatorname{argmax}_{j \in \mathcal{C}_t} \left(\langle \mathbf{w}_t, \mathbf{x}_{i_t,j} \rangle + \alpha_t \sqrt{\mathbf{x}_{i_t,j}^T \Sigma_t^{-1} \mathbf{x}_{i_t,j}} \right)$
- **Epoch Greedy:**
  - **Algorithm:** Explicitly separate exploration and exploitation rounds. Exploration - Pick a random item. Exploitation - Pick $j_t = \operatorname{argmax}_{j \in \mathcal{C}_t} \langle \mathbf{w}_t, \mathbf{x}_{i_t,j} \rangle$.
  - **Regret:** $R(T) = \tilde{O}\left(n^{1/3} \left(\frac{\text{Tr}(L^{-1})}{\lambda n} \right)^{\frac{1}{3}} T^{\frac{2}{3}} \right)$

Large Scale Thompson Sampling

- **Algorithm:** Obtain sample from posterior i.e. $\tilde{\mathbf{w}}_t \sim \mathcal{N}(\mathbf{w}_t, \Sigma_t^{-1})$. Pick $j_t = \operatorname{argmax}_{j \in \mathcal{C}_t} \langle \tilde{\mathbf{w}}_t, \mathbf{x}_{i_t,j} \rangle$.
- **Regret:** $R(T) = \tilde{O}\left(\frac{dn\sqrt{T}}{\sqrt{\lambda}} \sqrt{\log\left(\frac{3\,\text{Tr}(L^{-1})}{n} + \frac{\text{Tr}(L^{-1})T}{\lambda d n^2 \sigma^2} \right)} \right)$
- **Naive Sampling:** Using Cholesky factorization. Requires $O(n^2d^2)$ computation.
- **Proposed Sampling:** To obtain unbiased sample from a GMRF (Papandreou'10), solve $\Sigma_t \tilde{\mathbf{w}}_t = (L \otimes I_d) \tilde{\mathbf{w}}_0 + X_t^T \tilde{\mathbf{r}}_t$. Same computational complexity as MAP estimation.

Experiments

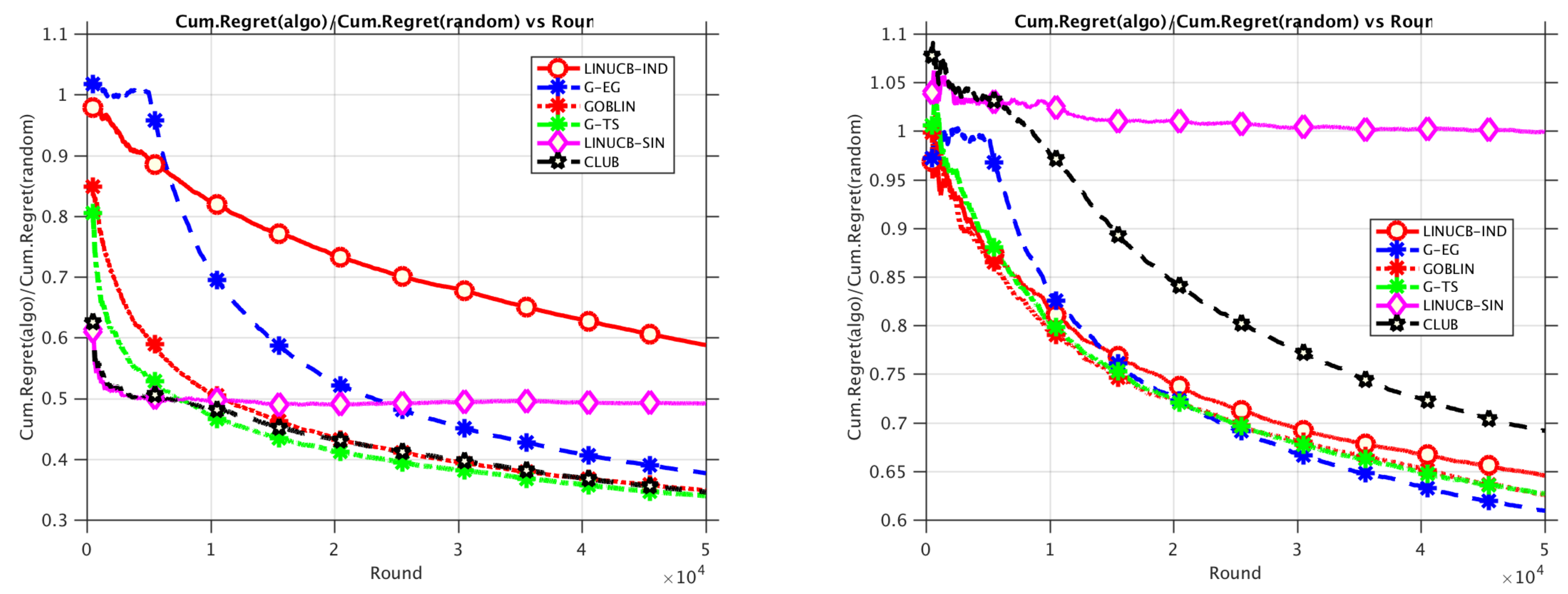
- **Graph Based:** G-EG, GOBLIN (Cesa-Bianchi'13), GOBLIN++ (scalable GOBLIN), G-TS
- **Baselines: No sharing:** EG-IND, LINUCB-IND, TS-IND; **No personalization:** LINUCB-SIN; **Clustering:** CLUB
- **Scalability:**



(a) (b)

Figure: Synthetic network: Runtime (in seconds/iteration) vs (a) Number of nodes (b) Dimension

- **Regret:**

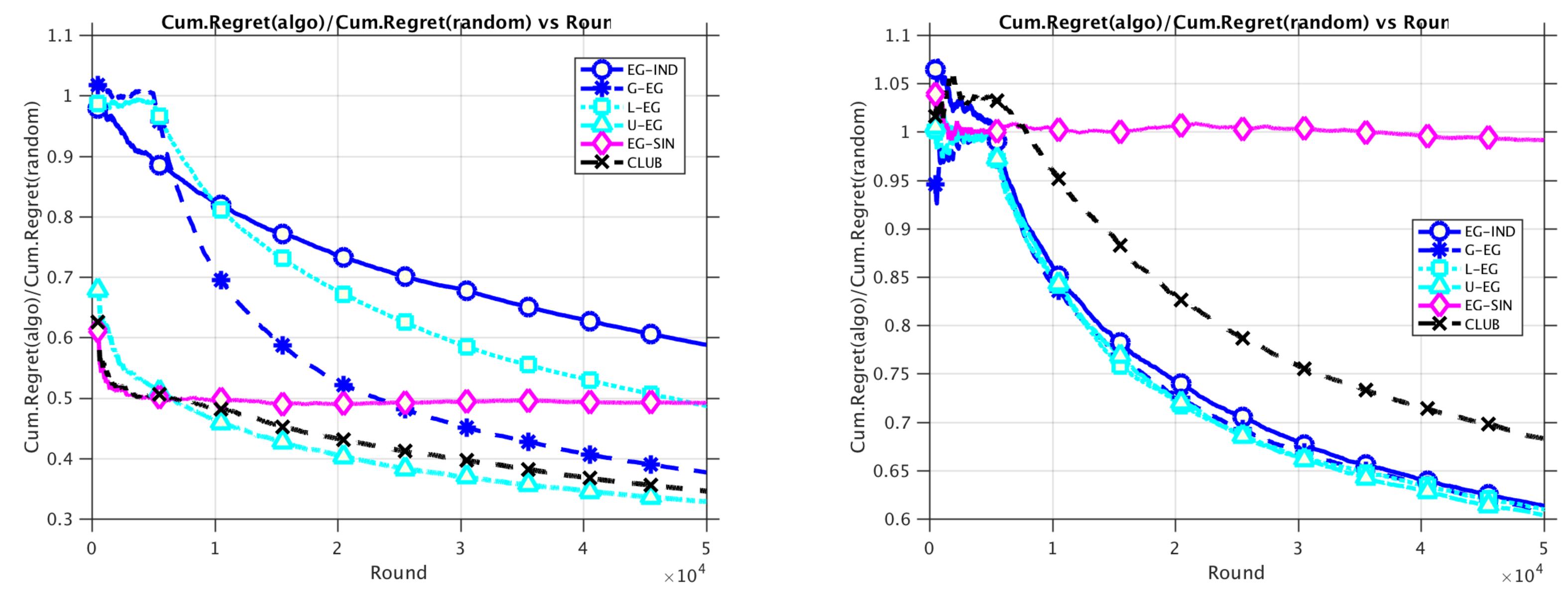


(a) Last.fm (b) Delicious

Figure: Regret Minimization

Learning the graph on the fly

- If the graph is not available, write a joint minimization w.r.t $\mathbf{w}_t$ and precision matrix V_t: $[\mathbf{w}_t, V_t] = \operatorname{argmin}_{\mathbf{w},V} ||\mathbf{r}_t - X_t \mathbf{w}||_2^2 + \text{Tr}\left(V(\lambda W^T W + V_{t-1}^{-1}) \right) + \lambda_2 ||V||_1 - (dn+1) \ln |V|$
- **Variants:** L-EG: Learning starting from empty graph; U-EG: Updating starting from given graph



(a) Last.fm (b) Delicious

Figure: Regret Minimization while learning the graph

Future Work

- Tighten the regret bound for Thompson Sampling and prove regret bounds for the learning the graph variant.