

移动边缘计算中基于联邦学习的视频请求预测和协作缓存策略

李光辉 李宜璟* 胡世红

(江南大学人工智能与计算机学院 无锡 214122)

摘要: 随着互联网社交平台的崛起和移动智能终端设备的普及, 自媒体短视频、直播等视频业务蓬勃发展, 人们对高质量视频服务的需求也急剧上升。与此同时, 连接到核心网络的大量智能设备增加了回程链路的负载, 传统的云计算难以满足用户对视频服务的低延迟要求。移动边缘计算(MEC)通过在网络边缘部署具有计算和存储能力的边缘节点, 通过在更靠近用户的边缘侧提高计算和存储服务, 降低了数据传输时延进而缓解了网络阻塞。因此, 基于MEC架构, 该文充分利用网络边缘资源, 提出了基于联邦学习的视频请求预测和视频协作缓存策略。通过利用多个边缘节点对提出的深度请求预测模型(DRPN)视频请求预测模型进行联邦训练, 预测视频未来的请求情况, 然后量化缓存内容所带来的时延收益并协作地以最大化该时延收益为目的进行缓存决策。该文分析了真实数据集MovieLens, 模拟了视频请求缓存场景并进行实验。仿真结果表明, 相比于其他策略, 所提策略不仅能有效降低用户等待时延, 在有限的缓存空间中提高内容多样性, 从而提高缓存命中率, 降低缓存成本, 还能降低整个系统的通信成本。

关键词: 边缘计算; 联邦学习; 视频缓存; 请求预测

中图分类号: TN929.5; TP181

文献标识码: A

文章编号: 1009-5896(2023)01-0218-09

DOI: 10.11999/JEIT211287

Video Request Prediction and Cooperative Caching Strategy Based on Federated Learning in Mobile Edge Computing

LI Guanghui LI Yijing HU Shihong

(School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China)

Abstract: With the rise of Internet social platforms and the popularization of mobile smart terminal devices, people's demand for high-quality and real-time data has risen sharply, especially for video services such as short videos and live streams. At the same time, too many terminal devices connected to the core network increase the load of the backhaul link, so that the traditional cloud computing is difficult to meet the low-latency requirements of users for video services. By deploying edge nodes with computing and storage capabilities at the edge of the network, Mobile Edge Computing (MEC) can calculate and store closer to the users, which will reduce the data transmission delay and alleviate the network congestion. Therefore, making full use of the computing and storage resources at the edge of the network under MEC, a video request prediction method and a cooperative caching strategy based on federated learning are proposed. By federally training the proposed model Deep Request Prediction Network (DRPN) with multiple edge nodes, the video requests in the future can be predicted and then cache decisions can be made cooperatively. The simulation results show that compared with other strategies, the proposed strategy can not only effectively improve the cache hit rate and reduce the user waiting delay, but also reduce the communication cost and cache cost of the whole system to a certain extent.

Key words: Mobile Edge Computing(MEC); Federated learning; Video caching; Request prediction

收稿日期: 2021-11-17; 改回日期: 2022-02-22; 网络出版: 2022-03-08

*通信作者: 李宜璟 6191914008@stu.jiangnan.edu.cn

基金项目: 国家自然科学基金(62072216)

Foundation Item: The National Natural Science Foundation of China (62072216)

1 引言

随着移动互联网的飞速发展以及智能终端功能的逐渐强大,拍摄、处理、上传、分享等功能大大降低了视频制作的难度,进而带动了自媒体短视频、直播等视频服务的兴起。越来越多的人可以通过各种视频网站或社交媒体平台传递信息。随着5G时代的到来,视频正在成为信息的主要载体。根据英特尔委托Ovum最新发布的《5G娱乐经济学报告》,预计未来10年5G用户的月均流量将是4G用户的7倍,到2028年消费者将在视频服务上花费近 1.5×10^{11} 美元^[1]。传统的云中心具有强大的计算和存储能力,但在5G应用场景下,也难以应对用户对数据急剧增加的需求。移动边缘计算(Mobile Edge Computing, MEC)^[2]作为一种有潜力的新型计算范式,通过在网络边缘部署计算和存储资源,在更靠近用户的位置提供服务或存储数据,减少数据传输延迟并缓解网络拥塞,弥补了云计算的不足。

内容缓存作为新兴网络体系结构(如以内容为中心的网络和以信息为中心的网络)的基本网络功能,并不是一项新技术,但视频流量的快速增长也导致业界和学术界正在重新设计内容缓存系统,从而适应这种巨大的流量^[3]。因此,越来越多的研究者考虑在移动边缘计算的框架下研究内容缓存的相关问题^[4-8]。

Thar等人^[4]提出将深度学习应用于预测内容的未来流行度,通过在云数据中心训练深度学习模型,做出缓存决策,将缓存决策发送到每个基站以主动存储流行内容。Rathore等人^[5]提出了一种基于深度学习的主动缓存框架,通过提取用户和内容的隐藏特征构造用户的内容流行度矩阵,并通过基于特征的协同过滤用该矩阵预测核心网络的流行内容进而缓存。杨静等人^[6]提出一种结合特征感知的内容社交价值预测方法,通过预测的价值计算每个内容的总流行价值进而选择缓存内容。上述研究涉及的模型训练都在数据中心完成,因此需要将用户请求的相关信息上传至数据中心,这个过程中需要占用网络流量,消耗通信成本,同时可能引起用户相关隐私泄露的安全问题。为了解决上述问题,Saputra等人^[7]提出了一种新颖的基于分布式深度学习框架来预测流行内容,每个边缘节点都从其覆盖区域的移动用户收集信息,并建立动态日志在本地进行预测。刘浩洋等人^[8]则提出一种流行度匹配边缘缓存策略,并通过建模实验验证了流行度参数匹配对应的文件缓存概率可以最大限度提升通信可靠

性并降低回程带宽压力。上述研究在仿真实验中均得到很好的验证,但与仿真实验中按概率分布生成的数据不同,现实世界的数据更加多样和稀疏,这对预测带来了更多挑战。Google首次提出了联邦学习的概念,不同于普通分布式训练的是,联邦学习的计算节点对数据具有绝对控制权,可以选择随时停止计算和通信,退出学习过程^[9]。中心节点无法直接或间接操作计算节点上的数据。此外,在分布式学习中,不同计算节点上的数据划分通常是均匀的,具有独立同分布的特点,这样的特性非常适合设计高效的训练算法,但真实场景下,各边缘节点收集的数据量由其覆盖的用户偏好决定,是不平衡的,因此各个节点的数据往往表现出不同的分布特征。

针对上述问题,本文在移动边缘计算的背景下,提出一种基于联邦学习的视频请求预测和视频协作缓存策略。利用已部署的多个边缘节点对提出的预测模型进行联合训练,在保证预测准确性的前提下提升了训练速度,减少通信成本,并使模型训练和直接访问原始数据解耦,减少隐私泄露的风险。同时,本文提出具有协作意识的缓存策略,进一步降低了时延,并通过减少不必要的冗余缓存提高了有限存储空间中的内容多样性,从而提高缓存命中率,降低缓存成本。为了验证方法的有效性,本文利用现实世界的公开数据集MovieLens^[10],模拟了视频请求缓存场景。实验结果表明,本文提出的策略能有效提高缓存命中率,减小小时延,并且降低通信成本。

2 系统模型

2.1 网络架构

考虑如图1所示的云边协同网络,边缘节点通过无线蜂窝网络为覆盖区域内大量不同地理位置的移动用户提供服务,也可通过回程链路连接到云服

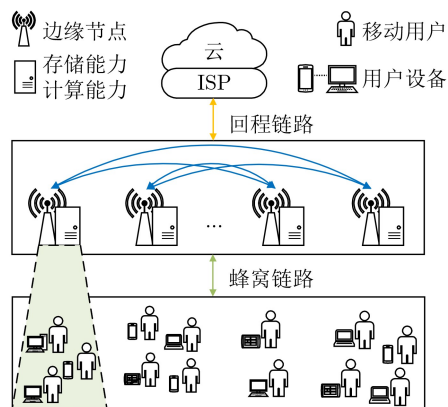


图1 网络结构

务中心。其中,每个边缘节点都部署了用于计算和存储的边缘服务器,因此均能通过计算进行预测,从而提前主动缓存一些用户可能请求的内容,降低用户等待时延,从而提高用户体验质量(Quality of Experience, QoE)。用户可以通过边缘节点在本地获取请求内容,或者经由互联网服务提供商(Internet Service Provider, ISP)从云中心获取请求内容。

令 $\mathcal{N} = \{1, \dots, n, \dots, N\}$ 表示边缘节点集合,每个边缘节点都设有一定的存储容量 C ,用于缓存内容。用户集合 $\mathcal{U} = \{1, \dots, u, \dots, U\}$ 随机分布在各个边缘节点的服务范围内。 $\mathcal{F} = \{1, \dots, f, \dots, F\}$ 表示用户请求内容的集合,每个内容 f 大小均为 s_f 。具有不同大小的内容始终可以拆分为相同大小的数据段,每个数据段视为一个“内容”,这也是现实中常用的做法。对于视频内容,用户可能仅观看其中一部分,因此只需传输相应部分而非全部内容,从而降低传输延迟^[11,12]。 $R_n^t = [r_1, r_2, \dots]$ 表示边缘节点 n 在时间步 t 内收到的请求列表,其中 r_i 表示第 i 个到达的请求($i = 1, 2, \dots$),并且每个请求 r 可表示为一个元组 (u, f, c, \hat{t}) , $u \in \mathcal{U}, f \in \mathcal{F}$,其中 c 表示请求的相关上下文信息, \hat{t} 为请求 r 到达的时间戳。在ISP可提供的内容库 \mathcal{F} 中,任意用户 u 均能在任意时间,任意服务区域内对内容 f 发出请求 r 。

2.2 单个边缘节点的功能模块

图2展示了多个边缘节点协同工作时每个边缘节点所具备的功能模块。每个边缘节点通过用户接口模块,以时间步 t 为一个时间单位接收服务区域内用户发出的请求 R_n^t ,并将请求信息存入本地数据库。边缘节点可以直接将接收的请求信息送入请求处理模块,并用训练好的预测模型对用户的未来请求进行预测(请求预测模块),然后联合邻居边缘节点利用预测信息协作地进行缓存决策(缓存决策模块),具体的协作缓存策略在3.2节详细描述。最后通过缓存管理模块决定在本地主动缓存的内容。由于用户可以不断进入或离开不同边缘节点的服务范围,并且不断有新的内容生成,用户对内容的需求不断变化,不同时期内容的流行度也不同,

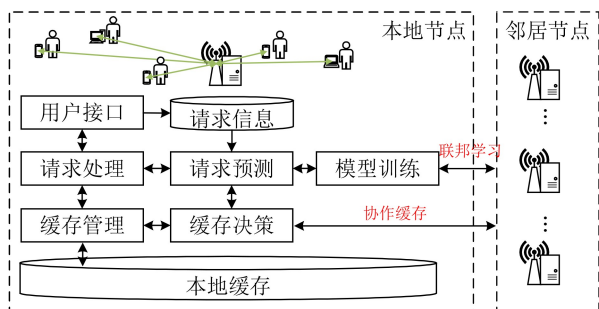


图2 单个边缘节点功能

因此请求预测模型是时间敏感的。为了确保预测模型的有效性,需要适时用新的数据训练更新预测模型。因此当前的本地节点可以联合多个邻居边缘节点,通过联邦学习训练得到新的模型以淘汰过时模型,基于联邦学习的模型训练的具体过程见3.1.2节。

3 基于联邦学习的视频请求预测和协作缓存策略

3.1 基于联邦学习的视频请求预测

本节将主要介绍提出的视频请求预测模型、构造用于训练的数据集处理过程以及模型的训练和更新过程。

3.1.1 视频请求预测模型及数据处理

本文首先建立一个视频请求预测模型,该模型可根据用户对视频发出的历史请求信息预测视频在未来被频繁请求的概率,从而确定视频未来的流行趋势并得到值得提前缓存的视频列表。在本文的请求预测任务中,在分析真实数据集的基础上,提出了一种新的深度请求预测模型(Deep Request Prediction Network, DRPN),具体结构如图3。选择卷积层作为特征抽取器,在特征空间中提供多维输入数据的抽象表示。由于输入请求信息的时序性,在卷积操作后,选择递归神经网络(Recurrent Neural Networks, RNN)学习特征序列间的时间依赖关系。RNN在文本处理、流量预测等领域已被证明有效^[13,14],Ale等人^[15]也通过双向深层递归网络(Bidirectional Deep Recurrent Neural Network, BDRNN)预测在线活动缓存。在各种类型的RNN中,本文选择长短期记忆网络(Long Short-Term Memory, LSTM)作为递归层,因为它扩展了存储单元,有助于学习长期的时间关系,并在请求预测的问题上能获得很好的性能。图4显示了LSTM单元的内部结构,其主要思想是通过3个门来控制时间轴上的信息更新,从而在每个时间步获取信息。为方便表达,令单元状态为 c ,输入状态为 x ,隐藏状态为 h , W 表示相应权重矩阵, b 表示相应偏置。在 t 时刻,一个LSTM单元包含3个输入:当前时刻的网络输入 x^t ,上一个时刻的输出 h^{t-1} ,以及上一时刻的单元状态 c^{t-1} ,同时包含两个输出:当前时刻的输出 h^t 和当前时刻的单元状态 c^t 。遗忘门控制是否继续保留长期状态,即决定了上一个时刻的单元状态 c^{t-1} 有多少保留到至 c^t 。输入门控制是否将由上一次输出和本次输入计算得到的当前单元状态 c^t 送到长期状态中,即决定当前时刻将多少网络的输入 x^t 保存到单元状态 c^t 。输出门控制单元状态是否输出,即决定有多少单元状态 c^t 将输出到

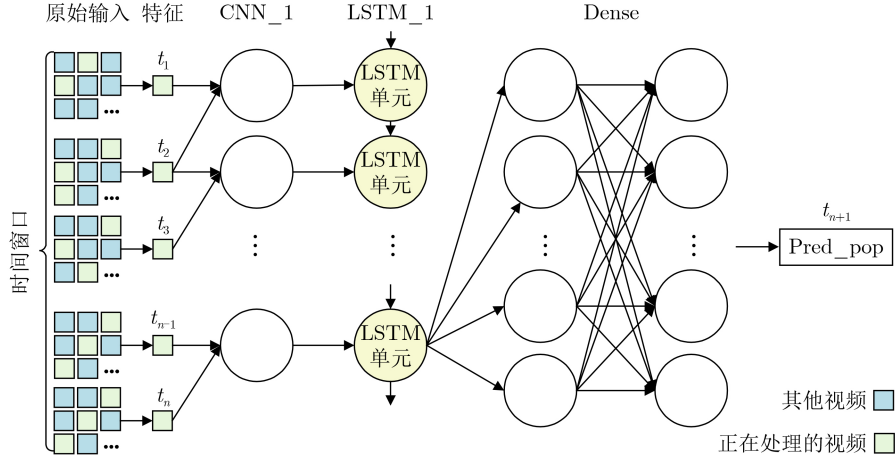


图3 DPRN 网络结构

LSTM的当前输出 h^t 中。遗忘门、输入门、输出门、当前单元状态和输出的具体更新规则为

$$f^t = \sigma(W_f \cdot [h^{t-1}, x^t] + b_f) \quad (1)$$

$$i^t = \sigma(W_i \cdot [h^{t-1}, x^t] + b_i) \quad (2)$$

$$o^t = \sigma(W_o \cdot [h^{t-1}, x^t] + b_o) \quad (3)$$

$$c^t = \tanh(W_c \cdot [h^{t-1}, x^t] + b_c) \quad (4)$$

$$c^t = f^t \odot c^{t-1} + i^t \odot c^t \quad (5)$$

$$h^t = o^t \odot \tanh(c^t) \quad (6)$$

DRPN模型包括卷积层、LSTM层和全连接层，用于模拟输入和输出之间的复杂非线性关系。图4展示了CNN_1将输入特征序列转换为结合了邻域信息的特征序列，并为LSTM_1提供输入的过程，也展示了LSTM_1的最后一次输出送到全连接层中进行分类的过程。最后，sigmoid函数将网络输出转换为以概率形式表示的视频未来流行度pred_pop (视频在未来被频繁请求的概率)。

对于每个边缘节点，首先获取最近一段时间的原始请求数据，其中包含多个时间步长。然后对于每个视频 i ，收集其统计信息，并构造适当的特征。计算一个时间步 t 内视频 i 的请求数量为 req_i ，并计算 t 内视频的请求概率 P_{req_i} ，其中 $P_{req_i} = req_i / \sum_{j \in I} req_j$ ， I 为该时间步内所有被请求过的视频集合。此外，时间步长 t 是一个可调参数，例如 $t = 2$ h。然后，对视频类型进行独热(one-hot)编码。

在对原始数据进行预处理后，进一步构造可以用于训练预测模型的数据集。文献[4]表明请求数据具有一定序列关系。因此，对于每个视频，用一个时间窗口的数据来预测其在下一个时间步的流程度。值得注意的是，一个时间窗口包含 \bar{t} 个连续的时间步长， \bar{t} 为可调参数。每个时间步中的样本特

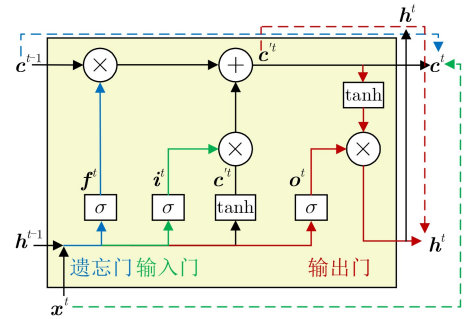


图4 LSTM 单元内部结构

征为 $(req, P_{req}, genres)$ 。定义流行内容为请求量排名在前20%的内容，则当某样本对应的视频在下一个时间步中符合流行内容的标准时，将该样本的标签记为1，否则记为0。因此，每个边缘节点 n 获得相应经过处理后的新数据集 D_n ，该数据集可用于训练预测模型。

3.1.2 模型训练及自动更新机制

由上一节可得数据集构造过程中，通过自主计算推断出数据特征和标签，而不需要人工打标签，并且每个边缘节点服务覆盖的用户不同，用户进行请求的频率也不同，所以每个节点 n 的数据集 D_n 在数据量上可能存在较大差距，即数据分布不平衡。此外，请求数据来自各个用户设备，本身带有隐私性，数量庞大。这些特性都非常符合用于联邦学习[9]的数据特点。除此之外，随着边缘计算的发展，越来越多的边缘节点被部署在更接近用户的地方以提高QoE。因此，本文提出利用联邦学习机制对预测模型进行训练。

首先，假设部署了包含 N 个边缘节点的边缘节点集合 \mathcal{N} ，每个边缘节点 n 都有相应的本地数据集 D_n ， $|D_n|$ 表示 D_n 的大小，所以总数据集 $\mathcal{D} = \bigcup_{n \in \mathcal{N}} D_n$ 。对于学习问题，目标是通过损失函数 $f(w)$ 的反向传播求得目标参数 w 。本文中的预测问

题是一个2元分类问题, 因此使用分类模型中常用的2元交叉熵(Binary Cross Entropy, BCE)作为损失函数。对于每个边缘节点 n , 损失函数定义为

$$F_n(\mathbf{w}) = \frac{1}{|D_n|} \sum_{i=1}^{|D_n|} f_i(\mathbf{w}) \quad (7)$$

而每个本地模型的优化目标为

$$\mathbf{w}_j^t = \arg \min F_n(\mathbf{w}_j | \mathbf{w}_j^{t-1}) \quad (8)$$

全局损失函数定义为

$$F(\mathbf{w}) = \sum_{n=1}^N \frac{|D_n|}{D} F_n(\mathbf{w}), \quad D = \sum_{D_n \in \mathcal{D}} |D_n| \quad (9)$$

最终目标在于求解

$$\mathbf{w}^* = \arg \min F(\mathbf{w}) \quad (10)$$

而每个边缘节点更新参数的规则为

$$\mathbf{w}_i^{t+1} = \mathbf{w}_i^t - \eta \nabla F_n(\mathbf{w}^t) \quad (11)$$

聚合器汇总更新参数的规则为

$$\mathbf{w}^{t+1} = \sum_{n=1}^N \frac{|D_n|}{D} \mathbf{w}_n^{t+1} \quad (12)$$

本文提出一个通用的梯度下降算法来求解参数, 参数的同步更新在通信过程中进行, 具体过程如**算法1**所示。在每轮更新开始时, 随机选择 K 个

边缘节点, 其中 $K \in [1, N]$ 。将当前的全局初始参数 \mathbf{w}^t 分发给这 K 个节点。这些被选中的节点基于全局初始状态参数 \mathbf{w}^t 和本地训练集 D_n 进行一定轮次 E 的本地训练, 根据式(11)更新 \mathbf{w}_n^{t+1} 。然后云中心作为聚合器收集参与训练的节点更新的参数并根据式(12)计算更新全局参数 \mathbf{w}^{t+1} , 再将计算所得的全局参数分发给下一轮参与更新的节点, 进行新一轮次的本地训练。重复该过程直至收敛。

由于数据具有时效性, 预测模型的精度会随着时间的推移而降低。此外, 在真实的无线网络环境中, 训练质量会受到数据包错误和无线资源可用性等因素的影响^[16], 这些因素都可能降低缓存命中率, 导致模型失效。于是, 提出一个自适应的模型更新规则以保证预测模型的准确性。用户发出请求后, 边缘节点记录该请求最终被响应的位置, 并以此计算一段时间内的平均缓存命中率。当缓存命中率低于设定阈值时, 系统认为模型失效, 并自动获取本地近期的请求信息进行数据处理, 构造新的训练集, 然后用新的数据去训练更新预测模型。

3.2 协作缓存策略

通过预测模型, 每个边缘节点 n 可以利用最近一个时间窗口内的本地数据去预测下一个时间步的视频流行度, 从而获得下一个时间步的视频预测列表 $L_n = [l_1, l_2, \dots]$, 每个 l 是一个元组 $(f, s_f, \text{pred_pop})$, 其中 f 为视频标识, s_f 为视频内容大小, pred_pop 为视频在未来被频繁请求的预测概率, L_n 是根据 pred_pop 降序排列得到的。相比于云中心, 边缘节点普遍离用户更近, 所以对用户来说, 即使当本地节点没有提前缓存用户请求的视频, 但其邻居边缘节点缓存了相应视频时, 用户通过较短的查询时延后, 仍然能以用户和边缘节点间较快的传输速度在邻居节点获取视频内容。为了尽可能减少时延, 我们定义时延的减少量为收益, 若采用枚举的方法来选出能带来最大收益的缓存策略, 假设有 F 部影片(视频), N 个节点, 且每个节点可以容纳 C 个内容, 那么对于每个节点来说, 该节点的缓存策略数有 C_F^C 种, 而对于 N 个节点, 则总共会有 $(C_F^C)^N$ 种缓存策略, 当 F, C, N 3个值增加时(尤其是 N)会使得枚举过程的耗时急剧上升。因此需要更快且有效的方法来进行缓存决策, 在联合多个边缘节点的情况下, 本文提出一种基于贪心算法的协作缓存策略(Greedy algorithm based Cooperative Caching Strategy, GCCS), 以尽可能降低时延为目的, 并使本地节点和邻居节点协作地缓存一些重复内容, 以节省更多空间缓存其他内容, 增加内容多样性, 进一步提高缓存效果。这对ISP来说, 也一定程度

算法1 DRPN模型的分布式联邦训练

输入: $\mathcal{N}, \mathcal{D}, k, \mathbf{w}_n^0, T, E, \eta$
 输出: \mathbf{w}^{t+1}

- (1) $N \leftarrow |\mathcal{N}|$
- (2) $K \leftarrow \max(\lceil kN \rceil, 1)$ # k 为每轮参与训练的边缘节点占比, $k \in (0, 1]$
- (3) for $t = 1, 2, \dots, T$ do #全局训练轮次
- (4) 从节点集合 \mathcal{N} 中随机选择 K 个节点形成子集 \mathcal{N}'
- (5) for $n \in \mathcal{N}'$ do #各节点并行执行
- (6) $\mathbf{w}_n^{t+1} \leftarrow \text{LOCALUPDATE}(n, \mathbf{w}^t)$
- (7) end for
- (8) 中心作为聚合器, 负责计算、汇总并分发下一轮权重 \mathbf{w}^{t+1}
- (9) $\mathbf{w}^{t+1} \leftarrow \sum_{n=1}^N \frac{|D_n|}{D} \mathbf{w}_n^{t+1}$
- (10) end for
- (11) function LOCALUPDATE(n, \mathbf{w}^t) #本地训练
- (12) for $e = 1, 2, \dots, E$ do #本地训练轮次
- (13) $\mathbf{w}_n^{t+1} \leftarrow \mathbf{w}_n^t - \eta \nabla F_n(\mathbf{w}^t)$ # η 为学习率
- (14) end for
- (15) return \mathbf{w}_n^{t+1}
- (16) end function

上减少邻近区域内不必要的冗余内容，节约了缓存成本。提出的协作缓存策略GCCS具体过程如算法2所示，提出的协作缓存策略在预测得到的pred_pop(算法描述中记作 c)的基础上利用贪心算法尽可能

算法2 基于贪心算法的协作缓存策略(GCCS)

输入: $\mathcal{N}_n, \mathcal{L} = [L_1, \dots, L_n, \dots, L_N], \mathcal{C} = [C_1, \dots, C_n, \dots, C_N]$
 输出: $\mathcal{L}' = [L'_1, \dots, L'_n, \dots, L'_N]$

- (1)初始化 \mathcal{C} 中的每一项为实际最大可用缓存容量
- (2)初始化 \mathcal{L}' 中的每一项为空的缓存列表
- (3)记 \mathcal{F} 为 \mathcal{L} 中每一个列表所涉及的内容标识符 f 的全集
- (4)repeat
- (5) best_gain \leftarrow 0
- (6) for n in \mathcal{N}_n
- (7) for f in \mathcal{F}
- (8) if f not in L' and $s_f \leq C_n$ then
- (9) gain \leftarrow CALCULATEGAIN(n, f)
- (10) best_gain \leftarrow max(best_gain, gain)
- (11) end if
- (12) end for
- (13) end for
- (14) if best_gain \neq 0 then
- (15) $f' \leftarrow$ 产生最大增益时所对应的内容标识符
- (16) $n' \leftarrow$ 产生最大增益时所对应的节点
- (17) Add f' to $L'_{n'}$
- (18) $C_{n'} \leftarrow C_{n'} - s_{f'}$
- (19) end if
- (20)until best_gain = 0
- (21)function CALCULATEGAIN(n', f')
- (22) total_gain \leftarrow 0
- (23) 记 \mathcal{F}' 为 \mathcal{L}' 中每一个列表所涉及的内容标识符 f 的全集
- (24) for n in \mathcal{N}_n
- (25) if f' in L_n then
- (26) $c \leftarrow L_n$ 中对于 f' 的流行度预测值pred_pop
- (27) if f' not in \mathcal{F}' then
- (28) total_gain \leftarrow total_gain + c #命中状态由云端命中转换为邻居命中
- (29) end if
- (30) if $n = n'$ then
- (31) total_gain \leftarrow total_gain + αc #命中状态由邻居命中转换为本地命中
- (32) end if
- (33) end if
- (34) end for
- (35) return total_gain
- (36) end function

降低请求时延。在每一轮迭代中，考虑每个视频 f 存储到每个节点上的操作可能性，并考虑任一操作实际发生后，在未来接收到与请求预测结果相似的请求序列时，总体响应时延total_gain所发生的变化。对每条预计发生的请求，若该操作会使命中情况发生变化，则该操作将带来相应的时延收益(c 或 c)。因此，总体响应时延total_gain为对每个节点上的每条预计发生的请求计算得到的总体时延收益。然后选出能带来最大时延收益best_gain的操作作为本轮迭代中所要执行的缓存操作。

4 仿真实验

4.1 数据集与实验设置

实验使用了公开数据集MovieLens，数据集包括283228个用户对58098部电影的27753444条评级以及电影的相关信息(如电影题材)。类似文献[3]，假设评级总数和电影的请求总数相同，且用户对电影评级的时间与用户对电影发出请求的时间相同。为了模拟用户移动性，设置用户移动系数 $m = 0.1$ ，因此每经过一个时间周期 T ，边缘节点服务的用户将有占比为 m 的用户被随机替换。用户和云中心之间的传输速率为1~3 MBps，用户和边缘节点之间的传输速率为3~5 MBps。在每个通信轮次，系统随机选择比例为 $k = 0.5$ 的边缘节点参与训练，并将其对应的数据集中80%用于训练，20%用于验证，防止过拟合。对于每个选中参与联邦训练的边缘节点，设置其本地训练的轮次 $E = 10$ ，batch size = 128，Adam为优化函数，学习率 $\eta = 10^{-6}$ 。

4.2 评价指标

4.2.1 预测准确性

本文提出的协作缓存策略基于未来的视频流行度，即DRPN模型预测未来视频频繁被请求的概率。基于这个预测结果，边缘节点利用存储空间主动缓存那些未来可能被频繁请求的视频。因此，预测模型的准确性会影响缓存策略的性能。为了评价预测模型的准确性，本研究选择BCE来衡量预测误差，并使用归一化折损累计增益(Normalized Discounted Cumulative Gain, NDCG)评估预测模型的性能。NDCG是一个常用的基于排名的指标，强调排名靠前的重要性。有较高请求的内容比较低请求内容更影响最终的指标得分，且高请求的内容出现在更靠前的位置时，指标越高，因此NDCG越高，指标越好。

假设共有 n 条样本数据， \hat{y}_i 表示第 i 个视频是未来流行内容的概率，与之对应的真实标签为 y_i ，那么

$$\text{BCE} = \frac{1}{n} \sum_{i=1}^n -[y_i \log_2(\hat{y}_i) + (1 - y_i) \log_2(1 - \hat{y}_i)] \quad (13)$$

预测结果的顺序有较大影响,因此假设每个视频的请求数为 z_i 。首先计算折损累计增益(Discouted Cumulative Gain, DCG),按照预测值 \hat{y}_i 降序排列得到对应的列表 $\hat{Y} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n]$ 和 $Z = [z_1, z_2, \dots, z_n]$ 。因此计算得到

$$DCG = \sum_{i=1}^n \frac{z_i}{\log_2(i+1)} \quad (14)$$

同时,为了计算NDCG,还需计算理想情况下的最大DCG值(Ideal Discouted Cumulative Gain, IDCG)。因此根据真实情况下视频的请求数 z_i 对列表重新降序排序计算IDCG

$$IDCG = \sum_{i=1}^n \frac{z_i}{\log_2(i+1)} \quad (15)$$

然后,根据式(16)计算得到NDCG

$$NDCG = \frac{DCG}{IDCG} \quad (16)$$

4.2.2 性能指标

本文选择缓存命中率和延迟降低率作为两个关键指标来评估GCCS策略的性能。缓存命中率通常由本地存储是否存在用户请求的视频确定。为了便于描述,对于每个边缘节点 n ,假设在一个时间步 t 内接收到的请求为 $R_n^t = [r_1, r_2, \dots]$,其中 $r = (u, f, c, \hat{t})$, f 的大小为 s_f 。 $x_n^r \in \{0, 1\}$ 表示缓存状态, $x_n^r = 1$ 表示边缘节点 n 中缓存了请求 r 的相应视频,否则 $x_n^r = 0$ 。在本文的协作缓存策略中,为了降低缓存成本,边缘节点将不在多个节点中重复缓存获益较小的视频。然而,由于可以在邻居节点获得请求内容,相比从云端下载内容仍然能获得较快的传输速度,因此缓存命中率可由式(17)计算

$$P^{hit} = \frac{\sum_{r \in R_n^t} (x_{local}^r + (1 - x_{local}^r) x_{neighbor}^r)}{|R_n^t|} \quad (17)$$

与传输视频的时间相比,查询本地节点缓存列表的时间几乎可以忽略不计。在文献[4]的启发下,考虑边缘节点之间的协作延迟成本,使用 τ 表示邻居节点之间连接和查询的时延,并令 v_{uc} 表示用户和边缘节点之间的传输速率, v_{ic} 表示用户和云中心的传输速率。因此,时延表示为

$$t = \left. \begin{aligned} & \sum_{r \in R_n^t} ((1 - x_{local}^r) \tau \\ & + X^r v_{uc} s_f \\ & + (1 - X^r) v_{ic} s_f) \\ & X^r = \text{sgn}(x_{local}^r + x_{neighbor}^r) \end{aligned} \right\} \quad (18)$$

其中, $\text{sgn}(x)$ 为阶跃函数,定义为

$$\text{sgn}(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (19)$$

为了更直观地展示算法的性能,定义了相比于边缘节点无缓存情况下,用户等待时延的降低率

$$P^{delay} = \frac{\sum_{r \in R_n^t} v_{uc} s_f - t}{\sum_{r \in R_n^t} v_{uc} s_f} \quad (20)$$

4.3 实验结果分析

为了证明基于联邦学习的视频请求预测的优势,本研究对比了DRPN模型在不同训练模式下的预测质量,包括中心式训练、联邦训练和本地训练。如图5所示,由联邦学习训练得到的模型能达到中心式训练得到的模型相近的预测质量。在请求数据规律性较强的时间段,联邦训练的效果甚至比中心式训练的效果更好,因为联邦学习在多个节点共同训练模型的同时权衡了全局数据特性和局部特性。本地训练只训练本地数据集,没有学习其他地区的视频流行度规律,因此无法适应用户移动性所带来的视频流行度的规律变化,整体效果不佳。由于用户请求的统计特性在不同时段存在变化,在请求数据规律性较弱的时段,如请求数量非常少的时段,NDCG的值也普遍会下降。

图6描述了DRPN预测模型在不同训练模式下的收敛过程。在中心式训练方法下,预测模型在第60个通信轮次才达到收敛,但在联邦训练方法下,

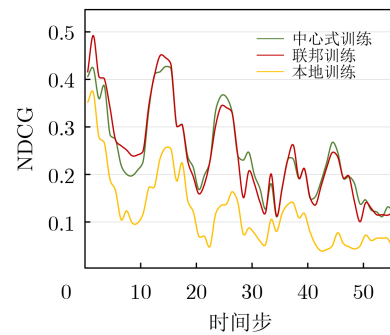


图5 不同训练模式下DRPN预测模型的训练质量(NDCG)

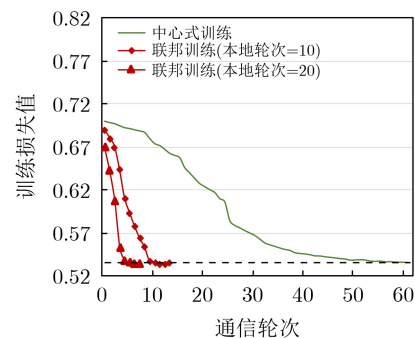


图6 DRPN预测模型在不同训练模式下的收敛情况

预测模型使用较少的通信轮次就能稳定收敛到相同效果。实验结果表明，联邦学习下的DRPN的收敛速度比中心式训练快，并且通过增加边缘节点上的本地训练轮次可以进一步减少模型收敛所需的通信轮次，从而降低通信成本。对于边缘节点而言，相比于总数据集，本地的数据集较小，计算成本也较小，因此可以通过增加本地计算的方式来降低通信成本。

图7和图8对比了各种缓存策略在不同条件下的性能表现，其中LRU, FIFO和 LFU为经典缓存替换方法，Ideal为在已知当天请求量排序的情况下，依次缓存之前出现过且请求量较高的内容的结果，决定了预测方法在缓存性能上的理论上限，但由于当天的内容请求情况无法提前获得，实际系统中无法实现。为了保证实验的公平，在测试对比方法时，本研究同样允许边缘节点间存在协作性，各边缘节点间可以协作地响应用户发出的视频请求。

图7是不同缓存策略在部署3个边缘节点，不同缓存容量下的性能变化。从图7(a)和图7(b)可以看出本文提出的GCCS策略的平均缓存命中率和平均时延降低率均高于经典缓存策略。首先，DRPN预测模型可以更好地预测视频未来被请求的概率，从而为缓存策略提供良好基础。其次，GCCS通过

贪心算法，以最大化时延收益为目的，让节点之间能够协作地完成本地缓存计划，不仅节省出更多空间来增加内容多样性以提高命中率，并且最大限度地降低了用户的等待时延。由于对时延的评价指标是降低率，因此即使是很小的提升，对于实际时延的影响依旧是很大的。此外，随着存储容量的增加，缓存性能指标也相应提高，这是因为有更多的空间来缓存相对不那么流行的视频，而在缓存空间受限时，GCCS策略很大程度缩小了缓存命中率与理论上限之间的距离。

图8则展现了不同缓存策略在固定缓存容量比为0.0011，不同边缘节点部署数量下的性能变化。从图8可以观察到边缘节点数目的增加对传统缓存策略的影响较小，但对于本文提出的GCCS策略，增加一定数目的边缘节点则能明显提升性能。因为随着节点的增加，每个节点服务的区域范围缩小，使其训练得到的预测模型更具有针对性，对此区域内关于视频未来请求的预测也会更准确。但当节点数量不断增加至多个节点的服务范围之间开始重叠时，再增加节点并不会带来额外的信息收益，反而增加了部署和维护节点的成本，因此并非部署越多的边缘节点就能获得越好的效果，实际情况下应该通过权衡成本和性能来决定部署边缘节点的数目。

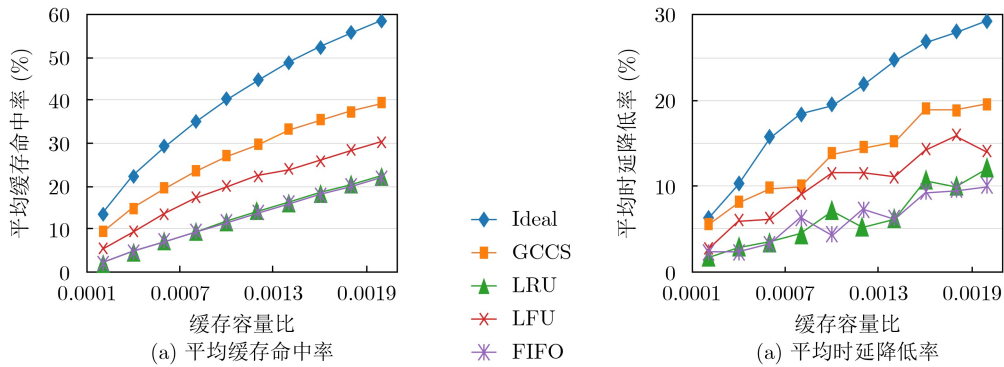


图 7 不同缓存策略在不同缓存容量比下的性能指标

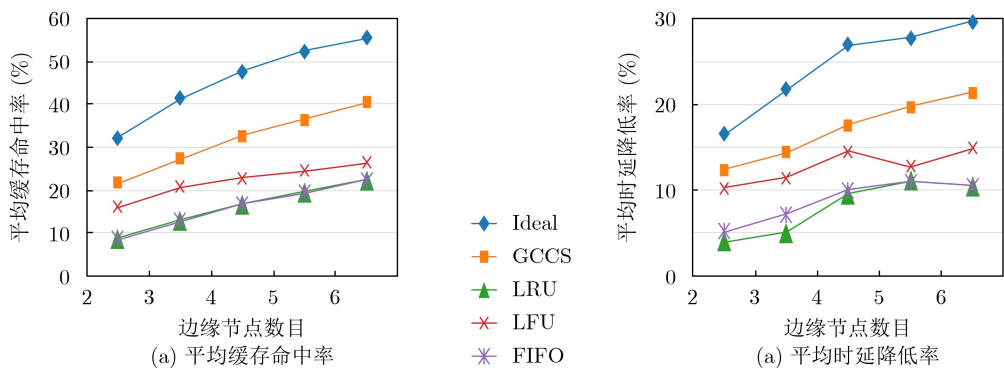


图 8 不同缓存策略在不同边缘节点数目下的性能指标

5 结束语

本文在移动边缘计算的背景下,提出了基于联邦学习的视频请求预测和视频协作缓存策略。联合已部署的多个边缘节点对提出的DRPN预测模型进行联合训练,从而降低通信成本,同时保护请求数据的隐私性,并根据后续真实请求情况的反馈主动更新模型,保证预测模型的准确性。此外,本文以最大化时延收益为目标,联合多个边缘节点协作地进行缓存,不仅有效降低了整体时延,还可以节约缓存成本,在有限的缓存空间中储存更多的内容,提高缓存命中率。由于在多个边缘节点进行协作时,需要无线通信,在真实的无线环境中存在的不确定性可能影响模型预测的准确性,我们将在下一步工作中考虑更真实场景限制下,提高方法的可用性,并通过与差分隐私、多方安全计算等方式的组合,提供更强的隐私保护能力。

参考文献

- [1] Intel. How 5G will transform the business of media and entertainment[R]. 2018.
- [2] MANSOURI Y and BABAR M A. A review of edge computing: Features and resource virtualization[J]. *Journal of Parallel and Distributed Computing*, 2021, 150: 155–183. doi: [10.1016/j.jpdc.2020.12.015](https://doi.org/10.1016/j.jpdc.2020.12.015).
- [3] LI Suoheng, XU Jie, VAN DER SCHAAR M, *et al.* Trend-aware video caching through online learning[J]. *IEEE Transactions on Multimedia*, 2016, 18(12): 2503–2516. doi: [10.1109/TMM.2016.2596042](https://doi.org/10.1109/TMM.2016.2596042).
- [4] THAR K, TRAN N H, OO T Z, *et al.* DeepMEC: Mobile edge caching using deep learning[J]. *IEEE Access*, 2018, 6: 78260–78275. doi: [10.1109/ACCESS.2018.2884913](https://doi.org/10.1109/ACCESS.2018.2884913).
- [5] RATHORE S, RYU J H, SHARMA P K, *et al.* DeepCachNet: A proactive caching framework based on deep learning in cellular networks[J]. *IEEE Network*, 2019, 33(3): 130–138. doi: [10.1109/MNET.2019.1800058](https://doi.org/10.1109/MNET.2019.1800058).
- [6] 杨静, 李金科. 带有特征感知的D2D内容缓存策略[J]. *电子与信息学报*, 2020, 42(9): 2201–2207. doi: [10.11999/JEIT190691](https://doi.org/10.11999/JEIT190691).
YANG Jing and LI Jinke. Feature-aware D2D content caching strategy[J]. *Journal of Electronics & Information Technology*, 2020, 42(9): 2201–2207. doi: [10.11999/JEIT190691](https://doi.org/10.11999/JEIT190691).
- [7] SAPUTRA Y M, HOANG D T, NGUYEN D N, *et al.* Distributed deep learning at the edge: A novel proactive and cooperative caching framework for mobile edge networks[J]. *IEEE Wireless Communications Letters*, 2019, 8(4): 1220–1223. doi: [10.1109/LWC.2019.2912365](https://doi.org/10.1109/LWC.2019.2912365).
- [8] 刘浩洋, 王钢, 杨文超, 等. 基于随机几何理论的流行度匹配边缘缓存策略[J]. *电子与信息学报*, 2021, 43(12): 3427–3433. doi: [10.11999/JEIT210493](https://doi.org/10.11999/JEIT210493).
LIU Haoyang, WANG Gang, YANG Wenchao, *et al.* Popularity matching edge caching policy based on stochastic geometry theory[J]. *Journal of Electronics & Information Technology*, 2021, 43(12): 3427–3433. doi: [10.11999/JEIT210493](https://doi.org/10.11999/JEIT210493).
- [9] MCMAHAN H B, MOORE E, RAMAGE D, *et al.* Communication-efficient learning of deep networks from decentralized data[C]. The 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, USA, 2017: 1273–1282.
- [10] GroupLens. Recommended for new research[EB/OL]. <https://grouplens.org/datasets/movielens/>, 2020.
- [11] MÜLLER S, ATAN O, VAN DER SCHAAR M, *et al.* Context-aware proactive content caching with service differentiation in wireless networks[J]. *IEEE Transactions on Wireless Communications*, 2017, 16(2): 1024–1036. doi: [10.1109/TWC.2016.2636139](https://doi.org/10.1109/TWC.2016.2636139).
- [12] JIANG Yanxiang, MA Miaoli, BENNIS M, *et al.* User preference learning-based edge caching for fog radio access network[J]. *IEEE Transactions on Communications*, 2019, 67(2): 1268–1283. doi: [10.1109/TCOMM.2018.2880482](https://doi.org/10.1109/TCOMM.2018.2880482).
- [13] GUO Dandan, CHEN Bo, LU Ruiying, *et al.* Recurrent hierarchical topic-guided RNN for language generation[C/OL]. The 37th International Conference on Machine Learning, 2020: 3810–3821.
- [14] CUI Zhiyong, KE Ruimin, PU Ziyuan, *et al.* Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values[J]. *Transportation Research Part C: Emerging Technologies*, 2020, 118: 102674. doi: [10.1016/j.trc.2020.102674](https://doi.org/10.1016/j.trc.2020.102674).
- [15] ALE L, ZHANG Ning, WU Huici, *et al.* Online proactive caching in mobile edge computing using bidirectional deep recurrent neural network[J]. *IEEE Internet of Things Journal*, 2019, 6(3): 5520–5530. doi: [10.1109/JIOT.2019.2903245](https://doi.org/10.1109/JIOT.2019.2903245).
- [16] CHEN Mingzhe, YANG Zhaoyang, SAAD W, *et al.* A joint learning and communications framework for federated learning over wireless networks[J]. *IEEE Transactions on Wireless Communications*, 2021, 20(1): 269–283. doi: [10.1109/TWC.2020.3024629](https://doi.org/10.1109/TWC.2020.3024629).
李光辉: 男, 教授, 博士生导师, 研究方向为边缘计算、物联网。
李宜璟: 女, 硕士生, 研究方向为边缘计算。
胡世红: 女, 博士, 研究方向为边缘计算。