META-LEARNING THE BEST CACHING EXPERT

Maik de Vries M.J.A.deVries-1@student.tudelft.nl

CSE3000 - RESEARCH PROJECT
PROFESSOR: DR. G. IOSIFIDIS
SUPERVISOR: NARAM MHAISEN

[0] BACKGROUND

In the field of computer science, few problems are as common as the caching problem. A **cache** has as its main goal to provide fast access to a size-limited subset of some library. The **problem** arises when it comes to deciding what is to be kept and what is to be removed from such a cache.

Over the years, various caching policies have been proposed, each with its strengths and weaknesses. Their main limitation is the need to know certain properties of the request pattern upfront in deciding the most suitable policy.

In recent years, the problem has been approached anew from an online linear optimisation setting **[1]**. This class of algorithms continuously adapt and learns the optimal decision policy based on the success of their previous decisions, avoiding the need for upfront knowledge.

[1] PROBLEM

These algorithms guarantee an optimal caching policy for any arbitrary request pattern **[1]**. Their performance is primarily affected by the tuning of their **hyperparameters** (e.g. learning rate), which is done under the assumption of **adversarial conditions**. Unfortunately, it remains **unclear how to tune these for non-adversarial request patterns**.

Therefore, the question remains: ***is there an effective way to learn the optimal tuning of such hyperparameters to derive the best caching policy?***

[2] GOALS

- Simulate the **Online Gradient Ascent** (**OGA**) **[1]** algorithm
- Study the effect of **learning rates** on cache performance
- Simulate the **Exponentiated Gradient** (**EG**) **[2]** algorithm
- Use **EG** as **meta-learner** which uses **OGA** as experts

[3] SYSTEM MODEL

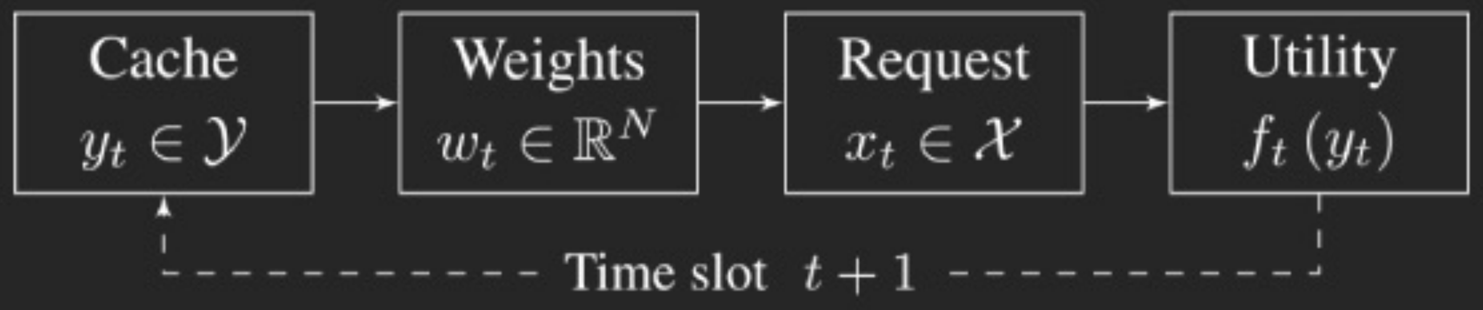

Figure 1: A cache configuration y_t is chosen; an adversary reveals weights w_t; an adversary reveals request x_t; the achieved utility $f_t(y_t)$ is computed; and the next time slot $t+1$ is processed.

The objective is to minimise **regret** of cache configuration y_t compared to the **best static configuration in hindsight** y^*:

$$R_T = \sum_{t=1}^{T} f_t(y^*) - \sum_{t=1}^{T} f_t(y_t)$$

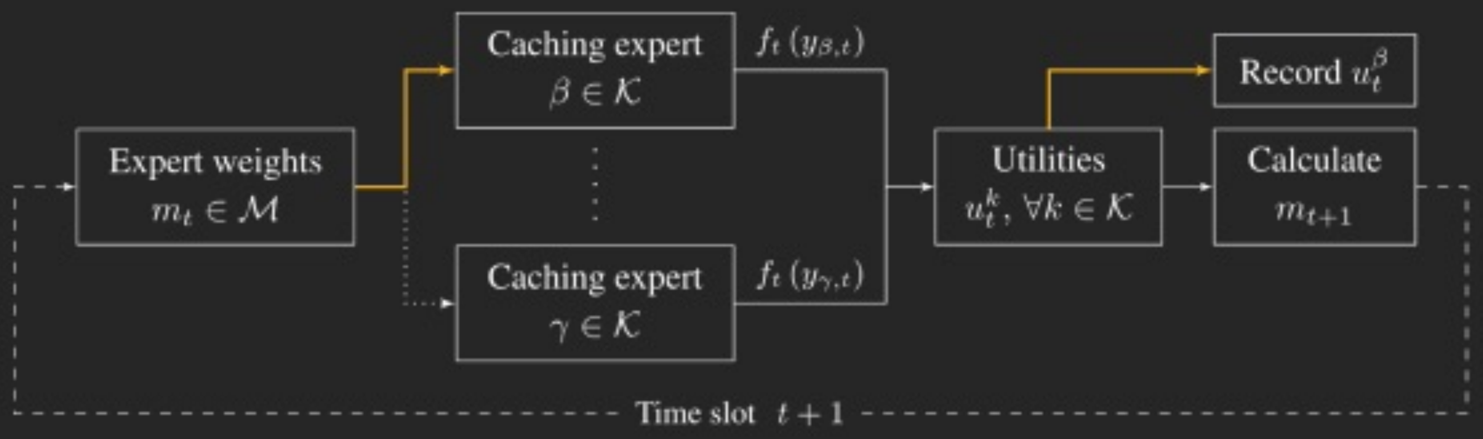

Figure 2: A caching expert β is randomly selected based on expert probability vector m_t; all caching experts process request x_t and output their obtained utilities $f_t(y_{k,t})$; the meta-learner records the utility of the selected expert u_t^β; the expert probability vector m_{t+1} is computed; and the next time slot $t+1$ is processed.

[4] RESULTS

TABLE 1
Performance relative to best-performing caching expert (%)

	Time slot			
ζ	0.25×10^5	0.5×10^5	1.0×10^5	2.0×10^5
0.6	3.49	0.09	−0.98	−0.55
0.8	3.06	0.46	−0.54	−0.36
1.0	3.31	0.94	−0.05	−0.25

Performance comparison between meta-learner σ^* and the best-performing caching expert at various time slots, expressed as a percentage (%) for each Zipfian distributed request model with parameter ζ.

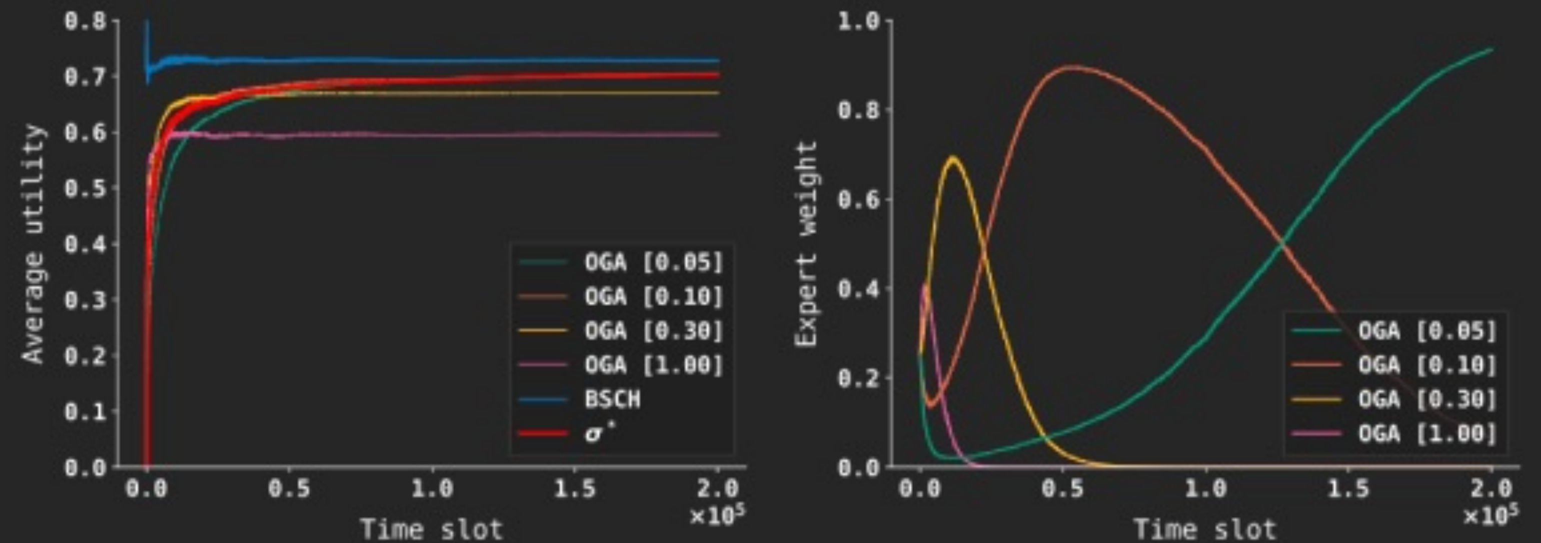

*Figure 3: Zipfian distributed request pattern (ζ = 1.0, N = 2500, C = 250); (a) average utility over time of **meta-learner**, BSCH and various OGA experts; (b) progression of meta-learner **expert weights** over time*

[5] CONCLUSIONS

The online meta-learner caching policy improves **upon the previously** known regret bounds of related works. Furthermore, a consistent improvement in performance is achieved when the individual caching experts' relative performance varies over time.

TU Delft

REFERENCES:
[1] G. PASCHOS, A. DESTOUNIS, L. VIGNERI, G. IOSIFIDIS, "LEARNING TO CACHE WITH NO REGRETS", IEEE INFOCOM, 2019
[2] F. ORABONA, "A MODERN INTRODUCTION TO ONLINE LEARNING", ARXIV:1912.13213, 2023