



(12)发明专利申请

(10)申请公布号 CN 110784760 A

(43)申请公布日 2020.02.11

(21)申请号 201910871317.5

(22)申请日 2019.09.16

(71)申请人 清华大学

地址 100084 北京市海淀区100084信箱82
分箱清华大学专利办公室

(72)发明人 徐明伟 孟子立 陈婧 郭雅宁
孙晨

(74)专利代理机构 北京聿宏知识产权代理有限公司 11372

代理人 吴大建 张杰

(51)Int.Cl.

H04N 21/437(2011.01)

H04N 21/44(2011.01)

H04N 21/466(2011.01)

H04N 21/647(2011.01)

权利要求书2页 说明书9页 附图4页

(54)发明名称

一种视频播放方法、视频播放器及计算机存储介质

(57)摘要

本发明公开了一种视频播放方法、视频播放器及计算机存储介质,该方法应用于移动终端,该方法包括:播放由多个连续的视频块组成的预定视频;针对预定视频中的每个视频块,计算该视频块的第一播放状态,并采用ABR算法并根据第一播放状态计算与第一播放状态对应的该视频块的下一个视频块的比特率决策;将预定视频中所有视频块的第一播放状态及其对应的动作作为训练数据集;采用CART算法基于训练数据集生成决策树;将决策树部署到移动终端的视频播放器中并使视频播放器根据由决策树得到的比特率播放视频。本发明能够将复杂的ABR算法转化为算法简单的决策树,移动终端的视频播放器根据由上述决策树得到的比特率播放视频时,能够极大地提高用户体验。



1. 一种视频播放方法,应用于移动终端,其特征在于,包括:
 播放预定视频,所述预定视频由多个连续的视频块组成;
 针对所述预定视频中的每个视频块,执行以下步骤:
 计算该视频块的第一播放状态;
 采用ABR算法根据所述第一播放状态计算与所述第一播放状态对应的动作,其中,与所述第一播放状态对应的动作为该视频块的下一个视频块的比特率决策;
 将所述预定视频中所有视频块的第一播放状态及其对应的动作作为第一训练数据集;
 采用CART算法基于所述第一训练数据集生成用于确定播放视频用的比特率的决策树;
 将所述决策树部署到移动终端的视频播放器中;
 在所述视频播放器向预设的视频服务器发送请求,并在接收到视频服务器反馈的请求通过的消息后根据由决策树基于当前网络状态得到的比特率播放视频服务器反馈的视频。
2. 根据权利要求1所述的视频播放方法,其特征在于,还包括:
 对所述决策树进行优化;
 将所述决策树部署到移动终端的视频播放器中,包括:将优化后的决策树部署到移动终端的视频播放器中,
 根据由决策树基于当前网络状态得到的比特率播放视频服务器反馈的视频,包括:根据由优化的决策树基于当前网络状态得到的比特率播放视频服务器反馈的视频。
3. 根据权利要求2所述的视频播放方法,其特征在于,所述采用CART算法基于所述第一训练数据集生成决策树包括:
 在CART算法中采用贪婪算法选择所述第一训练数据集中的播放状态作为数据特征构建叶节点,直到叶节点数达到第一预设阈值或者所述第一训练数据集的Gini系数小于第二预设阈值。
4. 根据权利要求3所述的视频播放方法,其特征在于,在生成决策树的过程中采用的损失函数为 $\ell(r; r_0)$:

$$\ell(r; r_0) = \frac{(r - r_0)^2}{(R_{\max} - R_{\min})^2}, R_{\min} \leq r \leq R_{\max}$$
 其中, $r = \pi(s)$, $r_0 = \pi^*(s)$, π 为当前生成的决策树, π^* 为ABR算法, s 为视频块的播放状态; $R_{\max}$ 为预设的最大比特率, $R_{\min}$ 为预设的最小比特率。
5. 根据权利要求4所述的视频播放方法,其特征在于,对所述决策树进行优化包括:
 S1: 播放所述预定视频;
 针对所述预定视频中的每个视频块,执行步骤S2和S3:
 S2: 基于所述决策树计算该视频块的第二播放状态;
 S3: 采用所述ABR算法根据所述第二播放状态计算与所述第二播放状态对应的动作,其中,与所述第二播放状态对应的动作为该视频块的下一个视频块的比特率决策;
 S4: 汇总所述预定视频中的所有视频块的第一播放状态和第二播放状态,获得优化方法的播放状态;
 S5: 汇总所述预定视频中的所有视频块的第一播放状态对应的动作和所有视频块的第二播放状态对应的动作,获得优化方法的决策动作;
 S6: 将所述优化方法的播放状态和所述优化方法的决策动作作为第二训练数据集;

S7:将采用CART算法基于所述第二训练数据集生成的决策树作为优化后的决策树；
重复所述S1至S7,直到达到预设的最大迭代次数。

6.一种视频播放器,应用于移动终端,其特征在于,包括:

视频播放模块,用于播放预定视频,所述预定视频由多个连续的视频块组成;

计算模块,用于针对所述预定视频中的每个视频块,执行以下步骤:

计算该视频块的第一播放状态;

采用ABR算法根据所述第一播放状态计算与所述第一播放状态对应的动作,其中,与所述第一播放状态对应的动作为该视频块的下一个视频块的比特率决策;

第一训练数据集获取模块,用于将所述预定视频中所有视频块的第一播放状态及其对应的动作作为第一训练数据集;

决策树生成模块,用于采用CART算法基于所述第一训练数据集生成用于确定播放视频用的比特率的决策树;

部署模块,用于将所述决策树部署到所述视频播放器中;

收发模块,用于在向预设的视频服务器发送请求,并接收到视频服务器反馈的请求通过的消息后通知所述视频播放模块根据由决策树基于当前网络状态得到的比特率播放视频服务器反馈的视频。

7.根据权利要求6所述的视频播放器,其特征在于,还包括:

优化模块,用于对所述决策树进行优化,

所述部署模块还用于将优化后的决策树部署到所述视频播放器中,

所述视频播放模块还用于根据由优化的决策树基于当前网络状态得到的比特率播放视频服务器反馈的视频。

8.根据权利要求7所述的视频播放器,其特征在于,所述决策树生成模块用于在CART算法中采用贪婪算法选择所述第一训练数据集中的播放状态作为数据特征构建叶节点,直到叶节点数达到第一预设阈值或者所述第一训练数据集的Gini系数小于第二预设阈值。

9.根据权利要求8所述的视频播放器,其特征在于,所述决策树构建模块采用的损失函数为 $l(r; r_0)$:

$$l(r; r_0) = \frac{(r - r_0)^2}{(R_{max} - R_{min})^2}, R_{min} \leq r \leq R_{max}$$

其中, $r = \pi(s)$, $r_0 = \pi^*(s)$, π 为当前生成的决策树, π^* 为ABR算法, s 为视频当前的播放状态; R_{max} 为预设的最大比特率, R_{min} 为预设的最小比特率。

10.一种计算机存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1至5任意一项所述的视频播放方法。

一种视频播放方法、视频播放器及计算机存储介质

技术领域

[0001] 本发明涉及互联网信息技术领域,尤其涉及一种视频播放方法、视频播放器及计算机存储介质。

背景技术

[0002] 在现有的网络系统中,视频流量占网络总流量的很大一部分。同时,在线视频传输的需求近年来迅猛增长。为了优化用户在线观看视频的视频质量,自适应比特率(Adaptive Bit-rate,ABR)技术应运而生。ABR技术最早于2011年由学术界首先提出,旨在优化用户的体验质量(Quality of Experience,QoE)。简单来说,ABR算法通过在客户端基于当前网络情况的估计来选择最适合该用户的比特率进行视频传输。利用ABR技术,用户在线观看视频时,可以充分利用当前可用的网络带宽,同时尽量避免卡顿等情况的出现,从而提升用户的体验质量。

[0003] 在实际部署中,ABR算法需要在QoE需求的差异(有的用户希望视频清晰度尽可能高而不在乎卡顿,有的用户则相反)、网络吞吐量的波动(未来网络吞吐难以准确预测)以及决策与决策之间的关联效应(序贯决策过程中决策是前后依赖的)的共同作用下进行复杂的优化。现有技术中已经有各种不同的ABR算法的优化方案,例如,混合整数规划(Mixed Integer Linear Programming,MILP)、李雅普诺夫优化和通过深度神经网络优化等,以使在线视频播放达到最优的性能。

[0004] 然而,上述各种优化方案使得ABR算法在实际部署中存在很多问题。目前大部分视频都是在移动终端上播放的,由于上述ABR优化算法较复杂,而播放视频的移动终端的计算资源通常十分有限,难以支持复杂优化问题的求解,因此,视频内容提供商很难像传统方法那样将ABR算法直接集成到HTML页面中,在客户端播放器内进行部署,这一问题会随着未来ABR优化方法的进一步复杂化而变得更加严峻。

发明内容

[0005] 本发明要解决的技术问题是:现有技术中的ABR优化算法较复杂,导致无法直接在客户端播放器内进行部署,从而使视频播放时用户的观看体验较差。

[0006] 为解决上述技术问题,本发明提供了一种视频播放方法,应用于移动终端,其包括:

[0007] 播放预定视频,所述预定视频由多个连续的视频块组成;

[0008] 针对所述预定视频中的每个视频块,执行以下步骤:

[0009] 计算该视频块的第一播放状态;

[0010] 采用ABR算法根据所述第一播放状态计算与所述第一播放状态对应的动作,其中,与所述第一播放状态对应的动作为该视频块的下一个视频块的比特率决策;

[0011] 将所述预定视频中所有视频块的第一播放状态及其对应的动作作为第一训练数据集;

[0012] 采用CART算法基于所述第一训练数据集生成用于确定播放视频用的比特率的决策树；

[0013] 将所述决策树部署到移动终端的视频播放器中；

[0014] 在所述视频播放器向预设的视频服务器发送请求，并接收到视频服务器反馈的请求通过的消息后根据由决策树基于当前网络状态得到的比特率播放视频服务器反馈的视频。

[0015] 进一步的，所述方法还包括：

[0016] 对所述决策树进行优化；

[0017] 将所述决策树部署到移动终端的视频播放器中，包括：将优化后的决策树部署到移动终端的视频播放器中，

[0018] 根据由决策树基于当前网络状态得到的比特率播放视频服务器反馈的视频，包括：根据由优化的决策树基于当前网络状态得到的比特率播放视频服务器反馈的视频。

[0019] 进一步的，所述采用CART算法基于所述第一训练数据集生成决策树包括：

[0020] 在CART算法中采用贪婪算法选择所述第一训练数据集中的播放状态作为数据特征构建叶节点，直到叶节点数达到第一预设阈值或者所述第一训练数据集的Gini系数小于第二预设阈值。

[0021] 优选的，在生成决策树的过程中采用的损失函数为 $l(r; r_0)$ ：

$$[0022] \quad \ell(r; r_0) = \frac{(r - r_0)^2}{(R_{\max} - R_{\min})^2}, R_{\min} \leq r \leq R_{\max}$$

[0023] 其中， $r = \pi(s)$ ， $r_0 = \pi^*(s)$ ， π 为当前生成的决策树， π^* 为ABR算法， s 为视频块的播放状态； $R_{\max}$ 为预设的最大比特率， $R_{\min}$ 为预设的最小比特率。

[0024] 进一步的，对所述决策树进行优化包括：

[0025] S1：播放所述预定视频；

[0026] 针对所述预定视频中的每个视频块，执行步骤S2和S3：

[0027] S2：基于所述决策树计算该视频块的第二播放状态；

[0028] S3：采用ABR算法根据所述第二播放状态计算与所述第二播放状态对应的动作，其中，与所述第二播放状态对应的动作为该视频块的下一个视频块的比特率决策；

[0029] S4：汇总所述预定视频中的所有视频块的第一播放状态和第二播放状态，获得优化方法的播放状态；

[0030] S5：汇总所述预定视频中的所有视频块的第一播放状态对应的动作和所有视频块的第二播放状态对应的动作，获得优化方法的决策动作；

[0031] S6：将所述优化方法的播放状态和所述优化方法的决策动作作为第二训练数据集；

[0032] S7：将采用CART算法基于所述第二训练数据集生成的决策树作为优化后的决策树；

[0033] 重复所述S1至S7，直到达到预设的最大迭代次数。

[0034] 本发明还提供了一种视频播放器，应用于移动终端，其包括：

[0035] 视频播放模块，用于播放预定视频，所述预定视频由多个连续的视频块组成；

[0036] 计算模块，用于针对所述预定视频中的每个视频块，执行以下步骤：

[0037] 计算该视频块的第一播放状态；

[0038] 采用ABR算法根据所述第一播放状态计算与所述第一播放状态对应的动作，其中，与所述第一播放状态对应的动作为该视频块的下一个视频块的比特率决策；

[0039] 第一训练数据集获取模块，用于将所述预定视频中所有视频块的第一播放状态及其对应的动作作为第一训练数据集；

[0040] 决策树生成模块，用于采用CART算法基于所述第一训练数据集生成用于确定播放视频用的比特率的决策树；

[0041] 部署模块，用于将所述决策树部署到所述视频播放器中；

[0042] 收发模块，用于在向预设的视频服务器发送请求，并接收到视频服务器反馈的请求通过的消息后通知所述视频播放模块根据由决策树基于当前网络状态得到的比特率播放视频服务器反馈的视频。

[0043] 进一步的，所述视频播放器还包括：

[0044] 优化模块，用于对所述决策树进行优化，

[0045] 所述部署模块还用于将优化后的决策树部署到视频播放器中，

[0046] 所述视频播放模块还用于根据由优化的决策树基于当前网络状态得到的比特率播放视频服务器反馈的视频。

[0047] 进一步的，所述决策树生成模块用于在CART算法中采用贪婪算法选择所述第一训练数据集中的播放状态作为数据特征构建叶节点，直到叶节点数达到第一预设阈值或者所述第一训练数据集的Gini系数小于第二预设阈值。

[0048] 优选的，所述决策树构建模块采用的损失函数为 $l(r; r_0)$ ：

$$[0049] \quad \ell(r; r_0) = \frac{(r - r_0)^2}{(R_{\max} - R_{\min})^2}, R_{\min} \leq r \leq R_{\max}$$

[0050] 其中， $r = \pi(s)$ ， $r_0 = \pi^*(s)$ ， π 为当前生成的决策树， π^* 为ABR算法， s 为视频当前的播放状态； $R_{\max}$ 为预设的最大比特率， $R_{\min}$ 为预设的最小比特率。

[0051] 本发明还提供一种计算机存储介质，其上存储有计算机程序，所述计算机程序被处理器执行时实现上述任意一项所述的视频播放方法。

[0052] 与现有技术相比，上述方案中的一个或多个实施例可以具有如下优点或有益效果：

[0053] 应用本发明的视频播放方法，由于将计算极其复杂的原始ABR算法转化为算法简单、轻量级的决策树，极大地降低了对计算资源的消耗，并缩短了决策延迟，因此，可直接将其部署到客户端的视频播放器中，该视频播放器根据由上述决策树得到的比特率播放视频时，能够极大地提高用户体验。

附图说明

[0054] 通过结合附图阅读下文示例性实施例的详细描述可更好地理解本公开的范围。其中所包括的附图是：

[0055] 图1为本发明实施例的方法流程图一；

[0056] 图2为本发明实施例的方法流程图二；

[0057] 图3为本发明实施例的系统结构图一；

- [0058] 图4为本发明实施例的系统结构图二；
 [0059] 图5为本发明实施例中决策树的结构示意图及其决策效果图；
 [0060] 图6为本发明实施例中未经过优化的决策树的预测效果示意图；
 [0061] 图7为本发明实施例的算法示意图。

具体实施方式

[0062] 为使本发明的目的、技术方案和优点更加清楚，以下将结合附图及实施例来详细说明本发明的实施方式，借此对本发明如何应用技术手段来解决技术问题，并达成技术效果的实现过程能充分理解并据以实施。

[0063] 本发明的设计目标为将复杂的ABR算法转化为轻量级的、高效的线上部署模型，且需保证转化后的部署模型在性能上与原始ABR算法无差异。这其中，线性拟合、非线性拟合、策略摘要等方法均可作为目标转化模型的备选方案。本发明最终基于以下原因，采用决策树作为目标转化模型：

[0064] (1) 决策树表达能力丰富。由于决策树是一种非参数化的表达形式，其可以表达复杂的决策逻辑。决策树丰富的表达能力使其在转化过程中可以保证ABR算法的性能。如图5所示，即使在状态空间中，决策边界的非线性程度很高，决策树依然可以高保真地近似决策边界，这是因为其可以在需要的时候灵活地调整其决策粒度。

[0065] (2) 决策树足够轻量级。由于二元决策树(二叉树)由一系列条件判断构成，在实际部署时网络管理员可通过JavaScript中的分支语句将其轻量级地实现。部署一个100个叶子节点的决策树为HTML页面带来的额外大小还不到1%。

[0066] (3) 决策树决策逻辑与ABR算法类似。ABR算法在决策时也一般由一系列的条件判断组合而成。例如，优化QoE需要在目前缓冲区大小和网络吞吐量均较高(以避免视频卡顿)、当前分辨率也较高(以避免分辨率抖动)的前提下才可以选择高比特率。

[0067] 然而，决策树是一种监督学习方法，其为优化特定的损失函数(一般为平均预测误差)而设计。其通常需要一个较大的带有标签的数据集，以在整个状态空间内优化。数学上，这一优化过程可表示为：

$$[0068] \quad \hat{\pi} = \operatorname{argmin}_{\pi \in \Pi} (\mathbb{E}_{s \sim d_{\pi}} [\mathbb{I}_{\pi(s) \neq a}]) \quad (\text{式1})$$

[0069] 其中， d_{π} 为当使用决策树策略 π 进行优化时的状态空间分布， s 与 a 分别为观察到的状态及其对应的决策。然而，由于分布与流量吞吐量、视频长度以及策略内容等相互耦合，因此很难直接计算状态的概率分布。最近的一些研究工作通过在整个状态空间中均匀采样来详尽地搜索每个状态的动作，实际上这一方法既低效又有偏。状态空间的维度通常很高(Pensieve的状态空间有25维)，使得所有组合的列举效率低下。同时，由于状态空间中的状态在现实世界中并不是均匀分布，因此在状态空间中的均匀采样可能并不是无偏的，因此会降低性能。因此，我们采用虚拟播放器的设计，并使用真实的网络流量数据来模拟ABR算法。与数据包级仿真相比，虚拟播放器快速高效，因为它们只计算视频块级别的信息。然后，我们在模拟播放过程中收集状态-动作对。因为这些数据是用真实世界的流量数据生成的，所以它们与真实生产环境比是无偏的。

[0070] 但是，通过利用虚拟播放器，基于给定数据集将ABR算法转换为决策树也具有挑战性。由于ABR系统的级联效应，即使最终决策树的整体预测精度很高，转换后的决策树的性

能仍可能很差。如图6所示,尽管整体准确率很高,但一次错误的决策可能会将决策树带入到在训练时没有经历过的状态空间中去。决策树因为不了解该子空间应该如何处理,可能会随之犯更多错误。这将进一步推动决策树离开轨道并使性能恶化。为了应对这一挑战,受到模仿学习的最新进展的启发,本发明不断对决策树进行仿真实验,并让原始ABR算法(教师)纠正该决策树(学生)做出的错误决策。在循环迭代中,决策树将逐步学习如何在整个状态空间做出决策。

[0071] 基于上述分析,本发明实施例的算法如图7所示,为了将复杂的原始ABR算法转化为决策树,本发明实施例采用虚拟播放器来有效地模拟真实视频播放器的系统动态性,并采用模仿学习来提高决策树的保真度。本发明通过连续模拟决策树的性能,并根据原始ABR算法的结果纠正决策树所产生的错误。以下为本发明实施例中算法代码的示意:

Input: ABR Algorithm π^* .

Output: Decision Tree π_M .

```

1 (S, A)  $\leftarrow$  VirtualPlay( $\pi^*$ )
[0072] 2 foreach  $i \in [1, \dots, M]$  do
3    $\pi_i \leftarrow$  TrainDecisionTree(S, A)
4    $(S_i, A_i) \leftarrow$  VirtualPlay( $\pi_i$ )
5    $A_i^* \leftarrow$  Predict( $\pi^*$ ,  $S_i$ )
6   Aggregate  $S \leftarrow S \cup S_i$ ,  $A \leftarrow A \cup A_i^*$ 

```

[0073] 基于上述代码,本发明提供一种视频播放方法,其应用于移动终端,如图1所示,本方法包括以下步骤:

[0074] S110、播放预定视频,所述预定视频由多个连续的视频块组成;

[0075] 针对上述预定视频中的每个视频块,执行步骤S120和步骤S130:

[0076] S120、计算该视频块的第一播放状态;

[0077] S130、采用待部署的ABR算法并根据所述第一播放状态,计算与所述第一播放状态对应的动作,与所述第一播放状态对应的动作为该视频块的下一个视频块的比特率决策;

[0078] S140、将所述预定视频中所有视频块的第一播放状态及其对应的动作作为第一训练数据集;

[0079] 本实施例中,采用预先设置的虚拟播放器来播放预定视频,该预定视频为划分为视频块的网络流量数据集和视频摘要。对于每个ABR算法,本实施例首先在虚拟播放器中模拟该ABR算法,以收集初始状态-动作对 (S, A) 用于后续的决策树训练(上述算法代码的第1行)。本实施例中的决策树训练过程也即决策树的生成过程。虚拟播放器是基于跟踪的块级模拟器,可以精确模拟具有痕迹和视频清单的实际视频播放器的行为。对于某个待部署的ABR算法,虚拟播放器使用网络流量数据集和视频摘要作为算法的输入来进行模拟。在实际部署中,内容提供商可以使用公共网络流量数据集或收集历史数据进行模拟。此外,我们的评估表明,即使在训练阶段使用的网络流量数据集在统计上与测试环境中的网络流量数据特征不同,本方法 also 具有很强的泛化能力。

[0080] 具体而言,虚拟播放器不断地计算当前视频块的播放状态(即第一播放状态),该播放状态具体包括当前缓冲区的大小、当前的下载时间等参数。然后,待部署的原始ABR算

法获取上述第一播放状态,为下一个视频块生成比特率决策,该决策即为上述第一播放状态对应的动作。将该动作发送回虚拟播放器,虚拟播放器根据该动作继续播放下一个视频块,然后继续计算下一个视频块的状态,如此循环,直到预定视频播放完成。在这个过程中,我们能够采集到每个视频块的播放状态及其对应的动作,将这些状态-动作对初始化为(S,A),作为后续生成决策树的训练数据集。

[0081] S150、采用CART算法基于所述第一训练数据集生成决策树;

[0082] 如上述算法代码的第3行所示,本实施例首先使用分类和回归树训练(Classification and Regression Tree,CART)算法基于初始化后的状态-动作对(S,A)生成决策树 π (也称为学生)。本实施例中,我们并没有采用现有技术中的0-1损失来预测准确度(式1),而是使用归一化的平方损失作为决策树生成期间的训练损失,本实施例所采用的损失函数如下:

$$[0083] \quad \ell(r; r_0) = \frac{(r - r_0)^2}{(R_{\max} - R_{\min})^2}, R_{\min} \leq r \leq R_{\max} \quad (\text{式 } 2)$$

[0084] 其中, $r = \pi(s)$, $r_0 = \pi^*(s)$, π 为当前生成的决策树, π^* 为待部署的ABR算法, s 为视频当前的播放状态; $R_{\max}$ 为预设的最大比特率, $R_{\min}$ 为预设的最小比特率。

[0085] 使用平方损失背后的原理是希望惩罚学生(决策树)策略中的那些与教师(原始ABR算法)策略相差甚远的比特率误差,因为它们对视频卡顿等有更大的影响。然后,在CART算法中采用贪婪算法选择上述第一训练数据集中的播放状态作为数据特征构建叶节点,以最小化损失函数,直到叶节点数达到第一预设阈值或者第一训练数据集的Gini系数小于第二预设阈值,其中,第一预设阈值由网络运营商设置。当第一训练数据集的Gini系数小于第二预设阈值时,表明所有的样本已经被完全分开。

[0086] S160、将所述决策树部署到移动终端的视频播放器中,使所述视频播放器向预设的视频服务器发送请求,并在接收到视频服务器反馈的请求通过的消息后根据由决策树得到的比特率播放视频服务器反馈的视频。

[0087] 将复杂的原始ABR算法转化为简单的决策树后,就可以像现有技术那样将决策树直接部署到移动终端的视频播放器中。通过决策树不断生成的比特率决策,对网络视频进行播放,以改善用户的视频体验。具体地,视频播放器中的收发模块向预设的视频服务器发送视频请求,视频服务器向视频播放器的收发模块反馈请求通过的响应消息和网络视频,上述收发模块接收到该网络视频后,将其传送给视频播放器中的视频播放模块,该视频播放模块根据当前计算出的比特率对该网络视频进行播放。这个播放过程同样是对组成该网络视频的视频块进行播放的过程。

[0088] 为了提升所获得的决策树的性能,如图2所示,本实施例还包括:S170、在将决策树部署到移动终端之前,还将上述获得的决策树进行优化;S180、将优化后的决策树部署到移动终端的视频播放器中,并使视频播放器向预设的视频服务器发送请求,并在接收到视频服务器反馈的请求通过的消息后根据由优化后的决策树得到的比特率播放视频服务器反馈的视频。

[0089] 本实施例中,对决策树进行优化包括:

[0090] S1:播放所述预定视频;

[0091] 针对所述预定视频中的每个视频块,执行步骤S2和S3:

- [0092] S2:基于所述决策树计算当前视频块的第二播放状态;
- [0093] S3:采用待部署的ABR算法并根据所述第二播放状态,计算与所述第二播放状态对应的动作,与所述第二播放状态对应的动作为下一个视频块的比特率决策;
- [0094] S4:汇总所述预定视频中的所有视频块的第一播放状态和第二播放状态,获得优化方法的播放状态;
- [0095] S5:汇总所述预定视频中的所有视频块的第一播放状态对应的动作和所有视频块的第二播放状态对应的动作,获得优化方法的决策动作;
- [0096] S6:将所述优化方法的播放状态和所述优化方法的决策动作作为第二训练数据集;
- [0097] S7:将采用CART算法基于所述第二训练数据集生成的决策树作为优化后的决策树;
- [0098] 重复上述S1至S7,直到达到预设的最大迭代次数。因此,本实施例对决策树的优化是一种循环优化过程,让老师循环地纠正学生作出的策略错误,从而提高决策树的性能。
- [0099] 具体地,本实施例在虚拟播放器中模拟决策树 π_i ,并收集一系列新的状态-动作对 (S_i, A_i) (上述算法代码中的第4行)。目前,虽然学生 π_i (即步骤S150中生成的决策树)已经知道如何在面对培训状态时做出决策,但独立模拟 π_i 可能会导致表现不佳。如图6所示,由于级联效应,模拟中学生 π_i 经历的 S_i 中的许多状态可能在本实施例中的迭代的训练期间 (即上述步骤S1至S7) 没有经历过。我们仍需要在后续步骤中更正决策树策略。
- [0100] 因此,我们将 S_i 中的状态提供给原始ABR算法 π^* (教师),并收集教师生成的决策数据集 A_i^* (上述算法代码中的第5行)。最后,我们用当前的状态-动作对 (S_i, A_i^*) 汇总到总的学生的状态和教师的行为 (S, A) 中,并返回算法的第2行继续下一次迭代 (上述算法代码中的第6行)。在这种情况下,当在下次迭代中训练决策树 π_{i+1} 时,它将从最后一次迭代所犯的误差中吸取教训。循环将这样继续,直到达到用户设置的最大迭代次数 (M) 。然后,最后一次迭代生成的决策树将被部署到客户端视频播放器中。
- [0101] 下面对本发明实施例进行理论分析:
- [0102] 如上所述,网络运营商需要设置两个超参数:最大迭代次数 (M) 和叶子节点数 (第一预设阈值)。因此,我们提供了本方法实际部署决策树时的平均损失函数界限 (失真度) 的理论分析。我们首先证明本实施例定义的损失函数同时具有Lipschitz性和强凸性:
- [0103] 结论1式2中的 $l(r; r_0)$ 同时具有Lipschitz性与强凸性。
- [0104] 证明: $\forall r_0, r_1, r_2 \in [R_{\min}, R_{\max}]$, 我们有:
- [0105] $|l(r_1; r_0) - l(r_2; r_0)| = |(r_1 - r_0)^2 - (r_2 - r_0)^2| =$
- [0106] $|r_1 + r_2 - 2r_0| \cdot |r_1 - r_2| \leq 2(R_{\max} - R_{\min}) |r_1 - r_2|$ (式3)
- [0107] 最后一个不等号成立是因为 $l(r; r_0)$ 的定义域在 $[R_{\min}, R_{\max}]$ 上。因此 $l(r; r_0)$ 具有Lipschitz性, Lipschitz常数为 $\mathcal{L} = 2(R_{\max} - R_{\min})$ 。类似地,我们同样可以证明 $l(r; r_0)$ 具有强凸性。 $\forall \lambda \in [0, 1]$, 有:

$$\ell(\lambda r_1 + (1 - \lambda)r_2; r_0) \leq \lambda \ell(r_1; r_0) + (1 - \lambda) \ell(r_2; r_0) - \left(\frac{\nu}{2}\right) \lambda(1 - \lambda)(r_1 - r_2)^2$$

[0108] (式4)

[0109] 其中,强凸性系数 $\nu=2/(R_{\max}-R_{\min})^2$ 。证毕。

[0110] 由于损失函数 $\ell(r; r_0)$ 同时具有Lipschitz性与强凸性,我们可以将Ross等人发表在AISTATS 2011上的工作进行扩展。因此,我们可以得到,当使用本方法产生的决策树独立处理视频时,平均损失函数的上界满足如下结论:

[0111] 结论2对任意 $\delta>0$,当训练时损失函数值为 ε_M 时,存在策略 $\hat{\pi} \in \{\pi_1, \dots, \pi_M\}$,使得平均损失函数满足:

$$\mathbb{E}_{s \sim d_{\pi}}[\ell(\hat{\pi}(s); \pi^*(s))] \leq \varepsilon_M + \Theta\left(\frac{1}{T}\right)$$

[0112] (式5)

[0113] 当 $M = \Theta\left(T \log\left(\frac{1}{\delta}\right)\right)$ 时,上式成立概率大于 $1-\delta$ 。T是模拟播放其中的视频块数。

[0114] 证明:令 $Q_t^{\pi'}(s, \pi)$ 为在初始状态s下采取动作a,并随后t步内都采用策略 π' 的成本,则:

$$Q_t^{\pi'}(s, \pi) = (a - \pi^*(s_1))^2 + \sum_{\tau=2}^t (\pi'(s_{\tau}) - \pi^*(s_{\tau}))^2$$

[0115] (式6)

[0116] 其中, s_{τ} 为在 τ 时刻的状态。所以有:

$$Q_{T-t+1}^{\pi^*}(s, a) - Q_{T-t+1}^{\pi^*}(s, \pi^*(s)) = \frac{(a - \pi^*(s))^2}{(R_{\max} - R_{\min})^2} \leq 1 \triangleq u$$

[0117] (式7)

[0118] 根据Ross等人的工作,结论2证毕。

[0119] $\hat{\pi}$ 可以通过不同迭代的决策树之间的交叉验证找到,这通常是我们实验中最后一次迭代的决策树 π_M 。因此,我们为本方法的失真度提供了上限。训练时的损失函数值 ε_M 与原始ABR算法的复杂性和叶节点的数量N(决策树的表达能力)有关。

[0120] 本发明还提供一种视频播放器,其应用于移动终端,如图3所示,该视频播放器包括:

[0121] 视频播放模块,用于播放预定视频,所述预定视频由多个连续的视频块组成;

[0122] 计算模块,用于针对所述预定视频中的每个视频块,执行以下步骤:

[0123] 计算该视频块的第一播放状态;

[0124] 采用待部署的ABR算法并根据所述第一播放状态,计算与所述第一播放状态对应的动作,与所述第一播放状态对应的动作为该视频块的下一个视频块的比特率决策;

[0125] 第一训练数据集获取模块,用于将所述预定视频中所有视频块的第一播放状态及其对应的动作作为第一训练数据集;

[0126] 决策树生成模块,用于采用CART算法基于所述第一训练数据集生成决策树;

[0127] 部署模块,用于将所述决策树部署到上述视频播放器中;

[0128] 收发模块,用于向预设的视频服务器发送请求,并在接收到视频服务器反馈的请求通过的消息后通知所述视频播放模块根据由决策树得到的比特率播放视频服务器反馈

的视频。

[0129] 进一步的,如图4所示,该视频播放器还包括:

[0130] 优化模块,用于对所述决策树进行优化,

[0131] 所述部署模块还用于将优化后的决策树部署到视频播放器中,

[0132] 所述视频播放模块还用于根据由优化的决策树得到的比特率播放视频服务器反馈的视频。

[0133] 在本播放器的实施例中,决策树生成模块用于在CART算法中采用贪婪算法选择所述第一训练数据集中的播放状态作为数据特征构建叶节点,直到叶节点数达到第一预设阈值或者所述第一训练数据集的Gini系数小于第二预设阈值。

[0134] 在本播放器的实施例中,决策树生成模块采用的损失函数为 $\ell(r; r_0)$:

$$[0135] \quad \ell(r; r_0) = \frac{(r - r_0)^2}{(R_{\max} - R_{\min})^2}, R_{\min} \leq r \leq R_{\max}$$

[0136] 其中, $r = \pi(s)$, $r_0 = \pi^*(s)$, π 为当前生成的决策树, π^* 为待部署的ABR算法, s 为视频当前的播放状态; $R_{\max}$ 为预设的最大比特率, $R_{\min}$ 为预设的最小比特率。

[0137] 本发明所述的视频播放器,其具体的工作原理、工作流程等可参见本发明所提供的视频播放方法,此处不再对相同的技术内容进行赘述。

[0138] 本发明还提供一种计算机存储介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现如本发明实施例所述的视频播放方法。

[0139] 本发明为ABR算法部署以用于视频播放的实用性设计,可通用于多种在线视频客户端设备,包括但不限于:个人电脑、智能手机、平板电脑、智能电视等。

[0140] 本发明提供了一种视频播放方法及视频播放器,以网络管理员所希望选取的ABR算法为输入,通过自动将复杂ABR算法(如MILP、神经网络)转化为轻量级、可直接部署的决策树算法,以降低ABR算法部署的资源消耗。本发明在传统的直接部署方案中增加了轻量化转化步骤,为复杂ABR算法的实际部署提供支持。同时,为确保转化时的性能保障,本发明提出通过模仿学习进行循环迭代拟合,使得转化后的决策树与转化前的复杂ABR算法性能相近,也有助于加速最新ABR技术的实际落地。

[0141] 本发明鉴别了复杂ABR算法直接部署的局限性,创新性地设计了轻量化部署转化方法,提高了ABR算法部署的实用价值;本发明通过对多种转化目标方案的分析,创新性地提出选用决策树作为实际上线部署的计算方案,以降低ABR算法决策延迟、内存消耗以及页面大小资源消耗;本发明创新性地ABR算法客户端直接部署前引入轻量化转化步骤,在不用改变管理员训练或设计新ABR算法的前提下,通过将实际部署的算法转换为决策树,对复杂ABR算法的资源消耗进行优化;本发明还分析了视频传输中序贯决策过程的依赖性,创新性地设计了基于模仿学习的循环迭代拟合转化算法,能确保转换前后ABR算法性能不损失。将本发明生成的决策树部署到移动终端的视频播放器中时,视频播放器向视频服务器请求视频并根据由决策树得到的比特率播放该视频,能够极大地提高用户的视频体验。

[0142] 虽然本发明所公开的实施方式如上,但所述的内容只是为了便于理解本发明而采用的实施方式,并非用以限定本发明。任何本发明所属技术领域内的技术人员,在不脱离本发明所公开的精神和范围的前提下,可以在实施的形式上及细节上作任何的修改与变化,但本发明的保护范围,仍须以所附的权利要求书所界定的范围为准。

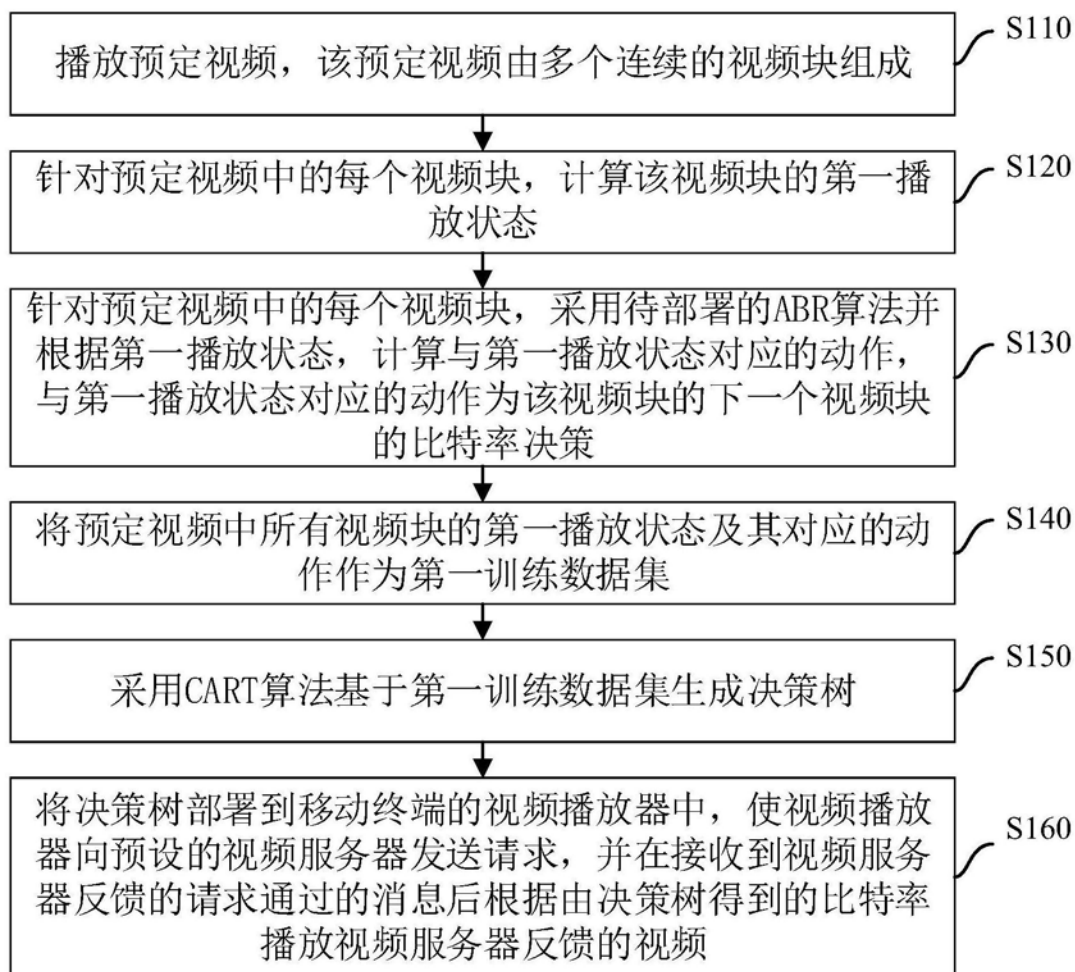


图1

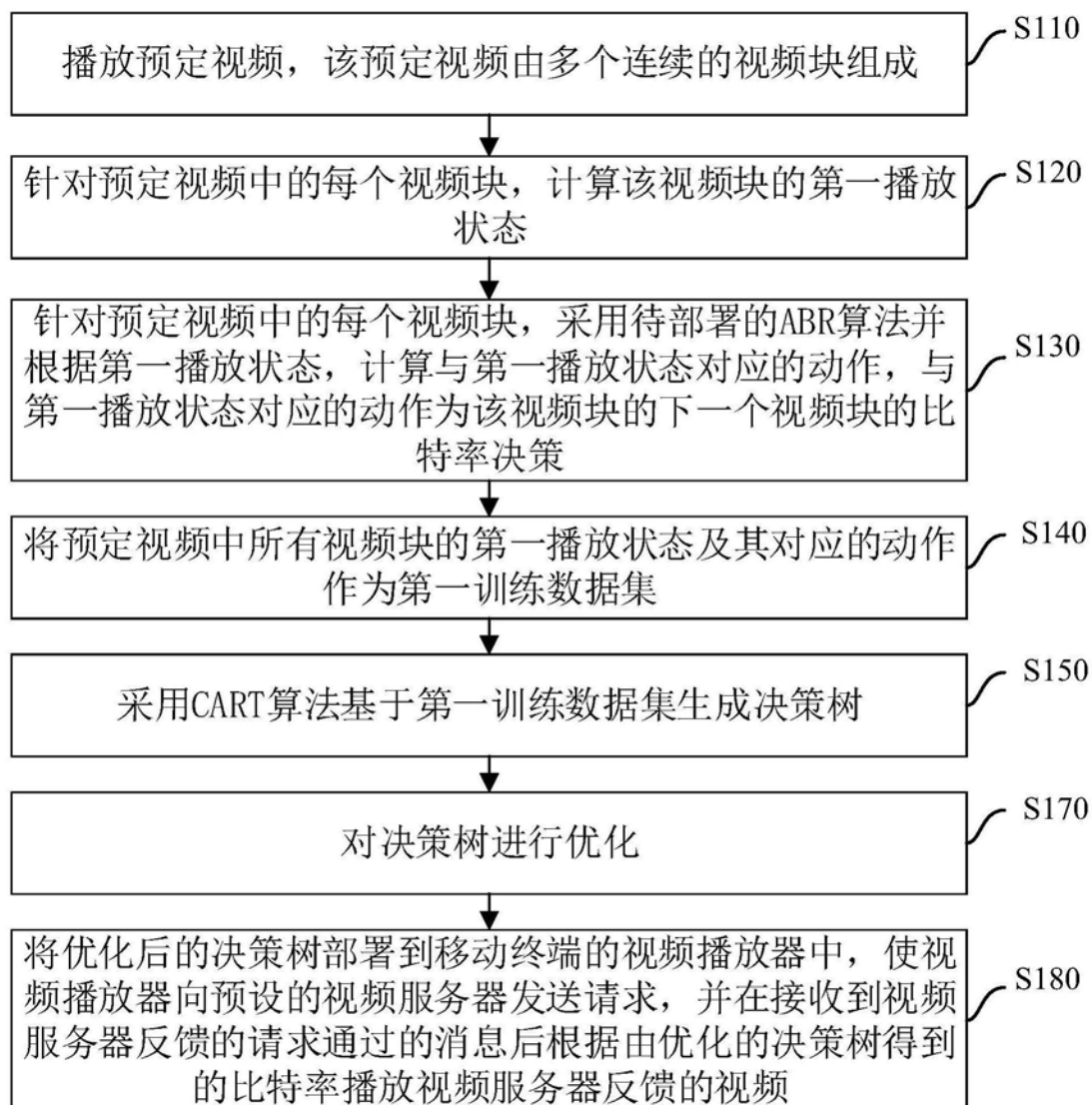


图2



图3

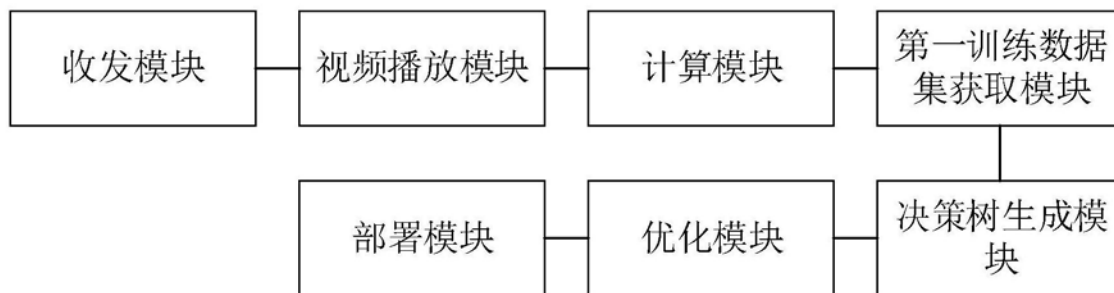


图4

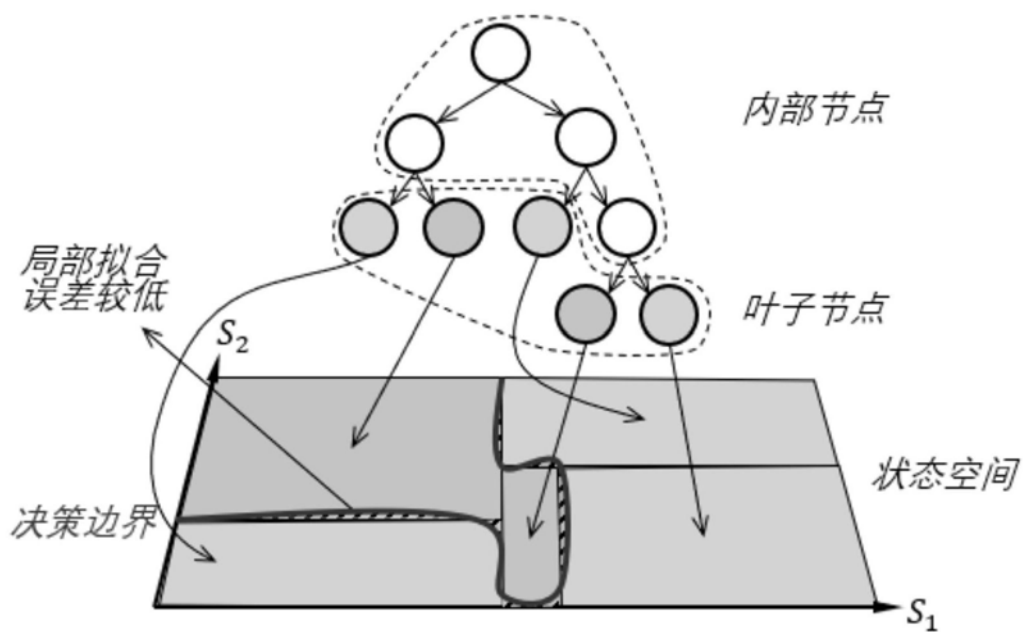


图5

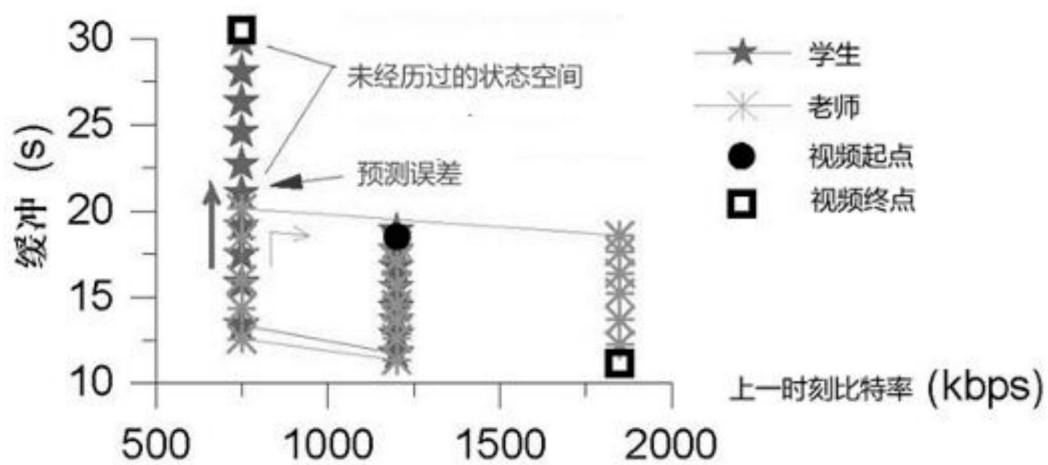


图6

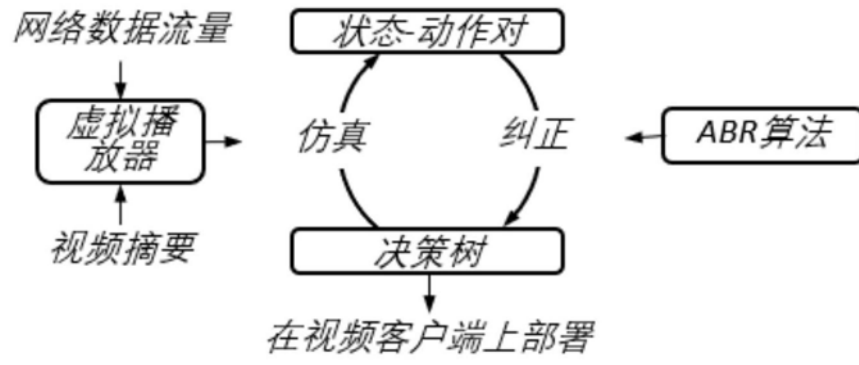


图7