Optimistic No-regret Algorithms for Discrete Caching

Mhaisen, Naram; Sinha, Abhishek; Paschos, Georgios; Iosifidis, George

DOI
10.1145/3606376.3593561

Publication date
2023

Document Version
Final published version

Published in
Performance Evaluation Review

Citation (APA)
Mhaisen, N., Sinha, A., Paschos, G., & Iosifidis, G. (2023). Optimistic No-regret Algorithms for Discrete Caching. *Performance Evaluation Review*, *51*(1), 69-70. https://doi.org/10.1145/3606376.3593561

Optimistic No-regret Algorithms for Discrete Caching

Naram Mhaisen
Delft University of Technology
Delft, The Netherlands

Abhishek Sinha
Tata Institute of Fundamental Research
Mumbai, India

Georgios Paschos
Amazon
Luxembourg, Luxembourg

George Iosifidis
Delft University of Technology
Delft, The Netherlands

ABSTRACT

We take a systematic look at the problem of storing whole files in a cache with limited capacity in the context of optimistic learning, where the caching policy has access to a prediction oracle. The successive file requests are assumed to be generated by an adversary, and no assumption is made on the accuracy of the oracle. We provide a universal lower bound for prediction-assisted online caching and proceed to design a suite of policies with a range of performance-complexity trade-offs. All proposed policies offer sublinear regret bounds commensurate with the accuracy of the oracle. In this pursuit, we design, to the best of our knowledge, the first optimistic Follow-the-Perturbed leader policy, which generalizes beyond the caching problem. We also study the problem of caching files with different sizes and the bipartite network caching problem.

CCS CONCEPTS

• **Networks → Network performance analysis**.

KEYWORDS

online algorithms; optimistic learning; caching; regret bounds.

ACM Reference Format:
Naram Mhaisen, Abhishek Sinha, Georgios Paschos, and George Iosifidis. 2023. Optimistic No-regret Algorithms for Discrete Caching. In *Abstract Proceedings of the 2023 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '23 Abstracts), June 19–23, 2023, Orlando, FL, USA*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3578338.3593561

1 INTRODUCTION

This extended abstract summarizes the main results of the original paper in [3]. The paper addresses the discrete caching (prefetching) problem: choose files to replicate in a local cache in order to maximize the probability that a new file request is served locally. *Hitting* the cache optimizes user experience in CDN's, and enhances the performance of wireless networks. This work aspires to advance our theoretical understanding of this fundamental problem and proposes new provably-optimal and computationally-efficient caching algorithms using a new approach based on *optimistic learning*.

SIGMETRICS '23 Abstracts, June 19–23, 2023, Orlando, FL, USA

ACM ISBN 979-8-4007-0074-3/23/06.
https://doi.org/10.1145/3578338.3593561

Common caching policies store the newly requested files and employ the Least-Recently-Used (LRU), Least-Frequently-Used (LFU) and other similar rules to evict files when the cache capacity is exhausted. Under certain statistical assumptions on the request trace, such policies maintain the cache at an optimal state. However, with frequent addition of new content to libraries and the high volatility of files popularity, these policies can perform arbitrarily bad. This has spurred research efforts for policies that operate under more general conditions. The goal of this work is to design robust caching policies that are able to learn effective caching decisions with the aid of a prediction oracle of *unknown quality* (Fig. 1 left).

2 MODEL

We formulate the caching problem as an online convex optimization (OCO) problem. At each slot $t = 1, 2, \dots, T$, a learner (the caching policy) selects a caching vector $x_t \in \mathcal{X}$ from the set of admissible cache states $\mathcal{X} \subseteq \{0,1\}^N$ for a cache of size C, where N is the library size. Then, a 1-hot vector $\theta_t \in \{0,1\}^N$ with value 1 for the requested file is revealed, and the learner receives a reward of $f_t(x_t) = \langle \theta_t, x_t \rangle$ for cache hits. The reward is revealed only after committing x_t, which naturally matches the dynamic caching operation where the cached files are decided before the next request arrives. Here, the learner makes no statistical assumptions and θ_t can follow any distribution, even one that is handpicked by an adversary. In the optimistic framework, the learner does not only consider its hit or miss performance so far when deciding x_t, but also the predictor's performance and output (Fig. 1 right). We characterize the policy's performance by using the static *regret* metric:

$$R_T(\{x\}_T) \triangleq \sup_{\{f_t\}_{t=1}^T} \left\{ \sum_{t=1}^T f_t(x^\star) - \sum_{t=1}^T f_t(x_t) \right\},$$

where $x^\star = \text{argmax}_{x \in \mathcal{X}} \sum_{t=1}^T f_t(x)$ is the (typically unknown) *best-in-hindsight* cache decision that can be selected only with access to future requests. An algorithm is said to achieve sublinear regret when its average performance gap R_T/T vanishes as $T \to \infty$.

3 ACHIEVABLE REGRET FOR CACHING WITH A PREDICTOR

Our first result demonstrates the best achievable regret in the setup we consider, which turns out to be $R_T = \Omega([\sum_t \|\theta_t - \tilde{\theta}_t\|]^{1/2})$, indicating a significant potential of obtaining a regret that scales with the predictor's error rather than the time horizon T. In general, the predictions refer to the next function $\tilde{f}_t(\cdot)$. However, since most OCO algorithms learn based on the observed gradients, it suffices to have predictions $\tilde{\theta}_t = \nabla \tilde{f}_t(x_t)$. And for caching, this coincides

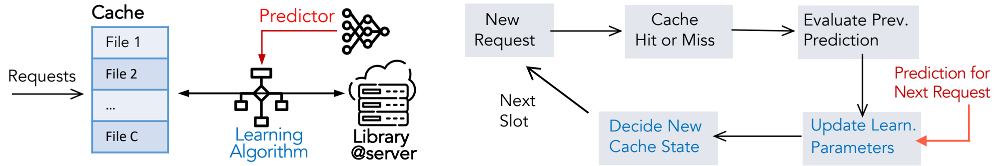

Figure 1: Optimistic online caching with predictions: system schematic (*left*) & algorithm template (*right*).

Algorithm 1: OFTRL-Cache

Input: $\sigma = 1/\sqrt{C}$, $\delta_1 = \|\theta_1 - \tilde{\theta}_1\|_2^2$, $\sigma_1 = \sigma\sqrt{\delta_1}$, $x_1 = \arg\min_{x\in\mathcal{X}}\langle x, \theta_1\rangle$
Output: $\{x_t \in \mathcal{X}\}_T$
for $t = 2, 3 \dots$ **do**
  $\tilde{\theta}_t \leftarrow$ prediction *//Obtain request prediction for slot t*
  $\hat{x}_t = \text{argmax}_{x\in\text{conv}(\mathcal{X})}\{-r_{1:t-1}(x) + \langle x, \Theta_{t-1} + \tilde{\theta}_t\rangle\}$
  $x_t \leftarrow MadowSample(\hat{x}_t)$
  $\Theta_t = \Theta_{t-1} + \theta_t$ *// Receive t-slot request and update total gradient*
  $\sigma_t = \sigma\left(\sqrt{\delta_{1:t}} - \sqrt{\delta_{1:t-1}}\right)$ *// Update the regularization parameter*
end

Algorithm 2: OFTPL-Cache

Input: $\eta_1 = 0$, $y_1 = \arg\min_{y\in\mathcal{X}}\langle y, \theta_1\rangle$
Output: $\{y_t \in \mathcal{X}\}_T$
$\gamma \overset{iid}{\sim} \mathcal{N}(0, \mathbf{1}_{N\times 1})$ *// Sample a perturbation vector*
for $t = 2, 3, \dots$ **do**
  $\tilde{\theta}_t \leftarrow$ prediction *//Obtain request prediction for slot t*
  $\eta_t = \frac{1.3}{\sqrt{C}}\left(\frac{1}{\ln(Ne/C)}\right)^{1/4}\sqrt{\sum_{\tau=1}^{t-1}\|\theta_\tau - \tilde{\theta}_\tau\|_1^2}$ *// Update the pert. param.*
  $y_t = \text{argmax}_{y\in\mathcal{X}}\langle y, \Theta_{t-1} + \tilde{\theta}_t + \eta_t\gamma\rangle$ *// Update the cache vector*
  $\Theta_t = \Theta_{t-1} + \theta_t$ *// Receive request for t and update total gradient*
end

with a prediction for the next request[1]. Here, $\tilde{\theta}_t$ is a probability distribution over the library.

Theorem 1. *For any online caching policy, there exist a sequence of requests $\{\theta_t\}_T$ and predictions $\{\tilde{\theta}_t\}_T$ for which the regret R_T satisfies*

$$\mathbb{E}[R_T] \geq \sqrt{\frac{C}{2\pi}}\sqrt{\sum_{t=1}^{T}||\theta_t - \tilde{\theta}_t||_2^2} - \Theta\left(\frac{1}{\sqrt{T}}\right).$$

4 CACHING THROUGH OPTIMISTIC REGULARIZATION (OFTRL-CACHE)

The gist of our approach here is that we use OFTRL to obtain $\hat{x}_t \in \text{conv}(\mathcal{X})$, and then apply Madow's sampling scheme to recover integral caching vectors $x_t \in \mathcal{X}$ which satisfy the hard capacity non-convex constraint. In other words, we define $\mathcal{X} = \{x \in \{0,1\}^N \mid \sum_{i=1}^N x_i \leq C\}$, where $\mathcal{N}$ is the set of unit-sized files (library) and C is the cache capacity (in file units); and $x_i = 1$ decides to cache file $i \in \mathcal{N}$. Let us define the prediction error at slot t as $\delta_t \triangleq \|\theta_t - \tilde{\theta}_t\|_2^2$, and introduce the proximal σ_t-strongly convex regularizer w.r.t. the Euclidean ℓ_2 norm $r_t(x) = \frac{\sigma_t}{2}\|x - x_t\|_2^2$. Following [2], we define parameters $\{\sigma_t\}_t$ using the accumulated prediction errors, namely:

$$\sigma_1 = \sigma\sqrt{\delta_1},\ \ \sigma_t = \sigma\left(\sqrt{\delta_{1:t}} - \sqrt{\delta_{1:t-1}}\right)\ \forall t \geq 2,\ \text{with}\quad \sigma = 1/\sqrt{C}.$$

The detailed steps are summarized in Algorithm 1. The regret guarantee of Algorithm 1 is described next.

Theorem 2. *Algorithm 1 ensures, for any time horizon T and $N \geq 2C$, the expected regret bound:*

$$\mathbb{E}[R_T] \leq 2\sqrt{C}\sqrt{\sum_{t=1}^{T}||\theta_t - \tilde{\theta}_t||_2^2}$$

Discussion. The bound in Theorem 2 *shrinks* with the prediction quality. If all predictions are accurate, we get $R_T \leq 0$; when all predictions fail, we get $R_T \leq 2\sqrt{2CT}$. That is, in the worst scenario, the regret bound is worse by a constant factor of $\sqrt{2}$ compared to the FTRL algorithm that does not use predictions and ~ 5 compared to the lower bound derived in Sec. 3. Moreover, the bounds are dimension-free and do not depend on the library size N.

[1] In fact this model can be readily generalized to other linear utilities beyond cache-hits, so as to incorporate e.g., file-specific caching gains, time-varying network conditions.

5 CACHING THROUGH OPTIMISTIC PERTURBATIONS (OFTPL-CACHE)

We propose next a new OFTPL algorithm that is of independent interest with potential applications that extend beyond caching to other $k-$set structured problems [1]. The steps of the proposed scheme are presented in Algorithm 2, where we denote the t-slot OFTPL decisions with $y_t \in \mathcal{X}$. The following theorem characterizes the performance of this new OFTPL algorithm.

Theorem 3. *Algorithm 2 ensures, for any time horizon T and $N \geq 2C$ with $C \geq 11$, the expected regret bound:*

$$\mathbb{E}_\gamma[R_T] \leq 3.68\sqrt{C}\left(\ln\frac{Ne}{C}\right)^{1/4}\sqrt{\sum_{t=1}^{T}||\theta_t - \tilde{\theta}_t||_1^2}.$$

Discussion. The regret bound here is also modulated with the quality of predictions: it collapses to zero when predictions are perfect, and maintains $R_T \leq 3.68\ \ln(\frac{Ne}{C})^{1/4}\sqrt{2CT}$ for arbitrary bad predictions. Comparing with Theorem 2, OFTPL achieves regret bounds worse by a factor of $\sim 1.9\ln(\frac{Ne}{C})^{1/4}$, which depends on N, albeit in a small order. However, Algorithm 2 does not involve the expensive projection operation that appears in Algorithm 1, but rather a simple quantile-finding operation (top C files).

REFERENCES

[1] Alon Cohen and Tamir Hazan. 2015. Following the Perturbed Leader for Online Structured Learning. In *Proc. of the ICML*.
[2] Naram Mhaisen, George Iosifidis, and Douglas Leith. 2022. Online Caching with no Regret: Optimistic Learning via Recommendations. arXiv:2202.10590
[3] Naram Mhaisen, Abhishek Sinha, Georgios Paschos, and George Iosifidis. 2022. Optimistic No-Regret Algorithms for Discrete Caching. *Proc. ACM Meas. Anal. Comput. Syst.* 6, 3, Article 48 (2022), 28 pages.