Introduction to data science & artificial intelligence (INF7100)

Arthur Charpentier

#123 A/B Testing

été 2020

A/B Testing, Bandits & Reinforcement

A/B Testing

500 users
12% conversion
60 sales
$6,000 sales

500 users
3% conversion
15 sales
$1,500 sales

difference = $4,500 (cost to find the 'good' button)

A/B Testing

$$\begin{cases}\text{first } 10\% & \text{random choice}\\ \text{next } 90\% & \text{best choice}\end{cases}$$

50+900
12% conversion
114 sales
\$11,400 sales

50 users
3% conversion
1.5 sales
\$150 sales

cost to find the 'good' button = (only) \$450

need to be sure that you really have the button after 100 trails...

A/B Testing

$$\begin{cases} \text{first } 10\% & \text{random choice} \\ \text{next } 90\% & \text{best choice but you got it wrong...} \end{cases}$$

50	50+900 users
12% conversion	3% conversion
6 sales	28.5 sales
\$600 sales	\$2,850 sales

cost of not findding the ‘good’ button = \$8,200 !

dilemma : exploration v.s. exploitation

Multi-arm Bandit & Reinforcement Learning

Multi-arm bandit: imaginary slot machine with multiple arms for the customer to choose from, each with different payoffs, (analogy for a multitreatment experiment)
Arm: treatment in an experiment (see button on the webpage).

See also Reinforcement Learning in Economics and Finance for a survey...

Thompson's sampling

Consider k arms, each produces reward $\begin{cases}\$1 \text{ with probability } \theta_j \\ \$0 \text{ probability } 1-\theta_j\end{cases}$

Mean reward of arm j (θ_j) is unknown.

Assume prior distributions $\mathcal{B}(\alpha_j,\beta_j)$ for θ_j

Posterior, at stage t, is also Beta distributed,

$$(\alpha_j,\beta_j) \leftarrow \begin{cases}(\alpha_j,\beta_j) \text{ if } j \text{ was not selected at step } t \\ (\alpha_j,\beta_j)+(r_t,1-r_t) \text{ if } j \text{ was selected at step } t\end{cases}$$

where r_t is the reward obtained at step t ($\in\{0,1\}$).

Thompson's sampling

Algorithm 1: Bernoulli Thompson's sampling

```
initialization : (α_1, β_1), ..., (α_k, β_k);
for t=1,2,... do
    for j=1,2,...,k do
        sample θ_{j,t} ← B(α_j, β_j);
    select j* ← argmax{θ_{j,t}};
    apply arm j* and observe r_t update
      (α_{j*}, β_{j*}) ← (α_{j*} + r_t, β_{j*} + 1 − r_t)
```

Greedy Algorithm

Algorithm 2: Greedy Bernoulli

initialization : $(\alpha_1, \beta_1), ..., (\alpha_k, \beta_k)$;
for *t=1,2,...* **do**
    for *j=1,2,...,k* **do**
        set $\theta_{j,t} \leftarrow \dfrac{\alpha_j}{\alpha_j + \beta_j}$;
    select $j^\star \leftarrow \text{argmax}\{\theta_{j,t}\}$;
    apply arm $j^\star$ and observe r_t update
    $(\alpha_{j^\star}, \beta_{j^\star}) \leftarrow (\alpha_{j^\star} + r_t, \beta_{j^\star} + 1 - r_t)$

see A Tutorial on Thompson Sampling