Bayesian Collaborative Bandits for Time Slot Inference in Maternal Health Programs

Arpan Dasgupta [1] **Arun Suggala** [1] **Karthikeyan Shanmugam** [1] **Milind Tambe** [1 2] **Aparna Taneja** [1]

Abstract

Mobile health programs have gained a lot of popularity recently due to the widespread use of mobile phones, particularly in underserved communities. However, call records from one such maternal mHealth program in India indicate that different beneficiaries have different time preferences, due to their availability during the day as well as limited access to a phone. This makes selection of the best time slot to call a beneficiary an important problem for the program. Prior work has formalized this as a collaborative bandit problem, where the assumption of a low-rank call pickup matrix allows for more efficient exploration across arms. We propose a novel Bayesian solution to the collaborative bandit problem using Stochastic Gradient Langevin Dynamics (SGLD) and Thompson Sampling for selection of time slots. We show that this method is able to perform better in scarce data situations where there are limited time steps for exploration, and has the ability to utilize prior knowledge about arms to its advantage. We also propose a faster version of the algorithm using alternative sampling which can potentially scale to a very large number of users such that it may be potentially deployable in the real world. We evaluate the algorithm against existing methods on simulated data inspired from real-world data.

1. Introduction

Maternal health is an important area of concern and for a lot of developing countries reducing the maternal mortality rate is one of WHOs Sustainable Development Goals (SDGs) (WHO). It is critical to provide timely and essential information to pregnant women to address this. Mobile

[1]Google DeepMind, Bengaluru, India [2]Harvard University, Boston, USA. Correspondence to: Arpan Dasgupta <arpandg@google.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Figure 1: A beneficiary of the Kilkari program

health (mHealth) programs run by non-profit organizations utilize the large scale availability of mobile phones to provide such critical information via automated voice calls on mobile phones (kil). ARMMAN (arm) is one such NGO based in India whose mission is to improve maternal and child health outcomes for underserved communities in India. ARMMAN partners with the Ministry of Health and Family Welfare in India to run the Kilkari program (kil) which provides this information to the mothers through automated voice calls sent at different times throughout the week. However, one challenge faced by the NGO is the low pick-up rate of the beneficiaries when called, requiring multiple retries (more than 6 on average). This is largely due to the fact that different beneficiaries prefer to listen to these calls at different time slots and on different days due to practical constraints such as shared phones, different working hours, household chores, and network reliability (Mohan et al., 2021; Lalan et al., 2023). Thus, carefully deciding on the time slots to call a beneficiary improves the probability of pickup of a call and hence the effectiveness of the program by reducing critical bandwidth currently spent on retries. The NGO combats low pickup rates by calling beneficiaries multiple times, and hence a more targeted approach should also help meet the bandwidth constraints of the NGO. This becomes even more crucial as the goal of the NGO is to scale operations to a nation-wide level eventually.

Finding the optimal time slots for each beneficiary can be done by formulating the pick-up problem in terms of a multi-agent multi-arm bandit problem by considering users as agents and time slots as arms. A simple multi-arm bandit solution requires exploration over the time slots for each beneficiary. (Pal et al., 2024) utilizes collaborative bandits to jointly learn preferences for each user assuming that the time slot preference matrix is low-rank and uses matrix fac-

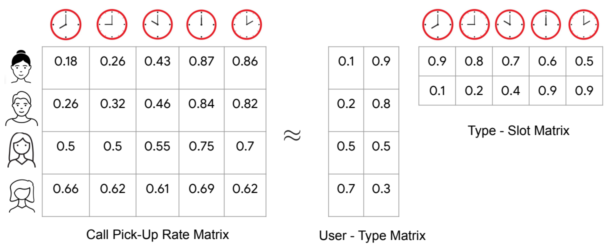

Figure 2: Illustration showing the pick-up matrix for 4 users and 5 time slots, which can be decomposed into 2 matrices assuming 2 implicit user types. Each user is some combination of the two types, and each user type has some preference across time slots.

torization to obtain an estimate for the preferences. Further, Boltzmann exploration (Cesa-Bianchi et al., 2017) is used to ensure that these preferences converge to optimal over time. The low-rank assumption is valid due to the fact that groups of users show similar preferences. Figure 2 shows a simple example of the low-rank pick-up problem. However, there were a few fundamental problems with the method used to tackle the problem. Since the method uses matrix completion, a large number of samples are required for a new arm to be effectively optimized. Also, in the presence of prior information, the number of samples can be significantly reduced, which this method has no provision to leverage.

We introduce a Bayesian formulation of the problem which alleviates the previously mentioned issues. Bayesian approaches (Bharadiya, 2023) allow the use of information present in priors to quickly provide an estimate even in unseen scenarios such as newly introduced arms, as well as scenarios where there is very little data. The latter is especially important due to the fact that in most practical settings, such as the maternal health application, there is very limited time to explore and then exploit (and thus a finite horizon). This is because if we do not act soon enough we risk beneficiaries getting dropped off from the program. Bayesian methods also allows the use of Thompson Sampling (Thompson, 1933) which is shown (Nakajima & Sugiyama, 2011) to be a very effective method and can provide much tighter bounds than the previously attempted Boltzmann exploration.

We show that in low-data settings, the Bayesian method outperforms (Pal et al., 2024), despite completely uninformative priors. We also propose a scalable solution with alternative sampling of user and time slot matrices. Gains are measured in terms of cumulative regret over not picking the best arm for each beneficiary as well as reduction in the number of required calls. We intend to evaluate this work in a real world field study in collaboration with the non-profit ARMMAN, for potential deployment in Kilkari.

2. Related Work

AI in Maternal Healthcare Limited resource allocation problems in maternal healthcare have previously been solved by restless multi-arm bandits (Mate et al., 2022; Nair et al., 2022; Verma et al., 2023). The time slot selection problem using collaborative bandits was previously studied by (Pal et al., 2024).

Multi-armed Bandits Multi-armed bandits are a highly studied and effective method for solving several resource allocation problems. Several methods such as phased elimination (Lattimore & Szepesvári, 2020; Slivkins et al., 2019), UCB (Auer et al., 2002), Thompson Sampling (Thompson, 1933; Agrawal & Goyal, 2012) and Best-arm Identification (Agrawal et al., 2020; Garivier & Kaufmann, 2016) have been studied in detail.

Collaborative Bandits Collaborative Bandits have garnered recent attention due to the widespread popularity of modern recommender systems (Bresler et al., 2016; Dadkhahi & Negahban, 2018). While several algorithms with strong bounds under special conditions have been proposed (Pal et al., 2023; Jain & Pal, 2022), (Pal et al., 2024) proposed an algorithm which works in approximate low-rank problems which is closer to our setting.

Bayesian Matrix Factorization While several methods have been proposed for Bayesian matrix factorization (Nakajima & Sugiyama, 2011), the ones which utilize MCMC (Salakhutdinov & Mnih, 2008) are of particular interest to us. (Ahn et al., 2015) proposes utilizing SGLD in a distributed manner using block partitioning to perform matrix factorization. While such methods have been applied in practice (Zhang et al., 2020; Li et al., 2016), none of these methods extend the solution in a bandit setting that forms our focus.

3. Problem Formulation

The problem of pickup in Kilkari can be characterized by the use of a pickup matrix $M \in R^{u,t}$ where element $m_{i,j}$ represents the probability of user i picking up the call when they are called at time slot j. u and t represent the number of users and time slots respectively.

The pickup matrix can be assumed to be low-rank because of similarity of pickup patterns across beneficiaries (Pal et al., 2024; Bresler et al., 2016). Our low rank models assumes that each of the users belongs to a simplex of k archetypes, while each archetype has a pattern of pickup across the different time slots (e.g. those with a shared phone may prefer morning and evening slots). Thus, we want to decompose matrix M into $U \in \mathrm{R}^{u,k}$ and $R \in \mathrm{R}^{k,t}$, $M = UR$, where U of dimension $u \times k$ and R of dimension $k \times t$ represent user to type matching and type to reward matching respectively. Each row of U is a valid probability mass function over the support $[k]$. Each entry of R is a scalar probability of pickup in $[0, 1]$.

The input data is provided as X, where X_i refers to the

i_{th} data point. In general $X_i = (u_i, t_i, p_i) \in [1, \cdots u] \times [1, \cdots t] \times \{0, 1\}$, which represents if the call was picked up or not (p_i) by user u_i at time slot t_j. N is the number of data points in total and n is the batch size.

3.1. Matrix Completion with Boltzmann Exploration

Prior work (Pal et al., 2024) utilizes matrix factorization (MF) using a nuclear norm optimization method. Given the data X, a matrix D is created with the estimated probabilities from the observed data. MF is run on D to get a completed matrix M'.

$$min_{M'} \sum_{\{i,j\} \in D} \left(D_{i,j} - M'_{i,j} \right) + \lambda ||M'||_*$$

M' estimates the original matrix M, following which, exploration is done by using Boltzmann exploration (Cesa-Bianchi et al., 2017). While this method performs well in comparison to UCB (Auer et al., 2002) on each individual arm, there is much scope for improvement, such as utilization of priors, and performance in scarce data scenarios.

4. Bayesian Collaborative Bandits

4.1. Bayesian Matrix Factorization

A Bayesian method on the contrary can perform well given prior information, or in low data settings. We aim to model the matrix factorization as a bayesian optimization problem similar to (Ahn et al., 2015). Since we need a way to compute the posterior distribution given the data, we use Stochastic Gradient Langevin Dynamic (SGLD) (Welling & Teh, 2011) which allows us to compute updates to the parameters in batches of data as computing parameter updates with the entire dataset is very compute intensive. Another advantage of using SGLD is that we can directly sample from the posterior using SGLD, which is later used along with Thompson sampling as a method to pick time slots to call for a given user. The general SGLD update looks like:

$$\Delta\Theta = \frac{\epsilon}{2}(\nabla \log p(\Theta) + \frac{N}{n}\Sigma_{i=1}^{n} \nabla \log p(X_i|\Theta)) + \eta \qquad \eta \sim N(0, \epsilon) \tag{1}$$

where, Θ are the parameters that characterize the likelihood function of X (observed data). $p(\Theta)$ is the Bayesian prior over these parameters. In our case, we assume that element $U_{u,k'}$ is filled with values $\frac{e^{\theta_{u,k'}}}{\sum_k (e^{\theta_{u,k}})}$ and each element $R_{k,t}$ is filled with $\frac{1}{1+e^{r_{k,t}}}$. Here, $\theta_{u,k}$ and $r_{k,t}$ are the parameters are sampled from independent priors each of which is an exponential distribution with pre-decided parameters $\lambda_{u,k}$ and $\alpha_{k,t}$ respectively. Note that, $\Theta = \{\theta_{u,k}\} \cup \{r_{k,t}\}$.

In our setting the prior $p(\Theta)$ from Equation (1) is:

$$p(\Theta) = p(\theta, r) = \left(\Pi_{u,k}\lambda_{u,k} e^{\lambda_{u,k}\theta_{u,k}}\right) \left(\Pi_{k,t}\alpha_{k,t} e^{\alpha_{k,t} r_{k,t}}\right) \tag{2}$$

Coordinates of the gradient $\nabla_\Theta \log p(\Theta)$ are given by:

$$\nabla_{\theta_{u,k}} \log p(\theta, r) = \lambda_{u,k}, \ \nabla_{r_{k,t}} \log p(\theta, r) = \alpha_{k,t} \tag{3}$$

The likelihood (second) term of Equation (1) involves the observed reward $x_{u,t}$ which is a mixture of Bernoulli random variables and is calculated as

$$\begin{aligned} P = p(x_{u,t}|\theta, r) &= \Sigma_k \left(p(k|\theta_u) p(x_{u,t}|k, r_t)\right) \\ p(k|\theta) &= \frac{e^{\theta_{uk}}}{\Sigma_j e^{\theta_{uj}}} \\ p(x_{u,t}|k, r_t) &= \frac{x_{u,t} e^{r_{k,t}}}{(1 + e^{r_{k,t}})} + \frac{(1 - x_{u,t})}{(1 + e^{r_{k,t}})} \end{aligned} \tag{4}$$

Here, $p(k|\theta)$ and $p(x_{u,t}|k, r_t)$ represent the probability of sampling archetype k given the user parameters and the Bernoulli likelihood of observing a Boolean variable $x_{u,t}$ given the archetype k and the reward parameters. Coordinates of $\nabla_\Theta \log p(x_{u,t}|\Theta)$ are given by:

$$\begin{aligned} &\nabla_{\theta_{u,k}} \log p(x_{u,t}|\theta, r) \\ &\quad = \frac{1}{P}(\delta_{\theta_{u,k}}(U_{u,k})) \cdot p(x_{u,t}|k, r_t) \end{aligned} \tag{5}$$

$$\nabla_{r_{k,t}} \log p(x_{u,t}|\theta, r) = \frac{1}{P} \frac{e^{\theta u,k}}{\Sigma_j e^{\theta_{u,j}}} \frac{(2x_{u,t} - 1)e^{r_{k,t}}}{(1 + e^{r_{k,t}})^2} \tag{6}$$

4.2. Thompson Sampling

While there are several solution paradigms for regret minimization in multi-arm bandit problems such as UCB, Phased Elimination, Thompson Sampling (TS) (Thompson, 1933; Nakajima & Sugiyama, 2011) is a popular method known for its simplicity and excellent empirical performance in several applications. Further, being applicable in Bayesian settings, TS is a popular method when prior information is available. TS works by choosing a sample of reward parameters from the posterior distribution, pulling the best arm and then evolving the belief according to the new data received. The most difficult part of the process is to sample from the posterior distribution in complex scenarios. In the limit, Markov Chain Monte Carlo (MCMC) methods (Andrieu & Thoms, 2008) such as SGLD converge to the posterior distribution, which allows us to directly sample from the converged SGLD matrices. Once we sample a row of the reward matrix $M_{u,:}$ in our context, we choose the best time slot for each user.

4.3. Scaling with Alternating Sampling

While we see that this works well in the experiments Figure 3, we realize that SGLD does not scale very well in practice to cases where the number of users is very large. This is not desirable as Kilkari program should potentially scale to millions of beneficiaries. We thus come up with a scalable formulation of the algorithm which can be further optimized via a distributed computation.

Number of examples	Bayesian MF		Data Matrix		Matrix Completion	
	Error	Error Fraction	Error	Error Fraction	Error	Error Fraction
500	**14.7**	**0.27**	52.807	0.991	49.127	0.922
5000	**12.69**	**0.215**	47.72	0.812	30.71	0.522
40000	**7.24**	**0.13**	27.61	0.51	23.82	0.441
400000	5.67	0.12	7.77	0.16	**5.05**	**0.1**

Table 1: Comparison of error of Bayesian matrix factorization on a 500×20 matrix with 4 types, compared to (1) The data matrix D as described in Section 3.1 (2) Completed matrix using matrix completion M'. The error E is measured as Frobenius norm of the difference with the original matrix M, and the fractional error is the error in ratio of the Frobenius norm of M, ie. $\frac{E}{||M||_F}$

4.3.1. Conditional Independence of Updates

We want to establish that updates of the conditional posterior of the parameters of the i_{th} user is independent of the parameters of the other users and the reward parameter r given data X. Since the update Equation (1) is a decomposition of the term $P(\theta|X)$, it is sufficient to show the following.

Theorem 4.1. *The updates in parameters for one user are independent from the other users.*

$$P(\theta_u|r, \theta_{u' \neq u}, X) = P(\theta_u|r, X) \tag{7}$$

The proof is provided in Appendix A

4.3.2. Parallelization Accross Batches of Users

Theorem 4.1 enables us to perform updates in θ in blocks of users, accumulate updated parameters and then perform the updates on r. This type of alternation is different from (Ahn et al., 2015) as it does not involve a complicated breakdown and recombination of the original matrix into blocks. The complete algorithms are given in Appendix B.

5. Experiments

We conduct experiments comparing raw matrix factorization, regrets on the bandit problem and in terms of the required number of retries (in Appendix C.2).

5.1. Matrix Factorization

Appendix C.1 describes the process of data generation and metrics for comparison in detail. Table 1 shows that Bayesian MF performs the best in low data setting and continues to do so until a very large number of samples are available, beyond which non-Bayesian MF is able to perform similarly. It must be noted that the priors provided to the method are mostly uninformative as they are set as 0.5 for each value indicating that the values are from a probability distribution. In the presence of informative priors, Bayesian factorization should require even lesser examples to reach the same loss.

5.2. Bandit Regrets

We show the cumulative regret on up to 1000 users on a randomly generated pickup matrix in Figure 3. Regret is calculated per batch of 1000 users and is calculated as the mean difference between the best arm probability and the probability of pickup in the chosen arm. We compare against non Bayesian matrix completion (Pal et al., 2024) and UCB for each arm (Auer et al., 2002). UCB performs worse than all the methods because it cannot leverage the collaborative structure of the problem. Our proposed bayesian method does very well and is able to start minimizing regret very quickly. The alternating sampling version, performs slightly worse than the direct sampling method, but is still able to outperform other methods by about 80% in terms of regret. We hypothesize that this is because joint sampling of U and R probably leads to a direct convergence to the posterior and the alternating sampling leads to a noisy path to convergence.

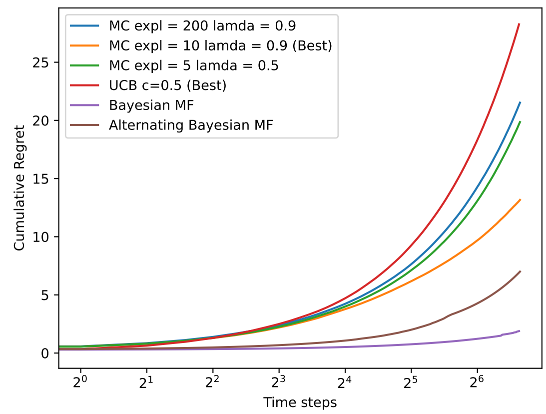

Figure 3: Cumulative regret on logarithmic scale for the different methods averaged over 5 random matrices. Results are on a 1000×20 matrix with 4 types. Every time step adds 1000 samples.

6. Conclusion and Future Directions

In this work, we show a significant improvement over the SOTA in terms of cumulative regret and pickup rates. We also propose an alternating method to scale better. We have shown that prior information can be incorporated in the problem, we intent to formalize this further in future work. We also aim to come up with theoretical regret bounds for Bayesian matrix factorization as well as convergence bounds for the alternative sampling method.

Impact Statement

The results shown in the paper are obtained in simulation. This will however be followed by a thorough evaluation through a field study, as for past works (Verma et al., 2023), to evaluate its effectiveness in the real world before deployment. Simulations show potential for saving a large number of calls compared to the current methodology used by the NGO, which means more effective utilization of bandwidth as well as potential to potentially enroll more beneficiaries. We rely on the expertise of our partner NGO to achieve these goals, and would like to acknowledge and thank our collaborators at ARMMAN.

References

SDG Target 3.1 Maternal mortality — who.int. https://www.who.int/data/gho/data/themes/topics/sdg-target-3-1-maternal-mortality. [Accessed 31-05-2024].

Armman Home - ARMMAN - Helping Mothers and Children — armman.org. https://armman.org. [Accessed 31-05-2024].

Kilkari - ARMMAN - Helping Mothers and Children — armman.org. https://armman.org/kilkari/. [Accessed 31-05-2024].

Agrawal, S. and Goyal, N. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pp. 39–1. JMLR Workshop and Conference Proceedings, 2012.

Agrawal, S., Juneja, S., and Glynn, P. Optimal $\delta - correctbest - armselectionforheavy - taileddistributions. InAlgorithmic$ *Learning Theory*, pp. 61– $-110. PMLR$, 2020.

Ahn, S., Korattikara, A., Liu, N., Rajan, S., and Welling, M. Large-scale distributed bayesian matrix factorization using stochastic gradient mcmc. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 9–18, 2015.

Andrieu, C. and Thoms, J. A tutorial on adaptive mcmc. *Statistics and computing*, 18:343–373, 2008.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256, 2002.

Bharadiya, J. P. A review of bayesian machine learning principles, methods, and applications. *International Journal of Innovative Science and Research Technology*, 8(5):2033–2038, 2023.

Bresler, G., Shah, D., and Voloch, L. F. Collaborative filtering with low regret. In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*, pp. 207–220, 2016.

Cesa-Bianchi, N., Gentile, C., Lugosi, G., and Neu, G. Boltzmann exploration done right. *Advances in neural information processing systems*, 30, 2017.

Dadkhahi, H. and Negahban, S. Alternating linear bandits for online matrix-factorization recommendation. *arXiv preprint arXiv:1810.09401*, 2018.

Garivier, A. and Kaufmann, E. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, pp. 998–1027. PMLR, 2016.

Jain, P. and Pal, S. Online low rank matrix completion. *arXiv preprint arXiv:2209.03997*, 2022.

Lalan, A., Verma, S., Sudan, K. M., Mahale, A., Hegde, A., Tambe, M., and Taneja, A. Analyzing and predicting low-listenership trends in a large-scale mobile health program: A preliminary investigation. *arXiv preprint arXiv:2311.07139*, 2023.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Lattimore, T. and Szepesvári, C. *Bandit algorithms*. Cambridge University Press, 2020.

Li, G., Chen, Q., et al. Exploiting explicit and implicit feedback for personalized ranking. *Mathematical Problems in Engineering*, 2016, 2016.

Mate, A., Madaan, L., Taneja, A., Madhiwalla, N., Verma, S., Singh, G., Hegde, A., Varakantham, P., and Tambe, M. Field study in deploying restless multi-armed bandits: Assisting non-profits in improving maternal and child health. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 12017–12025, 2022.

Mohan, D., Scott, K., Shah, N., Bashingwa, J. J. H., Chakraborty, A., Ummer, O., Godfrey, A., Dutt, P., Chamberlain, S., and LeFevre, A. E. Can health information through mobile phones close the divide in health behaviours among the marginalised? an equity analysis of kilkari in madhya pradesh, india. *BMJ Global Health*, 6(Suppl 5):e005512, 2021.

Nair, V., Prakash, K., Wilbur, M., Taneja, A., Namblard, C., Adeyemo, O., Dubey, A., Adereni, A., Tambe, M., and Mukhopadhyay, A. Adviser: Ai-driven vaccination intervention optimiser for increasing vaccine uptake in nigeria. *arXiv preprint arXiv:2204.13663*, 2022.

Nakajima, S. and Sugiyama, M. Theoretical analysis of bayesian matrix factorization. *The Journal of Machine Learning Research*, 12:2583–2648, 2011.

Pal, S., Suggala, A. S., Shanmugam, K., and Jain, P. Optimal algorithms for latent bandits with cluster structure. In *International Conference on Artificial Intelligence and Statistics*, pp. 7540–7577. PMLR, 2023.

Pal, S., Tambe, M., Suggala, A., and Taneja, A. Improving mobile maternal and child health care programs: Collaborative bandits for time slot selection. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pp. 1540–1548, 2024.

Salakhutdinov, R. and Mnih, A. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pp. 880–887, 2008.

Slivkins, A. et al. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

Verma, S., Singh, G., Mate, A. S., Verma, P., Gorantala, S., Madhiwalla, N., Hegde, A., Thakkar, D. H., Jain, M., Tambe, M. S., et al. Deployed saheli: Field optimization of intelligent rmab for maternal and child care. In *Innovative Applications of Artificial Intelligence (IAAI)*, 2023.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688. Citeseer, 2011.

Zhang, C., Yang, Y., Zhou, W., and Zhang, S. Distributed bayesian matrix decomposition for big data mining and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3701–3713, 2020.

A. Conditional independence of Updates

Theorem A.1. *The updates in parameters for one user are independent from the other users.*

$$P(\theta_i|r,\theta_{j\neq i},X) = P(\theta_i|r,X) \tag{8}$$

Proof. Let X be the data, θ and r be the user and reward matrix random variables. The likelihood function is:

$$L(X,\theta,r) = \prod_{i=1}^{N}\Big((1-X_{i(u,t)})(1-\theta_u r_t) + (X_{i(u,t)}\theta_u r_t)\Big) \tag{9}$$

where θ_u is the u_{th} row of θ and r_t is the r_{th} column of r.

We want to establish that the updates of the conditional posterior of the parameters of the i_{th} user is independent of the parameters of the other users.

$$P(\theta_i|r,\theta_{j\neq i},X) = \frac{L(\theta_i,r,\theta_{j\neq i},X)p(\theta_i)}{\int L(\theta_i,r,\theta_{j\neq i},X)p(\theta_i)d\theta} \tag{10}$$

We can separate out the items in X where user i is involved.

$$L(\theta_i,r,\theta_{j\neq i},X) = L(\theta,r,X) \tag{11}$$

$$= \prod_{i=1}^{N}\big((1-X_{i(u,t)})(1-\theta_u r_t) + (X_{i(u,t)}\theta_u r_t)\big) \tag{12}$$

$$= \prod\left((1-X_i)(1-\theta_i r_t) + (X_i\theta_i r_t)\right) \tag{13}$$

$$\prod\left((1-X_{j\neq i})(1-\theta_{j\neq i} r_t) + (X_{j\neq i}\theta_{j\neq i} r_t)\right) \tag{14}$$

The second term in the product comes out in both numerator and denominator (as it is not dependent on θ_i) and cancels out. Thus we are left with

$$P(\theta_i|r,\theta_{j\neq i},X) = \frac{L(\theta_i,r,X_i)p(\theta_i)}{\int L(\theta_i,r,X_i)p(\theta_i)d\theta} \tag{15}$$

$$= P(\theta_i|r,X_i) \tag{16}$$

Where X_i is the data where i is involved.

□

B. Algorithms

Algorithm 1 and Algorithm 2 explain the full and alternating sampling SGLD methods. The complete algorithm is decribed by Algorithm 3.

Algorithm 1 SGLD with Full Sampling

Input: Batch Size n, Data X
Hyperparameters: Learning Rate ϵ
Parameters: λ, α
Initialize θ and r (from Section 4.1)
repeat
  Select batch x of size n from X.
  Calculate terms from Equation (5) and Equation (6)
  Update θ, r using Equation (1).
until θ, r converge
Return θ, r =0

Algorithm 2 SGLD with Alternating Sampling

Input: Batch Size n, Data X, User Blocks b
Hyperparameters: Learning Rate ϵ
Parameters: λ, α
Initialize θ and r (from Section 4.1)
repeat
  Select batch x of size n from X.
  for b_i **in** b **do**
    Calculate terms from Equation (5) for b_i
    Update θ_{b_i} using Equation (1).
  end for
  Merge θ_{b_i} to get θ.
  Calculate terms from Equation (6)
  Update r using Equation (1).
until θ, r converge
Return θ, r =0

C. Experiments

C.1. Data Generation

We test the matrix completion on randomly generated low rank matrices to analyze the performance of Bayesian matrix factorization using SGLD. A random matrix of size $u \times d$ with rank k is generated by multiplying two random matrices G and H of shape $u \times k$ and $k \times d$ respectively and adding a small amount of Gaussian noise to it. This is followed by a normalization step which scales the values to probabilities between $[0, 1]$.

$$M = \textit{normalize}(G \cdot H + N(\mu, \sigma^2))$$

The random matrices are randomly generated with each element being from the range $[0, 1]$ and μ and σ are chosen to be 0.5 and 0.1 respectively. The *normalize* operation is

Algorithm 3 Proposed Algorithm

Input: Time steps T, Samples per time step s.
Generate data X_0 from random users and time slots.
for $i = 1$ **to** T **do**
  if Sampling Method == *full* **then**
    θ, r from Algorithm 1
  else if Sampling Method = *alternating* **then**
    θ, r from Algorithm 2
  end if
  Calculate U, R (Section 4.1).
  Generate X_i with s samples using TS (Section 4.2).
end for
Return U, R =0

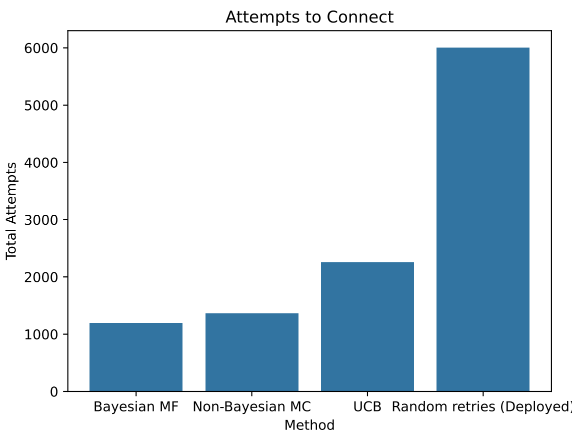

Figure 4: Average number of attempts required to connect to 1000 callers.

a scaling operation which maps the smallest and highest values to 0 and 1 and scales the other values accordingly. The value of k chosen is 4. This data is used in Section 5 for all experiments.

We compare the results with matrix completion using nuclear norm from (Pal et al., 2024). The criterion used for comparison is the Frobenius norm $||M||_F$ of the matrix from the original pick-up matrix.

C.2. Estimating Call Pickup Rates

We also measure expected number of attempts to reach a beneficiary for the different algorithms. The analysis shows that the proposed Bayesian method reduces the number of total calls required by 14% from the matrix completion method, by 47% from the UCB baseline and by about 81% from the current calling system employed by the algorithm.