A Multi-Armed Bandit Framework for Recommendations at Netflix

Jaya Kawale
Elliot Chow
Recommendations at Netflix

Personalized Homepage for each member

- **Goal**: Quickly help members find content they’d like to watch
- **Risk**: Member may lose interest and abandon the service
- **Challenge**: 117M+ members
- **Recommendations Valued at**: $1B*

---

Our Focus: Billboard Recommendation

Goal: Recommend a **single relevant title** to each member at the right time and **respond quickly** to member feedback.

Example Billboard of Daredevil on the Netflix homepage
Traditional Approaches for Recommendation

- Collaborative Filtering based approaches most popularly used.
  - Idea is to use the “wisdom of the crowd” to recommend items
  - Well understood and various algorithms exist (e.g. Matrix Factorization)
Challenges for Traditional Approaches

Challenges for traditional approaches for recommendation:

- Scarce feedback
- Dynamic catalog
- Non-stationary member base
- Time sensitivity
  - Content popularity changes
  - Member interests evolves
  - Respond quickly to member feedback
Multi-Armed Bandits
Increasingly successful in various practical settings where these challenges occur
Multi-Armed Bandit For Recommendation

- Multiple slot machines with unknown reward distribution
- A gambler with multiple arms
- Which machine to play in order to maximize the reward?
Bandit Algorithms Setting

For each round

- Learner chooses an **action** from a set of available actions
- The environment generates a response in the form of a real-valued **reward** which is sent back to the learner
- Goal of the learner is to maximize the cumulative reward or minimize the cumulative regret which is the difference in total reward gained in n rounds and the total reward that would have been gained w.r.t to the optimal action.
Multi-Armed Bandit For Recommendation

*Exploration-Exploitation tradeoff*: Recommend the optimal title given the evidence i.e. *exploit* or recommend other titles to gather feedback i.e. *explore.*
Principles of Exploration

- The best long-term strategy *may involve short-term sacrifices.*

- Gather information to make the best overall decision.
  - **Naive Exploration**: Add a noise to the greedy policy. [ε-greedy]
  - **Optimism in the Face of Uncertainty**: Prefer actions with uncertain values. [Upper Confidence Bound (UCB)]
  - **Probability Matching**: Select the actions according to the probability they are the best. [Thompson Sampling]
Numerous Variants

- Different Environments:
  - **Stochastic and stationary**: Reward is generated i.i.d. from a distribution specific to the action. No payoff drift.
  - **Adversarial**: No assumptions on how rewards are generated.

- Different objectives: **Cumulative** regret, **tracking** the best expert

- **Continuous or discrete** set of actions, finite vs infinite

- Extensions: Varying set of arms, Contextual Bandits, etc.
Epsilon Greedy for MABs

Epsilon Greedy

○ **Exploration:**
  ■ Uniformly explore with a probability $\epsilon$
  ■ Provides **unbiased data** for training.

○ **Exploitation:** Select the optimal action with a probability $(1 - \epsilon)$
Key Aspects of Our Framework

- Can support **different contextual bandit algorithms** i.e., Epsilon Greedy, Thompson Sampling, UCB, etc.

- **Closed-loop system** that establishes a link between how recommendations are made and how our members respond to them, important for online algorithms.

- Supports **snapshot logging** to log facts to generate features for offline training.

- Supports **regular updates** of policies.
System Architecture
Key Components

**Online**
- Apply explore/exploit policy
- Log contextual information
- Score and generate recommendations

**Offline**
- Attribution assignment
- Model training
Apply Explore/Exploit Policy

- **Generate** the candidate pool of titles
- **Select** a title from candidate pool
  - For uniform exploration, randomly select a title uniformly from the candidate pool
Log Contextual Information

- Exploration Probability
- Candidate pool
- Selected title
- Snapshot facts for feature generation
Attribution Assignment

- **Filter** for relevant member activity
- **Join** with explore/exploit information
- **Define** and construct sessions
- **Generate** labels
Billboard Candidate Titles

- Title A
- Title B
- Title C

Selected Billboard Title

Apply MAB Model

Title A

Homepage Construction

- Render Home Page
- Play Title A from Home Page
Feature Generation

- **Join** labels with snapshotted facts
- **Generate** features using [DeLorean](https://example.com)
  - Feature encoders are shared online and offline
Model Training and Publishing

- **Train** and validate model
- **Publish** the model to production
Metrics and Monitoring

- A/B test metrics
- Distribution of arm pulls
  - Stability
  - Explore vs. Exploit
- Take Rate
  - Convergence
  - Online v.s. Offline
  - Explore v.s. Exploit
Example Bandit Policies For Recommendation
Background and Notation

- Let $k = 1, \ldots, K$ denote the set of titles in the candidate pool when a member arrives on the Netflix homepage.

- Let $x_{ik} \in \mathbb{R}^d$ be the context vector for member $i$ and title $k$.

- Let $y_{ik}$ represent the label when member $i$ was shown the title $k$. 
Greedy Exploit Policy

- Learn a **model per title in the candidate pool** to predict the **likelihood of play on the title**
  \[
  Pr(y_{ik} = 1|x_{ik}, K) = \sigma(f(x_{ik}, \Theta))
  \]

- Pick a winning title:
  \[
  k = \arg \max \Pr(y_{ik} = 1|x_{ik}, K)
  \]

- Various models can be used to learn to predict the probability, for example, logistic regression, neural networks or gradient boosted decision trees.
Greedy Exploit Policy

Candidate Pool

Member

Features

Model 1

Model 2

Model 3

Model 4

Probability Of Play

Winner
Would the member have played the title anyways?
Causal Effect of an Advertisement

- Advertising: Target the user to increase the conversion.

- Causal Question: Would the user have converted anyways?*

Goal: Measure ad effectiveness.

Incrementality: The difference in the outcome because the ad was shown; the causal effect of the ad.

Goal: Recommend title which has the largest additional benefit from being presented on the Billboard

- Member could have played the title from anywhere else on the homepage or from search
- Popular titles likely to appear on the homepage via other rows e.g., Trending Now
- Better to utilize the real estate on the homepage for recommending other titles.

Define Policy to be incremental with respect to probability of play.
Incrementality Based Policy on Billboard

Goal: Recommend title which has the largest additional benefit from being presented on the Billboard

\[
\arg \max P(y_{ik} = 1|x_{ik}, K, b = 1) - P(y_{ik} = 1|x_{ik}, K, b = 0)
\]

Where \( b=1 \) → Billboard was shown for the title and \( b=0 \) → not shown.
Offline Evaluation: Replay  [Li et al, 2010]

- Relies upon **uniform exploration data**. For every record in the uniform exploration log \{context, title \(k\) shown, reward, list of candidates\}

- Offline Evaluation: For every record
  - Evaluate the trained model for **all the titles** in the candidate pool.
  - Pick the winning title \(k'\)
  - Keep the record in history if \(k' = k\) (the title impressed in the logged data) else discard it.
  - Compute the metrics from the history.
Offline Evaluation: Replay [Li et al, 2010]

Uniform Exploration Data - Unbiased evaluation

Evaluation Data

- context, title, reward
- context, title, reward
- context, title, reward

Train Data

- Reveal context x
- Winner title k'
- Use reward only if k' = k

Trained Model

Take Rate = \# Plays
\# Matches
Offline Replay

Exploit has higher replay take rate as compared to incrementality.

*Incrementality Based Policy sacrifices replay by selecting a lesser known title that would benefit from being shown on the Billboard.*

Lift in Replay in the various algorithms as compared to the Random baseline
Title A has a **low baseline probability of play**, however when the billboard is shown the probability of play increases substantially!

Title C has higher baseline probability and **may not benefit as much** from being shown on the Billboard.

Scatter plot of incremental vs baseline probability of play for various members.
Online Observations

- Online take rates for take rates follow the offline patterns.

- Our implementation of incrementality is able to shift engagement within the candidate pool.
Future Work

- Framework allows for easily plugging in different policies. Enables -
  - Policy exploration:
    - Different MAB policies TS, UCB, etc.
    - Other ways of combining causal inference with MABs.
  - Model exploration:
    - Different models like NN, LR, GBDT, etc.
  - Reward exploration.
    - Consider long term reward
    - Different kinds of rewards
Thank you.