Horde of Bandits using Gaussian Markov Random Fields

Sharan Vaswani **Mark Schmidt** **Laks V.S. Lakshmanan**
University of British Columbia

Abstract

The gang of bandits (GOB) model [7] is a recent contextual bandits framework that shares information between a set of bandit problems, related by a known (possibly noisy) graph. This model is useful in problems like recommender systems where the large number of users makes it vital to transfer information between users. Despite its effectiveness, the existing GOB model can only be applied to small problems due to its quadratic time-dependence on the number of nodes. Existing solutions to combat the scalability issue require an often-unrealistic clustering assumption. By exploiting a connection to Gaussian Markov random fields (GMRFs), we show that the GOB model can be made to scale to much larger graphs without additional assumptions. In addition, we propose a Thompson sampling algorithm which uses the recent GMRF sampling-by-perturbation technique, allowing it to scale to even larger problems (leading to a "horde" of bandits). We give regret bounds and experimental results for GOB with Thompson sampling and epoch-greedy algorithms, indicating that these methods are as good as or significantly better than ignoring the graph or adopting a clustering-based approach. Finally, when an existing graph is not available, we propose a heuristic for learning it on the fly and show promising results.

1 Introduction

Consider a newly established recommender system (RS) which has little or no information about the users' preferences or any available rating data. The unavailability of rating data implies that we can not use traditional collaborative filtering based methods [33]. Furthermore, in the scenario of personalized news recommendation or for recommending trending Facebook posts, the set of available items is not fixed but instead changes continuously. This new RS can recommend items to the users and observe their ratings to learn their preferences from this feedback ("exploration"). However, in order to retain its users, at the same time it should recommend "relevant" items that will be liked by and elicit higher ratings from users ("exploitation"). Assuming each item can be described by its content (like tags describing a news article or video), the contextual bandits framework [23] offers a popular approach for addressing this exploration-exploitation trade-off.

However, this framework assumes that users interact with the RS in an isolated manner, when in fact a RS might have an associated social component. In particular, given the large number of users on such systems, we may be able to learn their preferences more quickly by leveraging the relations between them. One way to use a social network of users to improve recommendations is with the recent gang of bandits (GOB) model [7]. In particular, the GOB model exploits the homophily effect [29] that suggests users with similar preferences are more likely to form links in a social network. In other words, user preferences vary smoothly across the social graph and tend to be similar for users connected with each other. This allows us to transfer information between users; we can learn about a user from his or her friends' ratings. However, the existing recommendation algorithm in the GOB framework has a *quadratic* time-dependence on the number of nodes (users) and thus can only be used for a small number of users. Several recent works have tried to improve the scaling of the GOB model by clustering the users into groups [15, 30], but this limits the flexibility of the model and loses the ability to model individual users' preferences.

In this paper, we cast the GOB model in the framework of Gaussian Markov random fields (GMRFs) and show how to exploit this connection to scale it to much larger graphs. Specifically, we interpret the GOB model as the optimization of a Gaussian likelihood on the users' observed ratings and interpret the user-user graph as the

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

prior inverse-covariance matrix of a GMRF. From this perspective, we can efficiently estimate the users' preferences by performing MAP estimation in a GMRF. In addition, we propose a Thompson sampling GOB variant that exploits the recent sampling-by-perturbation idea from the GMRF literature [31] to scale to even larger problems. This idea is fairly general and might be of independent interest in the efficient implementation of other Thompson sampling methods. We establish regret bounds (Section 4) and provide experimental results (Section 5) for Thompson sampling as well as an epoch-greedy strategy. These experiments indicate that our methods are as good as or significantly better than approaches which ignore the graph or that cluster the nodes. Finally, when the graph of users is not available, we propose a heuristic for learning the graph and user preferences simultaneously in an alternating minimization framework (Appendix A).

2 Related Work

Social Regularization: Using social information to improve recommendations was first introduced by Ma et al. [25]. They used matrix factorization to fit existing rating data but constrained a user's latent vector to be similar to their friends in the social network. Other methods based on collaborative filtering followed [32, 12], but these works assume that we already have rating data available. Thus, these methods do not address the exploration-exploitation trade-off faced by a new RS that we consider.

Bandits: The multi-armed bandit problem is a classic approach for trading off exploration and exploitation as we collect data [20]. When features (context) for the "arms" are available and changing, it is referred to as the *contextual* bandit problem [4, 23, 9]. The contextual bandit framework is important for the scenario we consider where the set of items available is constantly changing, since the features allow us to make predictions about items we have never seen before. Algorithms for the contextual bandits problem include epoch-greedy methods [21], those based on upper confidence bounds (UCB) [9, 1], and Thompson sampling methods [2]. Note that these standard contextual bandit methods do not model the user-user dependencies that we want to exploit.

Several graph-based methods to model dependencies between the users have been explored in the (non-contextual) multi-armed bandit framework [6, 27, 3, 26], but the GOB model of Cesa-Bianchi et al. [7] is the first to exploit the network between users in the contextual bandit framework. They proposed a UCB-style algorithm and showed that using the graph leads to lower regret from both a theoretical and practical standpoint. However, their algorithm has a time complexity that is quadratic in the number of users. This makes it infeasible for typical RS that have tens of thousands (or even millions) of users.

To scale up the GOB model, several recent works propose to cluster the users and assume that users in the same cluster have the same preferences [15, 30]. But this solution loses the ability to model individual users' preferences, and indeed our experiments indicate that in some applications clustering significantly hurts performance. In contrast, we want to scale up the original GOB model that learns more fine-grained information in the form of a preference-vector specific to each user.

Another interesting approach to relax the clustering assumption is to cluster both items and users [24], but this only applies if we have a fixed set of items. Some works consider item-item similarities to improve recommendations [34, 17], but this again requires a fixed set of items while we are interested in RS where the set of items may constantly be changing. There has also been work on solving a single bandit problem in a distributed fashion [18], but this differs from our approach where we are solving an individual bandit problem on each of the n nodes. Finally, we note that *all* of the existing graph-based works consider relatively small RS datasets ($\sim 1k$ users), while our proposed algorithms can scale to much larger RS.

3 Scaling up Gang of Bandits

In this section we first describe the general GOB framework, then discuss the relationship to GMRFs, and finally show how this leads to more scalable method. In this paper $\mathrm{Tr}(A)$ denotes the trace of matrix A, $A \otimes B$ denotes the Kronecker product of matrices A and B, I_d is used for the d-dimensional identity matrix, and $\mathrm{vec}(A)$ is the stacking of the columns of a matrix A into a vector.

3.1 Gang of Bandits Framework

The contextual bandits framework proceeds in rounds. In each round t, a set of items $\mathcal{C}_t$ becomes available. These items could be movies released in a particular week, news articles published on a particular day, or trending stories on Facebook. We assume that $|\mathcal{C}_t| = K$ for all t. We assume that each item j can be described by a context (feature) vector $\mathbf{x}_j \in \mathbb{R}^d$. We use n as the number of users, and denote the (unknown) ground-truth preference vector for user i as $\mathbf{w}_i^* \in \mathbb{R}^d$. Throughout the paper, we assume there is only a single target user per round. It is straightforward extend our results to multiple target users.

Given a target user i_t, our task is to recommend an available item $j_t \in \mathcal{C}_t$ to them. User i_t then provides feedback on the recommended item j_t in the form of a rating r_{i_t,j_t}. Based on this feedback, the estimated preference vector for user i_t is updated. The recommendation algorithm must trade-off between exploration

(learning about the users' preferences) and exploitation (obtaining high ratings). We evaluate performance using the notion of *regret*, which is the loss in recommendation performance due to lack of knowledge of user preferences. In particular, the regret $R(T)$ after T rounds is given by:

$$R(T) = \sum_{t=1}^{T} \left[\max_{j \in \mathcal{C}_t} (\mathbf{w}_{i_t}^{*T} \mathbf{x}_j) - \mathbf{w}_{i_t}^{*T} \mathbf{x}_{j_t} \right]. \tag{1}$$

In our analysis we make the following assumptions:

Assumption 1. *The ℓ_2-norms of the true preference vectors and item feature vectors are bounded from above. Without loss of generality we'll assume $||x_j||_2 \leq 1$ for all j and $||\mathbf{w}_i^*||_2 \leq 1$ for all i. Also without loss of generality, we assume that the ratings are in the range $[0, 1]$.*

Assumption 2. *The true ratings can be given by a linear model [23], meaning that $r_{i,j} = (\mathbf{w}_i^*)^T \mathbf{x}_j + \eta_{i,j,t}$ for some noise term $\eta_{i,j,t}$.*

These are standard assumptions in the literature. We denote the history of observations until round t as $\mathbb{H}_{t-1} = \{(i_\tau, j_\tau, r_{i_\tau, j_\tau})\}_{\tau=1,2\cdots t-1}$ and the union of the set of available items until round t along with their corresponding features as $\mathbb{C}_{t-1}$.

Assumption 3. *The noise $\eta_{i,j,t}$ is conditionally sub-Gaussian [2][7] with zero mean and bounded variance, meaning that $\mathbb{E}[\eta_{i,j,t} \mid \mathbb{C}_{t-1}, \mathbb{H}_{t-1}] = 0$ and that there exists a $\sigma > 0$ such that for all $\gamma \in \mathbb{R}$, we have $\mathbb{E}[\exp(\gamma \eta_{i,j,t}) \mid \mathbb{H}_{t-1}, \mathbb{C}_{t-1}] \leq \exp(\frac{\gamma^2 \sigma^2}{2})$.*

This assumption implies that for all i and j, the conditional mean is given by $\mathbb{E}[r_{i,j} | \mathbb{C}_{t-1}, \mathbb{H}_{t-1}] = \mathbf{w}_i^{*T} \mathbf{x}_j$ and that the conditional variance satisfies $\mathbb{V}[r_{i,j} | \mathbb{C}_{t-1}, \mathbb{H}_{t-1}] \leq \sigma^2$.

In the GOB framework, we assume access to a (fixed) graph $G = (\mathcal{V}, \mathcal{E})$ of users in the form of a social network (or "trust graph"). Here, the nodes $\mathcal{V}$ correspond to users, whereas the edges $\mathcal{E}$ correspond to friendships or trust relationships. The homophily effect implies that the true user preferences vary smoothly across the graph, so we expect the preferences of users connected in the graph to be close to each other. Specifically,

Assumption 4. *The true user preferences vary smoothly according to the given graph, in the sense that we have a small value of*

$$\sum_{(i_1, i_2) \in \mathcal{E}} ||\mathbf{w}_{i_1}^* - \mathbf{w}_{i_2}^*||^2.$$

Hence, we assume that the graph acts as a correctly-specified prior on the users' true preferences. Note that this assumption implies that nodes in dense subgraphs will have a higher similarity than those in sparse subgraphs (since they will have a larger number of neighbours).

This assumption is violated in some datasets. For example, in our experiments we consider one dataset in which the available graph is imperfect, in that user preferences do not seem to vary smoothly across all graph edges. Intuitively, we might think that the GOB model might be harmful in this case (compared to ignoring the graph structure). However, in our experiments, we observe that even in these cases, the GOB approach still lead to results as good as ignoring the graph.

The GOB model [7] solves a contextual bandit problem for each user, where the mean vectors in the different problems are related according to the Laplacian L[1] of the graph G. Let $\mathbf{w}_{i,t}$ be the preference vector estimate for user i at round t. Let $\mathbf{w}_t$ and $\mathbf{w}^* \in \mathbb{R}^{dn}$ (respectively) be the concatenation of the vectors $\mathbf{w}_{i,t}$ and $\mathbf{w}_i^*$ across all users. The GOB model solves the following regression problem to find the mean preference vector estimate at round t,

$$\mathbf{w}_t = \underset{\mathbf{w}}{\operatorname{argmin}} \left[\sum_{i=1}^{n} \sum_{k \in \mathcal{M}_{i,t}} (\mathbf{w}_i^T \mathbf{x}_k - r_{i,k})^2 + \lambda \mathbf{w}^T (L \otimes I_d) \mathbf{w} \right], \tag{2}$$

where $\mathcal{M}_{i,t}$ is the set of items rated by user i up to round t. The first term is a data-fitting term and models the observed ratings. The second term is the Laplacian regularization and equal to $\sum_{(i,j) \in \mathcal{E}} \lambda ||\mathbf{w}_{i,t} - \mathbf{w}_{j,t}||_2^2$. This term models smoothness across the graph with $\lambda > 0$ giving the strength of this regularization. Note that the same objective function has also been explored for graph-regularized multi-task learning [13].

3.2 Connection to GMRFs

Unfortunately, the approach of Cesa-Bianchi [7] for solving (2) has a computational complexity of $O(d^2 n^2)$. To solve (2) more efficiently, we now show that it can be interpreted as performing MAP estimation in a GMRF. This will allow us to apply the GOB model to much larger datasets, and lead to an even more scalable algorithm based on Thompson sampling (Section 4).

Consider the following generative model for the ratings $r_{i,j}$ and the user preference vectors $\mathbf{w}_i$,

$$r_{i,j} \sim \mathcal{N}(\mathbf{w}_i^T \mathbf{x}_j, \sigma^2), \quad \mathbf{w} \sim \mathcal{N}(0, (\lambda L \otimes I_d)^{-1}).$$

This GMRF model assumes that the ratings $r_{i,j}$ are independent given $\mathbf{w}_i$ and $\mathbf{x}_j$, which is the standard

[1] To ensure invertibility, we set $L = L_G + I_n$ where L_G is the normalized graph Laplacian.

regression assumption. Under this independence assumption the first term in (2) is equal up the negative log-likelihood for all of the observed ratings $\mathbf{r}_t$ at time t, $\log p(\mathbf{r}_t \mid \mathbf{w}, \mathbf{x}_t, \sigma)$, up to an additive constant and assuming $\sigma = 1$. Similarly, the negative log-prior $p(\mathbf{w} \mid \lambda, L)$ in this model gives the second term in (2) (again, up to an additive constant that does not depend on $\mathbf{w}$). Thus, by Bayes rule minimizing (2) is equivalent to maximizing the posterior in this GMRF model.

To characterize the posterior, it is helpful to introduce the notation $\boldsymbol{\phi}_{i,j} \in \mathbb{R}^{dn}$ to represent the "global" feature vector corresponding to recommending item j to user i. In particular, let $\boldsymbol{\phi}_{i,j}$ be the concatenation of n d-dimensional vectors where the i^{th} vector is equal to $\mathbf{x}_j$ and the others are zero. The rows of the $t \times dn$ dimensional matrix Φ_t correspond to these "global" features for all the recommendations made until time t. Under this notation, the posterior $p(\mathbf{w} \mid \mathbf{r}_t, \mathbf{w}, \Phi_t)$ is given by a $\mathcal{N}(\hat{\mathbf{w}}_t, \Sigma_t^{-1})$ distribution with $\Sigma_t = \frac{1}{\sigma^2}\Phi_t^T\Phi_t + \lambda(L \otimes I_d)$ and $\hat{\mathbf{w}}_t = \frac{1}{\sigma^2}\Sigma_t^{-1}\mathbf{b}_t$ with $\mathbf{b}_t = \Phi_t^T\mathbf{r}_t$. We can view the approach in [7] as explicitly constructing the dense $dn \times dn$ matrix Σ_t^{-1}, leading to an $O(d^2n^2)$ memory requirement. A new recommendation at round t is thus equivalent to a rank-1 update to Σ_t, and even with the Sherman-Morrison formula this leads to an $O(d^2n^2)$ time requirement for each iteration.

3.3 Scalability

Rather than treating Σ_t as a general matrix, we propose to exploit its structure to scale up the GOB framework to problems where n is very large. In particular, solving (2) corresponds to finding the mean vector of the GMRF, which corresponds to solving the linear system $\Sigma_t\mathbf{w} = \mathbf{b}_t$. Since Σ_t is positive-definite, the linear system can be solved using conjugate gradient [16]. Conjugate gradient notably does not require Σ_t^{-1}, but instead uses matrix-vector products $\Sigma_t\mathbf{v} = (\Phi_t^T\Phi_t)\mathbf{v} + \lambda(L \otimes I_d)\mathbf{v}$ for vectors $\mathbf{v} \in \mathbb{R}^{dn}$. Note that $\Phi_t^T\Phi_t$ is block diagonal and has only $O(nd^2)$ non-zeroes. Hence, $\Phi_t^T\Phi_t\mathbf{v}$ can be computed in $O(nd^2)$ time. For computing $(L \otimes I_d)\mathbf{v}$, we use that $(B^T \otimes A)\mathbf{v} = \text{vec}(AVB)$, where V is an $n \times d$ matrix such that $\text{vec}(V) = \mathbf{v}$. This implies $(L \otimes I_d)\mathbf{v}$ can be written as VL^T which can be computed in $O(d \cdot \text{nnz}(L))$ time, where $\text{nnz}(L)$ is the number of non-zeroes in L. This approach thus has a memory requirement of $O(nd^2 + \text{nnz}(L))$ and a time complexity of $O(\kappa(nd^2 + d \cdot \text{nnz}(L)))$ per mean estimation. Here, κ is the number of conjugate gradient iterations which depends on the condition number of the matrix (we used warm-starting by the solution in the previous round for our experiments, which meant that $\kappa = 5$ was enough for convergence). Thus, the algorithm scales linearly in n and in the number of edges of the network (which tends to be linear in n due to the sparsity of social relationships). This enables us to scale to large networks, of the order of 50K nodes and millions of edges.

4 Alternative Bandit Algorithms

The above structure can be used to speed up the mean estimation for any algorithm in the GOB framework. However, the LINUCB-like algorithm in [7] needs to estimate the confidence intervals $\sqrt{\boldsymbol{\phi}_{i,j}^T\Sigma_t^{-1}\boldsymbol{\phi}_{i,j}}$ for each available item $j \in \mathcal{C}_t$. Using the GMRF connection, estimating these requires $O(|\mathcal{C}_t|\kappa(nd^2 + d \cdot \text{nnz}(L)))$ time since we need solve the linear system with $|\mathcal{C}_t|$ right-hand sides, one for each available item. But this becomes impractical when the number of available items in each round is large.

We propose two approaches for mitigating this: first, in this section we adapt the epoch-greedy [21] algorithm to the GOB framework. Epoch-greedy doesn't require confidence intervals and is thus very scalable, but unfortunately it doesn't achieve the optimal regret of $\tilde{O}(\sqrt{T})$. To achieve the optimal regret, we also propose a GOB variant of Thompson sampling [23]. In this section we further exploit the connection to GMRFs to scale Thompson sampling to even larger problems by using the recent sampling-by-perturbation trick [31]. This GMRF connection and scalability trick might be of independent interest for Thompson sampling in other large-scale problems.

4.1 Epoch-Greedy

Epoch-greedy [21] is a variant of the popular ϵ-greedy algorithm that explicitly differentiates between exploration and exploitation rounds. An "exploration" round consists of recommending a random item from $\mathcal{C}_t$ to the target user i_t. The feedback from these exploration rounds is used to learn $\mathbf{w}^*$. An "exploitation" round consists of choosing the available item $\hat{j}_t$ which maximizes the expected rating, $\hat{j}_t = \text{argmax}_{j \in \mathcal{C}_t} \hat{\mathbf{w}}_t^T\boldsymbol{\phi}_{i_t,j}$. Epoch-greedy proceeds in epochs, where each epoch q consists of 1 exploration round and s_q exploitation rounds.

Scalability: The time complexity for Epoch-Greedy is dominated by the exploitation rounds that require computing the mean and estimating the expected rating for all the available items. Given the mean vector, this estimation takes $O(d|\mathcal{C}_t|)$ time. The overall time complexity per exploitation round is thus $O(\kappa(nd^2 + d \cdot \text{nnz}(L)) + d|\mathcal{C}_t|)$.

Regret: We assume that we incur a maximum regret of 1 in an exploration round, whereas the regret incurred in an exploitation round depends on how well we have learned $\mathbf{w}^*$. The attainable regret is thus proportional to the generalization error for the class of hypothesis functions mapping the context vector to an expected rating [21]. In our case, the class of hypotheses is a set

of linear functions (one for each user) with Laplacian regularization. We characterize the generalization error in the GOB framework in terms of its Rademacher complexity [28], and use this to bound the expected regret leading to the result below. For ease of exposition in the regret bounds, we suppress the factors that don't depend on either n, L, λ or T. The complete bound is stated in the supplementary material (Appendix B).

Theorem 1. *Under the additional assumption that $||w_t||_2 \leq 1$ for all rounds t, the expected regret obtained by epoch-greedy in the GOB framework is given as:*

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

Proof Sketch. Let $\mathcal{H}$ be the class of valid hypotheses of linear functions coupled with Laplacian regularization. Let $Err(q, \mathcal{H})$ be the generalization error for $\mathcal{H}$ after obtaining q unbiased samples in the exploration rounds. We adapt Corollary 3.1 from [21] to our context:

Lemma 1. *If $s_q = \left\lfloor \frac{1}{Err(q,\mathcal{H})} \right\rfloor$ and Q_T is the smallest Q such that $Q + \sum_{q=1}^{Q} s_q \geq T$, the regret obtained by Epoch-Greedy can be bounded as $R(T) \leq 2Q_T$.*

We use [28] to bound the generalization error of our class of hypotheses in terms of its empirical Rademacher complexity $\hat{\mathcal{R}}_q^n(\mathcal{H})$. With probability $1-\delta$,

$$Err(q, \mathcal{H}) \leq \hat{\mathcal{R}}_q^n(\mathcal{H}) + \sqrt{\frac{9\ln(2/\delta)}{2q}}. \tag{3}$$

Using Theorem 2 in [28] and Theorem 12 from [5], we obtain

$$\hat{\mathcal{R}}_q^n(\mathcal{H}) \leq \frac{2}{\sqrt{q}}\sqrt{\frac{12Tr(L^{-1})}{\lambda}}. \tag{4}$$

Using (3) and (4) we obtain

$$Err(q, \mathcal{H}) \leq \frac{\left[2\sqrt{12Tr(L^{-1})/\lambda} + \sqrt{\frac{9\ln(2/\delta)}{2}}\right]}{\sqrt{q}}. \tag{5}$$

The theorem follows from (5) along with Lemma 1. □

The effect of the graph on this regret bound is reflected through the term $\text{Tr}(L^{-1})$. For a connected graph, we have the following upper-bound $\frac{\text{Tr}(L^{-1})}{n} \leq \frac{(1-1/n)}{\nu_2} + \frac{1}{n}$ [28]. Here, ν_2 is the second smallest eigenvalue of the Laplacian. The value ν_2 represents the algebraic connectivity of the graph [14]. For a more connected graph, ν_2 is higher, the value of $\frac{\text{Tr}(L^{-1})}{n}$ is lower, resulting in a smaller regret. Note that although this result leads to a sub-optimal dependence on T ($T^{\frac{2}{3}}$ instead of $T^{\frac{1}{2}}$), our experiments incorporate a small modification that gives similar performance to the more-expensive LINUCB.

4.2 Thompson sampling

A common alternative to LINUCB and Epoch-Greedy is Thompson sampling (TS). At each iteration TS uses a sample $\tilde{\mathbf{w}}_t$ from the posterior distribution at round t, $\tilde{\mathbf{w}}_t \sim \mathcal{N}(\mathbf{w}_t, \Sigma_t^{-1})$. It then selects the item j_t based on the obtained sample, $j_t = \text{argmax}_{j \in \mathcal{C}_t} \tilde{\mathbf{w}}_t^T \boldsymbol{\phi}_{i_t,j}$. We show below that the GMRF connection makes TS scalable, but unlike Epoch-Greedy it also achieves the optimal regret.

Scalability: The conventional approach for sampling from a multivariate Gaussian posterior involves forming the Cholesky factorization of the posterior covariance matrix. But in the GOB model the posterior covariance matrix is a dn-dimensional matrix where the fill-in from the Cholesky factorization can lead to a computational complexity of $O(d^2n^2)$. In order to implement Thompson sampling for large values of n, we adapt the recent sampling-by-perturbation approach [31] to our setting, and this allows us to sample from a Gaussian prior and then solve a linear system to sample from the posterior.

Let $\tilde{\mathbf{w}}_0$ be a sample from the prior distribution and let $\tilde{\mathbf{r}}_t$ be the perturbed (with standard normal noise) rating vector at round t, meaning that $\tilde{\mathbf{r}}_t = \mathbf{r}_t + \mathbf{y}_t$ for $\mathbf{y}_t \sim \mathcal{N}(0, I_t)$. In order to obtain a sample $\tilde{\mathbf{w}}_t$ from the posterior, we can solve the linear system

$$\Sigma_t \tilde{\mathbf{w}}_t = (L \otimes I_d)\tilde{\mathbf{w}}_0 + \Phi_t^T \tilde{\mathbf{r}}_t. \tag{6}$$

Let S be the Cholesky factor of L so that $L = SS^T$. Note that $L \otimes I_d = (S \otimes I_d)(S \otimes I_d)^T$. If $\mathbf{z} \sim \mathcal{N}(0, I_{dn})$, we can obtain a sample from the prior by solving $(S \otimes I_d)\tilde{\mathbf{w}}_0 = \mathbf{z}$. Since S tends to be sparse (using for example [11, 19]), this equation can be solved efficiently using conjugate gradient. We can pre-compute and store S and thus obtain a sample from the prior in time $O(d \cdot \text{nnz}(L))$. Using that $\Phi_t^T \tilde{\mathbf{r}}_t = \mathbf{b}_t + \Phi_t^T \mathbf{y}_t$ in (6) and simplifying we obtain

$$\Sigma_t \tilde{\mathbf{w}}_t = (L \otimes I_d)\tilde{\mathbf{w}}_0 + \mathbf{b}_t + \Phi_t^T \mathbf{y}_t \tag{7}$$

As before, this system can be solved efficiently using conjugate gradient. Note that solving (7) results in an exact sample from the dn-dimensional posterior. Computing $\Phi_t^T \mathbf{y}_t$ has a time complexity of $O(dt)$. Thus, this approach is faster than the original GOB framework whenever $t < dn^2$. Since we focus on the case of large graphs, this condition will tend to hold in our setting.

We now describe an alternative method of constructing the right side of (7) that doesn't depend on t. Observe

that computing $\Phi_t^T \mathbf{y}_t$ is equivalent to sampling from the distribution $\mathcal{N}(0, \Phi_t^T \Phi_t)$. To sample from this distribution, we maintain the Cholesky factor P_t of $\Phi_t^T \Phi_t$. Recall that the matrix $\Phi_t^T \Phi_t$ is block diagonal (one block for every user) for all rounds t. Hence, its Cholesky factor P_t also has a block diagonal structure and requires $\mathcal{O}(nd^2)$ storage. In each round, we make a recommendation to a single user and thus make a rank-1 update to only one $d \times d$ block of P_t. This is an order $\mathcal{O}(d^2)$ operation. Once we have an updated P_t, sampling from $\mathcal{N}(0, \Phi_t^T \Phi_t)$ and constructing the right side of (7) is an $O(nd^2)$ operation. The per-round computational complexity for our TS approach is thus $O(\min\{nd^2, dt\} + d \cdot nnz(L))$ for forming the right side in (7), $O(nd^2 + d \cdot \text{nnz}(L))$ for solving the linear system in (7) as well as for computing the mean, and $O(d \cdot |\mathcal{C}_t|)$ for selecting the item. Thus, our proposed approach has a complexity linear in the number of nodes and edges and can scale to large networks.

Regret: To analyze the regret with TS, observe that TS in the GOB framework is equivalent to solving a single dn-dimensional contextual bandit problem, but with a modified prior covariance equal to $(\lambda L \otimes I_d)^{-1}$ instead of I_{dn}. We obtain the result below by following a similar argument to Theorem 1 in [2]. The main challenge in the proof is to make use of the available graph to bound the variance of the arms. We first state the result and then sketch the main differences from the original proof.

Theorem 2. *Under the following additional technical assumptions: (a)* $\log(K) < (dn-1)\ln(2)$*, (b)* $\lambda < dn$*, and (c)* $\log\left(\frac{3+T/\lambda dn}{\delta}\right) \leq \log(KT)\log(T/\delta)$*, with probability* $1-\delta$*, the regret obtained by Thompson Sampling in the GOB framework is given as:*

$$R(T) = \tilde{O}\left(\frac{dn\sqrt{T}}{\sqrt{\lambda}}\sqrt{\log\left(\frac{3\,\mathrm{Tr}(L^{-1})}{n} + \frac{\mathrm{Tr}(L^{-1})T}{\lambda dn^2\sigma^2}\right)}\right)$$

Proof Sketch. To make the notation cleaner, for the round t and target user i_t under consideration, we use j to index the available items. Let the index of the optimal item at round t be j_t^* whereas the index of the item chosen by our algorithm is denoted j_t. Let $s_t(j)$ be the standard deviation in the estimated rating of item j at round t. It is given as $s_t(j) = \sqrt{\boldsymbol{\phi}_j^T \Sigma_{t-1}^{-1} \boldsymbol{\phi}_j}$. Further, let $l_t = \sqrt{dn\log\left(\frac{3+t/\lambda dn}{\delta}\right)} + \sqrt{3\lambda}$. Let $\mathcal{E}^{\mu}(t)$ be the event such that for all j,

$$\mathcal{E}^{\mu}(t): \ |\langle \mathbf{w}_t, \boldsymbol{\phi}_j\rangle - \langle \mathbf{w}^*, \boldsymbol{\phi}_j\rangle| \leq l_t s_t(j)$$

We prove that, for $\delta \in (0,1)$, $\Pr(\mathcal{E}^{\mu}(t)) \geq 1-\delta$. Define $g_t = \sqrt{4\log(tK)}\rho_t + l_t$, where $\rho_t = \sqrt{9d\log\left(\frac{t}{\delta}\right)}$. Let $\gamma = \frac{1}{4e\sqrt{\pi}}$. Given that the event $\mathcal{E}^{\mu}(t)$ holds with high probability, we follow an argument similar to Lemma 4 of [2] and obtain the following bound:

$$R(T) \leq \frac{3g_T}{\gamma}\sum_{t=1}^{T} s_t(j_t) + \frac{2g_T}{\gamma}\sum_{t=1}^{T}\frac{1}{t^2} + \frac{6g_T}{\gamma}\sqrt{2T\ln 2/\delta} \tag{8}$$

To bound the variance of the selected items, $\sum_{t=1}^{T} s_t(j_t)$, we extend the analysis in [10, 35] to include the prior covariance term. We thus obtain the following inequality:

$$\sum_{t=1}^{T} s_t(j_t) \leq \sqrt{dnT} \times \sqrt{C\log\left(\frac{\mathrm{Tr}(L^{-1})}{n}\right) + \log\left(3 + \frac{T}{\lambda dn\sigma^2}\right)} \tag{9}$$

where $C = \frac{1}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}$. Substituting this into (8) completes the proof. □

Note that since n is large in our case, assumption (a) for the above theorem is reasonable. Assumptions (b) and (c) define the upper and lower bounds on the regularization parameter λ. Similar to epoch-greedy, transferring information across the graph reduces the regret by a factor dependent on $\mathrm{Tr}(L^{-1})$. Note that compared to epoch-greedy, the regret bound for Thompson sampling has a worse dependence on n, but its $\tilde{O}(\sqrt{T})$ dependence on T is optimal. If $L = I_{dn}$, we match the $\tilde{O}(dn\sqrt{T})$ regret bound for a dn-dimensional contextual bandit problem [1]. Note that we have a dependence on d and n similar to the original GOB paper [7] and that this method performs similarly in practice in terms of regret. However, as will see, our algorithm is much faster.

5 Experiments

5.1 Experimental Setup

Data: We first test the scalability of various algorithms using synthetic data and then evaluate their regret performance on two real datasets. For synthetic data we generate random d-dimensional context vectors and ground-truth user preferences, and generate the ratings according to the linear model. We generated a random Kronecker graph with sparsity 0.005 (which is approximately equal to the sparsity of our real datasets). It is well known that such graphs capture many properties of real-world social networks [22].

For the real data, we use the Last.fm and Delicious datasets which are available as part of the HetRec 2011 workshop. Last.fm is a music streaming website where each item corresponds to a music artist and the dataset consists of the set of artists each user has listened to. The associated social network consists of 1.8K users (nodes) and 12.7K friendship relations (edges). Delicious is a social bookmarking website, where an item corresponds to a particular URL and the dataset consists of the set of websites bookmarked by each user. Its corresponding social network consists of 1.8K users and 7.6K user-user relations. Similar to [7], we use the set of associated tags to construct the TF-IDF vector for each item and reduce the dimension of these vectors to $d = 25$. An artist (or URL) that a user has listened to (or has bookmarked) is said to be "liked" by the user. In each round, we select a target user uniformly at random and make the set $\mathcal{C}_t$ consist of 25 randomly chosen items such that there is at least 1 item liked by the target user. An item liked by the target user is assigned a reward of 1 whereas other items are assigned a zero reward. We use a total of $T = 50$ thousand recommendation rounds and average our results across 3 runs.

Algorithms: We denote our graph-based epoch-greedy and Thompson sampling algorithms as G-EG and G-TS, respectively. For epoch-greedy, although the theory suggests that we update the preference estimates only in the exploration rounds, we observed better performance by updating the preference vectors in all rounds (we use this variant in our experiments). We use 10% of the total number of rounds for exploration, and we "exploit" in the remaining rounds. Similar to [15], all hyper-parameters are set using an initial validation set of 5 thousand rounds. The best validation performance was observed for $\lambda = 0.01$ and $\sigma = 1$. To control the amount of exploration for Thompson sampling, we the use posterior reshaping trick [8] which reduces the variance of the posterior by a factor of 0.01.

Baselines: We consider two variants of graph-based UCB-style algorithms: GOBLIN is the method proposed in the original GOB paper [7] while we use GOBLIN++ to refer to a variant that exploits the fast mean estimation strategy we develop in Section 3.3. Similar to [7], for both variants we discount the confidence bound term by a factor of $\alpha = 0.01$.

We also include baselines which ignore the graph structure and make recommendations by solving independent linear contextual bandit problems for each user. We consider 3 variants of this baseline: the LINUCB-IND proposed in [23], an epoch-greedy variant of this approach (EG-IND), and a Thompson sampling variant (TS-IND). We also compared to a baseline that does no personalization and simply considers a single bandit problem across all users (LINUCB-SIN). Finally, we compared against the state-of-the-art online clustering-based approach proposed in [15], denoted CLUB. This method starts with a fully connected graph and iteratively deletes edges from the graph based on UCB estimates. CLUB considers each connected component of this graph as a cluster and maintains one preference vector for all the users belonging to a cluster. Following the original work, we make CLUB scalable by generating a random Erdos-Renyi graph $G_{n,p}$ with $p = \frac{3logn}{n}$.[2] In all, we compare our proposed algorithms G-EG and G-TS with 7 reasonable baseline methods.

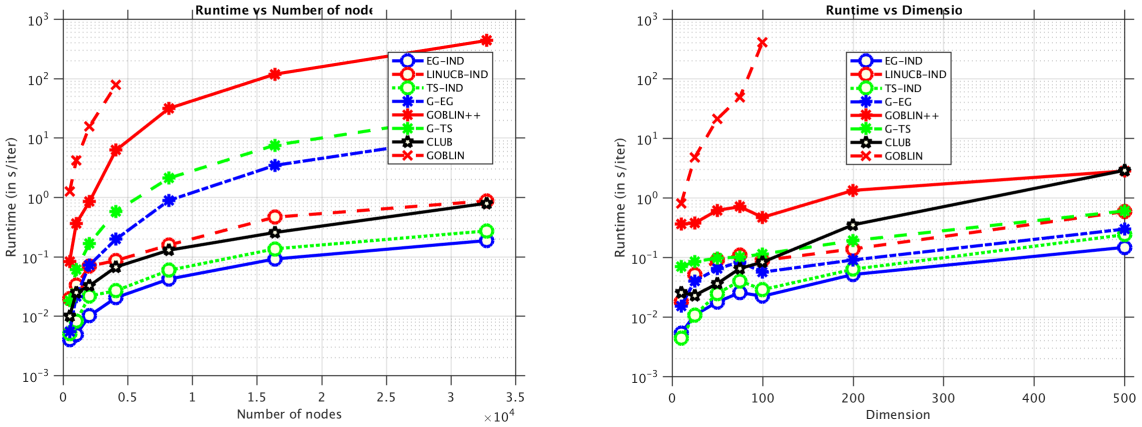
5.2 Results

Scalability: We first evaluate the scalability of the various algorithms with respect to the number of network nodes n. Figure 1(a) shows the runtime in seconds/iteration when we fix $d = 25$ and vary the size of the network from 16 thousand to 33 thousand nodes. Compared to GOBLIN, our proposed GOBLIN++ is more efficient in terms of both time (almost 2 orders of magnitude faster) and memory. Indeed, the existing GOBLIN method runs out of memory even on very small networks and thus we do not plot it for larger networks. Further, our proposed G-EG and G-TS methods scale even more gracefully in the number of nodes and are much faster than GOBLIN++ (although not as fast as the clustering-based CLUB or methods that ignore the graph).

We next consider scalability with respect to d. Figure 1(b) fixes $n = 1024$ and varies d from 10 to 500. In this figure it is again clear that our proposed GOBLIN++ scales much better than the original GOBLIN algorithm. The EG and TS variants are again even faster, and other key findings from this experiment are (i) it was not faster to ignore the graph and (ii) our proposed G-EG and G-TS methods scale better with d than CLUB.

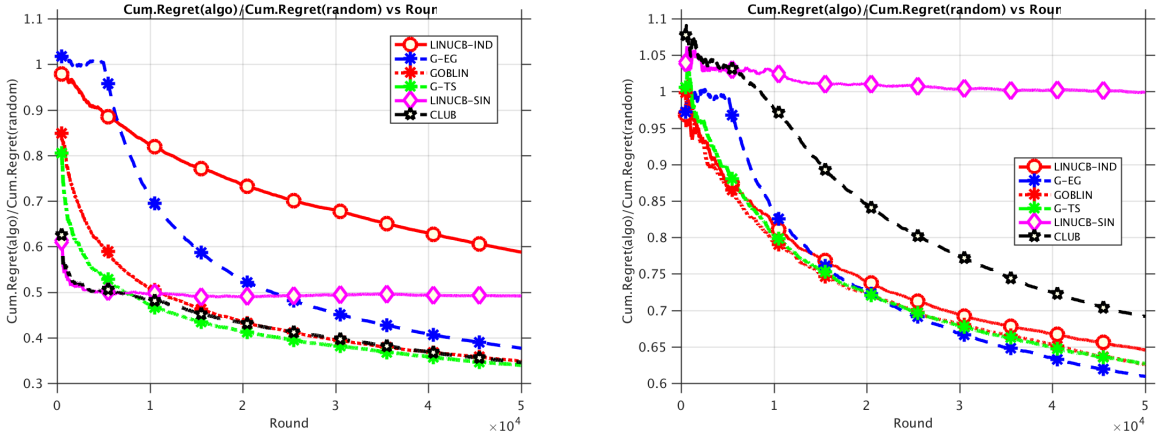
Regret Minimization: We follow [15] in evaluating recommendation performance by plotting the ratio of cumulative regret incurred by the algorithm divided by the regret incurred by a random selection policy. Figure 2(a) plots this measure for the Last.fm dataset. In this dataset we see that treating the users independently (LINUCB-IND) takes a long time to drive down the regret (we do not plot EG-IND and TS-IND as they had similar performance) while simply aggregating across users (LINUCB-SIN) performs well initially (but eventually stops making progress). We see that the approaches exploiting the graph help learn the user

[2]We reimplemented CLUB. Note that one of the datasets from our experiments was also used in that work and we obtain similar performance to that reported in the original paper.



(a) (b)

Figure 1: Synthetic network: Runtime (in seconds/iteration) vs (a) Number of nodes (b) Dimension



(a) Last.fm (b) Delicious

Figure 2: Regret Minimization

preferences faster than the independent approach and we note that on this dataset our proposed G-TS method performed similar to or slightly better than the state of the art CLUB algorithm.

Figure 2(b) shows performance on the Delicious dataset. On this dataset personalization is more important and we see that the independent method (LINUCB-IND) outperforms the non-personalized (LINUCB-SIN) approach. The need for personalization in this dataset also leads to worse performance of the clustering-based CLUB method, which is outperformed by all methods that model individual users. On this dataset the advantage of using the graph is less clear, as the graph-based methods perform similar to the independent method. Thus, these two experiments suggest that (i) the scalable graph-based methods do no worse than ignoring the graph in cases where the graph is not helpful and (ii) the scalable graph-based methods can do significantly better on datasets where the graph is helpful. Similarly, when user preferences naturally form clusters our proposed methods perform similarly to CLUB, whereas on datasets where individual preferences are important our methods are significantly better.

6 Discussion

This work draws a connection between the GOB framework and GMRFs, and uses this to scale up the existing GOB model to much larger graphs. We also proposed and analyzed Thompson sampling and epoch-greedy variants. Our experiments on recommender systems datasets indicate that the Thompson sampling approach in particular is much more scalable than existing GOB methods, obtains theoretically optimal regret, and performs similar to or better than other existing scalable approaches.

In many practical scenarios we do not have an explicit graph structure available. In the supplementary material we consider a variant of the GOB model where we use L1-regularization to learn the graph on the fly. Our experiments there show that this approach works similarly to or much better than approaches which use the fixed graph structure. It would be interesting to explore the theoretical properties of this approach.

References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[2] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. *arXiv preprint arXiv:1209.3352*, 2012.

[3] Noga Alon, Nicolo Cesa-Bianchi, Claudio Gentile, Shie Mannor, Yishay Mansour, and Ohad Shamir. Nonstochastic multi-armed bandits with graph-structured feedback. *arXiv preprint arXiv:1409.8428*, 2014.

[4] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[5] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, 3:463–482, 2003.

[6] Stéphane Caron, Branislav Kveton, Marc Lelarge, and Smriti Bhagat. Leveraging side observations in stochastic bandits. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, 2012.

[7] Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. In *Advances in Neural Information Processing Systems*, pages 737–745, 2013.

[8] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011.

[9] Wei Chu, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

[10] Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, pages 355–366, 2008.

[11] Timothy A Davis. Algorithm 849: A concise sparse cholesky factorization package. *ACM Transactions on Mathematical Software (TOMS)*, 31(4):587–591, 2005.

[12] Julien Delporte, Alexandros Karatzoglou, Tomasz Matuszczyk, and Stéphane Canu. Socially enabled preference learning from implicit feedback data. In *Machine Learning and Knowledge Discovery in Databases*, pages 145–160. Springer, 2013.

[13] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM, 2004.

[14] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[15] Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 757–765, 2014.

[16] Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. 1952.

[17] Tomáš Kocák, Michal Valko, Rémi Munos, and Shipra Agrawal. Spectral thompson sampling. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[18] Nathan Korda, Balázs Szörényi, and Shuai Li. Distributed clustering of linear bandits in peer to peer networks. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1301–1309, 2016.

[19] Rasmus Kyng and Sushant Sachdeva. Approximate gaussian elimination for laplacians-fast, sparse, and simple. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 573–582. IEEE, 2016.

[20] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[21] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pages 817–824, 2008.

[22] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker graphs: An approach to modeling networks. *The Journal of Machine Learning Research*, 11:985–1042, 2010.

[23] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[24] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, pages 539–548, 2016.

[25] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[26] Odalric-Ambrym Maillard and Shie Mannor. Latent bandits. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 136–144, 2014.

[27] Shie Mannor and Ohad Shamir. From bandits to experts: On the value of side-observations. In *Advances in Neural Information Processing Systems*, pages 684–692, 2011.

[28] Andreas Maurer. The rademacher complexity of linear transformation classes. In *Learning Theory*, pages 65–78. Springer, 2006.

[29] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

[30] Trong T Nguyen and Hady W Lauw. Dynamic clustering of contextual multi-armed bandits. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1959–1962. ACM, 2014.

[31] George Papandreou and Alan L Yuille. Gaussian sampling by local perturbations. In *Advances in Neural Information Processing Systems*, pages 1858–1866, 2010.

[32] Nikhil Rao, Hsiang-Fu Yu, Pradeep K Ravikumar, and Inderjit S Dhillon. Collaborative filtering with graph information: Consistency and scalable methods. In *Advances in Neural Information Processing Systems*, pages 2098–2106, 2015.

[33] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

[34] Michal Valko, Rémi Munos, Branislav Kveton, and Tomáš Kocák. Spectral bandits for smooth graph functions. In *31th International Conference on Machine Learning*, 2014.

[35] Zheng Wen, Branislav Kveton, and Azin Ashkan. Efficient learning in large-scale combinatorial semi-bandits. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1113–1122, 2015.