Federated Multi-armed Bandits with Personalization

Chesta Pahuja
Sunil Kahalekar

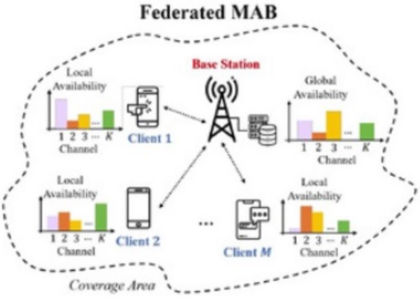
Industrial Engineering and Operation Research,
Indian Institute of Technology Bombay,
Powai Mumbai 400076, India

August 12, 2021

Outline

- Motivation: Federated MAB
- Generalization Vs Personlization
- PF-MAB Framework
- Learning Objective
- Lower Bound Analysis
- Algorithm Design:PF-UCB
    - PF-UCB Algorithm Phases
- Experiments
- Our approach
- Results and Observations
- Conclusion and Future Work

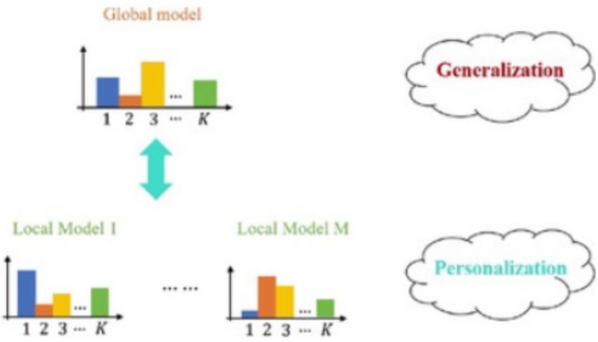
Motivation: Federated MAB



Key Observations :

- Server wants to learn the global model, but lacks direct access;
- Clients play heterogeneous (local) bandit games, but local observations only provide partial info of the global model;
- No one can solve the problem by itself → coordination (just like FL)

Generalization Vs Personlization



Recent advances in FL: seeking such personalization techniques.

$\boxed{\text{Federated MAB with Personalization}}$

PF-MAB Framework

- Client and Local model
  - M Clients
  - K arms;
  - Non-IID: $\mu_{k,m} = \mathbb{E}[X_{k,m}]$ for client m s arm k.
- Server and global model
  - K arms, $\mu_{k,m} = \mathbb{E}[X_k]$ for arm k
  - $\mu_k = \sum_{m=1}^{M} \mu_{k,m}$
- Constraint:
  - Server cannot directly interact with global model

Learning Objective

- Global Only : No personalization, full generalization
  - Only care about Global performance → Global cumulative reward

$$r_g(T) = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{m=1}^{M} X_{\pi_{m(t)}}(t)\right]$$

  - optimal choice: Global optimal arm;
  - solution : FMAB

Learning Objective

- Local Only : No generalization, full personalization
  - Only care about local performance → Local cumulative reward

$$r_l(T) = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{m=1}^{M} X_{\pi_m(t),m}(t)\right]$$

  - optimal choice: individual local optimal arm;
  - solution : Standard MAB → UCB, Thompson Sampling.

Learning Objective

- Global Only

$$r_g(T) = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{m=1}^{M} X_{\pi_{m}(t)}(t)\right]$$

- Local Only

$$r_l(T) = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{m=1}^{M} X_{\pi_{m}(t),m}(t)\right]$$

- Trading-off between generalization and personalization leads to a mixed learning objective :

$$r(T) = \alpha * r_l(T) + (1-\alpha) r_g(T)$$

$\alpha \in [0, 1]$ controls the "balance" between two competing objective

- Discussion :
  - $\alpha = 1$, local-only
  - $\alpha = 0$, global-only
  - $0 < \alpha < 1$, mixed

The Mixed Model

- The Mixed objective :

$$\alpha \in [0, 1], r(T) = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{m=1}^{M} X'_{\pi_m(t),m}(t)\right]$$

- Hypothetical mixed model for client m :
  - Mixed reward:

$$X'_{k,m}(t) = \alpha * X_{k,m}(t) + (1-\alpha)X_k(t)$$

  - Mixed mean reward:

$$\mu'_{k,m} = \underbrace{\left(\alpha + \frac{1-\alpha}{m}\right)\mu_{k,m}}_{\text{local info}} + \underbrace{\frac{1-\alpha}{m}\sum_{n\neq m}\mu_{k,m}}_{\text{global info}}$$

  - Challange : global info is determined by other clients

The Mixed Model

- Regret definition: w.r.t. the mixed learning objective

$$R(T) = \underbrace{T \sum_{m=1}^{M} \mu'_{*,m} - \mathbb{E}\left[\sum_{t=1}^{T} \sum_{m=1}^{M} X'_{\pi_m(t),m}(t)\right]}_{\text{exploration-exploitation}} + \underbrace{CMT_c}_{\text{communication}}$$

Lower Bound Analysis

Theorem 1

For any consistent algorithm, the regret R(T) can be lower bounded as

$$\liminf_{T\to\infty} \frac{R(T)}{\log(T)} \geq \sum_{m=1}^{M} \sum_{k\neq k'_{*,m}} \max\left\{ \frac{\Delta'_{k,m}}{\mathrm{kl}\left(Y_{k,m}, Y_{k'_{*,m},m}\right)}, \frac{\Delta'_{k,m}}{\min_{n:n\neq m, k'_{2,n}\neq k} kl\left(Z^{m}_{k,n}, Z^{m}_{k'_{*,n},n}\right)} \right.$$

- blue term: loss for local exploration
- green term: loss for global exploration
- minimum &maximum: worst case
- not a tight lower bound

Algorithm Design:PF-UCB

Three key principles inspired by lower bound analysis:

- "Being selfish": Learning local bandit model for its own benefit;
- "Being generous": Learning global model to help others;
- "Talk less and do more: only communicate periodically.

Algorithm Design:PF-UCB

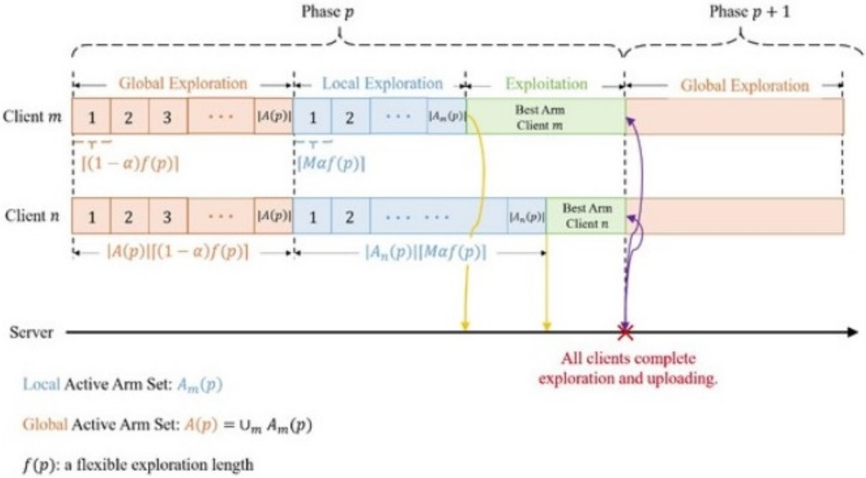


Figure: PF-UCB Algorithm Phases

Algorithm Design:PF-UCB

Each phase consists of three sub-phases: global exploration, local exploration, and exploitation.

- Global Exploration: Arms in the global active arm set to get global information. At phase $p, A_m(p)$ and $A(p) = \cup_{m\in[M]} A_m(p)$ denote the set of local and global active arms respectively.
- Local Exploration: Each arm $k \in A_m(p)$ is played by client m for $n^l_{k,m}(p) = \lceil M\alpha f(p) \rceil$ times. Total local exploration time slots at client m end: $K_m(p)\lceil M\alpha f(p) \rceil$. Due to heterogeneity clients have different local exploration lengths i.e., $K_m(p)$
- Model updates: Each arm $k \in A_m(p)$
  - Global exploration explored for $\lceil (1-\alpha)f(p) \rceil$ times
  - local exploration $\lceil M\alpha f(p) \rceil$ times
  - The total numbers of pulls by client m

$$n_{k,m}(p) = \lceil (1-\alpha)f(p) \rceil + \lceil M\alpha f(p) \rceil$$

Algorithm Design:PF-UCB

- After both global and local explorations, client m first sends "local model updates" to the server
- It contains updated local sample means of all global active arms $k \in A(p)$, as $\bar{\mu}_{k,m}(p)$ for arm k at phase p.
- Due to different local exploration length the server not receive the updates from all clients at the same time.
- Server has to wait until the updated sample means from all the clients are received and then sends the aggregated "global model" $\bar{\mu}_k(p) = \frac{1}{M}\sum_{m=1}^{M} \bar{\mu}_{k,m}(p)$ back to the clients.
- Synchronization among the clients required to minimize regret. Main drawback is all clients have to wait for the slowest client before the next iteration.
- Client can start exploitation once local updates send to server but server has to wait for other clients.

Algorithm Design:PF-UCB

- After sending local updates to server client can start exploitation but server has to wait for other clients.
- The global sample means $\bar{\mu}_k(p)$ are broadcast to the clients.
- Estimation for $\mu'_{k,m}$ is updated as $\bar{\mu}'_{k,m}(p) = \alpha\bar{\mu}_{k,m}(p) + (1-\alpha)\bar{\mu}_k(p)$. Then, a local arm elimination procedure is called to remove sub-optimal arm with a high probability.
- The local active set $A_m(p+1)$ for the next phase is updated as $A_m(p+1) = A_m(p) \backslash E_m(p)$.
- At end all the clients send $A_m(p+1)$ to the server and receive the global active set $A(p+1) = \cup_{m\in[M]} A_m(p+1)$ from the server.

Experiments from the paper

Synthetic Dataset

1. 4 clients, 9 arms
2. Local optimal arm for client $m \in \{1, 2, 3, 4\}$ is arm m
3. Global optimal arm is arm 9

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0.9 & 0.4 & 0.35 & 0.35 & 0.5 \\ 0 & 1 & 0 & 0 & 0.3 & 0.9 & 0.35 & 0.3 & 0.5 \\ 0 & 0 & 1 & 0 & 0.35 & 0.35 & 0.9 & 0.3 & 0.5 \\ 0 & 0 & 0 & 1 & 0.4 & 0.3 & 0.35 & 0.9 & 0.5 \end{bmatrix}$$

Regret curve

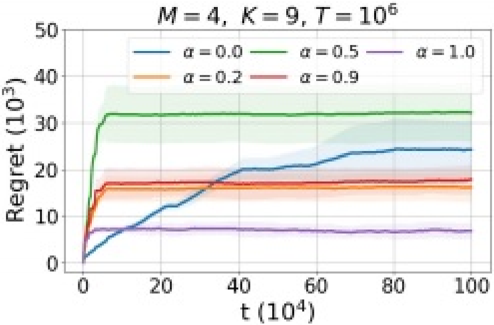


Figure: Synthetic Regret

MovieLens experiments

- 2113 clients and 10197 movies
- Randomly partitioned into 10 and 40 bunches

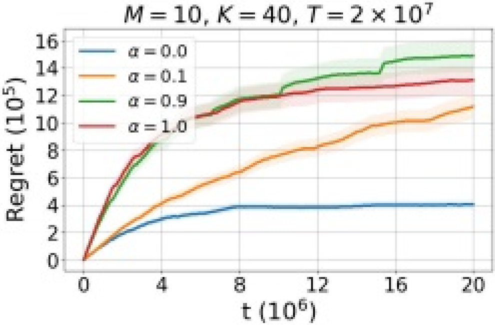


Figure: MovieLens regret

Our approach

Key Concepts:

- Cross silo vs. Cross-device federated learning (CDFL)
  - Data heterogeneity
  - Data transfer
  - Client participation
  - Client availability
- Extending PF-MAB to include CDFL
  - Only a fraction of clients are available at one time; we made changes in the algorithm to randomly sample a fraction from the available clients
  - Use only those clients to learn the mixture of local and the global model
  - Selected device **locally computes an update** using local data
  - server collects an **aggregate** of the device updates; integration point for secure aggregation for added privacy
  - Server locally updates the shared model from for all participants

Implementation

- Challenges:
  - Method for client sampling
  - Higher computation time
- Implementation:
  - Making changes to the official code provided by the authors
  - Movielens dataset with 2113 users and 10197 movies.
  - Pre-processing the MovieLens dataset- best movie genre both globally and locally to each client
  - Ratings of 10197 movies are averaged over 20 groups(arms) according to movie genres.
  - For simulation purposes the ratings were averaged over 1000 randomly formed user groups.
  - Algorithm randomly selects a partition(say 10/1000) of these 1000 users with probability 'p' at each time-step and calculates the regret.

Experiments

- Fix p = 0.08, clients are sampled randomly with 8 % probability (roughly 80-90 clients each time step).
- α = 0.5 , p = 0.08 and T = 200000.
- Time horizon is small and regret doesn't converge within this time horizon
- Possibly because client sampling is greedy in nature.
- Rigorous theoretical analysis- new client sampling

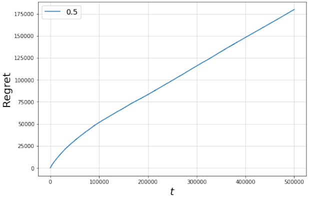


Figure: Regret with α = 0.5, p = 0.08, T = 500000

Results

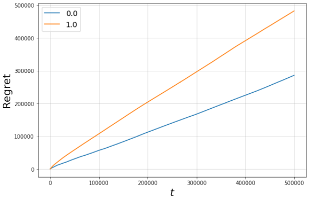


Figure: Regret with $\alpha = 0.0$ and 1.9, p $=$ 0.08, T $=$ 500000

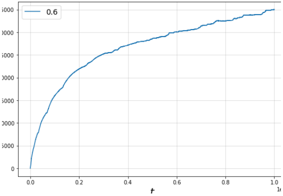


Figure: Regret with $\alpha = 0.6$, p $=$ 0.8

Results and Observations

- Regret still not converged, increasing slowly; elimination of sub-optimal arms
- 1000 users further randomly divided into 10 user groups where all the participants participate
- Time horizon is fixed at $T = 2 * 10^6$

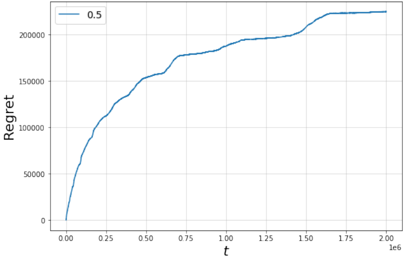


Figure: Regret with $\alpha = 0.5$, p = 1

Experiments

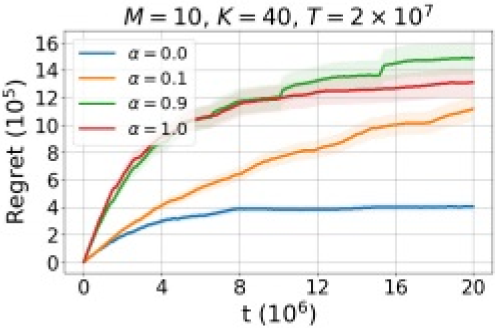


Figure: MovieLens regret

Results

- Our modified algorithm + modified dataset gives much lower regret($1.6 * 10^5$) than the regret recieved in the original experiments($12 * 10^5$) conducted by the author.

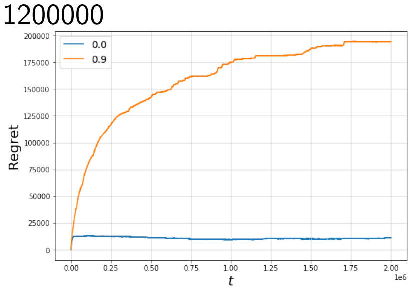


Figure: Regret with $\alpha = 0.0$ and 0.9, p $= 1$

Conclusion and Future Work

- Extended the PF-UCB method proposed in [1] to take a more realistic approach
- Novel extension's performance is measured using non-IID, real-world movie ratings - Dataset MovieLens.
- Change does actually enhance large-scale client engagement and customer sampling
- New research directions- Adding noise to the rewards for differential privacy; dropping lagging clients to save computation time
- More experiments should be undertaken
- Client sampling method can be improved as in [4]
- Thorough theoretical analysis must be undertaken to demonstrate the effectiveness of proposed extension.

References

- Li, T., Song, L. and Fragouli, C., 2020, June. Federated recommendation system via differential privacy. In 2020 IEEE International Symposium on Information Theory (ISIT) (pp. 2592-2597). IEEE.
- Shi, C., Shen, C. and Yang, J., 2021, March. Federated Multi-armed Bandits with Personalization. In International Conference on Artificial Intelligence and Statistics (pp. 2917-2925). PMLR
- Hanzely, F. and Richtárik, P., 2020. Federated learning of a mixture of global and local models. arXiv preprint arXiv:2002.05516.
- Shi, C. and Shen, C., 2021, January. Federated multi-armed bandits. In 35th AAAI Conference on Artificial Intelligence.
- Zhu, Z., Zhu, J., Liu, J. and Liu, Y., 2021. Federated bandit: A gossiping approach. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 5(1), pp.1-29.

References

Deng, Y., Kamani, M.M. and Mahdavi, M., 2020. Adaptive personalized federated learning. arXiv preprint arXiv:2003.13461.

Mansour, Y., Mohri, M., Ro, J. and Suresh, A.T., 2020. Three approaches for personalization with applications to federated learning. arXiv preprint arXiv:2002.10619.

Dubey, A. and Pentland, A., 2020. Differentially-Private Federated Linear Bandits. arXiv preprint arXiv:2010.11425.

Shi, C., Shen, C. and Yang, J., 2021, March. Federated Multi-armed Bandits with Personalization. In International Conference on Artificial Intelligence and Statistics (pp. 2917-2925). PMLR.

Thank you