Multi-armed Bandits with externalities

Viraj Nadkarni

IIT Bombay

Autumn 2019

The General Problem

- Similar to the stochastic Multi-armed bandit problem but with user types.
- Each arm has a corresponding type.
- The recommender gets the maximum reward when the type of the user and the arm matches.
- In each time slot, a user comes with a known type.
- $N \times N$ matrix B of reward means, with each row i having b_{ii} as the maximum value (w.l.o.g assume the b_{ii} decrease with i)
- **Aim** : Opinion Shaping , Reward Maximization

The model for two arms

Reward structure :

- A set of two arms and a corresponding set of two user types (each with populations $Z_0(t)$ and $Z_1(t)$ and proportions $z_0(t)$ and $z_1(t)$).
- We have a matrix $B = [b_{ij}]_{2\times 2}$ of the mean rewards where in each row i ,the element b_{ii} has the highest value.
- If the user arriving at time t is of type i and is shown the arm j, then the reward $R(t)$ is chosen from a Bernoulli distribution with mean b_{ij}.

The model for two arms

Updating the population : After reward $R(t)$ is obtained, the $Z_k(t)$'s are updated as follows -

- $Z_j(t+1) = Z_j(t) + R(t)$
- $Z_{-j}(t+1) = Z_{-j}(t) + (1 - R(t))$

where Z_{-j} is the population of the arm that was not recommended.
Policy : A tuple (p, q) denotes a unique recommendation policy, where p is the probability of recommending arm 0 given user of type 0 arrives and q is the same but for arm 1.
Aim : Choose policy to maximize z_0 when B is known and when it is unknown.

Known B, Fixed policy

Based on the model described in the previous slides, we arrive at the following ODE in expected z_0.

Differential equation :

$$\frac{dz}{dt} = \frac{-(d_1 + d_2)z + d_2}{A + t}$$

where $z = z_0(t)$,$A = z_0(0) + z_1(0)$(initial total population) and d_1, d_2 are given by :

$$d_1 = p(1 - b_{00}) + (1 - p)b_{01}$$

$$d_2 = q(1 - b_{11}) + (1 - q)b_{10}$$

Known B, Fixed policy

- **Solution to the ODE :**

$$z_0(t) = \frac{d_2}{d_1 + d_2} + (z_0 - \frac{d_2}{d_1 + d_2})(1 + \frac{t}{A})^{-(d_1+d_2)}$$

  Therefore, as t goes to ∞, z_0 approaches $d_2/(d_1 + d_2)$.

- The optimal policy (that maximizes long term population of type 0) is the (p, q) s.t $d_2/(d_1 + d_2)$ is maximized.

Known B, Fixed policy

Optimal policy : The optimal policy that maximizes the z_0 turns out to be :

$$p = I_{\{b_{00}+b_{01}>1\}}$$

$$q = I_{\{b_{10}+b_{11}<1\}}$$

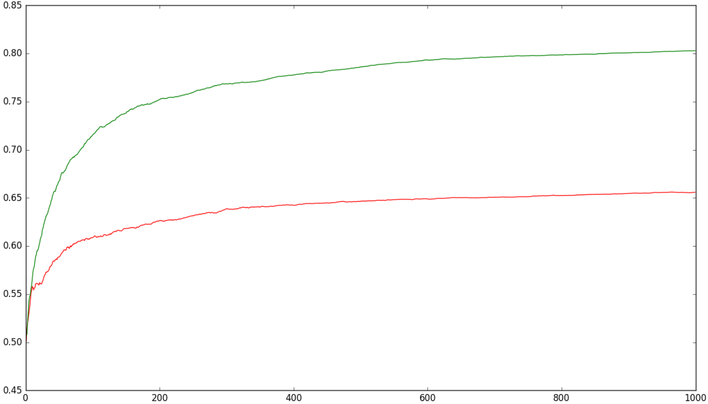
(where I_e is the indicator function of event e)
Intuition : To choose p, compare b_{00} (= probability that user would like arm 0) and $1 - b_{01}$ (= probability that user would dislike arm 1).

Known B, Fixed policy, Maximizing reward

- Suppose we wish to maximize cumulative reward instead.
- Optimal policy may not be always equal to the "Greedy policy" (showing each user type their preferred arm or $p = 1, q = 1$) .
- Next slides : Comparison when optimal policy is $p = 0, q = 0$ with greedy policy.

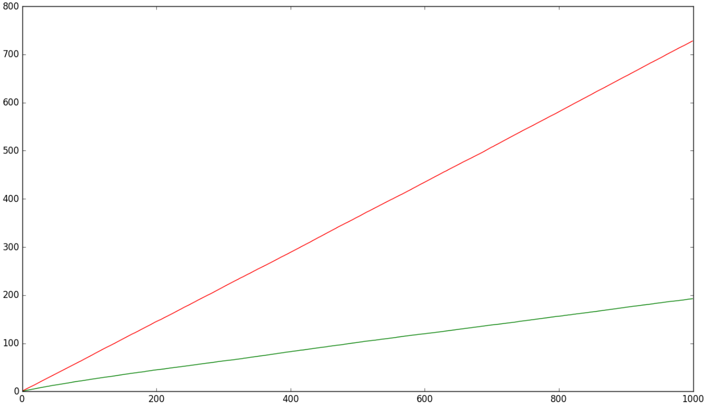
Known B, Fixed policy, Maximizing reward

Figure 1: Proportion of type 0 users vs time (Red : Greedy , Green : Optimal)



Known B, Fixed policy, Maximizing reward

Figure 2: Cumulative reward vs time (Red : Greedy, Green : Optimal)

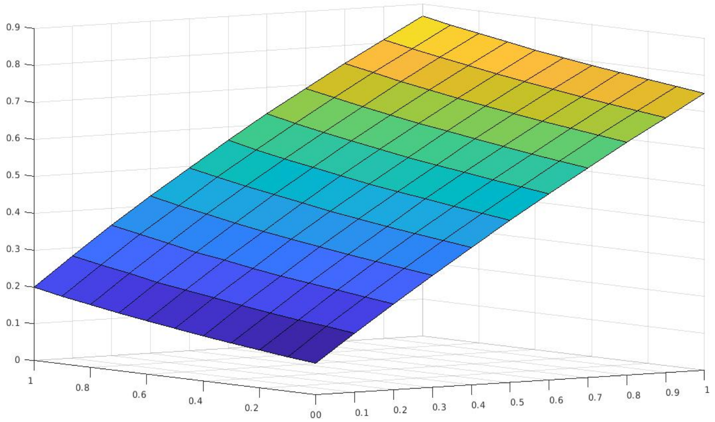


Known B, Fixed policy, Maximizing reward

Q : Is there a non-greedy (p, q) that maximizes cumulative reward ?
A : For the constraints that we have put on the matrix B the answer is NO.

Known B, Fixed policy, Maximizing reward

Figure 3: XY - plane : Values of (p,q), Z-axis : Value of equilibrium reward



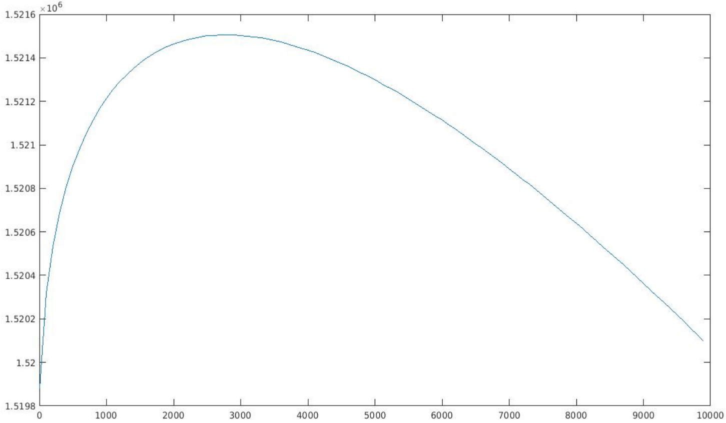
Known B, Fixed policy, Maximizing reward

Q : What about a mixed policy ? (do optimal policy for some time and greedy policy after that)

A : Improvement seen but only marginal.

Known B, Fixed policy, Maximizing reward

Figure 4: Cumulative reward at deadline Vs T_0 (deadline $T = 2 \times 10^6$)



Unknown B

What if the matrix B is unknown ?

- The general problem we seek to tackle in this is :
  Problem : Given a deadline T, find a policy that maximizes the population of type 0 users at $t = T$
- **A naive Explore-then-Commit policy :** Show arms uniformly at random and keep updating the estimate of matrix B till a time T_{thresh}. After this, show arms according to the optimal policy for the estimated matrix $\hat{B}$.

Analyzing the Explore-Then-Commit policy

Notation :

- Z_t =number of type 0 balls in urn at time t
- z_t =proportion of type 0 balls in the urn at time t

Define **regret** at time T as :

$$Regret(T) = E[\sum_{t=1}^{T}(\Delta Z_t^{opt} - \Delta Z_t^{pol})]$$

where the ΔZ_t^{opt} is the number of type 0 balls added at time t *given that the proportion of type 0 balls in the urn is* z_t^{pol}. We wish to minimize this regret.

Analyzing the Explore-Then-Commit policy

The above definition gives us the following expression for regret at time T.

$$Regret(T) = R_{explore} + R_{exploit}$$

where,

$$R_{explore} = 0.5(\sum_{1}^{2m} z_t)|b_{00} + b_{01} - 1| + 0.5(2m - \sum_{1}^{2m} z_t)|b_{11} + b_{10} - 1|$$

and

$$R_{exploit} = p'(\sum_{2m}^{T} z_t)|b_{00} + b_{01} - 1| + q'(T - 2m - \sum_{2m}^{T} z_t)|b_{11} + b_{10} - 1|$$

Here p' and q' are the probabilities of "bad" events happening. That is, if say $b_{00} + b_{01} > 1$ then $p' = Prob(\hat{b_{00}} + \hat{b_{01}} < 1)$.

Analyzing the Explore-Then-Commit policy

Deriving bounds on regret for a special case : Consider the case where we have $b_{00} = b_{11}$ and $b_{01} = b_{10}$. For this case we get the following two bounds on the regret as defined previously :

- **Gap-dependent bound :**

$$R_T \leq m + e^{\frac{-m\Delta^2}{2}}(T - 2m)$$

  where $\Delta = |b_{00} + b_{01} - 1|$

- **Gap-independent bound :**

$$R_T \leq \mathcal{O}(T^{2/3}(log(T))^{1/3})$$

ETC algorithm results using the Hoeffding bounds

- We now plot the analytical expressions for the population trajectory for the ETC algorithm.
- The expression used to plot this is :

$$z(t) = z_\infty + (z_{explore} - z_\infty)(1 + t/A)^{-(d_1+d_2)}$$

where d_1 and d_2 depend on the policy (p, q).

- Also, we use the following fact

$$\mathcal{E}(z(t)) = z_{right} * P(right) + z_{wrong} * P(wrong)$$

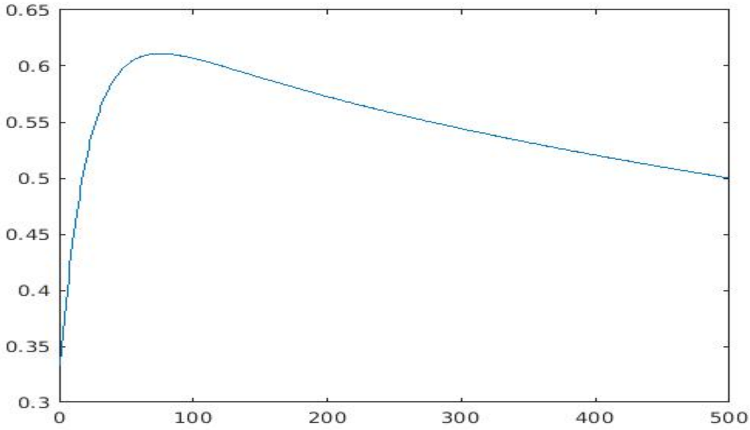
where the events "wrong" and "right" are the events that our estimates at the end of the exploration phase are wrong and right respectively.

- We use the following bound on $P(wrong)$:

$$P(wrong) \leq e^{-kT_{thresh}\Delta^2}$$

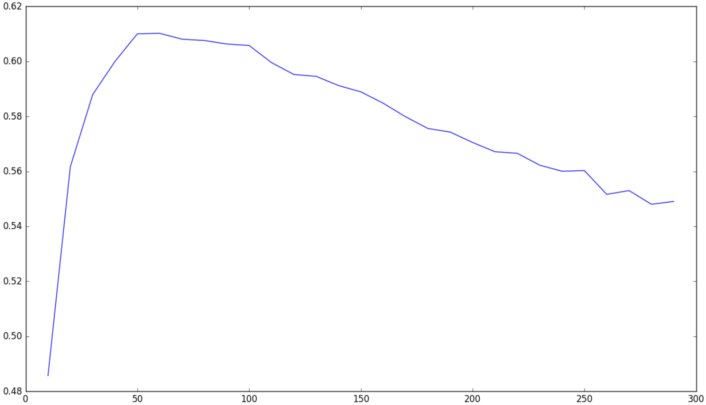
Unknown B, Explore-then-Commit

Figure 5: Proportion vs Value of threshold for ETC policy with deadline = 500



Unknown B, Explore-then-Commit

Figure 6: Proportion of type 0 users at deadline Vs T_{thresh} (deadline $T = 500$)



Need for a better policy

- Since the matrix B is unknown and the optimal threshold $2m$ in our Explore-Then-Commit policy depends heavily on the values in B (through Δ_i), we need a policy that keeps track of our confidence in the estimates of the elements of the matrix B.
- Therefore we try out two policies that use concepts similar to those used in the UCB and Thompson sampling algorithms.

UCB-LCB algorithm

This algorithm follows the following steps. In each time step, do :

- Keep an estimate matrix of the matrix B as $\hat{B}$.
- Define :

$$UCB_{ij} = \hat{b}_{ij} + \sqrt{\frac{klog(t)}{T_{ij}(t)}}$$

$$LCB_{ij} = \hat{b}_{ij} - \sqrt{\frac{klog(t)}{T_{ij}(t)}}$$

  where $T_{ij}(t)$ is the number of times b_{ij} was sampled till time t.
- If $UCB_{00} + UCB_{01} - 1 < 0$ then set $p = 0$. If $LCB_{00} + LCB_{01} - 1 > 0$ then set $p = 1$. If neither are true, set $p = 0.5$.
- Do a similar procedure for setting the value of q.

Thompson sampling based algorithm

This algorithm follows the following steps. In each time step, do :

- Keep two matrices $A_{2\times 2}(t)$ and $B_{2\times 2}(t)$ (initialised to all ones at $t = 0$).
- Sample a matrix of values $B_{sample}(t)$ such that :

$$B^{ij}_{sample} \sim \beta(A_{ij}(t), B_{ij}(t))$$

- Set $p = I_{B^{00}_{sample}+B^{01}_{sample}-1>0}$ and $q = I_{B^{11}_{sample}+B^{10}_{sample}-1<0}$.
- If reward in time slot t is R_t, then update the A and B matrices as :

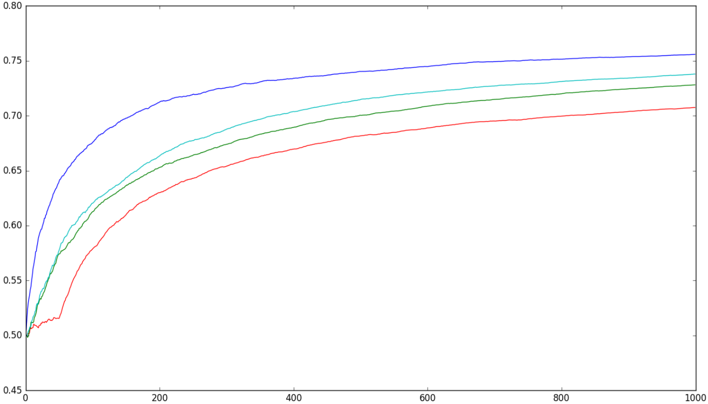
$$A_{ij}(t+1) = A_{ij}(t) + R_t$$

$$B_{ij}(t+1) = A_{ij}(t) + 1 - R_t$$

  where the arm preference i and a arm recommended j in the time slot t.

Results on UCB and Thompson

Figure 7: Proportion of type 0 users for various policies Vs Time (Blue : Optimal, Red : ETC, Green : UCB, Cyan : Thompson)



Conclusion

- Thompson sampling (appropriately tuned) performs better than UCB, which in turn performs better than ETC.
- Need to formally derive regret complexity for UCB and Thompson for comparison
- Need better concentration bounds for general cases
- Extend to multiple ($n > 2$) arms

THANK YOU