



一种基于视频分类的码率自适应算法

陈梓晗,叶进,肖庆宇

(广西大学 计算机与电子信息学院, 南宁 530004)

摘要: 流媒体的码率自适应算法依据网络状态动态调节视频块的码率,提升用户体验质量,但忽略了视频类型的差异对用户体验质量的影响,导致算法性能下降。提出区分视频类型特征的码率选择算法C-ABR。设计相应的用户体验质量效用函数,使用强化学习算法训练模型A3C,提升用户体验质量。实验结果说明,相对于典型的码率自适应算法Pensieve和MPC,C-ABR算法用户体验质量分别提升22.7%和50.4%。

关键词: 体验质量;强化学习;码率自适应算法;流媒体;激励函数

开放科学(资源服务)标志码(OSID):



中文引用格式: 陈梓晗,叶进,肖庆宇.一种基于视频分类的码率自适应算法[J].计算机工程,2021,47(12):118-121,130.
英文引用格式: CHEN Z H, YE J, XIAO Q Y. An adaptive bitrate algorithm based on video classification[J]. Computer Engineering, 2021, 47(12): 118-121, 130.

An Adaptive Bitrate Algorithm Based on Video Classification

CHEN Zihan, YE Jin, XIAO Qingyu

(School of Computer and Electronics Information, Guangxi University, Nanning 530004, China)

[Abstract] The Adaptive Bitrate (ABR) algorithms for streaming media can dynamically adjust the bitrate of video blocks according to network status, and thus improve the user Quality of Experience (QoE). However, the existing algorithms usually ignore the impact of video types on the QoE, resulting in performance degradation. This paper proposes a bitrate selection algorithm, C-ABR, which adapts to different types of videos. Corresponding utilization functions of user experience quality are designed, and the A3C model is trained by using the reinforcement learning algorithm to improve the QoE. The experimental results show that compared with the current typical bitrate self-adaption algorithms, including Pensieve and MPC, the C-ABR method improves the QoE by 22.7% and 50.4%, respectively.

[Key words] Quality of Experience (QoE); reinforcement learning; Adaptive Bitrate (ABR) algorithm; stream media; reward function

DOI: 10.19678/j.issn.1000-3428.0059790

0 概述

随着互联网技术的不断发展,视频流媒体已成为当前网络传输流量的主要组成部分,在网络应用中占据核心地位。根据第44次中国互联网统计报告显示^[1],目前中国视频类流量占因特网流量的80%,视频类APP应用产生了近30%的移动应用流量。思科公司的报告指出^[2]:全世界网络流量的67%是视频流量,预计到2021年将达到80%。视频流媒体的码率自适应(Adaptive Bitrate, ABR)算法的目标是让客户端所播放的视频块码率尽可能匹配当前网络带宽,从而最大化带宽的利用率,提升用户体验质量(Quality of Experience, QoE)。

依据工作原理的不同,码率自适应算法可以分为三类:第一类是基于吞吐量^[3-4]带宽预测^[5-6]的方法,依据网络吞吐量预测带宽;使得请求视频块的码率接近可用网络带宽;第二类是基于缓存^[7-9]的方法,将缓存分为若干区间,并对每个缓存区间采用不同的码率选择策略^[9-10];第三类方法是综合利用带宽预测^[11-13]以及缓存信息^[14-15]。而且随着人工智能技术的发展,此类策略应用机器学习^[16-17]方法实现更精确化的码率控制和QoE提升。下文具体分析各类方法的工作原理和特性。

基于吞吐量预测的方法核心是根据预测的网络吞吐率来选择码率。其中,客户端依据近段时间内的网络吞吐率来预测网络带宽,以此选择最佳码率,但受链

基金项目: 国家自然科学基金(6176030);广西自然科学基金(2018JJA170209)。

作者简介: 陈梓晗(1996—),男,硕士研究生,主研方向为流媒体传输;叶进(通信作者),教授、博士;肖庆宇,硕士研究生。

收稿日期: 2020-10-21 **修回日期:** 2020-12-05 **E-mail:** 1069879491@qq.com

路状态^[3]、带宽波动^[4]、网络环境异构性^[5]等因素的影响,选择最佳码率变得较困难,如在时变无线信道上吞吐量预测常常是不准确的,难以选择最优的码率。

不同于依据预测未来吞吐量来选择码率,文献[7]依据观察客户端视频缓冲区变化来确定最优码率,在视频块缓冲区长度较大时选择高码率,以提升视频质量;否则,选择低码率以避免视频卡顿。文献[8]提出的ABR算法在视频开始播放阶段基于历史吞吐量选择码率,在网络带宽平稳后根据当前的缓冲区占用情况选择视频码率。但总体而言,基于缓冲区占用的ABR算法由于客户端缓冲区的阈值界定难以选择,存在实际部署的问题。而基于带宽和缓存混合控制方法^[11-12]能充分利用网络带宽和缓冲区的状态^[13-14]选择码率。文献[15]将码率的选择建模为一个模型预测控制问题,依据吞吐量和缓冲区占用信息通过控制理论模型进行判断推理。然而,该方法的性能过度依赖于系统模型,对吞吐量预测误差和优化周期长度非常敏感。

随着人工智能技术飞速发展和不断应用,文献[16-17]提出基于强化学习框架的码率自适应方法。其中Pensieve^[16]借助于强化学习算法训练模型,不依赖于环境的假设,而是根据客户端采集数据为视频块选择码率,通过观察过去决策的结果来学习码率的选择策略。

现有研究的主流方法是构建统一的QoE模型,并设计启发式的码率自适应算法,考虑不同类型视频的特点。实际上,短视频、长视频和直播视频各自侧重不同的传输特性,具有不同的QoE需求。如果针对不同类型视频采用统一的QoE优化函数,则难以适应各种类型的个性化需求。例如,在网络带宽波动较大时,基于吞吐率的ABR算法也会频繁切换码率,虽然对长视频的影响不大,但会严重影响直播视频的观看体验。在Pensieve方案^[16]中,使用用户体验质量的相关指标作为激励(reward)函数反馈给码率决策系统,虽然兼顾了各个方面的性能需求,提高了网络适应性,但也并没有根据视频类型去制定更有针对性的reward函数,难以为各类视频提供满意的用户体验质量。

本文将视频分为长视频、短视频与直播视频三类,针对不同视频类型,通过强化学习方法得到更好的用户体验质量。使用不同类型视频的数据集训练强化学习模型,得到不同视频类型对应的reward函数参数,基于Asynchronous Advantage Actor-Critic (A3C)^[18]强化学习模型实现视频类型区分的码率自适应算法C-ABR。

1 算法设计

流媒体码率自适应算法的目标是提高用户体验质量。不同类型视频由于各自的特点和传输需求,对用户体验质量产生不同的影响。为解决该问题,本文提出视频类型敏感的C-ABR算法。

1.1 总体设计

C-ABR算法是基于DASH视频流框架的码率自适应算法。整体上包含网络参数处理模块、模型训

练模块和码率选择模块3个部分。C-ABR算法的部署如图1所示。首先网络参数处理模块采集网络状态和视频类型参数,提交给A3C模型训练模块,然后训练模块通过训练输出ABR码率选择的映射表,最终客户端依据映射表和当前网络参数选择码率。

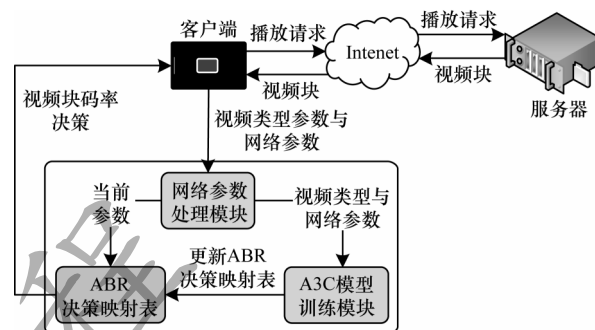


图1 C-ABR算法部署过程

Fig.1 Deployment process of C-ABR algorithm

在系统部署的过程中,本文选择A3C算法作为强化学习模型。与DQN^[19]等其他强化学习算法相比,A3C算法不需要使用经验池来存储历史样本,节约了存储空间。同时,A3C算法采用异步训练的模式,提升了训练速度。

1.2 激励函数

定义激励(reward)函数综合考虑码率、码率切换和卡顿时间对用户体验质量的影响。在强化学习过程中,reward函数是影响决策的重要因素,强化学习的每一步行动都会计算当前步数的reward函数值,reward函数值越大就表明当前步数对整个过程的贡献越大。具体计算公式如式(1)所示:

$$r_{\text{reward}} = \alpha \times q(R_k) - \beta \times |q(R_k) - q(R_{k-1})| - \gamma \times T_k \quad (1)$$

其中: R_k 是第 k 个视频块码率; $q(R_k)$ 是指第 k 个视频块reward函数中的码率得分,码率代表视频块的播放清晰度,码率越高,表示播放的视频块越清晰,进而reward得分越高; $\beta \times |q(R_k) - q(R_{k-1})|$ 为码率切换惩罚,是指在播放中对清晰度切换事件,在播放过程中,码率切换越多,惩罚越大; $\gamma \times T_n$ 为卡顿惩罚, T_n 是指在播放过程出现卡顿事件的时长,卡顿时间越长,reward函数值越低。

本文为不同视频类型设计相应的reward函数。首先将码率、码率切换与卡顿时间3项指标全部归一化。每个视频块的码率转化为 $q(R_k)/q(R_{\max})$,其中 R_{\max} 是指视频最高码率,视频切换转化为 $|q(R_k) - q(R_{k-1})| / (q(R_{\max}) - q(R_{\min}))$, R_{\min} 是视频最低码率。另外根据统计,通常视频块遭遇卡顿不会超过2 s,所以卡顿归一化为 $T_k/2$ 。最终得到归一化reward函数如式(2)所示:

$$r_{\text{reward}_1} = \lambda_1 \cdot \frac{q(R_k)}{q(R_{\max})} - \lambda_2 \cdot \frac{|q(R_k) - q(R_{k-1})|}{q(R_{\max}) - q(R_{\min})} - \lambda_3 \cdot \frac{T_k}{2} \quad (2)$$

然后将 $\lambda_1, \lambda_2, \lambda_3$ 全设定为1,使用 Pensieve 算法测试在3类视频中各项指标(码率、码率切换、卡顿时间)的影响程度。测试数据集包含各类视频500组样本,得到不同类视频各项指标的平均值。计算各项指标平均值的绝对值,得到各项指标对3项指标绝对值之和的比值,如图2所示。

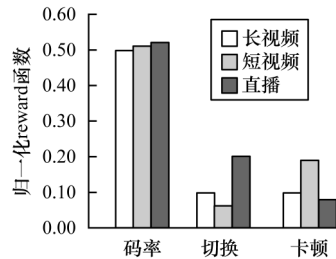


图2 不同视频类型下各指标的影响

Fig.2 Influence of each index under different video types

最后将长视频类型下 α, β, γ 参数分别设置为5、1、1。同时,依据图2所示指标项的影响程度,按各指标项之间的比例关系分别设置短视频和直播视频的 α, β, γ 参数,如表1所示。

表1 视频类型与 α, β, γ 的映射关系

Table 1 Mapping relation between video types and α, β, γ

视频类型	α	β	γ
长视频	5.0	1.0	1.0
短视频	5.1	0.6	1.9
直播视频	5.2	2.0	0.8

1.3 A3C 强化学习模型

本文使用A3C强化学习模型来选择码率。如图3所示,将不同视频类型的数据集分别输入A3C训练模型。输入数据为 $\{X_t, T_t, N_t, e_t, v, b_t\}$,其中: X_t 为上个视频块的下载吞吐率; T_t 为上个视频块下载的时间; N_t 为下个视频块的大小; e_t 为上个视频块的重传时间; v 为上个视频的视频类型; b_t 为当前的缓冲池大小。

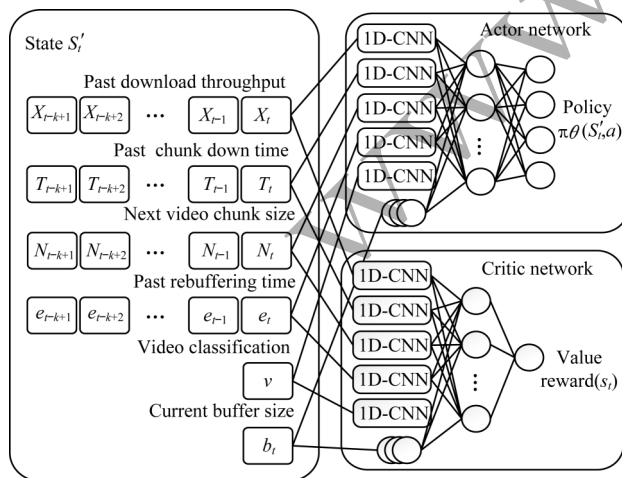


图3 A3C 强化学习模型

Fig.3 Reinforcement learning model of A3C

A3C 强化模型框架包含2个部分: Actor 行动模块和 Critic 评判模块。其中, Actor 行动模块是执行机构,输入外部的状态 S'_t ,然后输出动作 a ,在模型中表现为下个视频块的选择码率。而 Critic 评判模块则根据历史信息及反馈 reward 函数值进行自我调整,然后影响整个 Actor 行动模块。A3C 模型在 Actor-Critic 框架中还引入了异步训练的模式,可以同时启用多个训练环境进行训练,提升模型训练速度,并使样本分布更加均匀。

2 实验结果与分析

2.1 实验环境与数据集

实验配置: CPU 为 Intel i7 7700HQ, RAM 为 8 GB, GPU 为 Nvidia GTX 1060, GPU 加速库为 CUDA 10, 实验系统为 Ubuntu 16.04, 开发环境为 Pytorch, 开发工具为 Pycharm, 开发语言为 Python。

为便于对本文的码率自适应算法进行比较,本文使用 L3VTP 仿真器从真实环境中获取视频吞吐量数据来进行算法测试。仿真器的输入包含视频数据集和网络数据集,用来仿真模拟真实环境中网络的动态变化。本文视频数据集来自于真实环境采集,分别从真实环境中获取了短视频、长视频和直播视频进行仿真测试。其中各类视频的样本数量约为2 000,模型训练过程中迭代次数设置为20 000,表2所示为视频样本参数范围。

表2 视频样本参数范围

Table 2 Video sample parameter range

视频类型	视频长度/s	单个块长度/s	测试速度/(frame·s ⁻¹)
长视频	300~1 800	3~10	33~45
短视频	30~300	2~5	28~35
直播视频	30~1 800	1~3	24~32

2.2 结果分析

首先使用不同的强化学习模型来比较训练结果。从短视频、长视频、直播视频数据集各取10组数据分别进行测试,并取其平均结果。分别用 DQN、DDQN^[20]、A3C 3种不同的强化学习模型训练,并统计不同模型的 QoE 得分,测试结果如表3所示。

表3 不同强化学习模型的 QoE 得分

Table 3 QoE score of different reinforcement learning models

视频类型	DQN 模型	DDQN 模型	A3C 模型
长视频	2 508	2 697	2 813
短视频	1 146	1 317	1 696
直播视频	3 152	3 298	3 904

从测试结果可以看出, A3C 模型在测试中的 QoE 得分总体高于 DQN、DDQN 模型。其中,对于长视频,3种强化学习方法模型最后的得分是比较接

近;在直播视频得分对比中可以看出,A3C模型效果明显优于DQN与DDQN模型;而在短视频中,A3C模型的优势最为明显。

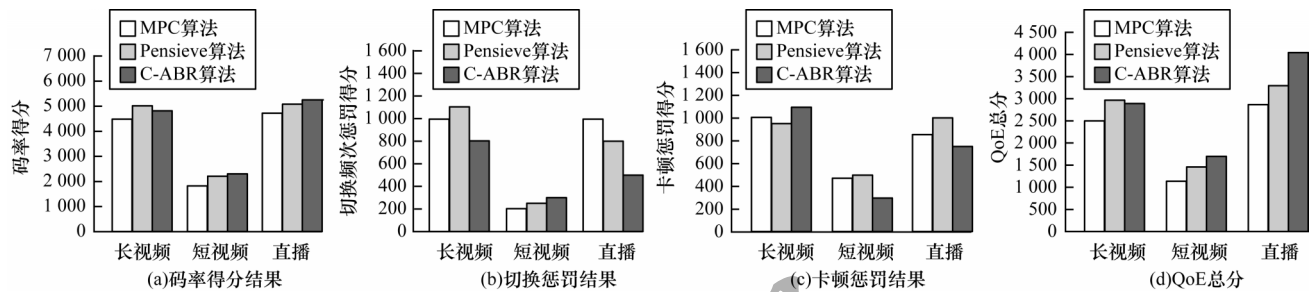


图4 不同算法在3类视频中的结果

Fig.4 Results of different algorithms in three kinds of videos

3种算法的码率得分情况如图4(a)所示,Pensieve和C-ABR算法相比MPC算法优势明显,原因在于:这2种算法都是基于强化学习的方法,能够在有限的带宽资源下,主动选择高码率来提升用户体验质量,使得码率得分提升。MPC缺乏准确的网络动态模型,它依赖于简单的和次优的启发式方法,导致保守的码率选择。而在C-ABR与Pensieve的对比中可以看出,短视频与直播视频C-ABR算法的得分更高。

切换惩罚与卡顿惩罚得分如图4(b)和图4(c)所示。由于C-ABR在训练过程中对视频进行了分类,在短视频中C-ABR将卡顿惩罚系数提高,因此在短视频的卡顿惩罚对比中,C-ABR的卡顿惩罚是最低的。而在直播视频中,C-ABR将切换频次惩罚系数提高,使得直播的切换频次惩罚明显低于其他算法。

QoE整体得分如图4(d)所示:在长视频中,C-ABR与Pensieve得分相差无几,在短视频的对比中,C-ABR比MPC提升了50.4%,比Pensieve提升了17.2%;在直播视频的对比中,C-ABR比MPC和Pensieve分别提升了42.1%和22.7%。

3 结束语

本文提出一种区分视频类型特征的码率选择算法C-ABR。该算法为不同类型视频设计了QoE优化函数,并使用A3C深度强化学习算法选择视频码率。C-ABR可广泛部署在视频传输框架中,不需要改动现有硬件,有效提升用户体验质量。实验结果表明,与Pensieve算法相比,C-ABR算法提高了14.7%的用户体验质量分数。下一步将考虑使用用户对不同视频源的主观观看体验作为选取码率的依据,制作精准的用户画像,并通过设置优先级调节网络带宽分配来提升用户体验。

参考文献

[1] 中国互联网络信息中心. 第44次中国互联网络发展状况统计报告[EB/OL]. [2020-09-10]. <http://www.cac.gov.cn>. China Internet Network Information Center. The 44th statistical report on Internet development in China[EB/OL]. [2020-09-10]. <http://www.cac.gov.cn>. (in Chinese)

然后测试本文C-ABR算法的性能。对比测试C-ABR、Pensieve与MPC 3种算法的性能,结果如图4所示

[2] Cisco visual networking index: forecast and methodology [EB/OL]. [2020-09-10]. <https://www.cisco.com/c/en/us/solutions/service-provider/index.html>.

[3] JIANG J, SEKAR V, ZHANG H. Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive [J]. IEEE/ACM Transactions on Networking, 2014, 22(1): 326-340.

[4] LIU C H, BOUAZIZI I, GABBOUJ M. Rate adaptation for adaptive http streaming [C]//Proceedings of ACM MMSys '11. New York, USA: ACM Press, 2011: 169-174.

[5] SUN Y, YIN X, JIANG J, et al. CS2P: improving video bitrate selection and adaptation with data-driven throughput prediction [C]//Proceedings of ACM SIGCOMM'16. New York, USA: ACM Press, 2016: 272-285.

[6] ABDELHAK B, ALI C B, ROGER Z. QoE-aware bandwidth broker for HTTP adaptive streaming flows in an SDN-enabled HFC network [J]. IEEE Transactions on Broadcasting, 2018, 64(2): 575-589.

[7] HUANG T Y, JOHARI R, MCKEOWN N. Downton abbey without the hiccups: buffer-based rate adaptation for HTTP video streaming [C]//Proceedings of ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking. New York, USA: ACM Press, 2013: 453-467.

[8] HUANG T Y, JOHARI R, MCKEOWN N, et al. A buffer-based approach to rate adaptation: evidence from a large video streaming service [C]//Proceedings of ACM SIGCOMM'14. New York, USA: ACM Press, 2014: 187-198.

[9] 潘鹏程,李国平,王国中. 基于HTTP的流媒体自适应码率切换算法[J]. 电子测量技术, 2017, 40(9): 141-145.

PAN P C, LI G P, WANG G Z. HTTP-based adaptive bit rate switching algorithm for streaming media [J]. Electronic Measurement Technology, 2017, 40(9): 141-145. (in Chinese)

[10] HUANG T C, ZHOU C, ZHANG R X, et al. Comyco: Quality-aware adaptive video streaming via imitation learning [C]//Proceedings of the 27th ACM International Conference on Multimedia. New York, USA: ACM Press, 2019: 429-437.

[11] SPITERI K, URGANONKAR R, SITARAMAN P K. BOLA: near-optimal bitrate adaptation for online videos [C]//Proceedings of the 35th Annual IEEE International Conference on Computer Communications. Washington D. C., USA: IEEE Press, 2016: 1-9.

(下转第130页)

(上接第 121 页)

- [12] SOBHANI A, YASSINE A, SHIRMOHAMMADI S. A video bitrate adaptation and prediction mechanism for HTTP adaptive streaming [J]. ACM Transactions on Multimedia Computing Communications & Applications, 2017, 13(2): 18-23.
- [13] VERGADOS D J, MICHALAS A, SGORA A, et al. FDASH: a fuzzy-based MPEG/DASH adaptation algorithm [J]. IEEE Systems Journal, 2016, 10(2): 859-868.
- [14] TIAN G B, LIU Y. Towards agile and smooth video adaptation in dynamic HTTP streaming [C]//Proceedings of CoNEXT'12. Washington D. C. , USA: IEEE Press, 2012: 109-120.
- [15] YIN X, JINDAL A, SEKAR V, et al. A control-theoretic approach for dynamic adaptive video streaming over HTTP [J]. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 325-338.
- [16] MAO H Z, NETRAVALI R, ALIZADEH M. Neural adaptive video streaming with pensieve [C]//Proceedings of SIGCOMM'17. New York, USA: ACM Press, 2017: 197-210.
- [17] HUANG T, YAO X, WU C, et al. Tiyuntsong: a self-play reinforcement learning approach for ABR video streaming [C]//Proceedings of IEEE International Conference on Multimedia and Expo. Washington D. C. , USA: IEEE Press, 2019: 1678-1683.
- [18] ZHANG Y, CLAVERA I, TAI B, et al. Asynchronous methods for model-based reinforcement learning [EB/OL]. [2020-09-10]. <https://arxiv.org/abs/1910.12453v1>.
- [19] CHUNG J. Playing ATARI with deep reinforcement learning [EB/OL]. [2020-09-10]. <https://arxiv.org/preprint/1312.5602>.
- [20] VAN HASSELT H, GUEZ A, SILVER D. Deep reinforcement learning with double q-learning [EB/OL]. [2020-09-10]. <https://arxiv.org/pdf/1509.06461.pdf>.