FORECASTING IN ONLINE CACHING

EXPLORATION OF THE EFFECTS OF FORECASTER METHODS ON AN ONLINE LEARNING CACHING POLICY.

Author: Gareth Kit Kye Ler
Contact: GarethKitKyeLer@student.tudelft.nl
Supervisor: Fatih Aslan
Responsible Professor: Georgios Iosifidis

INTRODUCTION

The caching problem: What files need to be stored in the cache such that cache hits are maximized.

Caching is relevant for:
- Computer's CPU
- Content Delivery Networks [1]
- Wireless networks [2]

Common caching policies like LFU, LRU
- Works well in practice.
- Limited to stationary requests and can perform sub-optimally in other contexts [3].

Recent developments focus on **online learning algorithms.**
- Algorithm that dynamically learns best solution.
- Assumes worst-case: adversarial requests.

Optimistic Follow The Regularized Leader (OFTRL) [3] is an online learning algorithm that not only uses historical data but also predictions to expedite its learning.

CONTRIBUTIONS

- For the first time in literature, we implemented real forecasters as part of the optimistic caching policy.
- Conducted analysis and experiments on real datasets to evaluate the benefits of utilizing the forecasters of interest.
- Showed that expedited learning not only depends on forecaster accuracy by also on other implementation details.

REGRET AND OFTRL

Static regret
The metric used to describe the performance of a caching policy. It is considered optimal when regret is $O(\sqrt{T})$.

Static regret is defined as the opportunity cost of utility between the cache states chosen by the caching policy and the best static cache state.

Optimistic Follow the Regularized Leader
OFTRL is an online learning algorithm that uses historical data and future predictions.

When employed as a caching policy, it finds the cache state by maximizing utility using the sum of previous requests and the predicted request.

OFTRL learns faster with more accurate forecasters and has a regret bound dependent on prediction error:

$$R_T \leq 2\sqrt{C}\sqrt{\sum_{t=1}^{T} ||\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}_t||_2^2}$$

- R_T is the regret.
- C is the cache size.
- θ_t is the request vector.
- $\tilde{\theta}_t$ is the prediction vector.
- T is latest time.

FORECASTERS

Recommender Forecaster
Recommender forecaster takes the latest request as input and recommends a file that it considers "similar" to the input.
Similarity based on the utility matrix created by a lagged time series of requests.

TCN Forecaster
TCN forecaster takes a list of requests in vector form from a particular time frame and outputs the next file request

This is analogous to image classification, where TCN receives an image (matrix of past requests) and classifies it (predicts next file).

EXPERIMENT SETUP AND RESULTS

- OFTRL performance is tested with different forecasters on requests extracted from MovieLens Dataset.
- Forecasters 4 - 5 are trained on train and validation set.
- Forecaster 6 trained on train set and tuned with validation set.
- All results obtained from requests in test set.
- Forecasters 5 - 6 are evaluated with different prediction representations (probabilities or one-hot)

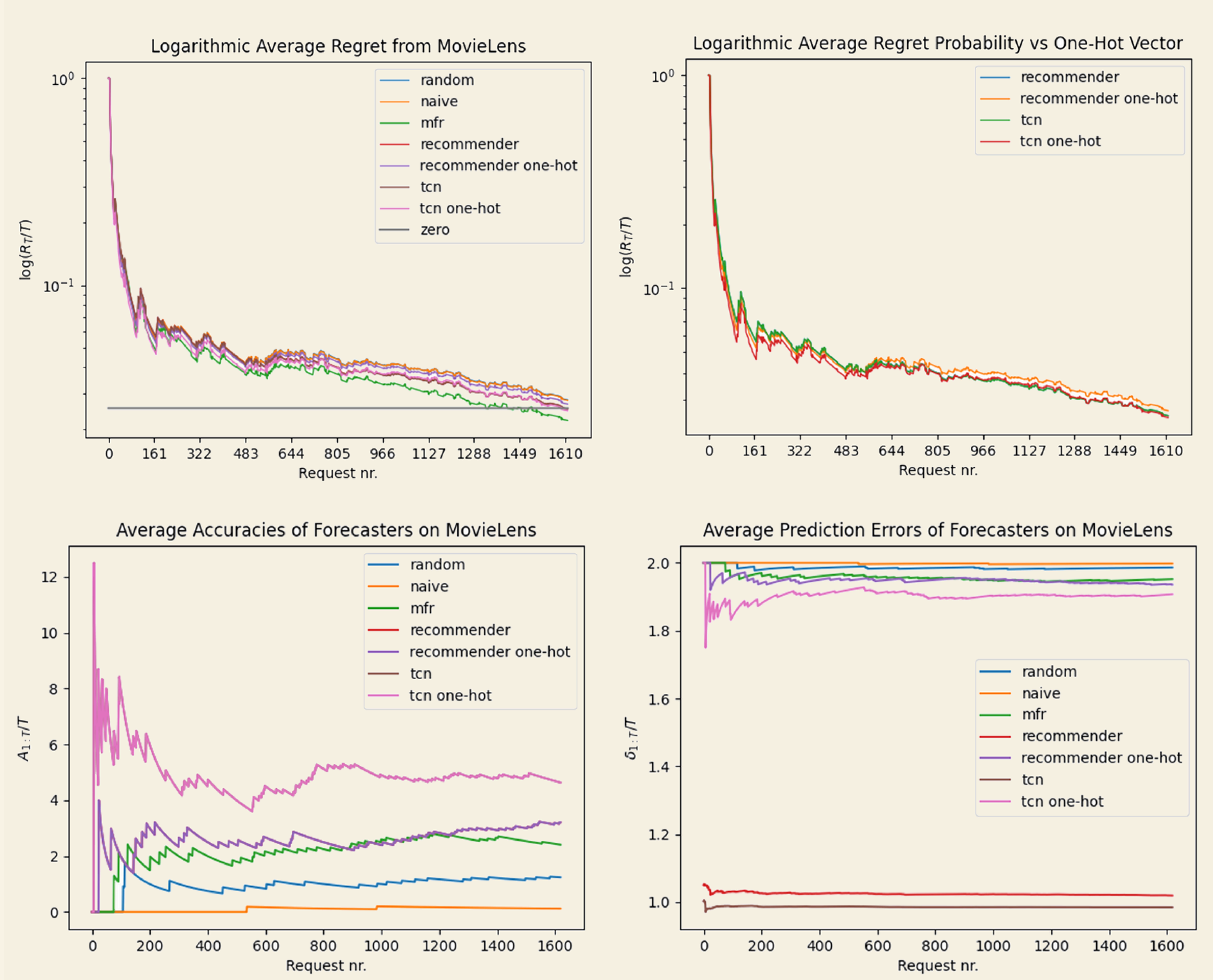
Forecasters:
1. Zero (no predictions)
2. Random
3. Naive
4. Most Frequently Requested
5. KNN Recommender
6. Temporal Convolutional Networks

Dataset
Movielens 10K dataset:
- Filtered to top 100 movies
- 16,000 requests (top 100)
- Train set 80%
- Validation set 10%
- Test set 10%

Metrics:
1. Average static regret over time.
2. Forecaster accuracy.
3. Forecaster accuracy over time.

RESULTS

	Accuracy	Prediction Error
random	0.9265 %	1.98147
naive	0.1235 %	1.99752
mfr	2.4089 %	1.95182
recommender	3.2119 %	1.01934
recommender one-hot	3.2119 %	1.93576
tcn	4.6325 %	0.98457
tcn one-hot	4.6325 %	1.90735



DISCUSSION

Low forecaster accuracy:
- All forecasters have accuracy below 5% with highest being 4.6% from TCN.
- Implies lack of autocorrelation in requests from MovieLens.
- Raises questions about whether MovieLens is suitable to represent content requests for caching research.

Forecasters yield higher regret than OFTRL with no forecaster:
- Low accuracy forecasters can slow OFTRL learning rather than expedite it.
- Raises the question if it is worth training a forecaster with such low accuracies.

MFR outperforms all other forecasters:
- MFR acts similarly to OFTRL. The highly similar function be cause of higher performance.
- Just speculation more research needs to be conducted.

No difference in performance with different prediction representation:
- Different prediction representation as probabilities or one-hot vector has no visible difference in performance. However, their prediction error is vastly different.
- Raises the question if regret bound should depend on prediction error or something else.

FUTURE WORK

- Experiment with different datasets to investigate if low performance is case specific or present for other datasets.
- Create dataset for caching research.
- Further research OFTRL regret bound and how non-one-hot predictions can affect it.
- Look into viability and effectiveness of time series forecasting requests on CDNs.
- Make comparison with popularity-based predictions.
- Extend by asking what would be the most cost-effective forecaster with respect to training resources and effective utility gained.

REFERENCES

[1] G. S. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caire, "The role of caching in future communication systems and networks," IEEE Journal on Selected Areas in Communications, vol. 36, no. 6, pp. 1111–1125, 2018.
[2] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," IEEE Transactions on Information Theory, vol. 59, no. 12, pp. 8402–8413, 2013.
[3] N. Mhaisen, A. Sinha, G. Paschos, and G. Iosifidis, "Optimistic no-regret algorithms for discrete caching," Proceedings of the ACM on Measurement and Analysis of Computing Systems, vol. 6, no. 3, pp. 1–28, 2022.