Multi-armed Bandits with externalities

VVN

2019

The Model

Reward structure :

- We have a set of N_a arms which give a fixed distribution of rewards if pulled for a user of a particular arm preference.
- The highest value of mean reward is obtained when a user of a particular preference is shown his preferred arm.
- Therefore we have a matrix $M = [\mu_{ij}]_{N_a \times N_a}$ of the mean rewards where in each row i ,the element μ_{ii} has the highest value.
- Also , the element μ_{ij} shows the mean reward obtained if the user with preference for arm i is shown the arm j.
- The rewards obtained when the preference matches the arm shown can take values in the set $\{0, 2\}$.
- The rewards obtained when the preference does not match the arm shown can take values in the set $\{-1, 1\}$.

The Model

Updating user preferences :

- We also have a user preference vector α with N_a components.
- The probability of a user arriving with preference for arm i is given by $\frac{\alpha_i}{\sum_j \alpha_j}$.
- If the arm i gets reward R from a user with preference for arm j, then α_i is incremented R and α_j is decremented by R.
- Thus the rewards and the policy together control the structure of the population α.
- The arm preference of the user is revealed to the recommender after he has chosen the arm to be shown and has obtained the reward.

The Aim

- We define the "best arm" to be the arm $argmax_i(\mu_{ii})$
- We aim to manipulate the population vector α through our policy so that the probability of a user arriving with preference for the best arm is maximized.
- If it is not possible to bring about the desired α, we also provide the minimal conditions on the mean matrix M so that our policy works.
- A secondary aim might be to minimize the cumulative regret accrued in the process of achieving the desired α.

Test policy 1

Our first test policy is to chose all arms uniformly at random. After simulating this policy many times and observing the trend for average α for fixed matrix M, we make the following observations :

- The values of α_i's arrange themselves in the order of the column sums of the matrix M.
- Even if the arm i is the best arm, α_i can drop to 0 very quickly if the i^{th} column sum is lower than that for other arms.
- The α trajectory for such a case is shown in the following figure.

Test policy 1 simulated

- The matrix we used is the following:

$$\begin{bmatrix} 0.7 & 0.1 & 0.3 & 0.3 & 0.4 \\ 0.1 & 0.9 & 0.4 & 0.4 & 0.4 \\ 0.2 & 0.05 & 0.6 & 0.4 & 0.5 \\ 0.6 & 0.2 & 0.2 & 0.65 & 0.2 \\ 0.2 & 0.2 & 0.55 & 0.2 & 0.55 \end{bmatrix}$$

- The color scheme is in the order : (r,g,b,c,m).

Test policy 1 simulated

Figure 1: Uniform policy

Test policy 2

Our second test policy is to chose the same one arm repeatedly.The following were the observations after the simulation:

- If we are repeatedly choosing arm i then α_j keeps reducing if μ_{ji} is positive.
- If an arm j has a large column sum (relative to other arms) has no positive elements in its row, then it is not possible to reduce α_j using any particular arm.

A Working policy

Based on the observations obtained after implementing the previous two policies, we now implement the following algorithm:

- **Step 1 - Estimation phase :**
  - We first need to estimate the matrix M so as to decide the best arm and then start manipulating the α_i's.
  - For the first T' iterations, we select arms uniformly at random and update our estimate $\hat{M}$ for the matrix.
  - The key tradeoff involved here is that if we take too long to estimate the matrices, then the α_i of the best arm may fall to zero by virtue of a low column sum.
  - Therefore choosing an appropriate value of T' is essential, otherwise we would not be able to guess the best arm with surety, or its α_i would just drop down to zero while estimation.
  - We also keep track of the α while doing all this.

A Working policy

- **Step 2 - Manipulation phase :**
  - We now focus on an arm i such that:

$$i = argmax_{j \neq bestarm}(\alpha_j)$$

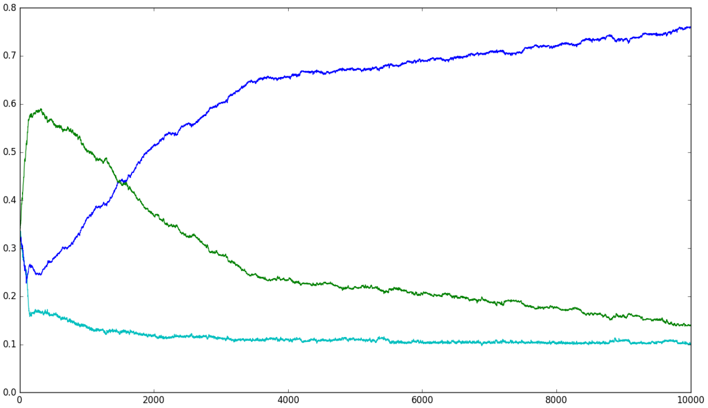
    Call this the target arm. This arm may change each with each iteration.
  - Then we look at the i^{th} row in the estimated matrix $\hat{M}$ and choose the arm k such that:

$$k = argmax_{j \neq bestarm, j \neq i}(\mu_{ij})$$

- **Step 3 : Exploitation phase :**
  - If $\alpha_{bestarm} > 1 - \delta$ then we choose the best arm.

A Working policy : Simulations

Figure 2: Policy with three arms



A Working policy : Simulations

Figure 3: Policy with five arms (averaged over 100 simulations)

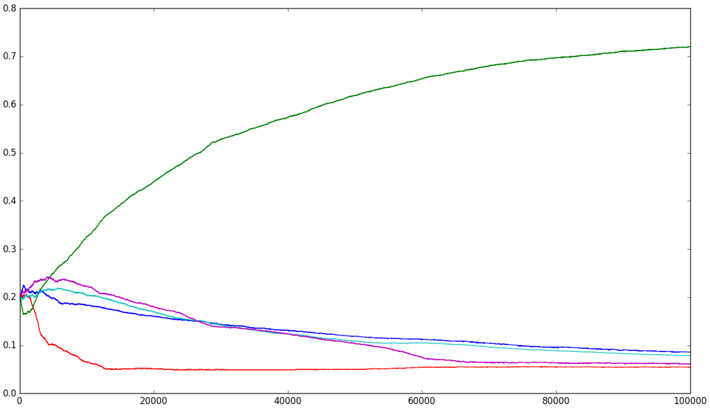
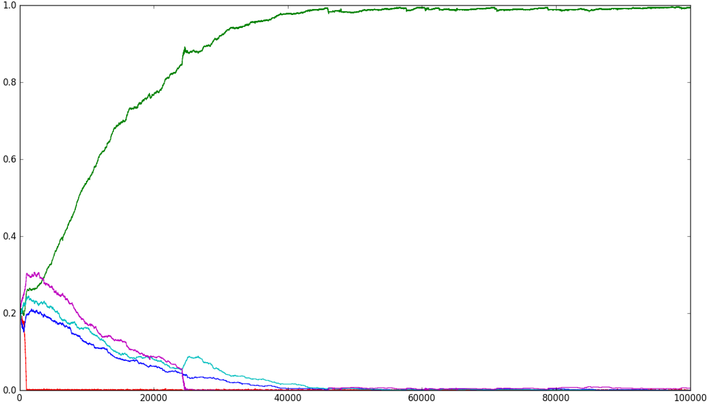


Figure 4: Policy with five arms (one particular simulation)



When does this not work?

- The policy that we discussed above does not work when there exists a row in the matrix M such that there is no positive mean reward in that row except for the diagonal element.
- In such cases there is no way to bring down the α value for this arm.
- Such a case is illustrated in the following simulation.

When does this not work?

Figure 5: Policy with five arms (not working)