Data Driven Approaches of QoE Modeling on FreeWheel Dataset

汤凯
tangkai@pku.edu.cn
kevintongcn@gmail.com

北京大学

2014年6月

目录

1 研究背景

随着互联网分发成本的不断降低，以及基于订阅和广告盈利模式的互联网视频商业模式的不断成功 [3]，互联网视频的规模和影响不断扩大。在这个背景下，无论是学术界还是工业界都有这样的共识，用户对于视频质量的期望正在持续升高，提升用户的体验（QoE，即Quality of Experience）对于维系这些盈利模式是非常重要的。只有用户的体验好了，才会有更多的用户加入视频的付费应用中来。只有当用户的体验好了，才会吸引到更多的用户，从而提高广告价格。只有当用户体验好了，用户看视频的时长才会提高，才会更有机会看到视频中插入的广告。

尽管人们对QoE有着很长的研究历史 [1, 2, 4]，但是对于互联网视频特别是动态自适应流媒体QoE 的认识还是非常有限的。这是由于动态自适应流媒体的互联网视频在QoE的质量（Quality Metric）和体验（QoE）的定义都和传统视频的定义不尽相同。

具体而言：首先，传统视频质量的指标（Quality Metric）是PSNR（即Peak Signal-to-Noise Ratio）。PSNR能够反映用户看到的视频和原始视频之间的相似程度，从PSNR中我们可以得到视频编码效果和传输中码率控制的综合在一起对于用户所看到的视频质量的影响，但这并不适用于目前动态自适应流媒体的互联网视频。这是由于：(i) 动态自适应流媒体视频是通过HTTP 传输，传统的码率控制机制、应用层重传和加冗余的方法不再使用。同样，用于在有限的带宽下进行关键帧进行非对称保护（UEP）从而提升视频质量的机制也不再使用。使用了HTTP后，所有的视频数据都被均等地对待，并通过TCP可靠地传输。当网络变差时，不再通过码率降低码率、控制清晰度，而是消耗缓冲区中的缓存视频，最终反映在视频是否出现卡顿上。(ii) 互联网视频不再是单码率的视频，而是多种码率在不同尺寸分辨率设备上播放的视频，这就意味着，以单一源做参考，对不同码率的视频计算PSNR，从而衡量其质量的做法不再有效。(iii) 互联网视频在播放过程中会出现码率之间的切换，其切换次数和频率都是传统视频中质量评估中不再具有的特性。正因为以上的原因，动态自适应流媒体视频的质量更关注与传输相关的指标，例如卡顿频率、平均码率、码率切换，初始缓冲时间等 [5, 7]。

另一方面，传统视频体验的指标一般通过统计用户的平均意见得分（MOS，即Mean Option Score）的方式，来完成对于用户主观感受的量化。但是在动态自适应流媒体的互联网视频情境下，MOS指标不再适用。这是由于(i) 互联网视频有大量用户，去统计所有用户或者有代表性的用户的MOS面临着成本过高，可操作性也不强的问题。(ii)更重要的是用户的MOS与互联网视频盈利模型的业务目标之间没有直接关系。正因为以上的原因，动态自适应流媒体的互联网视频更加关注那些能够直接影响到业务营收的、可量化评估用户参与程度的指标，例如观看的时长、访问次数和广告观看

次数等指标等 [8]。

在以上背景下，本部分研究的主要目标在于构建一个能够预测用户观看动态自适应流媒体互联网视频时QoE 的模型。本文本报告希望这个模型满足两个要求，首先是以用户的体验为中心，能够准确地预测用户参与度（Engagement）的模型。其次，这个模型应该是直观可操作，能够用于指导流媒体系统的设计，同时能够给出特定场景下码率、缓冲、切换之间的如何取舍(Trade Off)方案。

为此本报告提出了一个数据驱动的预测用户参与度的建模方法。通过利用互联网上的大规模用户行为数据，通过机器学习的方法，来找到QoE模型中Quality与Experience之间的关系。

2 数据集与数据分析

本报告打算依托互联网视频上大规模用户行为数据，完成QoE的建模。在本节，本文主要介绍数据集的构成，并且从中抽取出数据作为本文模型（Model）的特征（Feature，即模型的输入）和标签（Label，即模型的输出）。

2.1 数据集

本报告使用Freewheel的数据集，通过其日志数据，本文可以获取在播放ESPN直播时用户观看互联网视频时各种行为，以及用户观看广告内容时的反馈情况。

数据集中包括了42,983,194行日志，每一行日志的格式都符合NCSA通用日志格式。这种格式被广泛地用于Web Server中，用于记录HTTP 请求的标准格式。分为两大类：

- 第一类属于用户观看视频时的用户行为数据，如下所示：

  10.1.1.6 prod-freewheel.xxxx.xx.com - [10/Nov/2013:04:02:43 +0000] "GET /ad/g/1?_dv=2&nw=87146&csid=watchxxxx:ios:phone:xxx3&caid=15902488&vdur=100&vprn=846897&pvrn=134525&afid=49515690&sfid=801283&flag=+rema+slcb+aeti+exvt+sltp&prof=87146:watchxxxx_ios_hls&resp=m3u8;_fw_syncing_token=4843856&_fw_lpi=-170744411012240778,930000&_fw_lpu=http%3Axxx.xxx.com/ls04/xxx/2013/11/10/XXX_VIDEO_5S_M3U8_1215795_20131110/800K/800_slide_ads.m3u8 HTTP/1.1" 200 2013 "-" "AppleCoreMedia/1.0.0.11B511 (iPhone; U; CPU OS 7_0_3 like Mac OS X; en_us)" 108.202.128.193

- 第二类属于客户端返回的用户观看广告的Pingback，用于记录用户观看广告的情况，如下所示：

```
10.1.1.6 prod-freewheel.xxxx.xx.com - [10/Nov/2013:04:02:39 +0000]
"GET /ad/l/1?ct=6&metr=7&cn=midPoint&et=i&s=a005&n=87146%3B87146%3B9
4047%3B147530%3B376521&t=1384056094432715014&f=&r=87146&adid=2991384
&reid=2114302&arid=0&iw=&uxnw=87146&uxss=s241337&uxct=16 HTTP/1.1" 2
00 0 "-" "Mozilla/5.0 (iPhone4,1; iOS 7.0.3) FreeWheelAdManager/5.5.x
-r9841-201305070735;com.xxx.  XXXX XXXX/302" 75.175.76.253
```

从中可以看出数据由以下几个域构成：

1. `75.175.76.253` 是HTTP请求来源（HTTP客户端）的IP地址。
2. `[10/Nov/2013:04:02:39 +0000]` 是请求的时间戳，代表HTTP请求的日期、时间和时区。
3. `GET /requested?param=sth HTTP/1.1` 是HTTP客户端发起请求的内容。
4. `200` 是响应HTTP请求的状态代码。
5. `2326` 是请求的大小，单位是字节bytes。
6. `AppleCoreMedia/1.0.0.11B511 (iPhone; U; CPU OS 7_0_3 like Mac OS X; en_us)` 是客户端的UserAgent，用于标识客户端的类型，包括客户端的设备，操作系统，语言等信息。

其中HTTP请求的内容可以包括以下参数：

1. 以`/ad/g/1`开头的请求是对内容视频的m3u8下载的请求
   - 通过`_fw_lpu`参数，我们可以得到请求内容视频的码率。
2. 以`/ad/l/1`开头的请求是对广告视频的观看pingback的反馈
   - 通过`adid`参数，我们可以获得请求广告的ID
   - 通过`cn`参数，用户广告观看百分比的反馈，包括以下几个比例
     (a) firstQuartile 广告观看到25%
     (b) midPoint 广告观看到50%
     (c) thirdQuartile 广告观看到75%
     (d) Complete 广告观看到100%

有了对于数据集初步的认识，本文就能对数据集进行分析，并从中找到用于描述模型的特征和标签的指标。

2.2 数据处理

在上一节分析的基础上，我们理清了数据集的可用数据域。数据集是由42,983,194用户行为构成，每一条用户行为都包括用户的IP，Device或OS、请求的App Name、用户行为的时间戳、用户观看视频的码率和用户观看广告的比例。

因而我们首先对数据集进行清洗，获得以上数据域。我们以HTTP请求的返回代码为200，用户UserAgent和请求非空作为条件，将满足条件的用户行为的以上数据域清洗出来，此时25G的数据缩减到19G。

在清洗完数据后，我们需要对数据进行聚合，将相同用户的行为按照时间序排序。要确定unique user，我们拿用户的IP和Device Name构成tuple作为用户的ID，构建一个词典。词典的key是用户的ID，value是一个list里面列了该用户按照时间序排序的所有行为，每个行为包括以下三个数据域：1.用户行为的时间2.用户行为的类型：是广告反馈还是观看内容视频。3.用户观看广告的比例或者永固观看内容视频的码率。

在对所有4200万用户行为进行聚合后，我们获得19万7千个独立用户，其中既看过视频又看过广告的用户约为5万5千个。对于这5.5万名用户，我们拿到他们第一次请求的时间作为他们开始观看的时间；通过计算第一个请求和最后一个请求的时间差获得他们观看的时长；通过计算码率切换的次数，获得单位时间内码率切换的速率；通过计算每个单次广告观看中最大比率的平均值获得该用户观看广告的平均比例。我们统计这5个重要指标的CDF，如下文所示。

2.2.1 开始观看时间

每个用户开始请求的时间来自于其第一个HTTP请求的时间戳，由于时间戳是服务器上的时间，要得到用户的当地时间，本文利用IP库得到每个IP的时区，从而获得了用户开始观看的时间。

本文画出按全体用户开始观看时间的CDF图以及按设备分类的各个类别用户开始观看时间的CDF图，分别如图1和图2所示。从图中可以看出，不同设备的开始观看视频的时间基本是完全一致的。比较特别的是Android用户早晨（5am-12pm）观看的人数比例更高，这可能与Android用户的移动性更好有关。Apple TV用户的CDF曲线不那么平滑，这可能是由于样本数量少所导致的。

2.2.2 平均码率

用户的平均码率计算方法是每个用户观看行为中观看内容视频码率的算术平均值。用户的平均码率越高，意味着用户观看越高质量的视频。

本文画出全体用户平均码率的CDF图以及按用户设备分类的各个类别用户平均码

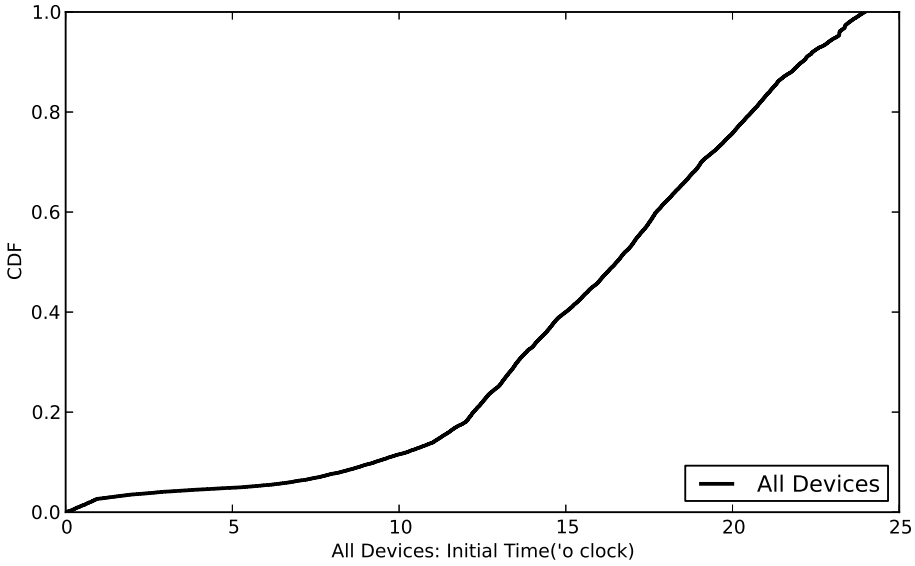

图 1: 所有用户的开始观看时间

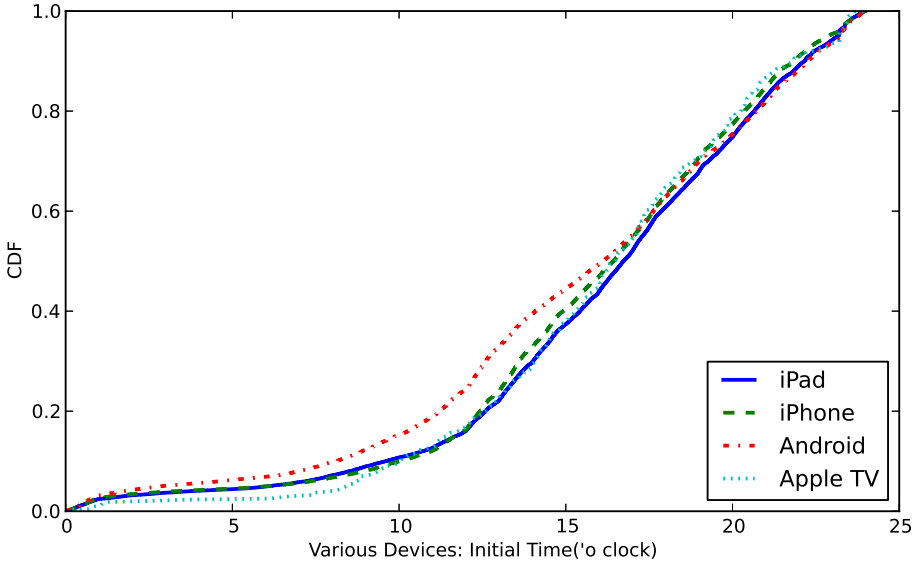

图 2: 各类设备用户的开始观看时间

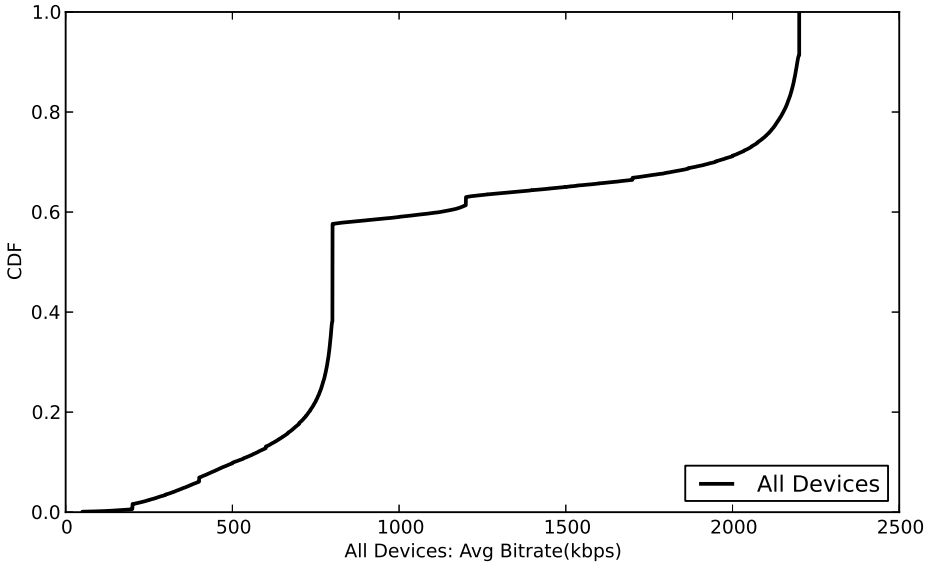

图 3: 所有用户观看的视频的平均码率

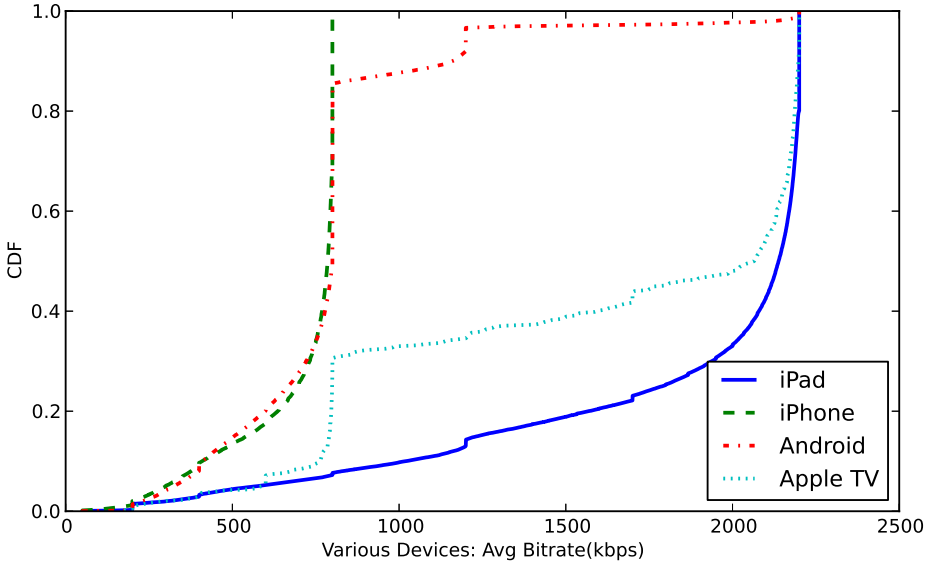

图 4: 各类设备用户观看的视频的平均码率

率的CDF图，分别如图3和图4所示。

从这些图中可以看出，不同设备的平均码率不尽相同，iPad用户的平均码率集中在2000kbps以上，而iPhone用户的平均码率集中在800kbps 以上。这就从侧面说明了iPad用户的网络条件更好，屏幕尺寸更大，更多地观看高码率的视频，而iPhone用户的移动性更强，屏幕尺寸更好，会更多地观看低码率的视频。而Android用户观看的视频的平均码率集中在800kbps和2000kbps两个峰值，这是因为数据集中的UserAgent 没能把Android的平板用户和手机用户分开，Android用户兼具iPhone用户和iPad用户的特点。

2.2.3 码率切换

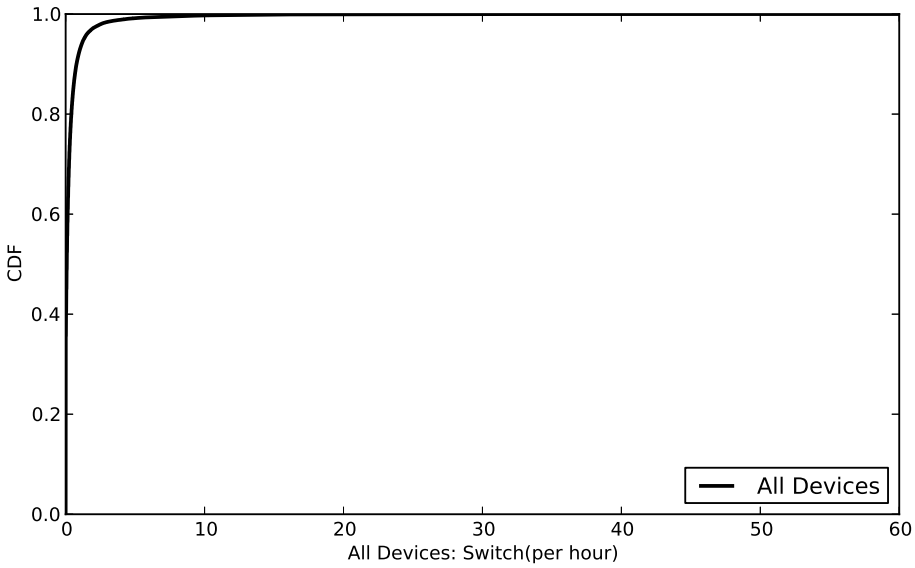

图 5: 所有用户的码率切换次数

用户的单位时间内码率切换次数体现了用户观看动态自适应流媒体视频质量的稳定程度。视频质量不同清晰度之间切换次数越多，用户就越容易不满。本文通过计算用户观看内容视频时码率的单位时间切换次数获得该值。

本文画出按全体用户码率切换次数的CDF图和按设备分类的各个类别用户码率切换次数的CDF图，分别如图5和图6所示。这些图说明，iPad和iPhone用户的码率切换和全部用户的总体趋势一致。相比iOS用户，Android用户的码率切换次数更多一些，这也许是由于Android用户的无线网络更加不稳定一些，不足以支撑高码率的视频导致码率切换频繁；也有可能是Android 客户端实现的码率适应算法不同，由于内存和CPU运算能力的不同，Android客户端可能缓存的视频时长更少，从而更容易发送码率切换。而Apple TV 的码率切换次数最少，这样从侧面反映客厅用户的网络最为稳定

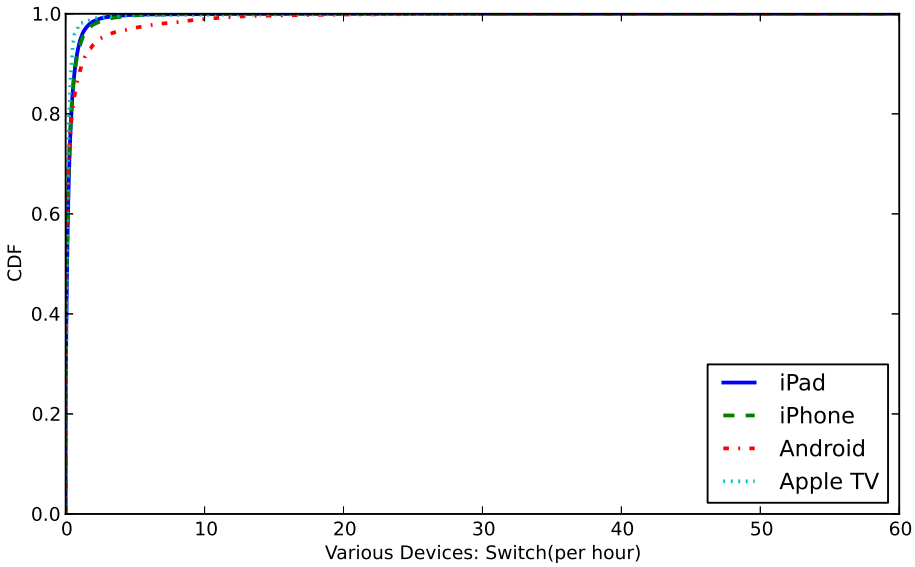

图 6: 各类设备用户的码率切换次数

的。

2.2.4 观看时长

本文通过计算每个用户第一次请求内容视频的时间戳到最后一个一次请求内容视频的时间戳之间的差值获得用户的观看视频的时间长度。本文认为时间长度可以反映用户对视频服务的参与度。

本文画出全体用户观看时长的CDF图和按设备分类的各个类别用户观看时长的CDF 图，如图7和8所示。上图说明，iPad、iPhone、Apple TV和Android用户观看时长和总体趋势基本一致，iPhone用户观看时间最少，Apple TV观看时长最长。

2.2.5 广告比例

用户在观看内容流的时候会从中插入广告流。内容流中插入的每一段广告，都有用户观看的百分比，本文将这些百分比计算平均值得到用户观看广告的比例。观看广告比例的高低反映了广告商向内容提供商支付的数额高低。

本文画出全体用户观看广告比例的CDF图和按设备分类的各个类别用户观看广告比例的CDF图，如图9和图10 所示。上图说明，相比总体情况，iPad、iPhone观看广告的比例远远高于Android用户，他们分别有40%和30%的用户看完了所有的广告，而Android用户仅仅有5%不到，这意味着Android用户的为广告商创造的价值更低。值得注意的是Apple TV的用户观看广告的均为100%，不过Apple TV 的样本数也比较少

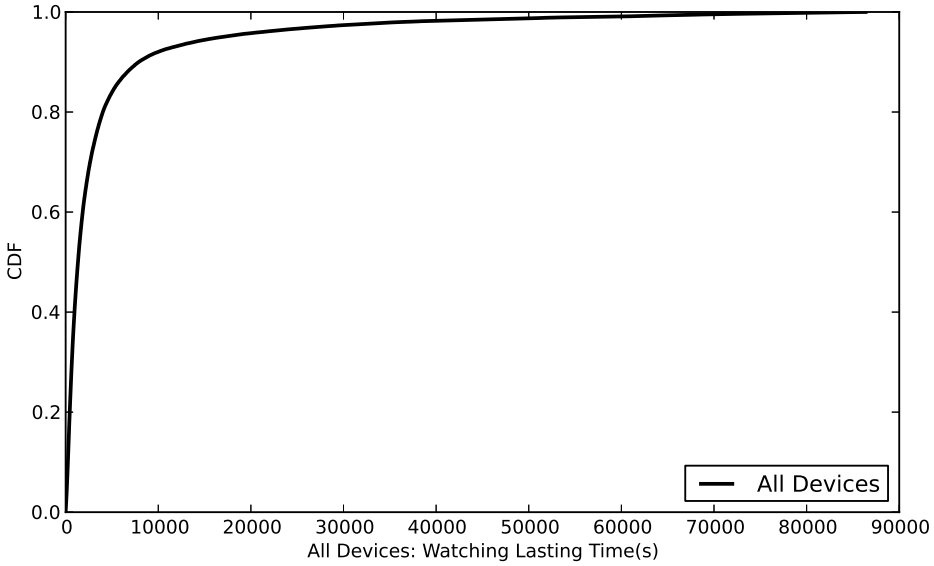

图 7: 所有用户的观看时长

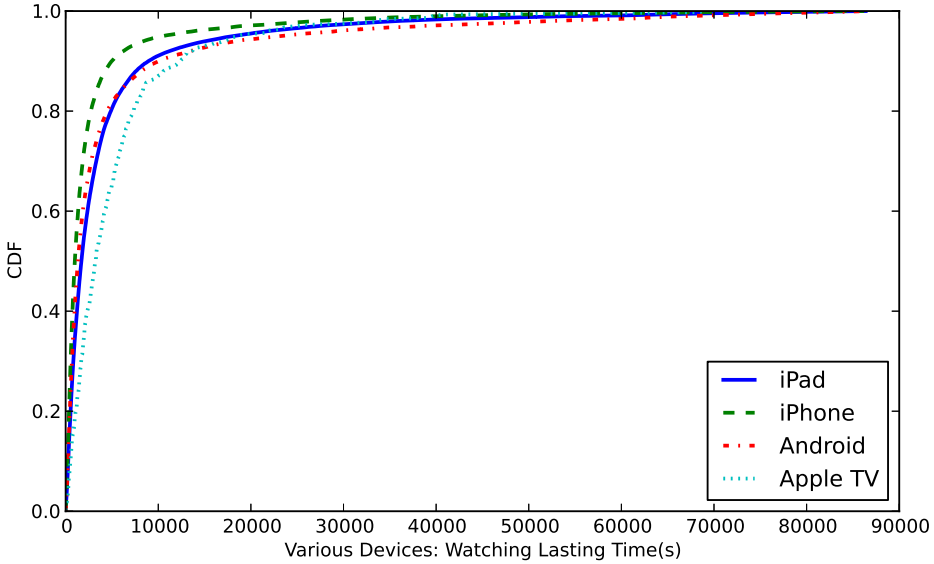

图 8: 各类设备用户的观看时长

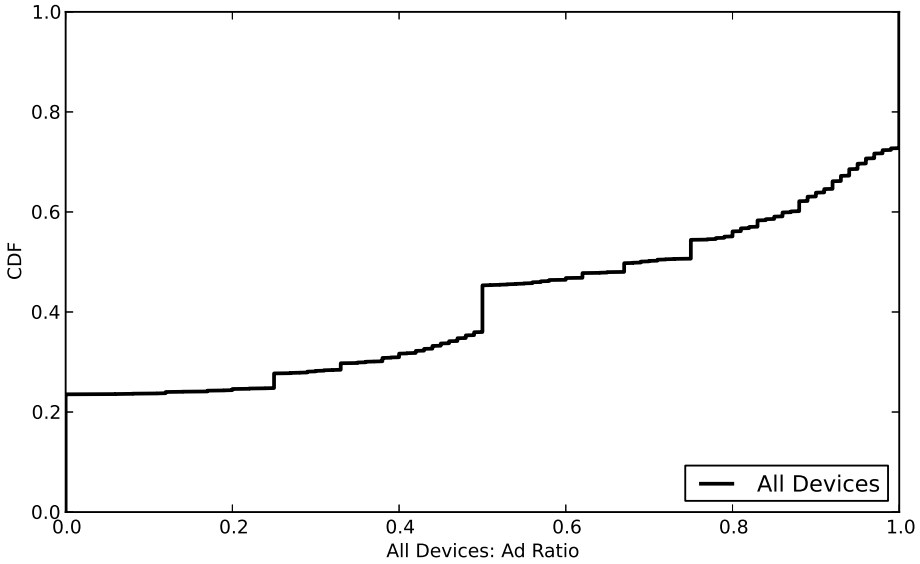

图 9: 所有用户的观看广告比例

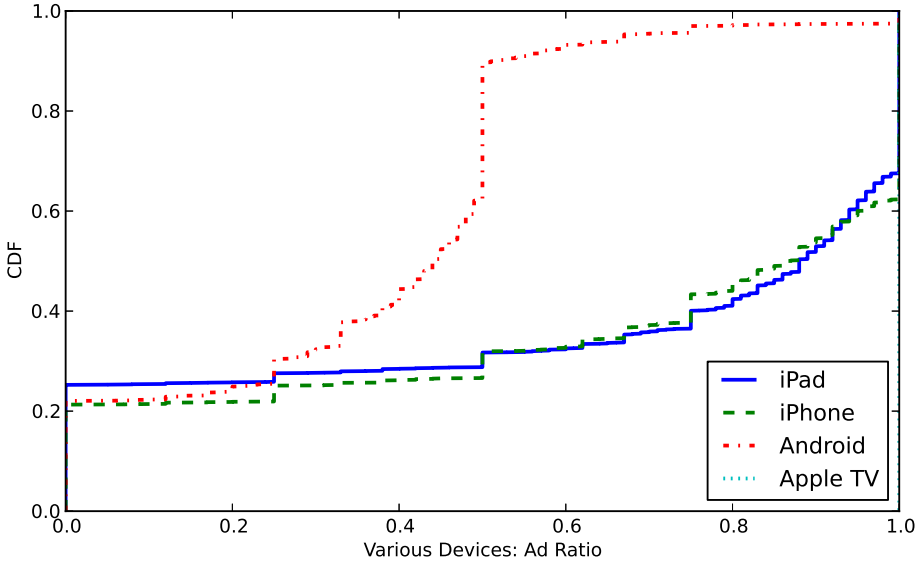

图 10: 各类用户的观看广告比例

（占总用户数的1%左右）。在QoE建模的过程中，本文不再预测Apple TV用户广告观看比例的QoE 模型，因为无论使用什么方法，其准确率都为100%，没有太大的实际意义。

3 QoE评估指标与特征抽取

上一章节中，本文介绍了所用的数据集，并对数据进行了分析。在这一章节中，本文将介绍QoE的指标，即模型输出；并将介绍建模过程中将用到的特征。

3.1 QoE评估指标

本文在前文中提到，QoE指标应与用户的参与度相关，同时也应考虑对网络视频提供商营收的影响的因素。在本研究中，根据数据集中已有的数据，本文考虑使用以下数据作为评估QoE 的指标，即模型的输出。

- 用户观看视频的时长，单位是秒。
  由于数据都来源于同一个频道直播过程中的用户行为数据，故观看的绝对时长相比观看时比例反映用户在视频中的参与度（Engagement）。
- 用户一次观看视频中，所看到广告的平均比例，以百分比计数。
  Freewheel将广告观看的比例作为广告计费中的重要指标。

3.2 特征抽取

3.2.1 质量因素

根据已有的研究 [5, 7]，在选取QoE模型特征的时候，本文考虑选择以下因素作为评估互联网视频质量的指标：

- 平均码率，单位是kbps。
  播放器会根据网络状况在多个视频清晰度之间切换，平均码率指的是在一次视频观看过程中所有选择码率以及对应时间的加权平均值。
- 码率切换速率，单位是次/小时。
  播放器会根据网络状况在多个视频清晰度之间切换，码率切换速率是指码率在单位时间内（每小时）切换的次数。
- 视频开始观看的时间
  即开始看视频的时候用户所在地的当地时间。虽然此参数代表视频质量，但是它

会影响用户观看时长这个模型的输出参数。例如：深夜的用户可能由于时间太晚不会看太长的时间，清晨和工作时间的观看ESPN的用户也会压缩自己看视频的长度。

在这里本文没有选取视频卡顿作为QoE模型的特征，原因有两点：第一，本文数据集是由服务端搜集的，而客户端向请求是周期性发出的，没法从服务端的日志中周期地推断出是否出现视频卡顿。第二，MSS会通过调整码率的方式尽量减少视频卡顿，对数据集的初步分析显示，大部分用户的码率切换相当少，传输带宽非常稳定，从侧面推断出在整个数据集中几乎没有出现卡顿现象。

3.2.2 外部因素

研究 [6]表明，在QoE模型中除了视频质量会影响模型的输出以外，一些外部因素，虽然不能刻画视频观看的质量，但也会对QoE模型有影响。本文通过对数据集中已有数据字段的甄选，选出以下因素作为QoE模型的外部因素。

- 设备类型
  发起看视频请求的设备类型，对数据的初步分析可知，设备可能包括：iPad, iPhone, Android, Apple TV等，他们的数量分别是23415、17102、13760、348。

本文之所以要选择这个因素作为QoE模型的外部因素是由于以下原因：

1. 这个因素本身是离散的。QoE建模作为分类问题，其特征需要离散化，目前的三维特征都是连续的，他们可以选择不同的离散化的粒度，但是设备分类这个参数没办法跟着其他特征一起增加或减少分类个数，导致尺度上的不一致。
2. 不同的设备类型对应不同的视频观看场景。使用iPhone看视频的用户更容易在移动中，或者在工作时间的小憩。使用iPad看视频的用户则更容易坐下来，看更长时间的视频等等。只有引入了外部因素才能更好地刻画用户的行为。
3. 不同的设备类型在相同的质量特征下有不同的表现。在动态自适应流媒体视频中，码率直接反映了分发（Deliver）给用户的质量，但这并不意味着更高质量的视频就对应更好的用户感受。一个低码率视频在iPhone上的观看效果会好于一个中码率视频在Apple TV上的效果。同样更加移动的iPhone用户会比在客厅的Apple TV 的用户有着更好对码率切换的忍耐力。当用户观看视频的设备并不是一致的时候，就需要引入外部因素，来理清不同设备下，特征对输出的影响关系。

综上所述，外部因素会直接影响到模型的输出以及模型特征对模型输出直接的影响关系，本文需要引入外部因素来理清这种影响。

4 模型概览

在以上已有数据分析的基础上，本文考虑对QoE进行建模。建模第一个应确立的问题，是明确任务的类型，是作为回归问题还是分类问题。在这个问题上，研究 [6]讨论了这个问题，他们的结论是定义为分类问题，因为本文不需要对模型的输出（用户参与度和广告观看比例）进行数值上的预测，只需要在给定一组特征的情况下，定性地得到模型输出的范围即可，因此，本文将问题转换成一个分类问题，采用三种成熟的分类器：朴素贝叶斯分类器、逻辑斯蒂回归（Logistic Regression）分类器和决策树分类器，对QoE 进行建模。

本文将QoE建模过程形式化如下。

本文定义第i个用户行为特征向量$\mathbf{x}_i$满足$\mathbf{x}_i = [x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)}]$，其中$x_i^{(j)}$第$i$个用户行为的第$j$维特征，$n$是特征总数量。为了更好进行模型的分类训练，本文将输入特征进行k值离散化。

$$x_i^{(j)} = \lfloor \frac{x_i^{(j)}}{Interval\ x_i^{(j)}} \rfloor \tag{1}$$

其中，

$$Interval\ x_i^j = \frac{Max\ x_i^{(j)} - Min\ x_i^{(j)}}{k} \tag{2}$$

本文进一步定义用户行为样本集合为$\mathbf{X}$，矩阵$\mathbf{X}$满足$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m]^T = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n]$，其中$\mathbf{x}_i$表示第$i$ 个用户行为的特征向量，$\mathbf{f}_j$表示所有用户第j维特征构成的向量，m为样本的总数量。

单输出QoE模型的被输出定义为Y，本文记Y的k个取值可以是$\{y_1, y_2, \ldots, y_k\}$。由于在3.1中所定义的模型输出：用户观看视频的时长和用户每次观看视频时观看广告的比例的取值都是连续的值，因此本文对原始数据集中的用户观看视频的时长和每次观看视频时观看广告的比例进行k值离散化以得到模型的确切输出标签。

4.1 朴素贝叶斯分类器

朴素贝叶斯分类器通过训练数据集学习联合概率分布$P(Y = y_i|\mathbf{X})$, 其表达方式如公式3所示。

$$P(Y = y_i|\mathbf{X}) = \frac{P(\mathbf{X}|Y = y_i) \cdot P(Y = y_i)}{P(\mathbf{X})} \tag{3}$$

其中，$P(Y=y_i|\mathbf{X})$表示该样例$\mathbf{X}$的归于y_i的类的后验概率。$P(\mathbf{X}|y_i)$表示该样例归于$Y=y_i$的类的先验条件概率。

本文采用极大似然估计来估计模型的参数。

4.2 逻辑斯蒂回归分类器

多项逻辑斯谛回归模型(Multi-nominal Logistic Regression Model)的原理如下：

$$P_\omega(y=y_i|\mathbf{x})=\frac{e^{\boldsymbol{\omega_i}\cdot\mathbf{x}+b}}{1+\sum_{i=1}^{k-1}e^{\boldsymbol{\omega_i}\cdot\mathbf{x}+b}}, i=1,2,...,k-1 \tag{4}$$

$$P_\omega(y=y_k|\mathbf{x})=\frac{1}{1+\sum_{i=1}^{k-1}e^{\boldsymbol{\omega_i}\cdot\mathbf{x}+b}} \tag{5}$$

其中，$\boldsymbol{\omega}$是模型的参数；b表示模型的偏置；$\mathbf{x}$表示模型的特征；$y\in\{y_1,,y_2,\cdot,n\}$是输出属于哪个类别。

本文采用梯度下降法来估计模型参数。

4.3 决策树

决策树算法是一种多级判决的方法。使用该算法的时候，先使用某种判别方法把待分类的样本分到某几个大组之一，这些大组中可能仍然包括几个不同的类。再对分到各组中的文本进一步判别，循此进行，直到把样例到某个确定的类为止。

由于决策树分类是分成几步来进行的，因此精确度比一次判决要高些，并且每一次的判决规则可以设置简单一些，可以只使用少数有效的特征，减少每一步的工作量。但是在分类器的建立上要使用比较多的时间。不同层次的特征选择也是一个比较困难的工作。另外,对不同的层次上使用的特征和分类方法进行选择以达到最优也是一个比较困难的工作。

本文选择CART算法来生成决策树。

决策树生成的过程中使用信息增益来进行特征选择。其公式如下：本文可以先计算数据集的总信息熵。

$$H(y)=\sum_{q=1}^{k}P[Y=y_q]log\frac{1}{P[Y=y_q]} \tag{6}$$

然后计算数据集在该特征f上的信息熵。

$$H(y|f)=\sum_{q=1}^{k}P[f=f_q]H(y|f=f_q) \tag{7}$$

最后得到数据集在特征f上的信息增益。

$$InfoGain = \frac{H(y) - H(y|f)}{H(y)} \tag{8}$$

4.4 支持向量机

支持向量机（即SVM）是一种二分类分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器。支持向量机的还包括有核技巧，不同的核函数让SVM既可以成为线性分类器又可以非线性分类器。

当采用线性核SVM分类器作为模型时，实验结果相比逻辑斯蒂回归分类器差别不大。这是由于采用线性核SVM分类器时，依然无法克服特征之间线性无关的假定。

对于非线性的SVM模型，本文并未做尝试，这是由于当使用非线性核SVM分类器时，样本首先会被映射到高维空间，再进行最大间隔的超平面选择，其结果可能很突出，但是成因难以解释，很难指导实际DASH 系统的设计，与本文的初衷相违背。

5 模型训练与实验结果

5.1 模型训练

在使用分类器的过程中，本文采用正则项限制模型复杂度的形式避免训练过程中的过拟合。在确定好模型的正则项系数后，本文采用5 折交叉检验计算出模型预测的准确率，选择其平均值作为模型训练的最终结果。

如3.2.2中所述，外部因素虽然不能刻画视频观看的质量，但是也对对QoE模型产生影响。本文选择设备类型作为外部因素。外部因素有两种使用方法：

1. 最简单的使用方法就是将设备类型作为机器学习方法中的一个特征；
2. 另外一种使用外部因素的方法就是，根据设备类型分割数据集，对使用不同设备类型的用户分别建模。

这两种方法分别有其优点和缺点。如果将设备类型作为机器学习方法中的一个特征，那么机器学习的框架不需要做任何的修改。对于不同的设备类型，所建立得到的QoE模型都是统一的。但是，根据本文所分析得到的结果，不同设备类型下用户行为具有差异性。因此，所训练得到的统一的QoE模型不是很直观，有可能不能刻画不同设备类型的不同用户行为。相反，如果根据设备类型分割数据集，对使用不同设备类型的用户分别进行QoE建模。对于每种设备类型，本文都对其用户群建立一个QoE模型，这样就能解决不同设备类型用户之间的行为差异性带来的误差。

为了体现出这两种做法的差别，本文分别计算增加一维特征的处理方法和分割数据集的处理方法下，不同的分类器统计其分类的准确率。本文用All标注前一种做法的结果，用设备的类型iPad、iPhone、Android、Apple TV 标注后一种做法的结果。

5.2 模型验证

在这一节中，本文主要验证不同分类器的预测效果。为此本文不使用外部因素，仅仅使用三个质量特征得到使用不同分类器的实验结果。由于模型有两个输出：观看时长和广告比例，因此本文构建两个独立的模型，分别训练出他们的结果。

由于本文将QoE建模问题转化成了分类问题，因为本文依据1 2式，将特征和输入做k值化处理（在本节$k = 2, 5, 10, 20$）。

对于模型的输出，本文做以下处理：

1. 当使用观看时长作为QoE模型的输出时，本文将输出按$86400/k$s间隔进行离散化，将时长转换成$1, 2, ..., k$等离散的值。
2. 当使用用户观看的广告比例作为QoE模型的输出时，本文将输出按$100/k$%间隔进行离散化，将平均比例转换成$1, 2, ..., k$等离散的值。

对于模型的特征，本文做以下处理：

1. 本文将平均码率按照$2500/k$ kbps间隔进行离散化，将平均码率转换成$1, 2, ..., k$等离散的值。
2. 本文将码率切换速率按照$60/k$ times/h间隔进行离散化，将码率切换速率转换成$1, 2, ..., k$等离散的值。
3. 本文将开始观看时间按照$24/k$ o'clock的间隔进行离散化，将开始观看时间转换成$1, 2, ..., k$等离散的值。

5.2.1 预测观看时长的结果

图11中给出了四种分类器在不考虑外部因素情况下，预测用户观看时长时的准确率。表1给出了不同分类器的参数。当朴素贝叶斯、逻辑斯的会贵、LinearSVM在处理多分类问题是，采用的是One VS All的方法，因而其参数数目和其分类数相等，为了简便起见，本文只列出第一组参数，在预测广告比例时我采用相同的处理，下文不再赘述。

从图中本文可以看出，决策树相比朴素贝叶斯、逻辑斯蒂回归和线性SVM有更好的性能。

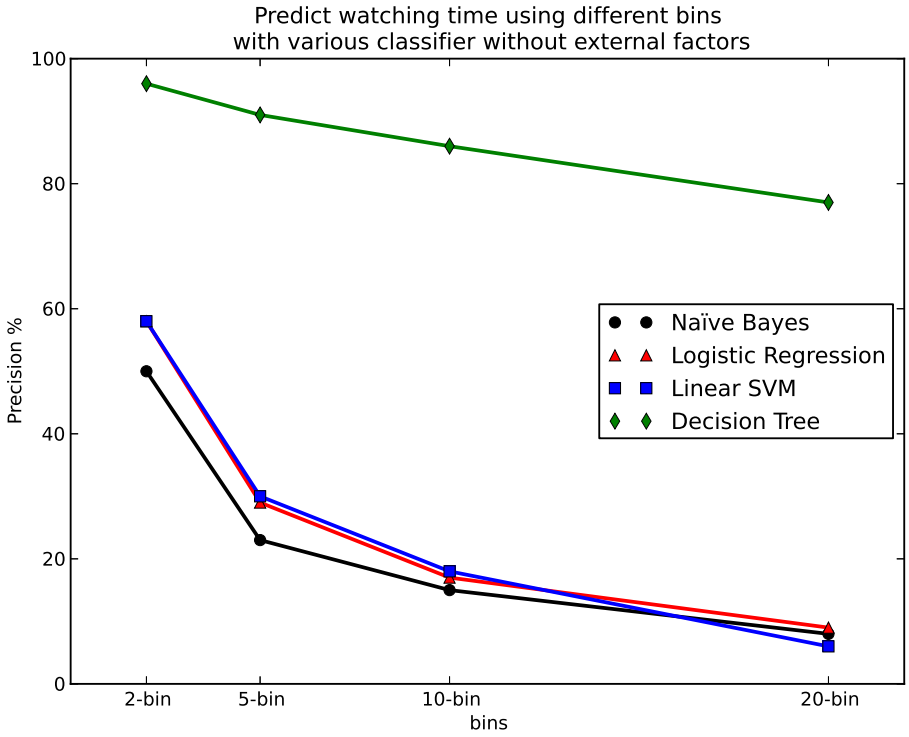

图 11: 各种分类器在不考虑外部因素情况下预测观看时长的结果

表 1: 不同分类器在预测观看时长的模型参数和结果

分类数	分类器	模型参数	准确率
2	朴素贝叶斯	[[0.3826546 0.25725701 0.36008839]] [0.4997]	50.19%
	逻辑斯蒂回归	[[-0.35131904 -1.62269564 -0.35392512]] [2.8500]	58.12%
	LinearSVM	[[-0.16710676 -0.7010655 -0.16423]] [1.2795]	58.79%
	决策树	[0.24217903 0.00018867 0.75763227]	96.56%
5	朴素贝叶斯	[[0.30676445 0.36404476 0.32919079]...] [0.2002]...	23.73%
	逻辑斯蒂回归	[[-0.04792862 0.1569339 0.04788364]...] [-1.8878]...	29.89%
	LinearSVM	[[-0.00125938 0.07055629 0.02288094]...] [-0.8917]...	30.39%
	决策树	[0.0624371 0.00193633 0.93562657]	91.91%
10	朴素贝叶斯	[[0.31672517 0.3505962 0.33267864]...] [0.1006]...	15.34%
	逻辑斯蒂回归	[[0.01074894 0.04247164 0.04054859]...] [-2.7387]...	17.83%
	LinearSVM	[[0.00079981 0.00824172 0.00665654]...] [-0.9006]...	18.04%
	决策树	[0.11034654 0.00970758 0.87994588]	86.25%
20	朴素贝叶斯	[[0.337753 0.32783861 0.33440839]...] [0.0506]...	8.25%
	逻辑斯蒂回归	[[0.01914012 -0.00258457 0.0224263]...] [-3.3771]...	9.61%
	LinearSVM	[[-0.01229381 -0.00509657 0.0200771]...] [-0.9409]...	6.87%
	决策树	[0.2106921 0.01837615 0.77093175]	77.97%

5.2.2 预测广告观看比例的结果

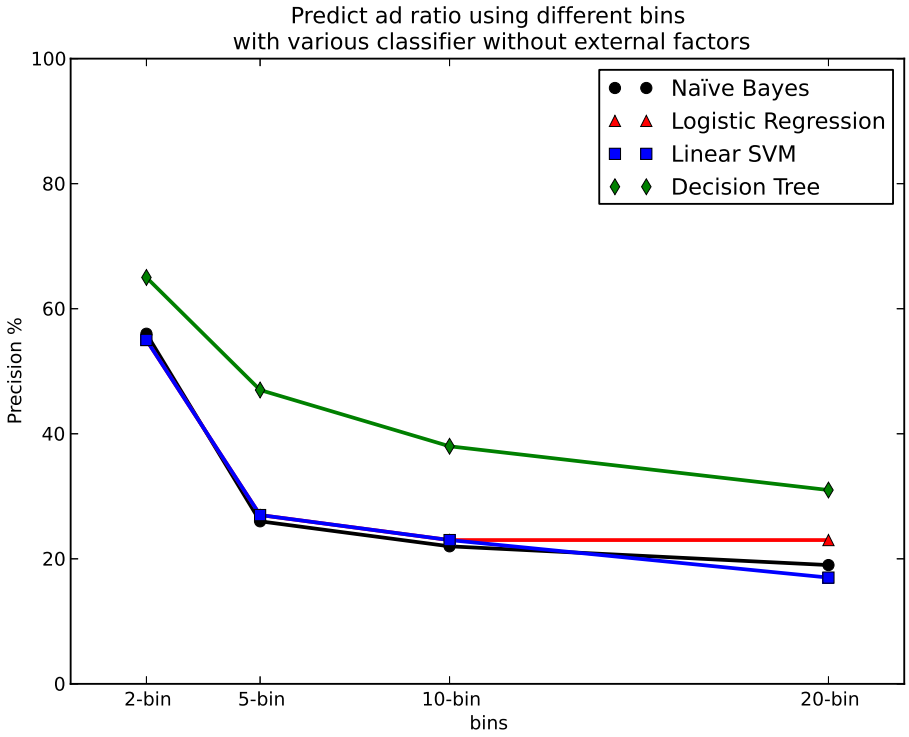

图 12: 各种分类器在不考虑外部因素情况下预测广告观看比例的结果

图12中给出了四种分类器在不考虑外部因素情况下，预测用户广告观看时的准确率。表2给出了不同分类器的参数。从图中本文可以看出，决策树相比剩下朴素贝叶斯、逻辑斯蒂回归和线性SVM有更好的性能。但是在预测广告时决策树的优势并没有在预测广告长度时明显。本文认为，这是由于影响用户观看视频的因素不仅仅是用户观看视频时的质量因素，还有可能是受到广告与视频内容切合度，以及用户是否对广告感兴趣等其他因素的影响。受限于数据集，本文目前无法得到广告和视频内容的关系以及用户的兴趣偏好。

由以上三个模型的两组结果可知，二分类的情况下，决策树相比剩下两个分类器有更好的性能，这是因为在逻辑斯蒂回归、贝叶斯分类器和线性支持向量机中，本文假定输入参数之间是没有关系的，但是时间是输入参数之间不是正交的，他们存在一定的依赖关系。因而使用决策树能够更好地刻画QoE模型。

5.3 验证外部因素的作用

在上一节中，本文发现决策树在所有本文使用的分类器中预测效果最好，为了进一步地提高模型的预测效果，本文考虑使用外部因素，即用户的设备信息。因而在这一节

表 2: 不同分类器在预测广告比例的模型参数和结果

分类数	分类器	模型参数	准确率
2	朴素贝叶斯	[[0.38806034 0.25446562 0.35747404]] [0.5048]	56.29%
	逻辑斯蒂回归	[[0.34383397 -0.78576721 -0.03635231]] [0.3938]	55.45%
	LinearSVM	[[0.16385808 -0.37209252 -0.02320682]] [0.1958]	56.30%
	决策树	[0.9600697 0.01229609 0.02763421]	65.55%
5	朴素贝叶斯	[[0.32275038 0.34483499 0.33241463]...] [0.2000]...	26.29%
	逻辑斯蒂回归	[[0.01473707 0.00070846 0.07873821]...] [-1.6730]...	27.37%
	LinearSVM	[[0.00614196 0.00056298 0.02392485]...] [-0.6896]...	28.12%
	决策树	[0.8893682 0.02231456 0.08831724]	47.16%
10	朴素贝叶斯	[[0.3231414 0.34249696 0.33436164]...] [0.1993]...	22.58%
	逻辑斯蒂回归	[[-0.0580255 -0.0787792 0.00771679]...] [-0.5893]...	23.58%
	LinearSVM	[[0.00257957 -0.00038547 0.01430317]...] [-0.7000]...	23.92%
	决策树	[0.73786345 0.0458058 0.21633075]	38.14%
20	朴素贝叶斯	[[0.32585736 0.33881047 0.33533216]...] [0.1994]...	19.99%
	逻辑斯蒂回归	[[-0.04647809 -0.05812802 -0.00602582]...] [-0.0999]...	27.59%
	LinearSVM	[[0.00700679 -0.01397234 0.0242150]...] [-0.7224]...	23.62%
	决策树	[0.60375824 0.05118562 0.34505614]	31.28%

中，本文主要验证第3.2.2节中提到的外部因素的影响。本文考虑以下三种情况：

1. 完全不使用外部因素

2. 将外部因素作为一维特征

3. 按外部因素不同将数据集分割成多个不同的数据子集，分别训练。

由于模型有两个输出：观看时长和广告比例，因此本文构建两个独立的模型，分别训练出他们的结果。

5.3.1 外部因素在预测观看时长的效果

图13中给出了外部因素在决策树预测观看时长时的准确率。表3 总结了在外部因素在决策树预测观看时长时的模型参数。本文可以看出，随着类别的增多，预测精度的提高，出现了预测的准确率降低现象，这是正常的现象。

我还将分离数据集后的准确率按照每种设备的比例做了加权平均，如在图13中，如图中蓝色线条所示。该图显示，从总体上看，外部因素在预测用户观看时长时没有提升效果。但是在分离数据集后，iPhone、iPad的准确率高于总体准确率。因此本文建议，在预测用户观看时长时，当用户为Android和Apple TV用户时，将不使用外部因素训练

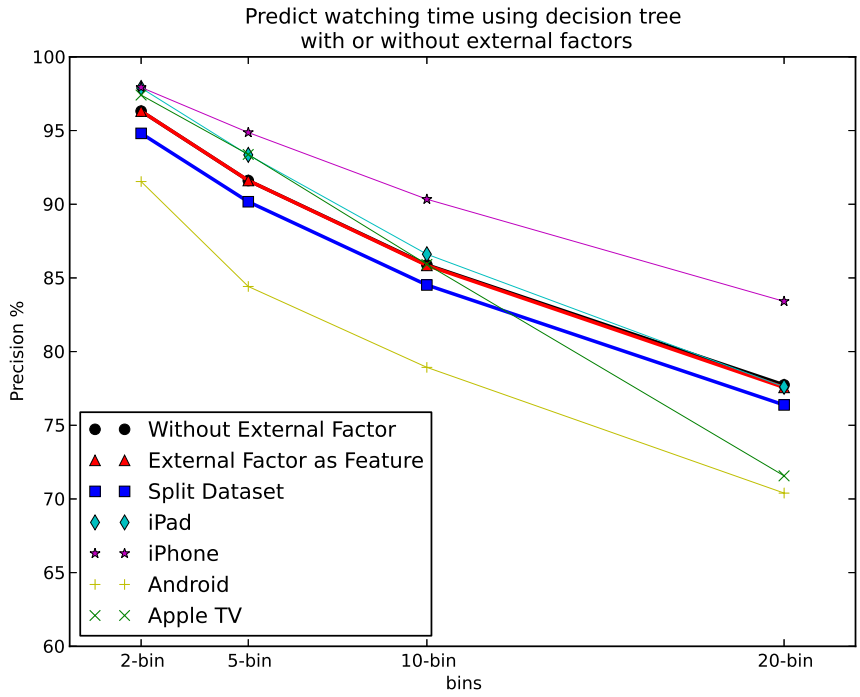

图 13: 外部因素对使用决策时预测观看时长时的效果

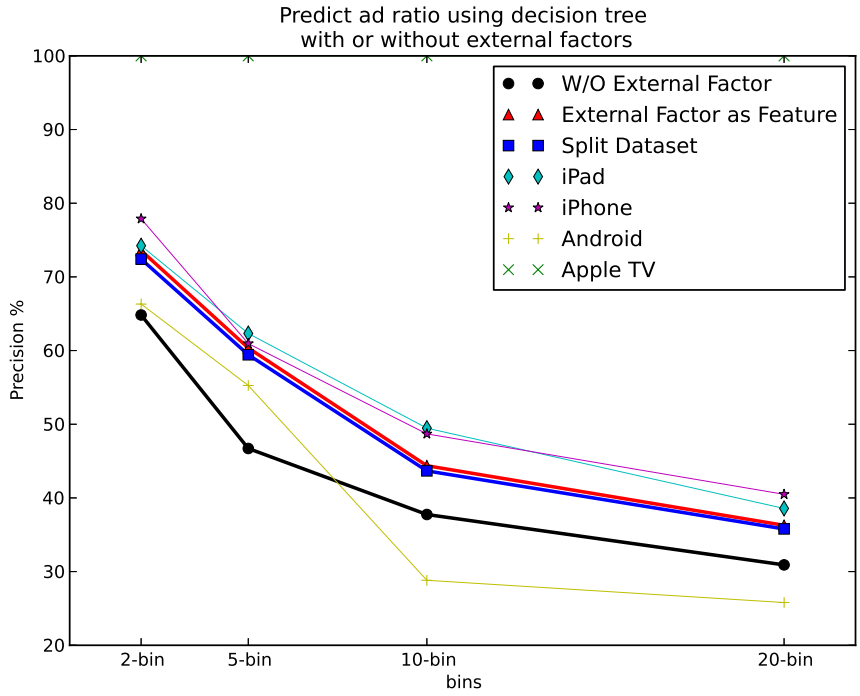

图 14: 外部因素对使用决策时预测广告比例时的效果

表 3: 外部因素对使用决策时预测观看时长时结果及模型参数

	外部因素		**gini系数**	**准确率**
二分类	不考虑外部因素		[0.24217905 0.00018867 0.75763227]	96.32%
	将外部因素作为特征		[0.00613085 0.00016647 0.38761940 0.60608325]	96.32%
	分离数据集	iPad	[0.06125813 0.00006108 0.93868078]	97.89%
		iPhone	[0 0.00008743 0.99991256]	97.93%
		Android	[0.00510934 0.000066089 0.99422975]	91.54%
		Apple TV	[0.28685399 0.00023255 0.71291345]	97.41%
五分类	不考虑外部因素		[0.0624371 0.00193633 0.93562657]	91.61%
	将外部因素作为特征		[0.03344623 0.00308303 0.73432779 0.22914295]	91.61%
	分离数据集	iPad用户	[0.04722643 0.00068895 0.95208461]	93.35%
		iPhone	[0.01768157 0.00044284 0.9818755]	94.87%
		Android	[0.05000375 0.01049921 0.93949704]	84.42%
		Apple TV	[0.39089338 0 0.60910662]	93.38%
十分类	不考虑外部因素		[0.11034654 0.00970758 0.87994588]	85.89%
	将外部因素作为特征		[0.10185455 0.01113534 0.77515905 0.11185106]	85.85%
	分离数据集	iPad	[0.0669185 0.00131707 0.93176444]	86.61%
		iPhone	[0.08278063 0.00098997 0.91622941]	90.34%
		Android	[0.19915799 0.03979507 0.76104694]	78.93%
		Apple TV	[0.48820032 0 0.51179968]	85.93%
二十分类	不考虑外部因素		[0.21059361 0.01845457 0.77095182]	77.74%
	将外部因素作为特征		[0.20886644 0.01712161 0.63907838 0.13493357]	77.57%
		iPad	[0.24841672 0.00617336 0.74540992]	77.61%
		iPhone	[0.14017758 0.00194281 0.85787961]	83.41%
		Android	[0.34821639 0.05432133 0.59746227]	70.40%
		Apple TV	[0.46189305 0 0.53810695]	71.57%

出的模型作为其QoE模型；对于iPad和iPhone用户，用iPad和iPhone用户单独分离出数据集训练出的独立模型作为其QoE模型。

图15、图16、图17给出了在2分类的情况下，我们使用到的预测观看时间决策树的可视化。对于5分类以上的决策树，其可视化图示过于复杂，分支过多也不便于辨认，我们通常使用内存中已经训练好的决策树直接用于预测。

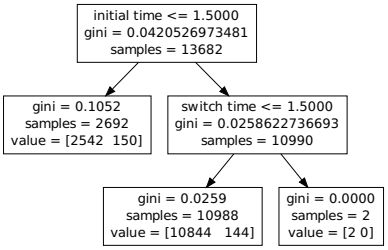

图 15: iPhone设备上对于观看时长预测的决策树

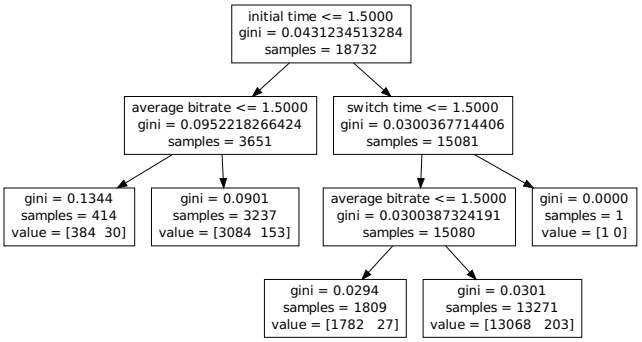

图 16: iPad设备上对于观看时长预测的决策树

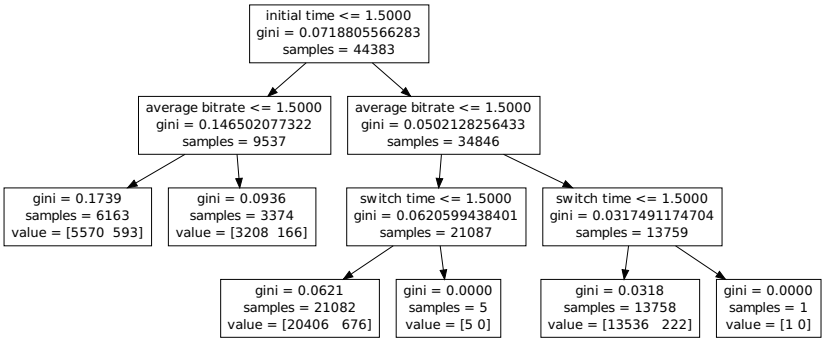

图 17: 不使用设备因素获得的用于预测观看时长的决策树

5.3.2 预测广告观看比例的多分类结果

图14中给出了外部因素在决策树预测广告比例时的准确率。表4 总结了在外部因素在决策树预测广告比例时的模型参数。本文可以看出，随着类别的增多，预测精度的提高，出现了预测的准确率降低现象，这是正常的现象。

表 4: 外部因素对使用决策时预测广告观看比例时结果及模型参数

	外部因素		**gini系数**	**准确率**
二分类	不考虑外部因素		[0.9600697 0.01229609 0.02763421]	64.83%
	将外部因素作为特征		[0.00898362 0.00173611 0.00207893 0.98720134]	73.58%
	分离数据集	iPad	[0.93365963 0.06141148 0.0049289]	74.23%
		iPhone	[0 0.80379381 0.19620619]	77.89%
		Android	[0.04312393 0.08224426 0.8746318]	66.32%
		Apple TV	[0 0 0]	100%
五分类	不考虑外部因素		[0.8893682 0.02231456 0.08831724]	46.71%
	将外部因素作为特征		[0.03393103 0.003994 0.02174967 0.9403253]	60.36%
	分离数据集	iPad	[0.67681782 0.03837541 0.28480677]	62.32%
		iPhone	[0.68519917 0.08927053 0.2255303]	60.95%
		Android	[0.23893653 0.17869226 0.58237122]	55.26%
		Apple TV	[0 0 0]	100%
十分类	不考虑外部因素		[0.73769324 0.04632991 0.21597685]	37.75%
	将外部因素作为特征		[0.07380063 0.01272863 0.07031321 0.84315754]	44.38%
	分离数据集	iPad	[0.53040188 0.05084647 0.41875165]	49.49%
		iPhone	[0.52503621 0.05411949 0.4208443]	48.69%
		Android	[0.39755281 0.1581613 0.44428589]	28.82%
		Apple TV	[0 0 0]	100%
二十分类	不考虑外部因素		[0.60333272 0.04953195 0.34713533]	30.91%
	将外部因素作为特征		[0.14816732 0.01644683 0.16216409 0.67322176]	36.28%
		iPad	[0.47663817 0.02858106 0.49478078]	38.57%
		iPhone	[0.34836277 0.04228076 0.60935647]	40.49%
		Android	[0.47122535 0.08119662 0.44757803]	25.80%
		Apple TV	[0 0 0]	100%

我还将分离数据集后的准确率按照每种设备的比例做了加权平均，如在图14中，如图中蓝色线条所示。该图显示，从总体上看，外部因素在预测用户广告比例时有提升效果。但是在分离数据集后，Apple TV、iPhone、iPad的准确率高于总体准确率。因此本文建议，在预测用户观看时长时，当用户为Android用户时，将使用整个数据集且将外部因素作为一维特征训练出的模型作为其QoE模型；对于iOS用户，用iPad、iPhone、Apple TV 用户单独分离出数据集训练出的独立模型作为其QoE模型。

图18、图19、图20给出了在2分类的情况下，我们使用到的预测广告比例的决策树的可视化图示。对于5分类以上的决策树，其可视化图示过于复杂，分支过多也不便于辨认，我们通常使用内存中已经训练好的决策树直接用于预测。

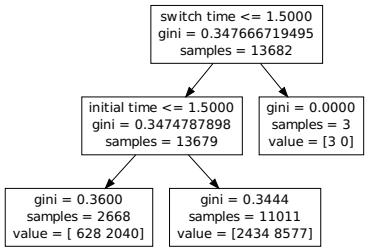

图 18: iPhone设备上对于广告比例预测的决策树

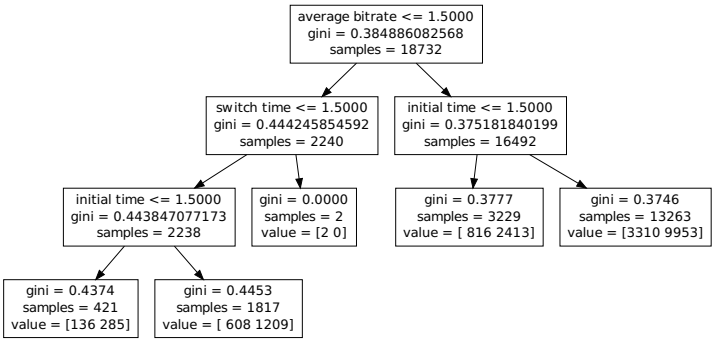

图 19: iPad设备上对于广告比例预测的决策树

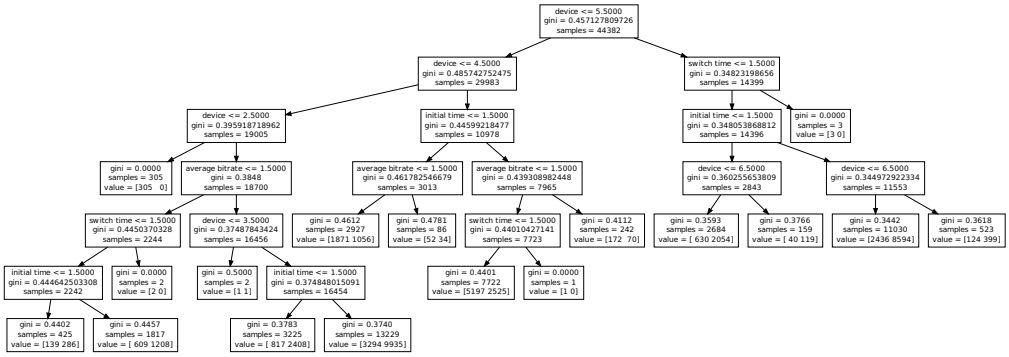

图 20: 将设备因素作为额外一维特征训练到的用于预测广告比例的决策树（请放大）

可以看到，图20看起来是图18和图19分别在device=iPhone和device=iPad分支上拼

接而来的，即在2分类情况下，依据外部因素分离数据集和将外部因素作为额外以为特征获得的总的决策树是完全相同的。事实上，这是一个正常现象。根据决策树的训练过程，系统总会将熵增益更大的特征放在更顶层的位置。而在相似数据分布的情况下，拥有更多分类的特征的熵增益比其他特征的熵增益更加大。在2分类的情况下，设备分类是分为4类的，因而设备分类有最高的熵增益，处于决策树最高层，相当于分离数据集的多个子决策树的拼合。

经过验证，当分类数大于5时，最高熵增益的特征就不再是设备信息，此时依据外部因素分离数据集训练获得的子决策树的聚合和将外部因素作为额外以为特征获得的决策树就不再相同。

6 总结

在本章中，本文对互联网视频上的大规模用户行为数据进行了分析，提出了一个全新的数据驱动的动态自适应流媒体QoE的建模方法。本文选择视频的平均码率、码率切换速率、视频开始观看时间三个因素作为评估互联网视频的质量因素；选择用户观看视频的设备类型作为评估视频质量的外部因素，并将用户的视频观看时长和广告观看比例作为用户QoE的量化标准。在对QoE进行建模的时候，本文将该任务视为分类问题，采用用三种成熟的分类器：朴素贝叶斯分类器、逻辑斯蒂回归（Logistic Regression）分类器和决策树分类器进行建模，获得了质量因素和外部因素对用户QoE的影响。实验结果表明，决策树能够更好地刻画QoE模型，因为作为输入特征的质量因素和外部因素不是正交的，它们之间存在一定的依赖关系。实验结果中还表明，当设备类型为iOS类用户时，本文倾向于分割用户后单独训练模型来预测用户的QoE。当设备类型为非iOS类用户时，本文倾向于使用整个数据集同时将外部因素作为一维特征来预测用户的QoE。

参考文献

[1] P.800 : Methods for subjective determination of transmission quality. http://www.itu.int/rec/T-REC-P.800-199608-I/en. Accessed: 2013-11-15.

[2] P.910 : Subjective video quality assessment methods for multimedia applications. http://www.itu.int/rec/T-REC-P.910-200804-I/en. Accessed: 2013-11-15.

[3] Streaming is going mainstream: The upward arc of online video, driven by consumers. https://www.cisco.com/web/about/ac79/docs/sp/Online-Video-Consumption_Consumers.pdf. Accessed: 2013-11-15.

[4] Video quality experts group (vqeg). http://www.its.bldrdoc.gov/vqeg/vqeg-home.aspx. Accessed: 2013-11-15.

[5] Athula Balachandran, Vyas Sekar, Aditya Akella, Srinivasan Seshan, Ion Stoica, and Hui Zhang. A quest for an internet video quality-of-experience metric. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, HotNets-XI, pages 97–102, New York, NY, USA, 2012. ACM.

[6] Athula Balachandran, Vyas Sekar, Aditya Akella, Srinivasan Seshan, Ion Stoica, and Hui Zhang. Developing a predictive model of quality of experience for internet video. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 339–350, New York, NY, USA, 2013. ACM.

[7] 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 SIGCOMM 2011 Conference*, SIGCOMM '11, pages 362–373, New York, NY, USA, 2011. ACM.

[8] M Watson. Http adaptive streaming in practice. In *In MMSys - Keynote*, 2011.