深圳大学研究生课程论文
专业英语

题目 Title A Survey on Bitrate Adaptation algorithm using machine learning

成绩 Final Score：

专业 Major： software engineering

课程名称、代码 Course Code： 2706011、专业英语

年级 Grade： 1st year graduate

姓名（Name） Sun wenbin

学 号 Student ID 1810273007

时间 Date： 2019 年 6 月

任课教师 ： Basker George

评分项	Title Page/ Abstract / Report Format (15%)	TOC/ Section/Citation /Acknowledgement (15%)	Discussion/ Method/ Analysis/ User Research Data (40%)	Conclusion/ Recommendation (10%)	Defense/ Oral Presentation (20%)
得分					
评分人	Basker George				
教师评语：					

一、修改后的最终稿 Your Final Version：

A Survey on Bitrate Adaptation algorithm using machine learning

Wenbin Sun

***Abstract*—Optimizing the user's experience when watching video is a very important issue. Adaptive bitrate(ABR) algorithm deployed on the client can improve the user's quality of experience (QoE). The traditional heuristic ABR algorithm may perform poor results for network bandwidth variety during video transmission. This paper is a summary of the application of machine learning in the bitrate adaptive algorithm. The purpose of combining machine learning with ABR algorithm is to ensure high quality experience for users in the case of bandwidth fluctuations. The experimental results show that the machine learning-based ABR algorithm perform better than the traditional heuristic algorithm, and make user's QoE higher.**

***Index Terms*—adaptive bitrate(ABR), video streaming, reinforcement learning**

I. Introduction

Internet video forms a major fraction of Internet traffic today , and delivering high quality of experience (QoE) is critical since it correlates with user engagement and revenue [9, 2, 3]. To deliver high quality video across diverse network conditions, most Internet video delivery uses adaptive bitrate (ABR) algorithms [12], combined with HTTP chunk-based streaming protocols (e.g., Apple's HTTP Live Streaming, Adobe's HTTP Dynamic Streaming). ABR algorithms (a) chop a video into chunks, each of which is encoded at a range of bitrates (or qualities); and (b) choose which bitrate level to fetch a chunk at based on conditions such as the amount of video the client has buffered and the recent throughput achieved by the client. Within this general frame work, ABR algorithms differ in how bitrate level selection decisions are made, and these decisions impact metrics such as the average bitrate or the rebuffering ratio. We call these QoE metrics, because they have been shown to correlate well with QoE [13].

The majority of traditional ABR algorithms develop fixed control rules for making bitrate decisions based on estimated network throughput ("rate-based" algorithms [14],[15]), playback buffer size ("buffer-based" schemes [16],[17]), or a combination of the two signals [18]. These schemes require significant tuning and do not generalize to different network conditions and QoE objectives. In this paper, we conclude three solutions which use state-of-the-art machine learning algorithm to optimize ABR algorithm, these solutions always perform better than traditional ABR algorithm. Pensieve [19] was proposed to improve accuracy and speed of bitrate decision estimations using Deep Reinforcement Learning (Deep-RL). Pensieve is a framework that is built based on observations collected by DASH clients (i.e., throughput estimation and buffer occupancy) across large video streaming experiments. It does not rely on pre-programmed models or assumptions about the environment, but, in fact, gradually learns the best policy for bitrate decisions through observation and experience. NAS [20] that explored a similar quality-enhancing problem but in the context of on-demand video streaming. NAS aims to learn a mapping between the low-quality and the high quality versions of video streaming, and uses the spare compute resources at the receiver to enhance the video quality at run time. Nas presents a new video delivery framework that utilizes client computation and recent advances in deep neural networks (DNNs) to reduce the dependency for delivering high-quality video. The use of DNNs enables it to enhance the video quality independent to the available bandwidth. Akhtar *et al.* designed Oboe [21] which allows the automatic tuning of configuration parameters to different network conditions for an ABR scheme. Consequently, these configuration parameters are applied at run-time to match the current network state. The proposed system significantly improves the bitrate decision of clientbased adaptation schemes like BOLA [23] and FastMPC [24], and it offers a 24% on average better viewer QoE compared to Pensieve.

II. BACKGROUND

A. Reinforcement learning

Reinforcement learning, including Q learning, has been integrated with advanced machine learning techniques to tackle difficult high-dimensional problems [13]–[15]. In 2013, Google DeepMind used a deep neural network, called DQN,

to approximate Q values in Q learning that overcomes the limitation of the traditional look-up table approach, and provide an end-to-end approach to allow an agent to learn a policy from its observations.

An RL agent learns through interacting with its environment over time. At each time step t, the agent observes a state s_t in a state space S about its environment, and chooses an action a_t from an action space A following a behavior policy π = $P(a_t|s_t)$, which is a mapping from state s_t to a probability of choosing action a_t. Then the agent obtains a reward r_t and transitions to a new state s_{t+1}, according to the environment dynamics, or model, for reward function R(s,a) and state transition probability $P(s_{t+1}|s_t, a_t)$ respectively. The accumulated reward is defined as return $\mathrm{R}_{\mathrm{t}} = \sum_{t=0}^{\infty} \gamma^{\mathrm{t}} r_{\mathrm{t}}$ with a discount factor $\gamma \in (0,1]$ .The goal of learning is to maximize the expected cumulative discounted reward.

The goal of the agent is to find an optimal policy π , which achieves the maximum expected return from all states. The state-value function $V^{\pi}(s,a) = E[R_t | s_t = s]$ and the action-value function $\mathrm{Q}^{\pi}(s,a) = E[R_t | s_t = s, a_t = a]$ can measure how good π is. $\mathrm{V}^{\pi}(s)$ represents the expected return for following policy π from state s, and $\mathrm{Q}^{\pi}(s,a)$ represents the expected return for selecting initial action a in state s and then following π . The advantage function $\mathrm{A}^{\pi}(s,a)=Q^{\pi}(s,a)-V^{\pi}(s,a)$ represents the relative advantage of actions.

B. Bitrate adaptation algorithm

Adaptive streaming (e.g., Apples HLS [10], DASH [11]) is designed to handle unpredictable bandwidth variations in the real world. Video is encoded into various bitrates (or resolutions) and divided into fixed length chunks, typically 2 – 10 seconds. An adaptive bitrate algorithm (ABR) decides the bitrate for each video chunk. Traditional ABR algorithms select bitrates using heuristics based on the estimated network bandwidth and/or the current size of the client-side playback buffer. Pensieve select the bitrates by using RL and outcome than heuristics-based approaches.

Figure 1 summarizes how RL can be applied to bitrate adaptation. As shown, the decision policy guiding the ABR algorithm is not handcrafted. Instead, it is derived from training a neural network. The ABR agent observes a set of metrics including the client playback buffer occupancy, past bitrate decisions, and several raw network signals (e.g., throughput measurements) and feeds these values to the neural network, which outputs the action, i.e., the bitrate to use for the next chunk. The resulting QoE is then observed and passed back to the ABR agent as a reward. The agent uses the reward information to train and improve its neural network model.

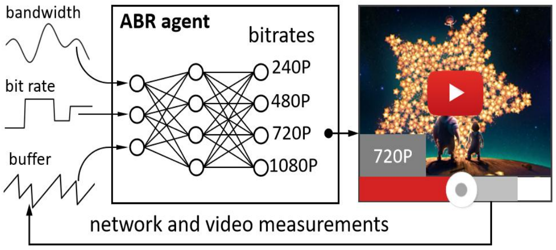

Figure 1: Applying reinforcement learning to bitrate adaptation.

III. ML-BASED BITRATE ADAPTION

In this section, I will conclude three unique solutions about using machine learning such as reinforcement learning and deep learning on bitrate adaptation algorithm.

A. Pensieve

Traditional ABR algorithms select bitrates using heuristics based on the estimated network bandwidth and/or the current size of the client-side playback buffer, it’s different from heuristic algorithm, Pensieve, learns ABR algorithms automatically, without using any pre-programmed control rules or explicit assumptions about the operating environment. Pensieve uses modern reinforcement learning (RL) techniques [24, 25, 26] to learn a control policy for bitrate adaptation purely through experience. During training, Pensieve starts knowing nothing about the task at hand. It then gradually learns to make better ABR decisions through reinforcement, in the form of reward signals that reflect video QoE for past decisions.

Pensieve, a system that optimize ABR algorithms using reinforcement learning (RL). Pensieve trains a neural network model that selects bitrates for future video chunks based on observations collected by client video players. Pensieve does not rely on pre-programmed models or assumptions about the environment. Instead, it learns to make ABR decisions solely through observations of the resulting performance of past decisions. As a result, Pensieve automatically learns ABR algorithms that adapt to a wide range of environments and QoE metrics. In all considered scenarios, Pensieve outperforms the best state-of-the-art scheme, with improvements in average QoE of 12%–25%. Pensieve demonstrate that directly optimizing for the desired QoE objective delivers better outcomes than heuristics-based approaches. In particular, Pensieve uses deep reinforcement learning and learns through "observations" how past decisions and the current state impact the video quality. It is the first time that apply RL to bitrate adaptation algorithm.

B. NAS

Inspired by the ever-increasing clients' computational power and recent advances in deep learning, NAS identifies an alternative and complementary approach to enhancing the video quality. We apply a deep neural network (DNN)-based quality enhancement on video content utilizing the client computation to maximize user QoE. In particular, a deep learning model learns a mapping from a low-quality video to a high-quality version, e.g., super resolution. This enables clients to obtain high-definition (e.g., 1080p) video from lower quality transmissions, providing a powerful mechanism for QoE maximization on top of bitrate adaption.

NAS, the first video delivery frame work that applies DNNs on video content using client's computational power to maximize user QoE. To guarantee reliable quality enhancement powered by DNN, it takes a content-aware approach in which a DNN is trained for each content separately. The idea is to leverage the DNN's overfitting property and use the training accuracy to deliver predictable high performance, instead of relying on the unpredictable test accuracy. Next, to meet the real-time constraints on heterogeneous environments, NAS use multiple scalable DNNs that provide anytime prediction [27, 28]. Such DNN architectures can adaptively control their computational cost given resource budget. NAS clients choose a DNN (from multiple options) that best fits their resources and adapt to temporal variations in computing power at each time epoch. The scalable DNN also enables the use of a partially downloaded DNN, bringing an incremental benefit in downloading a DNN model. Finally, to reconcile the ABR-based QoE optimization and DNN-based quality enhancement, NAS devise a content enhancement-aware ABR algorithm for QoE optimization. To this end, it integrate the design into the state-of-the-art ABR algorithm [29] that uses reinforcement learning [30]. The algorithm decides when to download a DNN model and which video bitrate to use for each video chunk.

C. Oboe

State-of-the-art ABR algorithms like BOLA and MPC rely on parameters that are sensitive to network conditions, so may perform poorly for some users and/or videos. In this paper, we propose a technique called Oboe to auto-tune these parameters to different network conditions. Oboe pre-computes, for a given ABR algorithm, the best possible parameters for different network conditions, then dynamically adapts the parameters at run-time for the current network conditions. Using testbed experiments, it shows that Oboe significantly improves BOLA,
MPC, and a commercially deployed ABR. Oboe also betters a recently proposed reinforcement learning based ABR, Pensieve, by 24% on average on a composite QoE metric, in part because it

is able to better specialize ABR behavior across different network states.

Oboe aims to ensure good QoE for all users by enabling ABR algorithms to perform better across a wide range of network conditions. The configurations of many ABR algorithms are sensitive to network state, specifically to the value and variability of the available throughput between the client and the video server. For example, β in MPC should be smaller when available throughput is highly variable, while in BOLA should be higher. This explains why the algorithms perform differently for different values of parameters on a given client trace. However, a line of prior work [4, 5, 6, 7, 8] has observed that network connections are piecewise stationary: that is, connections can be in one of several distinct states, where each state is distinguished by stationarity in the statistical sense (informally, a process is stationary if its statistical properties including mean and variance do not change over time - see for a more formal definition). Oboe leverages the piecewise stationarity of network connections to address the key challenge of sensitivity of configurations to network conditions. It does so using a two stage design: (a) an offline stage where it pre-computes the best configuration choice for each (stationary) network state, and (b) and an online stage, where during a session, it detects changes in network state and applies the pre-computed best configuration for the current (stationary) state. Oboe can also accommodate publisher specifications such as session type (live vs. video-on-demand, time-to-live requirements), bitrate levels or any explicit QoE tradeoffs (e.g.,preference between rebuffering and average bitrate) , by using these to influence the selection of the best configuration for each (stationary) network state in the offline stage.

IV. DISCUSSION

Deploying Pensieve in practice: In author's current implementation, Pensieve's ABR server runs on the server-side of video streaming applications. This approach offers several advantages over deployment in client video players. First, a variety of client-side devices are used for video streaming today, ranging from multi-core desktop machines to mobile devices to TVs. By using an ABR server to simply guide client bitrate selection, Pensieve can easily support this broad range of video clients without modifications that may sacrifice performance. Additionally, ABR algorithms are traditionally deployed on clients which can quickly react to changing environments.

Nas uses the "super-resolution" technology to map low-quality video frames to high-quality video frames using deep neural networks. The advantage of it is that we can transmit a low-quality video on the server side, and using the client's computing resources， it is converted into a high-quality video, which saves a lot of bandwidth, but there are two problems about it. First, it needs high requirements for the computing resources of client. For example, the mobile client don't have a powerful compute resource, so the result of video recovery on the mobile client may perform poor. Secondly during model training, a model can only adapt to a video clip with a relatively high content relevance. If the content changes a lot, then it is necessary to retrain a new model. This is also don't work.

Oboe improves the performance for most but not all sessions relative to the ABR algorithm it tunes. For instance, MPC+Oboe typically improves performance relative to RobustMPC by reducing rebuffering and/or the magnitude of bitrate changes, but at the expense of slightly lower bitrates. The resulting QoE is improved for most sessions, indicating Oboe does a good job of properly balancing the various factors, but some sessions see lower QoE. More generally, designing an ABR approach that can optimize the performance of all sessions is a hard problem that needs more research.

V. CONCLUSION

This paper mainly introduces three state-of-the-art optimization methods for ABR algorithm. The traditional ABR algorithm is heuristic. It uses a fixed algorithm to control the choice of bitrate, but the bandwidth fluctuation in the network is very large, so we need better solutions to optimize the ABR algorithm, Pensieve, a system which generates ABR algorithms using reinforcement learning. Unlike ABR algorithms that use fixed heuristics or inaccurate system models, Pensieve's ABR algorithm is generated using observations of the resulting performance of past decisions across a large number of video streaming experiments. This allows Pensieve to optimize its policy for different network characteristics and QoE metrics directly from experience. Over a broad set of network conditions and QoE metrics, the result shows that Pensieve outperformed existing ABR algorithms by 12% – 25%. NAS, a video delivery system that utilizes client computation to enhance the video quality. Unlike existing video delivery that solely relies on the bandwidth resource, NAS uses client-side computation powered by deep neural networks (DNNs). NAS introduces new system designs to address practical problems in realizing the vision on top of DASH. the evaluation over real videos on real network traces shows NAS delivers improvement between 21.89 – 76.04% in user quality of experience (QoE) over the current state of the art. Oboe is a system for automatically tuning ABR algorithms by adapting ABR configurations in real-time to match the current network state. Picking configurations in a manner informed by network state and publisher preferences distinguishes Oboe's approach from heuristics used today that do not consider these factors. Oboe significantly improves the performance of BOLA, HYB and RobustMPC, further, for nearly 80% of the sessions in their dataset, Oboe integrated with RobustMPC improves QoE relative to Pensieve and the improvements exceed 20% for 25% of the sessions.

REFERENCES

[1] Maxim Claeys, Steven Latré, Jeroen Famaey, Tingyao Wu, Werner Van Leekwijck, and Filip De Turck. Design and Optimisation of a (FA)Q-learning-based HTTP Adaptive Streaming Client. Connection Science, 26(1):25–43, 2014.

[2] Florin Dobrian, Vyas Sekar, Asad Awan, Ion Stoica, Dilip Joseph, Aditya Ganjam, Jibin Zhan, and Hui Zhang. Understanding the Impact of Video Quality on User Engagement. In Proceedings of the ACM Conference on Special Interest Group on Data Communication, SIGCOMM, 2011.

[3] Te-Yuan Huang, Nikhil Handigol, Brandon Heller, Nick McKeown, and Ramesh Johari. Confused, Timid, and Unstable: Picking a Video Streaming Rate is Hard. In Proceedings of the ACM Conference on Internet Measurement Conference, IMC, 2012.

[4] Hari Balakrishnan, Mark Stemm, Srinivasan Seshan, and Randy H Katz. Analyzing Stability in Wide-area Network Performance. ACM SIGMETRICS Performance Evaluation Review, 25:2–12, 1997.

[5] James Jobin, Michalis Faloutsos, Satish K Tripathi, and Srikanth V Krishnamurthy. Understanding the Effects of Hotspots in Wireless Cellular Networks. In Proceedings of the Conference of the IEEE Computer and Communications Societies, INFOCOM, 2004.

[6] Dong Lu, Yi Qiao, Peter A Dinda, and Fabian E Bustamante. Characterizing and Predicting TCP Throughput on the Wide Area Network. In IEEE International Conference on Distributed Computing Systems, ICDCS, 2005.

[7] Guillaume Urvoy-Keller. On the Stationarity of TCP Bulk Data Transfers. In Proceedings of the Passive and Active Measurement Conference, PAM, 2005.

[8] Yin Zhang and Nick Duffield. On the Constancy of Internet Path Properties. In Proceedings of the ACM SIGCOMM Workshop on Internet Measurement, 2001.

[9] Cisco: It Came to Me in a Stream. https://www.cisco.com/web/about/ac79/docs/sp/Online-Video-Consumption_Consumers.pdf.

[10] M. Abadi et al. 2016. TensorFlow: A System for Large-scale Machine Learning. In OSDI. USENIX Association.

[11] Akamai.2016.dash.js.https://github.com/Dash-Industry-Forum/dash.js/. (2016).

[12] Huang T Y, Handigol N, Heller B, et al. Confused, timid, and unstable:picking a video streaming rate is hard[C]// Acm Conference on Internet Measurement Conference. 2012.

[13] Dobrian F, Sekar V, Awan A, et al. Understanding the impact of video quality on user engagement[J]. Acm Sigcomm Computer Communication Review, 2011, 41(4):362-373.

[14] Jiang J, Sekar V, Hui Z. Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE[C]// 2012:326-340.

[15] Yi S, Yin X, Jiang J, et al. CS2P:Improving Video Bitrate Selection and Adaptation with Data-Driven Throughput Prediction[C]// Conference on Acm Sigcomm Conference. 2016.

[16] Huang T Y, Johari R, Mckeown N, et al. A buffer-based approach to rate adaptation: evidence from a large video streaming service[J]. Acm Sigcomm Computer Communication Review, 2014, 44(4):187-198.

[17] Spiteri K , Urgaonkar R , Sitaraman R K . BOLA: Near-Optimal Bitrate Adaptation for Online Videos[J]. 2016.

[18] Zhi L, Zhu X, Gahm J, et al. Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale[J]. IEEE Journal on Selected Areas in Communications, 2014, 32(4):719-733.

[19] Mao H, Netravali R, Alizadeh M. Neural Adaptive Video Streaming with Pensieve[C]// Conference of the Acm Special Interest Group on Data Communication. 2017.G. O. Young, “Synthetic structure of industrial plastics,” in Plastics, 2nd ed., vol. 3, J. Peters, Ed. New York, NY, USA: McGraw-Hill, 1964, pp. 15–64.

[20] Hyunho Yeo, Youngmok Jung, Jaehong Kim, Jinwoo Shin, and Dongsu Han. 2018. Neural Adaptive Content-aware Internet Video Delivery. In OSDI. 645–661.

[21] Akhtar Z, Nam Y S, Govindan R, et al. Oboe: auto-tuning video ABR algorithms to network conditions[C]//Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. ACM, 2018: 44-58.

[22] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, “BOLA: Near-optimal bitrate adaptation for online videos,” in Proc. IEEE INFOCOM 35th Annu. Int. Conf. Comput. Commun., Apr. 2016, pp. 1–9.

[23] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, “A control theoretic approach for dynamic adaptive video streaming over HTTP,” SIGCOMM Comput. Commun. Rev., vol. 45, no. 4, pp. 325–338, Aug. 2015. [Online]. Available: http://doi.acm.org/10.1145/2829988.2787486.

[24] H. Mao, M. Alizadeh, I. Menache, and S. Kandula. 2016. Resource Management with Deep Reinforcement Learning. In HotNets. ACM.

[25] V. Mnih et al. 2016. Asynchronous methods for deep reinforcement learning. In International Conference on Machine Learning. 1928–1937.

[26] R. S. Sutton and A. G. Barto. 1998. Reinforcement Learning: An Introduction. MIT Press.

[27] BOLUKBASI, T., WANG, J., DEKEL, O., AND SALIGRAMA, V.Adaptive neural networks for effificient inference. In Proceedings of the International Conference on Machine Learning (ICML) (2017), pp. 527–536.

[28] HU, H., DEY, D., HEBERT, M., AND BAGNELL, J. A. Anytime neural network: a versatile trade-off between computation and accuracy. arXiv preprint arXiv:1708.06832 (2018).

[29] MAO, H., NETRAVALI, R., AND ALIZADEH, M. Neural adaptive video streaming with pensieve. In Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM) (2017), pp. 197–210.

[30] SUTTON, R. S., AND BARTO, A. G. Reinforcement learning: An introduction. MIT press Cambridge, 1998.

二、诚信承诺：

本人郑重承诺在该课程论文完成的过程中不发生任何不诚信现象，一切不诚信所导致的后果均由本人承担。

Integrity Commitment:

I/We solemnly promise that I/we has/have not plagiarised during the completion of this course paper, and all the consequences caused by plagiarism will be borne by me/us.

签名拍照附件:

Signature with photo attachment: