Federated Multi-Arm Bandit

Adaptation of MABs by Samuel Gerstein

Project 1 Part 2 Review

- 10000 User Ad Clicks Given
- R = 1 for each click
- MAB Maximizes User Clicks

Ad 1	Ad 2	Ad 3	Ad 4	Ad 5	Ad 6	Ad 7	Ad 8	Ad 9	Ad 10
1	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0

What are the drawbacks of the Multi-Arm Bandit in Part 2?

- Transmission of Sensitive User Data
- MAB can not be scaled for more than one entry at a time
- MAB cannot be run locally for devices with poor connectivity

Proposed Federated Bandit Framework

- Learning happens solely on the devices, no user data is transmitted
- Bandit pulls can run in parallel, easily scalable for large online data
- Quicker convergence to the optimal policy

Federated Bandit Client Pseudocode

Algorithm 1: Federated MAB Client

Input: Global Action-Values $Q^{'}$, Number of Pulls k
Output Mean Differential of Action-Values wrt the Global Model $\Delta Q^{'}$
initialize $Q \leftarrow Q^{'}$;
initialize $N(j) \leftarrow 0$ for j = 1...k;
for i = *1*...k **do**

$$A \leftarrow \begin{cases} \arg\max_a Q(a) & \text{with probability } 1 - \varepsilon \\ \text{a random action} & \text{with probability } \varepsilon \end{cases}$$

display Ad A to user
observe click R
$N(A) \leftarrow N(A) + 1$
$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}[R - Q(A)]$

end
$\Delta Q^{'} \leftarrow Q - Q^{'}$
return $\Delta Q^{'}$

Central Server Pseudocode

Algorithm 2: Federated MAB Server

Input: Number of Communication Rounds C

initialize $Q(j) \leftarrow 0$ for j = 1...$actions$;

for i = *1*...C **do**

  for *each client m in server* **do**

    send global model Q

    run client

    receive $\Delta Q(m)$

  end

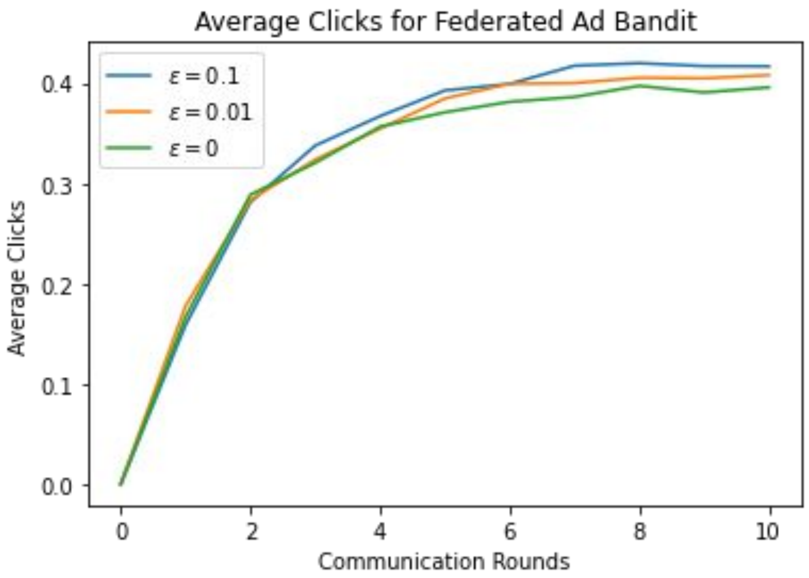
  $Q \leftarrow Q + \overline{\Delta Q}$

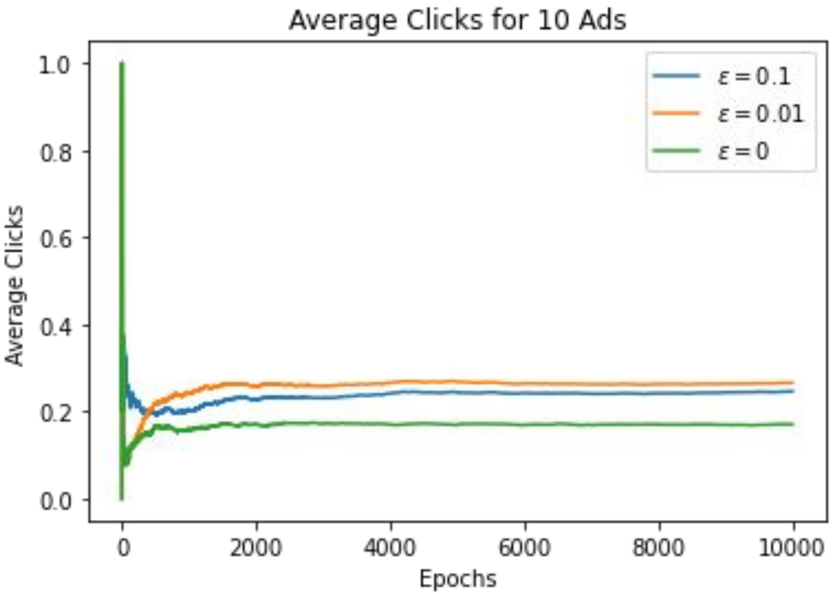
end

Experiment Design

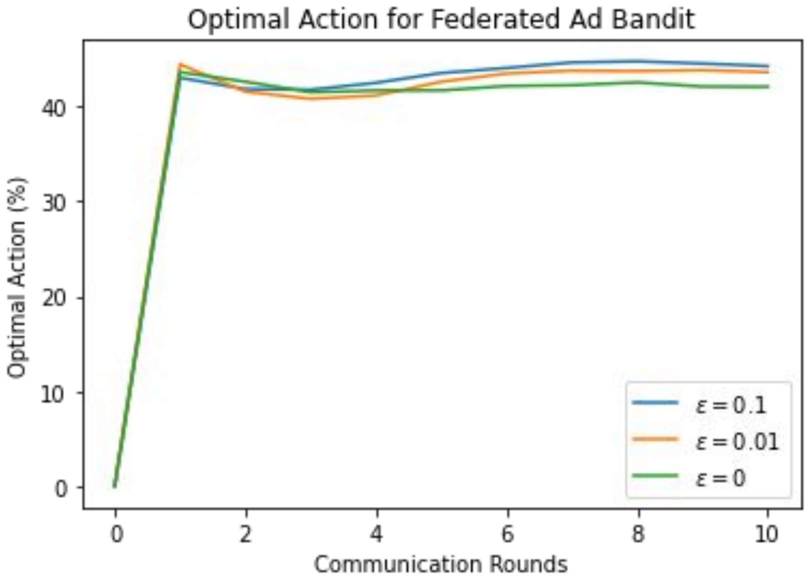
- 10000 Ad Entries Divided Equally into 100 Clients
- 10 Communication Rounds
- 10 Local Pulls

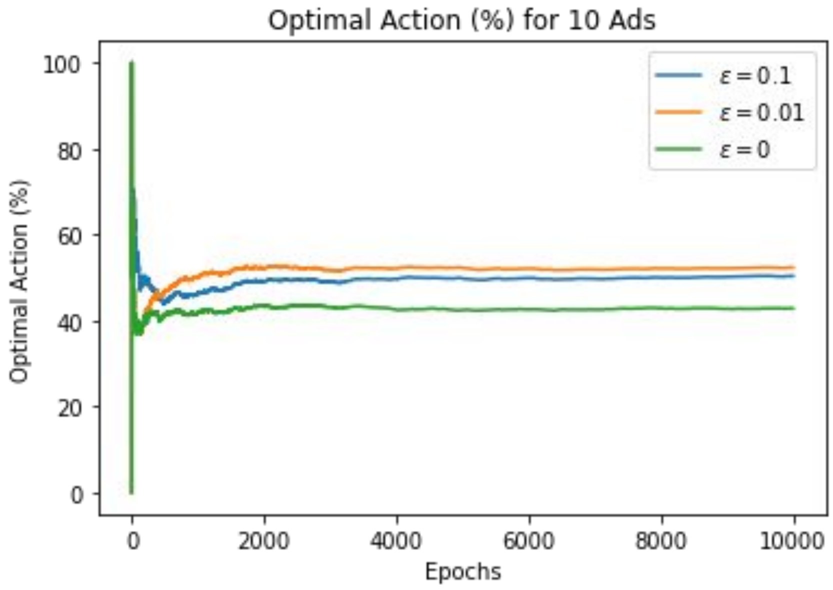
Average Clicks Comparison





Optimal Action Comparison

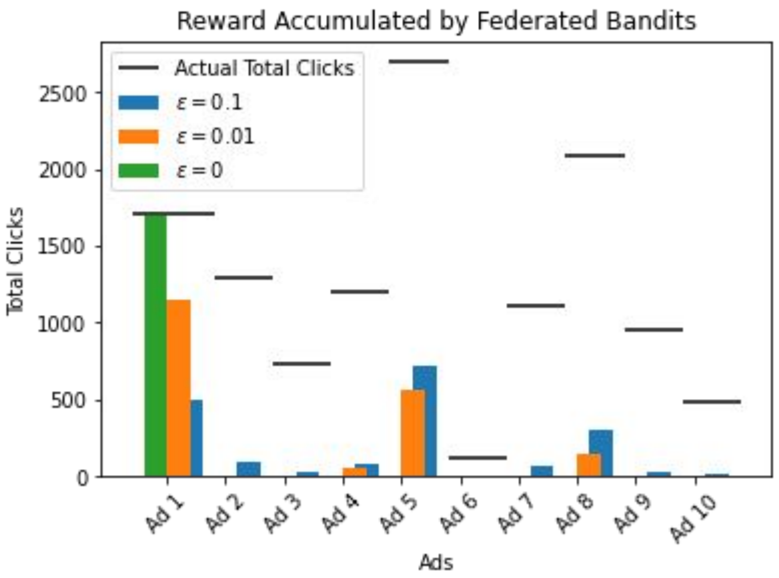


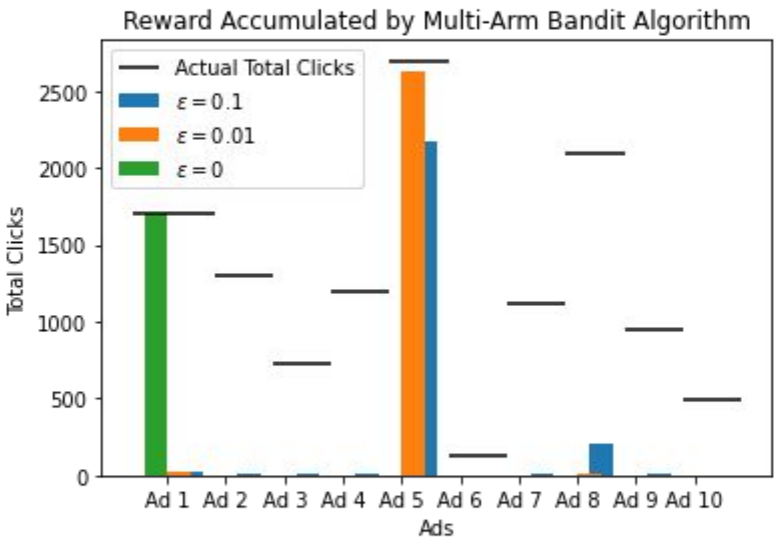


Discussion

- Modest Increase in Average Clicks
  - Federated: 0.3 - 0.4
  - Centralized: 0.15 - 0.3
- Decrease in Optimal Action Percentage
  - Centralized: 40 - 60%
  - Federated: 40 - 45%
- Local Epsilon-Greedy Policy Not as Effective
  - 1-3% Optimal Action Difference between greedy and epsilon-greedy approach
  - 0.05 Average Clicks Difference between greedy and epsilon-greedy approach

Total Reward Comparison





How to Remedy Federated Exploration Problem

- Weigh Exploring Agents Larger
  - Simple averaging may drown out outliers
- Federated Averaging of Gradient-Decay Epsilon
  - Treat Epsilon as a learned parameter, average in communication rounds
- Federated Upper Confidence Bound Algorithm
  - Learn Upper Interval, average in communication rounds

Conclusion

- Explained the issues with centralized MAB
  - Privacy
  - Sequential Computing
  - Long Convergence Time
- Proposed Federated MAB Framework
  - Global model updated with mean differential of local means
- Found the drawbacks of local exploration in Federated MAB
  - Little difference in optimality with greedy policy
  - Discussed Federated Gradient Epsilon and UCB