

基于在线视频学习的用户观看行为抽取方法

王东^a, 林宏^b, 左欣^c

(贵州师范学院 a. 数学与计算机科学学院; b. 教育信息网络中心; c. 继续教育学院, 贵阳 550018)

摘要:针对在线视频学习中用户观看行为的记录与呈现问题,对在线视频播放器进度条进行了扩展设计,实现用户观看行为的采集、存储与呈现。首先使用 div 标签作为进度条容器,再用 span 标签记录用户播放行为,定时将用户观看行为数据异步写入服务端。该方法能完整抽取用户观看行为,使学习者精确观察了解自己的学习过程,提高了用户体验,也为后期学习行为的分析和大数据处理储备数据。

关键词:在线学习;视频播放器;观看行为;进度条

中图分类号:TP391

文献标志码:A

引言

随着互联网 Web2.0 和云计算等技术的成熟,网络在线学习得到了前所未有的发展,近年来出现了一批在线学习平台。学习者通过在线学习平台参与各种学习活动,其中,观看视频是最核心的学习环节。平台通过自动记录学习者的观看行为,将其作为评价或考核学习者的主要依据^[1-3]。目前,多数在线学习平台记录观看行为的方法仍比较简单,仅记录学习者播放视频的时间长度,该方法未完整记录下学习者的各种观看行为。为了强制学习者必须观看所有视频片段,一些学习平台隐藏了视频播放器自带的进度控制功能,视频只能从头到尾按时间顺序播放。这种方式限制了学习者的学习自主性,容易使学习者产生观看疲劳,失去学习兴趣。本文对视频播放器的进度条进行了扩展设计,实现对学习者观看行为的实时采集、存储和呈现,基本思路是:(1)重新设计播放器进度条,用不同颜色分块表示视频片段的学习状态,学习者对观看过程有完全的播放控制权,以视频片段被播放的覆盖率作为学习的

评价标准;(2)播放进度条自动记录和存储学习者的详细观看行为数据,为学习者学习行为的跟踪与分析提供基本依据;(3)进度条提供播放详情的直观呈现,让学习者了解自己的学习过程。

1 视频播放进度条设计

视频播放器自带的播放进度条只能显示当前播放的时间点,不能更加详细的呈现已播放的各个片段。因此,我们对视频播放器的进度条进行扩展设计。用一个 div 作为自定义进度条的容器,其背景色作为进度条的背景色;用户的每个观看行为定义为一个播放片段,每个播放片段用一个 span 标签表示,span 标签置于 div 容器中,span 标签的背景色作为进度条的前景色;用 span 标签的 left 属性(左边距)代表播放起点,width 属性表示播放片段的时间长度。每个播放片段在用户触发播放器的播放行为事件时被初始化,每当一个播放片段开始时,在 div 容器中追加一个新的 span 标签,span 标签在 div 容器中的位置代表播放开始时间点,span 标签的初始长度为 0。随着视频播放进度的推进,通过实时同步改变 span 标签的 width 属性值,

收稿日期:2015-10-23

基金项目:贵州省 2014 年本科教学质量工程项目(黔教高发[2014]378 号)

作者简介:王东(1978-),男,贵州贵阳人,副教授,主要从事自然语言处理方面的研究,(E-mail)wd_gz@163.com

从而模拟出播放进度不断增加的效果。图 1 为进度条生成流程图。

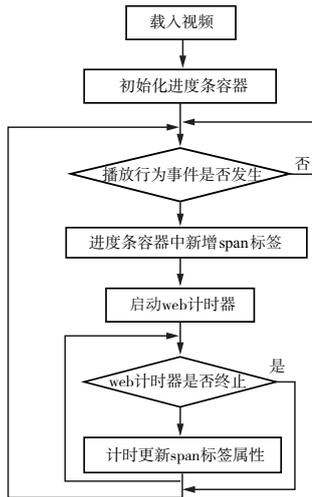


图 1 进度条生成流程图

实时同步需要用 Web 计时器采集播放器当前播放点时间,因此,Web 计时器的准确性很关键。JavaScript 提供的计时器最常用,但易受环境干扰,计时不准确,因此,选用了 jQuery^[4]的 timer 插件,计时准确性较好。当通过计时器采集到当前播放点后,由于进度条和视频播放器自带进度条所使用的度量单位不相同(分别为像素和时间),需要进行单位转换,转换关系为:

$$PxtoTime = px * div.width / totaltime \quad (1)$$

$$TimetoPx = time * totaltime / div.width \quad (2)$$

式(1)将进度条当前进度转换为播放器的当前播放时间, px 为当前 span 标签的 $width + left$, $div.width$ 为进度条总长度, $totaltime$ 为视频总时长;式(2)将播放器的当前播放时间转换为进度条当前进度, $time$ 为视频已播放时长。

视频播放过程中用户执行一系列播放控制,将在 div 容器中生成多个对应的 span 标签,它们分散或者重叠地分布在 div 容器中,由于 span 标签的背景色是相同的,在用户看来,是一个整体,从而完全模拟出分块进度条的效果。图 2 描述了多个播放行为后进度条的内部结构。

图 2 是一次视频观看会话中分块进度条生成的分解图。状态①表示进度条容器为空,用户未执行任何观看行为;状态②表示视频载入后用户执行了播放并持续观看了一个时间片段;状态③表示用户执行暂停后再继续播放了一个时间片段;状态④表示用户执行了拖动播

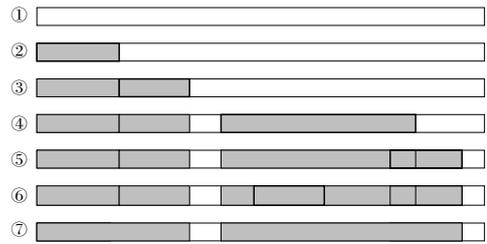


图 2 视频播放观看会话分解图

放,并持续观看了一个时间片段;状态⑤表示用户将播放器进度往前拖动播放,观看的时间片段与之前的时间片段有相交;状态⑥表示用户将播放器进度往前拖动播放,观看的时间片段包含于之前的时间片段中;状态⑦表示状态叠加后呈现的整体效果。

2 用户观看行为数据抽取

在线视频观看行为数据对进行教育大数据分析具有不可估量的价值,通过对海量学习行为进行深入挖掘,对用户的学习行为进行跟踪、记录并建模分析,就能全面跟踪、掌握学习者的学习行为、学习过程和学习特点,从而有针对性地改善课程视觉效果和教学方法^[5-7]。所设计的进度条本身具备存储用户学习行为的功能,但通过网页上的 html 标签存储用户观看行为数据是暂时的,当页面被刷新或重新载入时数据会被清除。因此,需要将用户观看行为数据存储到服务端数据库中。为了减少客户端与服务端连接的次数,客户端并不会实时向服务器提交行为数据,采取定时提交行为数据的方法。通过定义一个 DBLOCK 计时器,每隔一定时长向服务器提交一次行为数据。客户端向服务器提交的行为数据为进度条容器中所有 span 标签上的属性值集合,通过遍历 div 容器中的 span 标签并获取属性值,得到一个二维数组,再用 JSON.stringify()方法处理后作为 \$.post 提交方法的参数传递给后台处理程序。后台处理程序用 json_decode 方法解析后还原成二维数组,将二维数组中的每行插入到行为数据表中。在向服务器提交数据时,为了不刷新页面,采用 AJAX^[8-9]技术提交数据,用 JSON^[10-11]作为数据交换格式。同时为了避免写入重复数据,写入前先将同次视频观看会话的所有记录删除,再将前端播放数据写入数据库。

在 span 标签自定义 6 个属性 ($action, playdatatime, playstarttime, playendtime, dragstarttime, dragendtime$),

用以记录用户的观看行为。其中, *action* 表示播放类型,其取值为{‘play’, ‘pauseplay’, ‘dragplay’}, 分别表示载入后播放、暂停后播放和拖动播放; *playdatetime* 为初始化视频时系统的当前时间; *playstarttime* 存储当前片段播放的开始时间点, *playendtime* 存储当前片段播放的结束时间点; *dragstarttime* 表示拖动起点; *dragendtime* 表示拖动结束点。*dragstarttime* 和 *dragendtime* 只在 *action* 取值 *dragplay* 时有效,代表拖动起点和拖动结束点;当 *action* 取值 *play* 或 *pauseplay* 时, *dragstarttime* 和 *dragendtime* 被定义为0。相应地,在数据库中建立学习行为记录表存储每个 *span* 标签的属性值(表1)。

表1 学习行为记录表

字段名	数据类型	字段描述
videoID	Int	视频 ID 号
userID	Int	用户 ID 号
action	Varchar	观看行为类型
playdatetime	Datetime	视频观看会话时间
left	Int	观看时间片段距进度条容器左边距
width	Int	观看时间片段长度
playstarttime	Float	观看时间片段开始时间点
playendtime	Float	观看时间片段结束时间点
dragstarttime	Float	拖动开始时间点
dragendtime	float	拖动结束时间点

表1中每一条记录代表用户观看的一个时间片段, *videoID* 唯一标识一个视频, *userID* 标识一个用户。给定 *videoID* 和 *userID* 条件下, *playdatetime* 值相同的所示记录表示一个用户的一次观看会话的所有观看行为。通过 *left* 字段和 *width* 字段,可以在客户端页面上生成历史播放详情。通过统计 *playstarttime* 和 *playendtime* 字段可以获得时间片段长度,统计: *dragstarttime* 和 *dragendtime* 可以获得播放过程中被跳过的视频片段。当海量数据记录形成后,这些数据将成为大数据分析的有力证据,比如:当某个视频片段经常被用户跳过,就有可能是这部分视频制作不清,或者是课程内容安排不当。

要记录视频观看行为数据,就要准确捕获用户的播放行为。大多数视频播放器都提供一系列监听事件接口,只需要在监听事件中编写好相应的处理程序即可。本文所使用的是 *ckplayer* 播放器,表2中列出了使用到的监听事件及该事件中要执行的主要动作。

表2 监听事件列表

监听事件	执行动作
playHandler: 播放	a. 如果是首次单击播放,开始一个播放会话,新增一个 <i>action</i> 为 ‘play’ 的 <i>span</i> 标签,初始化 <i>span</i> 标签;b. 如果是暂停后播放,增加一个 <i>action</i> 为 <i>pauseplay</i> 的 <i>span</i> 标签,初始化 <i>span</i> 标签;c. 开启数据保存计时器;d. 开启 <i>web</i> 计时器。
endedHandler: 视频播放结束	a. 关闭 <i>web</i> 计时器;b. 关闭数据保存计时器
sendNetStream Handle:接收到 视频流,说明视 频加载成功, 随时可以播放	a. 初始化 <i>web</i> 计时器;b. 初始化数据保存计时器
bounceHandler: 鼠标在进度条上 松开时的时间点	a. 记录拖放点时间;b. 如果是首次直接拖动播放,开始一个播放会话;c. 增加一个 <i>action</i> 为 ‘ <i>pauseplay</i> ’ 的 <i>span</i> 标签,初始化 <i>span</i> 标签;c. 开启数据保存计时器;d. 开启 <i>web</i> 计时器;d. 清除拖放点时间。
seekingHandler: 跳转的时间	a. 获取跳转点时间;b. 如果拖动起点为0,表示视频载入后直接拖动播放,设置本次视频观看会话时间,新增当前播放 <i>span</i> 标签,否则如果跳转点不为0,新增一个 <i>span</i> 标签。

3 历史观看行为数据呈现

载入历史播放数据的过程就是将数据库中符合条件的记录在 *Web* 端生成相应标签的过程。主要呈现内容包括两个方面:一是历史总播放进度,将当前观看者、当前视频的所有历史记录进行合并后在前端呈现一条进度条;二是历史播放详情,将当前观看者、当前视频的所有记录按播放时间会话分组合并,每个会话形成一条进度条呈现在前端。

用户在一次观看会话中往往会触发若干观看行为,数据表中存储了许多记录,如果将每条记录在前端页面上生成一个 *span* 标签,在初始化前端页面时会增加负载。因此,在后台处理程序中将每次播放的历史数据进行合并后再输出到前端。由于数据表中的每一条记录对应一个 *span* 标签,采用 *span* 标签的合并来描述数据表记录的合并操作。分析观看会话中产生的 *span* 标签, *span* 标签之间可能产生三种关系:(1)相离关系:*span1* 和 *span2* 无重叠;(2)相交关系:*span1* 和 *span2* 部分区域重叠;(3)包含关系:*span2* 完全位于 *span1* 区域内,如图3所示。

合并后的状态:对于(1)来说,不需要合并;对于(2),修改 *span1* 的 *width* 值,删除 *span1*;对于(3),直接

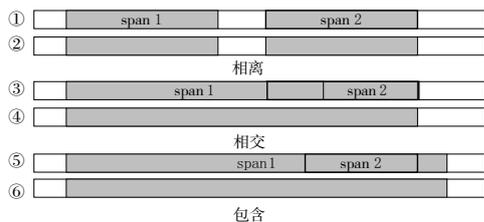


图 3 span 标签关系图

删除 span2。

观看行为记录的合并过程描述如下:

Step 1:对所有 span 标签按 *left* 属性值升序排列;

Step2:若 span 标签数大小 2,反复执行:

Step 2. 1 从第 1 个标签开始至第 $n-1$ 个标签,取相邻 span 标签,反复执行:

Step 2. 2 若两个 span 标签为包含关系,删除第二个 span 标签,跳 step2;

Step 2. 3 若两个 span 标签为相交关系,更新第一个 span 标签的 *width* 属性值,删除第二个 span 标签,跳 step2。

4 进度条标尺及参考线设计

带标尺及参考线的进度条能显著提高用户体验,用户可直观地观察到视频的时间刻度及播放进度。生成标尺首先需要准备一张带刻度线的图作为标尺容器的背景图,背景图水平平铺横向填充标尺容器。标尺的刻度值则需要根据视频时长动态生成。

参考线的主要作用包括两个方面:一方面作为对齐播放片段的垂直参考线,当鼠标移动到进度条容器区域内会跟随鼠标移动,方便用户察看观看片段的时间点;另一个方面是作为当前播放头实时向后移动。为协调好参考线的呈现,当鼠标位于进度条内时,使参考线跟随鼠标移动,当鼠标离开进度条区域外时,使参考线作为播放头实时向后移动。

5 用户观看行为抽取实证分析

为验证本文所述方法的有效性,准备了一个时长为 3.37 分钟的视频文件,在播放视频过程中多次模拟学习者的各种观看行为,分别执行载入视频后播放、播放中暂停、暂停后继续播放、播放中拖动、暂停中拖动、不同方向的拖动播放等观看行为,并在此过程中进行人工记载。通过历史观看行为数据的重新呈现。如图 4 所示,与人工记载的学习断点完全吻合,学习者的各种观看行

为被准确的记录了下来。

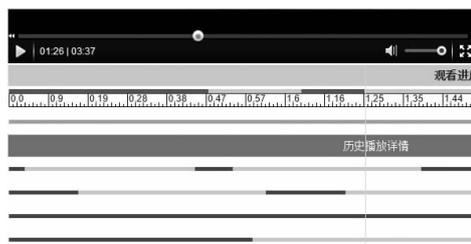


图 4 历史播放详情运行图

6 结束语

知识全球化的 MOOC 运行,以一种新型的学习方式正在兴起,这种以学习者在线观看视频为主的学习形式,不仅要求视频内容制作要有过硬的质量,也对播放过程的用户体验及用户观看行为数据的分析提出了很高的要求。本文基于开源播放器扩展了播放进度条的设计,有效解决了用户观看行为数据的记录与呈现问题,使学习者能够精确地观察了解自己的学习过程,增强了播放器的用户体验,也为用户学习行为大数据分析储备了数据。

参考文献:

- [1] 杨金来,洪伟林,张翼翔.网络学习行为的实时监控研究与实践[J].开放教育研究,2008,14(4):87-92.
- [2] 杨玉芹.MOOC 学习者个性化学习模型建构[J].中国电化教育,2014(6):6-10.
- [3] 吕佳宁,魏芳.大数据下网络视频类用户行为分析[J].新闻爱好者,2014(7):38-40.
- [4] 潘杰,周传生.基于 jQuery 框架的 Web 研究与实现[J].沈阳师范大学学报:自然科学版,2015,33(1):96-99.
- [5] 蒋卓轩,张岩,李晓明.基于 MOOC 数据的学习行为分析与预测[J].计算机研究与发展,2015,52(3):614-628.
- [6] 胡艺龄,顾小清,赵春.在线学习行为分析建模及挖掘[J].开放教育研究,2014,20(2):102-108.
- [7] 彭文辉,杨宗凯,黄克斌.网络学习行为分析及其模型研究[J].中国电化教育,2006(10):31-35.
- [8] 吕林涛,万经华,周红芳.基于 Ajax 的 Web 无刷新页面快速更新数据方法[J].计算机应用研究,2006,23

- (11):199-223. [J].西安石油大学学报:自然科学版,2011,26(1):95-98.
- [9] 熊文,熊淑华,孙旭,等.Ajax技术在Web2.0网站设计中的应用研究[J].计算机技术与发展,2012,22(3):145-148. [11] 李志秀,张军,陈光,等.jQuery Ajax异步处理JSON数据在项目管理系统中的应用[J].云南大学学报:自然科学版,2011,33(S2):247-250.
- [10] 屈展,李婵.JSON在Ajax数据交换中的应用研究

Method of User Watching Behavior Extraction Based on Online Video

WANG Dong^a, LIN Hong^b, ZUO Xin^c

(a. School of Mathematics and Computer Science; b. Educational Information and Network Technology Center; c. School of Continuing Education, Guizhou Education University, Guiyang 550018, China)

Abstract: For online video learning user watching behavior data recording and presenting problems, the online video player progress bar is designed to extend, to achieve the user viewing behavior data acquisition, storage and presentation. Firstly, it uses the div tag as the progress bar container, and then use span tag to record user play behavior, user watching behavior data is written asynchronously to server. This method can extract user watching behavior completely and make learners observing their learning process accurately, it enhance the experience of the user, and reserve data for later learning behavior analysis and large data processing.

Key words: online learning; video player; watching behavior; progress bar