Critique of "Neural Adaptive Video Streaming with Pensieve [1]"

In this day and age where HTTP-based video streaming volume is exploding across devices and networks of variable conditions and properties, heuristic-centered adaptive bitrate(ABR) streaming algorithms are not sufficient to accurately estimate network throughput and playback buffer occupancy to decide bitrates for future video chunks. The authors here propose a reinforcement learning(RL) approach - Pensieve - to optimize video streaming performances across a broad set of quality objectives and network conditions. The inability of heuristic-based ABR algorithms to handle variability of network throughput, conflicting video quality of experience(QoE) requirements, cascading bitrate decision effects, and, compromised smoothness due to the coarse-grained nature of bitrate selection, is the driving motivation for the development discussed in this paper. Using a reward-based learning approach eliminates the need for pre-programmed models and assumptions (which require significant tuning and are usually not generalizable across networks) by leveraging knowledge solely from observations of performance from past bitrate decisions. The paper further reinforces its motivation by showing how the state-of-the-art robustMPC [4], a heuristic ABR algorithm, fails to account for throughput fluctuations and prediction errors, and also fails to look far enough into the future to decide on an optimal bitrate value. Pensieve, on the other hand, is shown to be flexible enough to handle such difficulties of varying nature to output bitrate decisions that result in better QoE values.

Reinforcement learning algorithms rely on a learning agent that explores the state-action space to continually learn good decisions from bad ones with motivation from reward signals. Pensieve uses the *asynchronous advantage actor-critic(A3C)* [2] model to learn its control policy. A3C is the state-of-the-art RL algorithm that spawns multiple learning agents in parallel to speed up the learning process and expand the exploration space spanned. Due to its asynchronous nature, A3C does not need to maintain an experience relay like traditional RL algorithms - thereby saving on memory resources and compute time [2]. This asynchronism also helps in the application of video streaming where multiple users may concurrently send their video quality feedback to the central learning agent for online training. Actor-critic models in deep RL employ two different neural networks - an actor that learns the actual control policy, and, a critic that helps the actor learn the control policy. The actor takes as input a representation of the current state that the learner agent observes and outputs a probability distribution over the action space - *the policy*. RL algorithms that directly estimate a policy from input states employ the *policy gradient* method which updates the network weights in the direction that maximizes the expected total reward that can be earned by following the current policy from the current input state. The gradient of the expected cumulative reward gained on taking action a_t is estimated as the log-probability of choosing a_t when in state s_t scaled by a factor known as the *advantage*. The advantage reflects how much better choosing a_t is compared to taking the average action that is sampled from the current policy. Intuitively, the advantage moves the network weights more in the direction of actions that empirically lead to better QoE measurements. This is where the critic network comes in. The critic takes the same input as actor but outputs the expected cumulative reward that can be achieved by taking actions sampled from the current policy(the state-value). This value is then used to calculate the advantage which in turn updates the weights of the actor network. Once training is complete the critic is no

longer required and only the actor is used to make bitrate decisions. A3C is further enhanced by regularizing the actor update with an entropy measure which essentially moves the actor weights in the direction of higher entropy to encourage exploration of the state-space. The contribution of this regularizer is set to high at the start of the training phase and is gradually lowered over time.

Pensieve represents the state of its learning agents as a combination of past network throughput measurements, past chunk download durations, next chunk sizes to choose from, current playback buffer size, number of remaining video chunks, and the bitrate at which the last video chunk was downloaded. These numbers are fed into the actor and critic networks to obtain a distribution over next bitrates to choose from (i.e. the policy), and, the expected cumulative QoE achievable on choosing bitrates sampled from this distribution (i.e. the state-value) respectively. Asynchronism is achieved by running 16 such agents in parallel that continuously send their {state, bitrate, QoE} tuples to a central agent which updates the actor and the critic networks. Since, videos can be encoded at different bitrate levels, Pensieve tweaks the input and output structure of the actor-critic networks to result in a single model that can take in variable-sized inputs and produce variable-sized set of outputs. The features that will differ across videos is the list of available bitrates for the next chunk and therefore the list of next video chunk sizes to choose from. Pensieve gets around this problem by mapping each available bitrate value to the nearest bitrate value from a list of canonical values that span the range of bit rates expected to be seen in practice. The resulting array after this mapping is then fed into the network for training. How they come up with the list of canonical values is not mentioned, instead the authors just give an example that the DASH reference client video list can be covered by a range of 13 bitrate levels.

Pensieve has been thoroughly evaluated across a broad spectrum of experiments that, 1) compare it to state-of-the-art ABR algorithms in terms of different QoE measures, 2) test if Pensieve if generalizable to unseen network conditions, and, 3) check its sensitivity to neural network parameters and latency between video clients and ABR servers. All experiments are performed on data collected from two public datasets: a broadband dataset provided by the FCC and a 3G/HSDPA mobile dataset from Norway. 80% of the entire dataset is used for training the Pensieve neural network, while the remaining 20% is used as a test for all ABR algorithms. Pensieve is compared to five ABR algorithms which include one buffer-based method, one rate-based method, BOLA [3] which is a buffer-based method that performs an optimization step, MPC [4] which takes into account both buffer size and throughput estimates, and, robustMPC [4] which accounts for errors seen by MPC by normalizing predictions with the maximum error encountered in the last five chunks. The QoE metric used is one that favours high perceived video quality and penalizes both rebuffering time and abrupt changes in quality to favour smoothness. The perceived video quality term is varied across a linear metric, a logarithmic metric, and, one that favours HD videos over anything else.

For each dataset, the authors present graphs showing the performance of all six algorithms across the three different QoE metrics. The QoE achieved by Pensieve beats all other QoEs by significant margins (12.1% - 24.6%). The paper plots the cumulative distribution functions of average QoE and there Pensieve performs second only to the offline optimal scheme, which has complete future

throughput information and is therefore considered as the unobtainable upper bound on QoE measures. These figures show that heuristic-based ABR algorithms are unable to optimize bitrate decisions for varying QoE objectives as they follow fixed control laws whereas Pensieve learns its control laws with respect to the QoE objective to be optimized. A lot of Pensieve's performance gains come from its ability to minimize re-buffering, which are shown in figures that plot the individual components of all three QoE metrics. Next, the authors test Pensieve's generalizing capabilities by testing it in the wild on two real world networks and one purely synthetic dataset. Even here Pensieve outperforms other ABR algorithms. Finally, Pensieve is benchmarked by comparisons with tabular RL bitrate choosing algorithms, across varying neural network parameters like number of neurons and hidden layers, and, across networks with different latency between video clients and ABR servers. It is clear that the authors have left no stone unturned to prove, via experimentation, that Pensieve is indeed a better adaptive bitrate algorithm than every other kind of ABR algorithm proposed in the past.

A few interesting directions for future work could be to explore:

1. To what extent can the gap between Pensieve and the offline omniscient scheme be closed,
2. Deploying Pensieve directly on client video players instead of ABR servers using compressed neural networks, and how will this affect its performance,
3. If online training is an option to deal with the long offline training time and, to adapt to changing network conditions as new data arrives. If so, how can one deal with the computational overhead that online training will have on the client, how to modify Pensieve so that it can learn good policies from smaller amounts of data, and how to determine when and how frequently must a deployed model be retrained.

The paper does a good job in convincing how neural architectures for adaptive video streaming algorithms have the upper hand over traditional heuristic-based adaptive video streaming algorithms. It modifies and applies the state-of-the-art reinforcement learning method to run a wide number of experiments to prove the above and discusses future directions, some of which intersect with current machine learning research problems (training on resource-constrained devices, compressed architectures etc). In terms of application this work is significant to both multimedia content providers as well as multimedia telecommunication service providers. The paper is highly descriptive, thorough and fits in well in the domains of both networking and applied machine learning research.

References

[1] Mao, Hongzi, Ravi Netravali, and Mohammad Alizadeh. "Neural adaptive video streaming with pensieve." *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 2017.

[2] Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." *International Conference on Machine Learning*. 2016.

[3] Spiteri, Kevin, Rahul Urgaonkar, and Ramesh K. Sitaraman. "BOLA: Near-optimal bitrate adaptation for online videos." *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*. IEEE, 2016.

[4] Yin, Xiaoqi, et al. "A control-theoretic approach for dynamic adaptive video streaming over HTTP." *ACM SIGCOMM Computer Communication Review*. Vol. 45. No. 4. ACM, 2015.