**Ad Recommendation using Click Through Rates**

Minimax is a decision rule used in decision theory for minimizing the possible loss for the worst case(maximum loss) scenario. Regret is the negative emotion experienced when learning that an alternative course of action would have resulted in a more favorable outcome. Minimax decision making is based on opportunistic loss. It tries to minimize the maximum regret ( opportunistic loss) observed for all actions and states an action can take.

Lets consider a small scenario for Ads A1 , A2 , A3 with the current state of CTRs as shown:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Action** | A1 | A2 | A3 |
| **State of Nature** |  |  |  |  |
| Ad Show + hit (CTRs increase) |  | 11 | 11 | 7 |
| Ad Show + miss (CTRs decrease) |  | 7 | 6 | 5 |
| We don't show the ad (CTRs remain same) |  | 9 | 10 | 6 |

We then use this table to calculate the regret that we will observe. I have considered pay off directly proportional to CTRs ( the probability of an Ad being clicked if shown in a page view). Since the data provided about ads is not sufficient , it might happen that clicking Ad A1 might give more profit than clicking Ad A2 . So this payoff will accommodate this behavior if required and if the data is available.

The CTRs are being calculated as **[ number of clicks / number of impressions]** for each Ad.)

The payoff is constant for each Ad - **CTRs\*100 (doesn't matter as payoff is constant)**

Since in this case, all Ads are equal, payoff per click is constant for all Ads.

Regret(Opportunistic Loss) = best-payoff (for each state of nature) – payOff Received

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Action** | A1 | A2 | A3 |
| **State of Nature** |  |  |  |  |
| Ad Show + hit |  | 11 -11 = 0 | 11 – 11 = 0 | 11 – 7 = 4 |
| Ad Show + miss |  | 7 – 7 = 0 | **7 – 6 = 1** | 7 – 5 = 2 |
| We don't show the ad (Same as current CTR) |  | **10 – 9 = 1** | 10 – 10 = 0 | **10 – 6 = 4** |

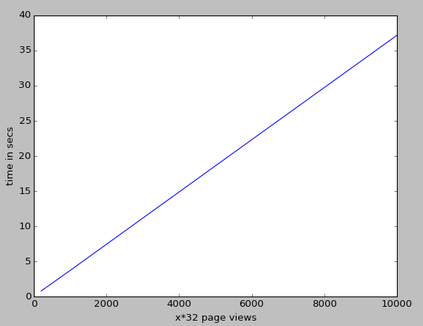
**Maximum Regret for all actions : 1 1 4**

**Minimum of all maximum regrets :**  1 **(Ad A1 and A2 in this example)**

The algorithm as described above considers all the states an action can take . For example , for Ad A1, all states for the next recommendation would be , 1-> we recommend A1 and A1 is actually clicked (CTR increases, payoff increases) , 2-> we recommend A1 and it is not clicked (CTRs decrease) and 3-> not recommend Ad A1 for next page view. It then considers what pay off will each of the action(Ads to show at next page view) we have, give, for all states of nature the action can take. It then finds the maximum regret by choosing the best payoff and minimizing the maximum regret for all options.

**Runtime**

Here is a view of runtime of the algorithm. The runtime is linear to the number of page views.



# 200 \* 32 page views .78 secs

# 500 \* 32 page views 1.88 secs

# 1000 \* 32 page views 3.71 secs

# 5000 \* 32 page views 18.6 secs

# 10000 \* 32 page views 37.18 secs

**Notes**

The first row in the csv file contains contains ad names . I have hard coded the Ad names in the script and skipped the execution of the 1st row ( row\_num = 0). Also , within a row , I have considered the page views randomly.

I am also sending a csv file of a sample run considered for row 100 and column number 13. The recommendation is Ad D3 which seems correct based on the data I have changed in the CSV file. There are more number of 1s in D3 column around row 100.

You can try adding a lot of 1s in one of the columns and you should see the recommendation for that Ad .

**Here are a few more data points:**

- To reiterate, the algorithm's runtime is linear with respect to the number of ads . It can process 3,20,000 (10,000 rows) page views and recommend Ads in 37.16 secs. This can be even reduced further by removing the randomization of columns within a row as I think the randomization does not do any good here.

- Also, to make sure it particularly works for the case of banner blindness, here is what I did

I considered rows 1-200 (200\*32 page views)

-Add lot of 1s in the column for B6 from row 2 to 54.

-Add lot of 1s in the column for D3 from row 31 to 122.

-Add lot of 1s in the column for A8 from row 77 to 200.

There is sufficient overlap between the Ads when popularity of one Ad started dying and other started becoming more popular.

Here is what the recommendation looks like:

|  |  |  |
| --- | --- | --- |
| **Page View** | **Row Num** | *Recommendation* |
| 1-17 | 1 | A2 |
| 18-24 | 1 | A6 |
| 25-2510 | 1-78 | B6 |
| 2511-5255 | 78-164 | D3 |
| 5255-6368 | 164-200 | A8 |

So , the algorithm quickly learns about an Ad becoming 'unpopular' and other becoming 'popular' and recommends that Ad.

You can simulate this behavior by modifying the csv file. Hope this helps.

Also, let me know if you need me to share the csv file with the modifications I made.