Aalto-yliopisto
Perustieteiden korkeakoulu

Optimizing budget allocation in creator marketing campaigns

Aleksi Päkkilä

17.06.2024

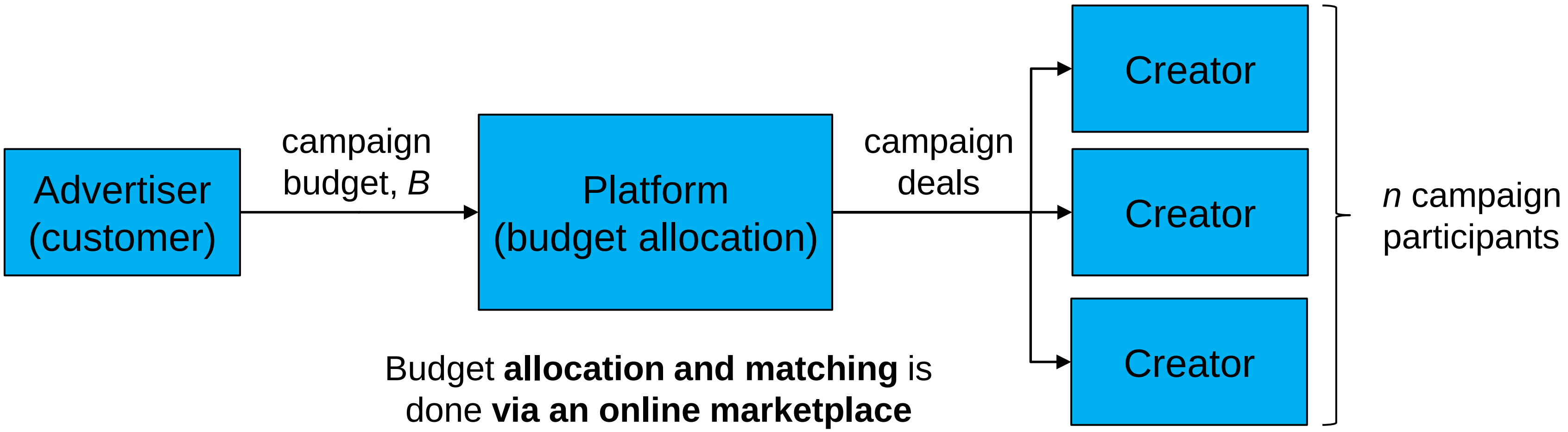
Instructor: *BSc. Leo Lännenmäki*

Supervisor: *Prof. Kai Virtanen*

Systeemianalyysin laboratorio

Objective of the thesis

- Dynamic budget allocation method for creator marketing platform
- Model the budget allocation problem as multi-armed bandit problem
- Solve the problem with BB-MAB-TS algorithm
- Maximizing the campaign performance measured with total views



The budget allocation problem

- Budget *B* allocated into multiple deals
- Deal acceptance rate as a function of price hypothetically S-shaped curve
- Optimal price point for budget allocation maximizes campaign performance measured with the total views of the advertisements

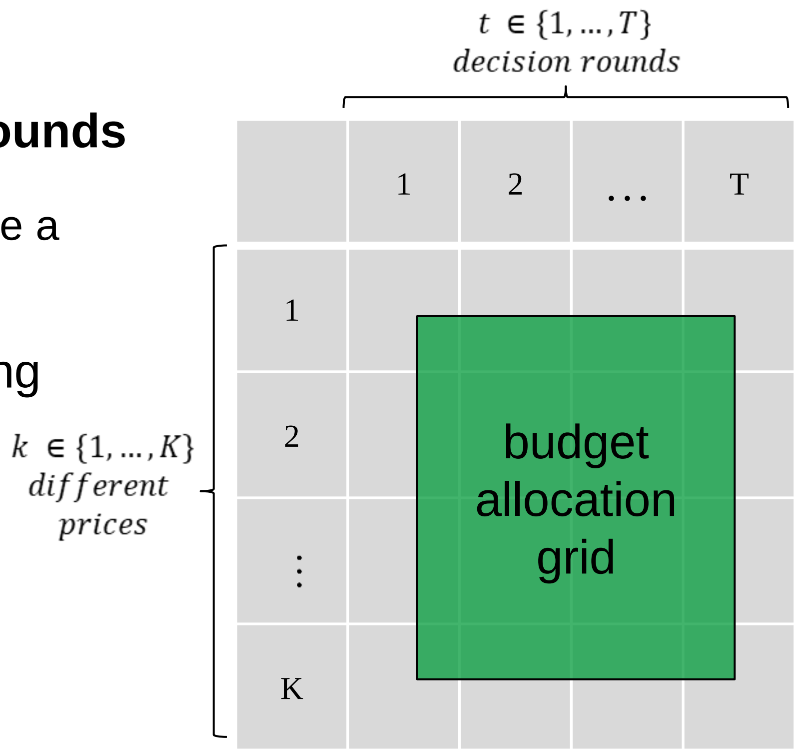
- Objective of optimizing budget allocation:
  1. allocate the budget over multiple decision rounds
  2. learn about acceptance rate with different prices
  3. find the lowest price with which the entire budget can be used

Introduction to multi-armed bandit (MAB)

- Sequential resource allocation problem
- Agent playing slot machine with $k \geq 2$ arms (see pic)
- Pull of an arm leads to a reward
- Reward distribution of different arms unknown
- Agent explores the optimality of different arms
- Objective is to maximize cumulative rewards
- Agent achieves optimal outcome with a decision policy

Budget allocation problem as BB-MAB

- Prices modeled as ***K* arms** of the MAB
- Budget allocation done over ***T* decision rounds**
  - Deals sent during one decision round create a batch
- Agent optimizes budget allocation by finding optimal price over the decision rounds
- **Budget allocation grid** for optimization
- Budget *B* limits the learning process
- Budget allocation problem = **batched & budgeted multi-armed bandit** problem (**BB-MAB**)



Objective of the BB-MAB solution

- Maximize cumulative rewards
  - Observed from reward function *f* after each decision round
- Minimize cumulative regret *r* (= loss due to non-optimal decisions)

$$\max \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{j=1}^{J} f_{kt}(b_{ktj}),$$
$$\text{s.t.} \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{j=1}^{J} b_{ktj} = B.$$

min $$r_t = \sum_{k=1}^{K} \sum_{t=1}^{T} (f_*(b_{kt}) - f_k(b_{kt}))$$

Active learning system set up for solving the BB-MAB

- To find the optimal price point, we need an active learning system (see grid on right)
- Budget is divided into b_t, $t \in \{1, \dots, T\}$ parts
- Allocation between different **prices *k*** across **decision rounds *t*** is determined by **weighs $\boldsymbol{\pi_{kt}}$**
  - $k \in \{1, \dots, K\}$
  - $t \in \{1, \dots, T\}$
- **Decision policy $\boldsymbol{\pi}$** determines the weighs

	1	2	…	T
1	$b_1\pi_{11}$	$b_2\pi_{12}$	…	$b_T\pi_{1T}$
2	$b_1\pi_{21}$	$b_2\pi_{22}$	…	$b_T\pi_{2T}$
⋮	⋮	⋮	⋱	⋮
K	$b_1\pi_{K1}$	$b_2\pi_{K2}$	…	$b_T\pi_{KT}$

Determining the budget allocation weighs with Thompson sampling

- Thompson Sampling = allocating pulls (budget) to different arms (prices) according to the current belief of their optimality

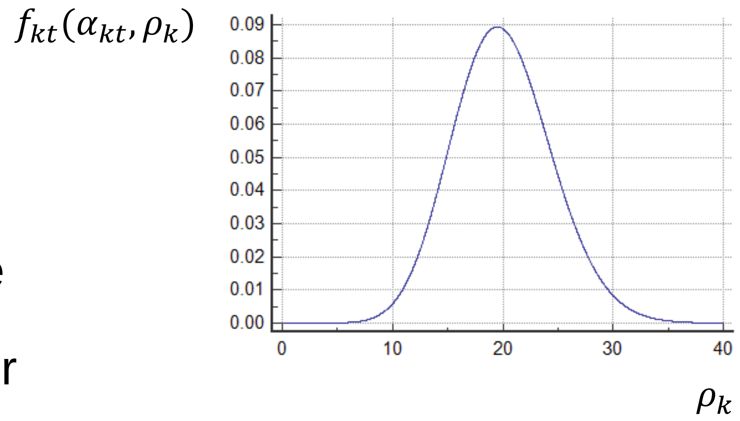
- Rewards are modeled as

$$f_{kt}(\alpha_{kt}, \rho_k) = \frac{\alpha_{kt}}{\rho_k},$$

where α_{kt} is acceptance rate and ρ_k price

- Weighs π_{kt} are sampled from the posterior distribution of the reward function

The posterior distribution is modeled with Poisson distribution



Posterior updating of the reward distribution

- Reward distribution modeled with Poisson distribution
- After each decision round acceptances and declines of deals is observed
  - Data used for posterior updating
- Assuming that acceptances are Poisson distributed: $a_{kt} \sim Pois(\lambda_{kt})$
- λ_{kt} modeled with Gamma conjugate:

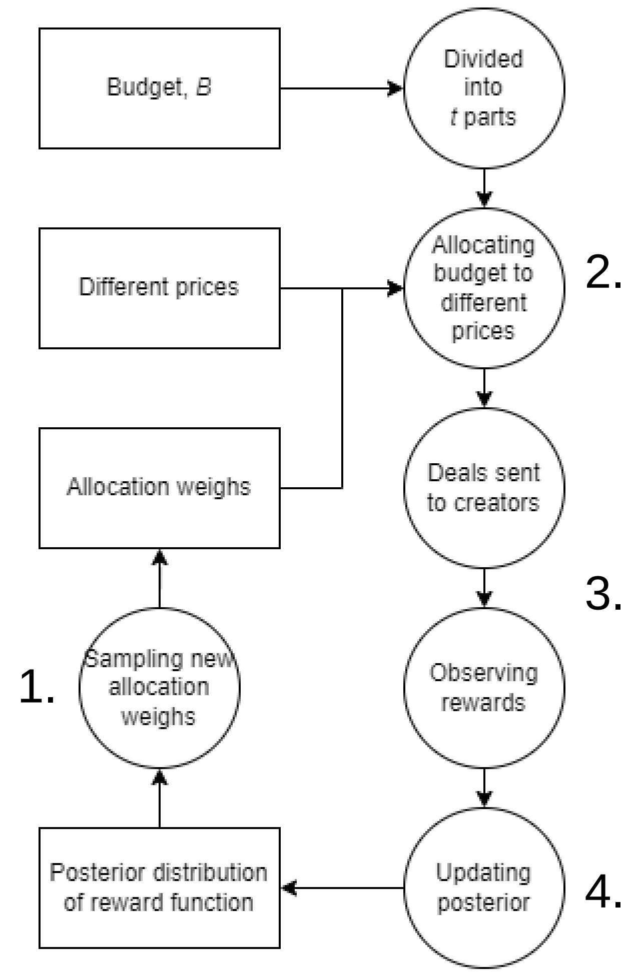
$$\lambda_{kt} \sim Gamma(\alpha_{kt}, \beta_{kt}),$$

where α_{kt} is the historical acceptance data and β_{kt} historical decline data

BB-MAB-TS algorithm as a solution for budget allocation problem

Computational steps of the algorithm at one decision round *t*:

1. Sampling budget allocation weighs π_{kt} from the posterior distribution of the reward function
2. Allocating budget b_t between prices ρ_k according to the weighs π_{kt}
3. Sending deals according to the allocation & observing rewards
4. Updating the posterior distribution of the reward function f_{kt} according to the observed data
5. Repeating the steps 1-4 until all of the budget B is used



Further research topics

- Convergence time to optimal price
- Computational efficiency of the algorithm
- Optimizing budget allocation between the decision rounds
- Optimizing the budget allocation on deal level
- Applying BB-MAB-TS to different resource/budget allocation problems

Conclusion

- Budget allocation problem modeled as a batched & budgeted multi-armed bandit (BB-MAB) problem
- Thompson Sampling based BB-MAB-TS algorithm suggested as the solution for the problem
- The algorithm requires further testing and research on its computational performance

References

Ding, W., Qin, T., Zhang, X., and Liu, T. (2013). Multi-Armed Bandit with Budget Constraint and Variable Costs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 27(1):232–238.

Gao, Z., Han, Y., Ren, Z., and Zhou, Z. (2019) Batched Multi-armed Bandits Problem. http://arxiv.org/abs/1904.01763

Russo, D., Van Roy, B., Kazerouni, A., Osband, I., and Wen, Z. (2018). A Tutorial on Thompson Sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96.

Scott, S. (2010). A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658.

Scott, S. (2015). Multi-armed bandit experiments in the online service economy. *Applied Stochastic Models in Business and Industry*, 31(1):37–45.

Thompson, W. (1933). On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 25(3/4):285–294.