

# 网络用户视频体验影响因素的探究

## 摘要

在本文中，我们主要对用户视频体验建立模型。根据题目中所给出的大量数据，我们利用了神经网络，决策树，及回归分析等方法探索了不同变量之间的函数关系。最终可以根据网络侧变量对用户视频体验给出一个合理的预测。

首先，我们对网络侧变量和用户体验变量两两做定性分析，通过画图 and 求相关系数的方法，大致确定变量之间基本的相关性，得到一些初步的结论。

接着，考虑到原数据集较为冗杂，我们借助决策树的想法，对一些“边界数据”（定义见 3.2）进行筛选并剔除，为后续的神经网络模拟及回归分析做一些铺垫工作。

之后，我们用神经网络进行了一些拟合，用以判断题目所给出的自变量和因变量之间是否有明确的函数关系。事实证明，一个因变量和自变量之间有很强的关联，而另一个的相关性则相对较弱。这一部分使得我们在探寻的时候有方向可寻，而不是去猜测两者是否真的有函数关系。这为模型的搭建奠定了基础。

最后，我们利用 Matlab 回归分析的方法，给出了各用户体验变量随网络侧变量之间的具体表达式，并做相关的检验，画出了拟合的效果图，并设计方法进行模拟。最后我们还求出了各项得分，与相应的体验变量之间的具体函数关系。

**关键词：**用户视频体验，神经网络，机器学习，回归分析，随机模拟

## 一、问题背景及文献综述

### 1.1 问题背景

随着无线宽带网络的升级，以及智能终端的普及，越来越多的用户选择在移动智能终端上用应用客户端APP观看网络视频。看网络视频影响用户体验的两个关键指标是初始缓冲等待时间和在视频播放过程中的卡顿缓冲时间，我们可以用初始缓冲时延和卡顿时长占比来定量评价用户体验。研究表明影响初始缓冲时延和卡顿时长占比的主要因素有初始缓冲峰值速率、播放阶段平均下载速率、端到端环回时间（E2ERTT）等。

我们的目标是根据附件所给出的大量数据，建立用户体验评价变量（初始缓冲时延，卡顿时长占比）与网络侧变量（初始缓冲峰值速率，播放阶段平均下载速率，E2ERTT）之间的函数关系。

## 1.2 文献综述

我们按照本题“网络侧估计用户视频体验”检索后得到与本题内容相关的文献有：

- （1） 康亚谦，无线视频流业务的用户体验质量估计模型及其应用，浙江大学
- （2） 于新，无线网络中端到端视频流业务的用户体验质量预测及优化技术，浙江大学
- （3） 李俐莹，网络视频的用户体验质量评价，石家庄铁道大学
- （4） 焦阳，无线网络中端到端视频流业务的用户体验质量预测及优化技术分析，河北联强通信科技有限公司

其对应研究内容大致如下：

- （1） 通过一种基于径向基函数的神经网络来得出端到端跨层参数与（E2ERTT）用户体验质量之间的关系。并由此提出了一种视频流传输控制优化机制以提高传输稳定性。
- （2） 针对无线网络中端到端视频流业务的用户体验质量预测问题，论文引入机器学习理论中模型组合的思想，建立了一种基于梯度提升决策树（GBDT）算法的用户体验评价预测模型。并由此提出了视频流的动态码率适配机制，使得播放更加流畅。
- （3） 通过对网络参数和视频内容对视频质量的影响、考虑视频内容的体验质量评价模型和基于机器学习的QoE与QoS评价模型的分析研究，本文

利用主成分分析、回归分析、深度学习等方法,使用NS2和Evalvid仿真工具开展网络视频质量的用户体验质量评价的研究。

(4) 也是用了基于GBDT算法的用户体验评价预测模型。

综上所述,我们可以看出类似(1)这样的神经网络工作并不是我们希望的,因为神经网络模型并不能给出一个确切的函数关系式;(3)中所做的工作在本题的题设中就已经完成,即我们的主要因素就是初始缓冲峰值速率、E2ERTT及播放阶段平均下载速率,我认为这三个变量已经足够少了,用不到所谓的主成分分析或者深度学习这类算法,而其中的回归分析则是平凡的;(2)和(4)的预测模型都是基于GBDT算法的机器学习模型,该模型本质上就是一个分类回归树模型,本文中分类和回归分开做,达到的效果也是差不多的。另一方面,由于我们与其数据集有很大的差异,例如其中丢包率是一个很重要的数据但是本题没有,所以他们的方法也不具有参考性。

故,现有的对“网络侧估计用户视频体验”的预测模型都不能很好地解答本题,因此我们接下来的工作将会是兼具挑战性与创新性的。

## 二、 假设

为了方便研究课题,根据题意及附件中所给出的数据,我们做如下的假设:

1. 整个视频观看过程分为2个阶段,即初始缓冲阶段和播放阶段。
2. 播放阶段的总时长为卡顿时长和播放时长之和,且给出的数据中,每次播放阶段的总时长均为30s左右。因此可固定取30000ms来呈现。
3. 定义卡顿占比为卡顿时长除以播放时长。
4. 题目中要求初始缓冲量为4s,单位为秒,乘以视频码率可以得到初始缓冲阶段需要下载的数据量。
5. 根据附件中所给数据,视频码率只有2903, 2934, 2966这3个值,波动范围很小,可认为是个常数。可取一个中间值来代替。
6. 认为卡顿门限是0,即在播放过程中,若缓冲区数据量为0时,则进入卡顿状态。

7. 根据题意，重播放门限为2.7s，即在卡顿时，若缓冲区数据量足以播放超过2.7s，则重新进入播放阶段。

### 三、研究方法及初步研究成果

为了方便理解，先陈述一个问题的等价描述。

有一个水槽（相当于数据缓存区），水槽有一个进水口S1（相当于客户端）和一个出水口S2（相当于播放器）。S1的进水量（相当于下载的数据量）是随机的。但是S2的出水量（相当于视频播放量）按照如下方式决定：

(1)最初时，只有当储水量能够放4s时，出水口才打开（相当于初始缓冲需要有4s的缓冲量）。

(2)出水口打开时，出水速度是固定的（播放速度是固定的）。

(3)出水口一旦打开，只有当水槽内的水全部流完，出水口才关闭（相当于视频播放时的卡顿）。

(4)出水口关闭后，要等水槽中的水能够放2.7s时，出水口才重新打开（相当于重播放门限为2.7s）。

#### （一）数据的预处理和定性分析

首先，为了方便我们建立模型，得到更加准确，更有价值的结论，我们需要对原始数据做一些预处理，即对已有数据做一定筛选，仅保留有价值的部分来进行操作。

根据题意及简化假设4，由于初始缓冲量为4s。那么乘以视频码率得到缓冲阶段需要下载的数据量为：

$$\begin{aligned}\text{需要下载数据量} &= \text{初始缓冲量} \times \text{视频码率} \\ &= 4\text{s} \times 2934\text{kbps} \times \frac{128\text{bytes}}{\text{kb}} \\ &= 1502208\text{bytes}\end{aligned}$$

但是从附件中所给的数据来看，附表中第N列给出了初始缓冲阶段实际下载的数据量，并不是所有的记录中都与我们实际求得需要下载的数据量一致，甚至有一些数据中实际的初始缓冲阶段下载数据量，与需要下载的数据量差距较远，可能会对我们之后的分析造成干扰，影响到我们之后分析的准确性。

附表给出的数据共有 89266 条，这里我们列出实际下载数据量位于不同区

间值的记录数，并做出相应的直方图：

表 1 实际下载数据量不同区间值的数据条数

区间值（单位：byte）	纪录条数
<200000	47
1500000-1550000	22195
1550000-1600000	45092
1600000-1650000	15121
>1650000	6811
总计	89266

从表中数据可以看到，下载数据量小于 200000 的纪录很少，并且与我们的假设条件不符，可认为是异常数据，可以将其剔除。而剩下的绝大多数的下载量位于 1500000-1650000byte 之间。因此我们画出这一区间实际初始下载量的直方图如下：

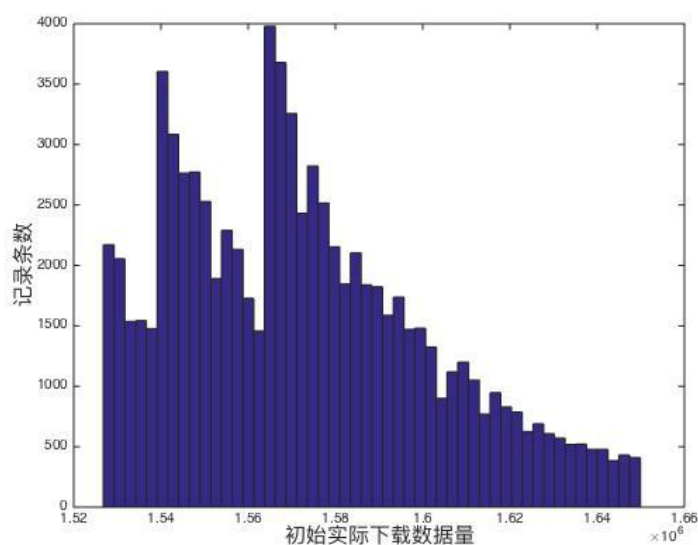


图 1 初始缓冲实际下载数据量直方图

我们需要对所给的数据进行了筛选。由于缓冲阶段实际下载数据量会直接影响到对缓冲阶段回归分析的效果，为了使得到的结果更准确，且符合题目中假设的要求，在对缓冲阶段做回归分析时，我们只保留与我们在上式求得的值较为接近的数据，这些数据对我们的研究有较高的价值。在这里我们人为保留了下载

数据量在 1500000 – 1550000bytes 之间的数据。

在这一节，我们通过画图 and 求相关系数的方法，来对各变量做初步的定性分析。

由题意，网络侧变量为初始缓冲峰值速率（B 列）、E2E RTT（C 列），播放阶段平均下载速率（D 列）。而用户体验变量为初始缓冲时延（E 列）、卡顿占比（F 列）。但根据简化假设，播放时长与卡顿时长之和固定为 30s，因此播放时长（J 列）可以唯一确定卡顿占比。

因此我们选 B、C、D 列为自变量，E、J 列为因变量。并分别考虑它们之间的关系。

并作以下变量表：

$x$	初始缓冲峰值速率（B 列）
$y$	E2E RTT（C 列）
$z$	播放阶段平均下载速率（D 列）
$u$	初始缓冲时延（E 列）
$v$	卡顿占比（F 列）
$w$	播放时长（J 列）

根据假设 2、3，我们知道  $v$  和  $w$  存在一定的关系，即， $v = \frac{L}{w} - 1$ ，其中  $L=30000\text{ms}$ ，表示视频时间，是个常数。

### 3.1.1 初始缓冲时延（E 列）和初始缓冲峰值速率（B 列）之间的关系。

首先画出这两个变量的散点图( $x - u$ )如下：

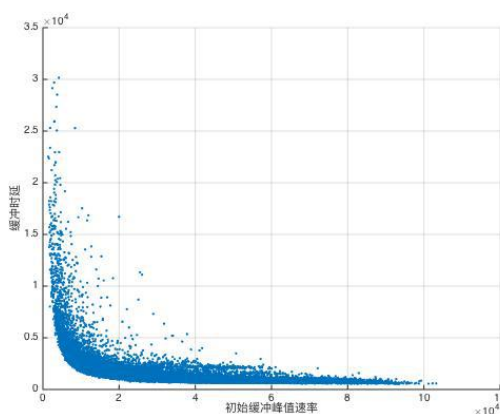
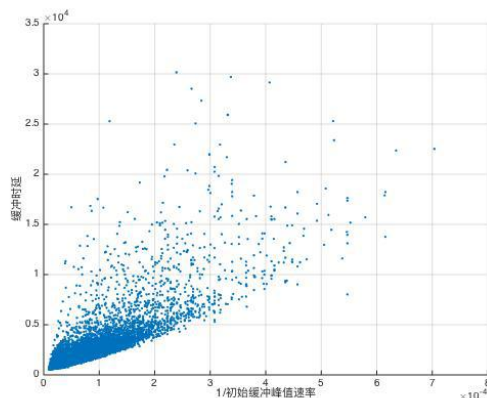


图 2  $u$  和  $x$  之间的图像

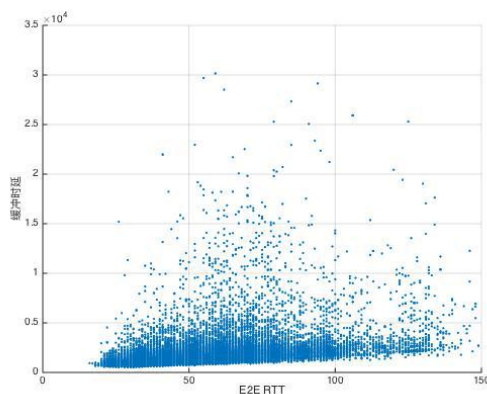
从图中来看， $u$  与  $x$  大致呈反比例关系。因此我们推测  $u$  和  $\frac{1}{x}$  应该有较强的线性关系。因此我们画出  $u$  和  $\frac{1}{x}$  的图像如下：

图 3  $u$  和  $\frac{1}{x}$  之间的图像

并计算它们的相关系数为 0.8301。认为有较强的线性相关性。

### 3.1.2 初始缓冲时延（E 列）和 E2E RTT（C 列）之间的关系

同样的，我们画出这两个变量的散点图( $y - u$ )如下：

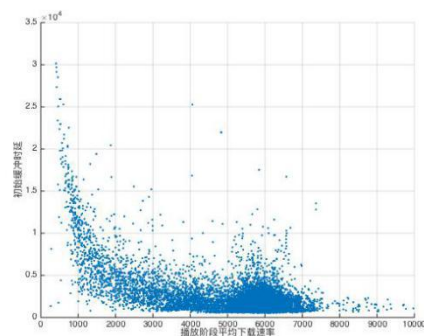
图 4  $u$  和  $y$  之间的图像

经计算，它们的相关系数为 0.3799，呈较弱正相关关系。点的分布较为混乱。

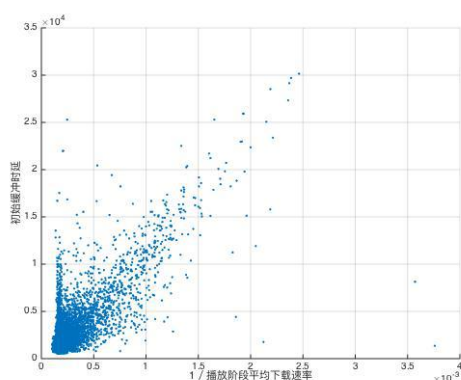
### 3.1.3 初始缓冲时延（E 列）和播放阶段平均下载速率（D 列）的关系

首先我们做出散点图如下。这里由于绝大多数的点的横坐标值不超过 10000，为了更清晰的看出图像的情况，我们仅保留这些点来画图( $z - u$ )。



图 5  $u$  和  $z$  之间的图像

和 1.1 中一样，从图中看可能呈反比例关系。因此我们画出  $u$  和  $\frac{1}{z}$  的图像如下：

图 6  $u$  和  $\frac{1}{z}$  之间的图像

从图中看大概呈正相关关系，计算相关系数为 0.7600，有较强相关性。从直观理解，似乎播放阶段的变量可能对缓冲时延不会造成什么影响，但播放阶段的下载速率能一定程度上反映总体网速的快慢，从而与缓冲时延有一定的关系。

### 3.1.4 播放时长（J 列）与播放阶段平均下载速率（D 列）的关系

画出散点图( $z-w$ )如下：

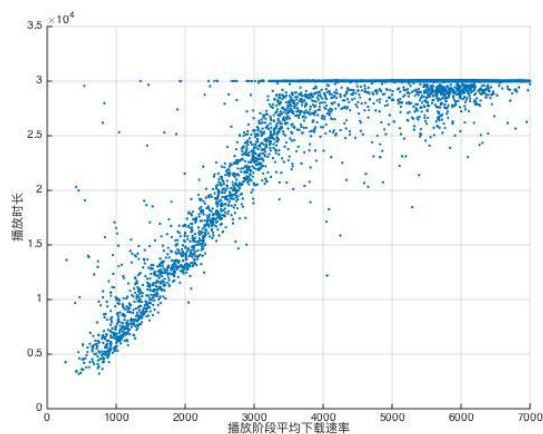


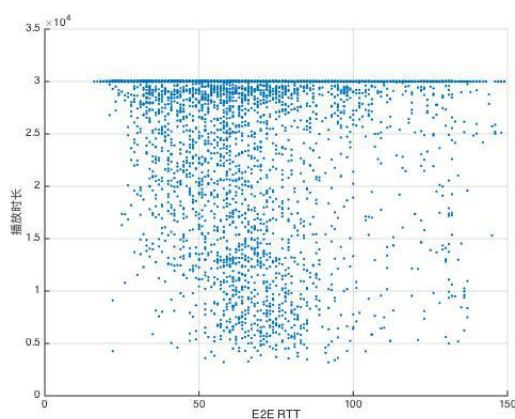


图 7  $w$  和  $z$  之间的图像

从直观上看，发现当下载速率小于 4000 时，它们大致呈线性关系，通过计算其相关系数，达到 0.9429，有非常强的线性相关性。而大于 4000 时，播放时长几乎分布在 25000ms 到 30000ms 之间，即卡顿的时间是比较短的。这也是我们直观上能够理解的。

### 3.1.5 播放时长（J 列）与 E2E RTT（C 列）之间的关系

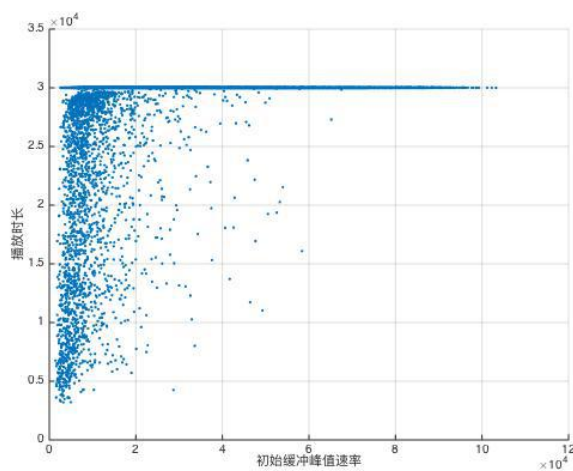
画出散点图( $y-w$ )如下：

图 8  $w$  和  $y$  之间的图像

从图中看，点的分布比较混乱，计算其相关系数为 -0.2602，因此认为这两个变量之间的关系不大。

### 3.1.6 播放时长（J 列）和初始缓冲峰值速率（B 列）之间的关系。

画出散点图( $x-w$ )如下：

图 9  $w$  和  $x$  之间的图像

图像分布较为散乱，相关系数仅有 0.2978 因此认为这两个变量之间关系不大，这也是符合常理的。

### 3.1.7 小结：

从以上的定性分析可以看到，初始缓冲时延（E列）和初始缓冲峰值速率（B列）呈较强反比例关系，和E2E RTT（C列）有很弱的正相关性。播放时长（J列）主要和播放阶段平均下载速率（D列）有较强线性关系，与其他两个自变量的关系不大。

即：

$u$  和  $\frac{1}{x}$  有很强的线性关系

$u$  和  $y$  有较弱的线性关系

$w$  和  $z$  有一定的线性关系

由关系式  $v = \frac{L}{w} - 1$  其中  $L = 30000\text{ms}$  为常数

得到：

$\frac{1}{v}$  和  $z$  有一定的线性关系

## （二） 决策树

从前面定性分析中的三点图中我们可以看出，数据之间的关系颇为复杂，故我们不妨作这样一种尝试，即我们可不可以通过一种简单有效的方法来“去除”一部分数据，可以理解为一种“分类”的操作，于是我们便想到了机器学习中的决策树算法（大致的思路<sup>[3]</sup>如图 9），其最终的结果可以看成是一些分段函数的复合。

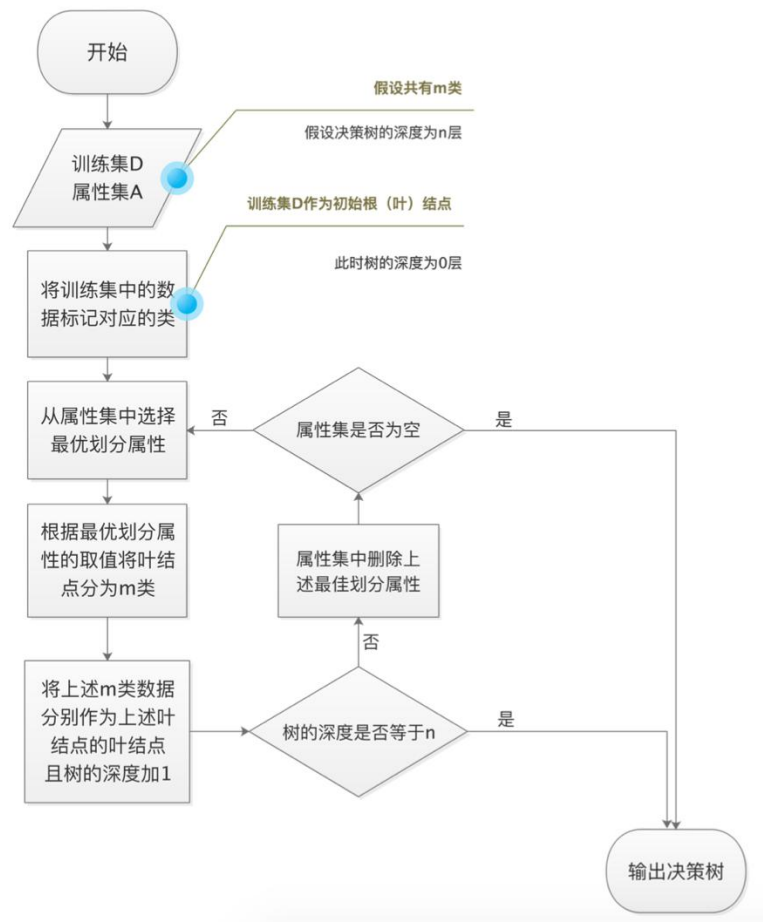


图 10 决策树算法简化流程图

在这里，我们定义边界数据为以下三种：

缓冲时延  $> 10000\text{ms}$ ，即缓冲得分  $= 1$  的数据

卡顿占比  $\geq 0.3$ ，即卡顿得分  $= 1$  的数据

卡顿占比  $= 0$ ，即卡顿得分  $= 5$  的数据

首先需要声明一点，即我们本题的最终目标是要找到  $x, y, z$  与  $u, v$  之间的函数关系式，但是在这里我们不得不承认，我们所定义的“边界数据”从一定程度上规避了一部分原问题，即我们在去除这些边界数据后，我们不可能再得到原题所求的完整的函数关系式。但是这不失为一种舍小为大的可行的方法，如果我们能做出不错的结果，那么这也算是一份可作参考的答卷了。

## 具体结果

### 3.2.1 缓冲阶段

根据边界数据，我们将数据按照缓冲得分分为得分为 1（class=1）的与得分>1（class=~）的两类，图 11 就是决策树深度为 2 时的结果：

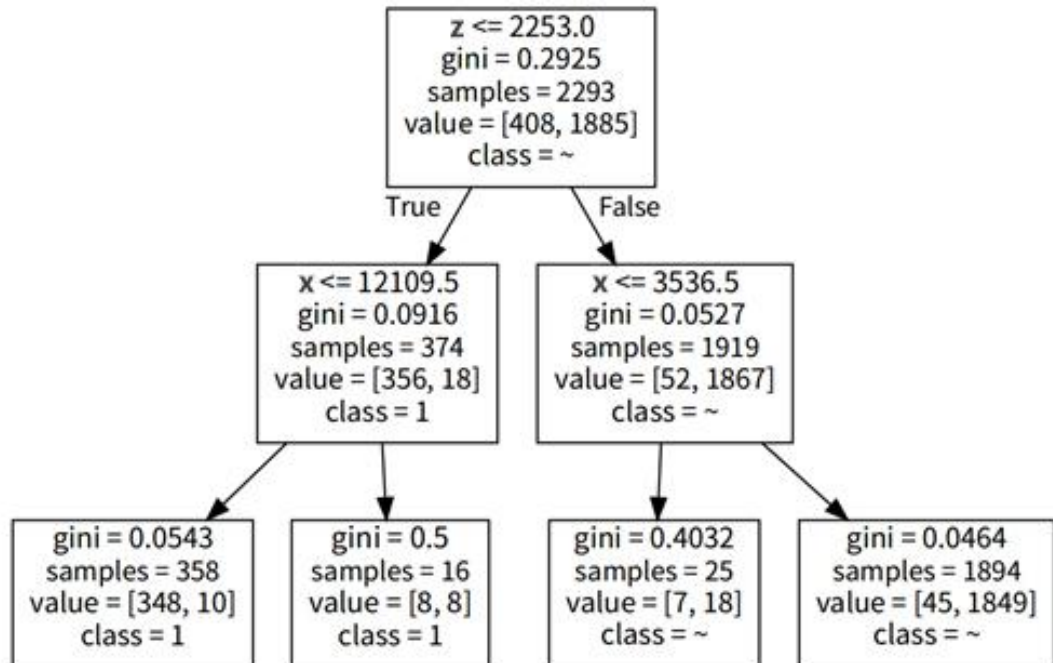


图 11 缓冲阶段决策树分类结果

得到关系如下：

$z \leq 2253.0$  时，缓冲得分=1，缓冲时延 < 10000ms

$z > 2253.0$  时，缓冲得分>1，缓冲时延  $\geq 10000$ ms

按照上述关系我们在测试集上得到的正确率达到了 **94%**

### 3.2.2 播放阶段

根据边界数据，我们将数据按照卡顿得分分为得分=1，得分=5，以及剩余部分这三类，图 12 是决策树深度为 2 时的结果：

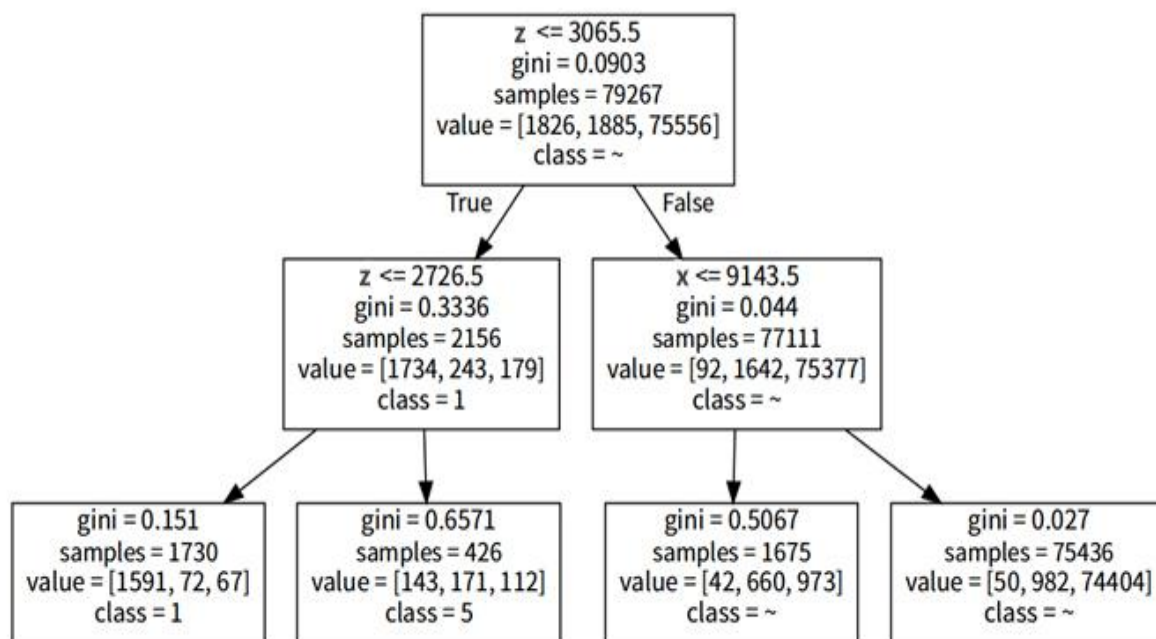


图 12 播放阶段决策树分类结果

得到关系如下：

$z \leq 2726.5$  时，卡顿得分=1，卡顿占比  $> 0.3$

$2726.5 < z \leq 3065.5$  时，卡顿得分=5，卡顿占比=0

$z > 3065.5$  时， $1 < \text{卡顿得分} < 5$ ， $0 < \text{卡顿占比} < 0.3$

按照上述关系我们在测试集上得到的正确率达到了 **97%**

### 3.2.3 注：

我们从测试集得到的准确率来看，上述的分类效果还是非常鼓舞人心的。其中需要特别说明的一点就是关于决策树深度的选择，即为什么选择深度为 2。我们在分析的时候是测试过不同的深度的，但是最后发现深度越深（大于 3 层），其在测试集上的正确率开始下降（应该由于过拟合造成的），然后在决策树深度为 2 或 3 中我们综合考虑了模型的泛化误差以及易解释性，即深度为 2 时的测试正确率仅仅是略低于深度为 3 的情形，但是模型更为简洁，故选择深度为 2。

另外，我们需要再强调一下，经过如此分类过后，我们在之后的分析中便会把这些边界数据去除，如此一来，我们便不能得到例如卡顿占比  $> 0.3$  时对应的确切的函数关系式了，但是从卡顿得分这个评价标准来看，当卡顿得分=1 时，

对于用户来说，体验就已经很差了，所以其实我们再得到其确切的函数关系式的意义也就很有限了，因此，虽然我们此举一定程度上回避了原问题，但是从“预测用户体验”的角度来看，我们的这个模型是起到了一定的作用的。

### （三）神经网络

我们用matlab的神经网络工具箱neural net fitting来对数据进行了拟合。其中将70%的数据作为训练集（training set），15%的数据作为验证集（validation set），15%的数据作为测试集（test set）。

这里简单介绍一下三个集合的作用：

训练集：学习样本数据集。主要是用来训练模型的。

验证集：对学习出来的模型，调整参数，确定网络结构，控制模型的复杂程度。

测试集：测试训练好的模型的分辨能力。

我们使用了含有一层隐藏神经元的神经网络结构。众所周知，对于在  $R^n$  中的任意紧集上的连续函数（或可积函数），都可以由单隐层前向神经网络一致逼近该函数至任意给定精度。由于题目是为了寻求 初始缓冲时延、卡顿时长占比 与 初始缓冲峰值速率、播放阶段平均下载速率、E2ERTT 之间的函数关系，因此我们搭建了两个神经网络，第一个用来寻找初始缓冲时延 与 初始缓冲峰值速率、播放阶段平均下载速率、E2ERTT 之间的关系，第二个用来寻求卡顿时长占比 与 初始缓冲峰值速率、播放阶段平均下载速率、E2ERTT 之间的关系。

#### 3.3.1 关于初始缓冲时延的神经网络的训练结果：







Results			
	 Samples	 MSE	 R
 Training:	62177	197370.38886e-0	8.57786e-1
 Validation:	13323	212128.37935e-0	8.55969e-1
 Testing:	13323	202150.71306e-0	8.51986e-1

图13 初始缓冲时延的神经网络训练结果



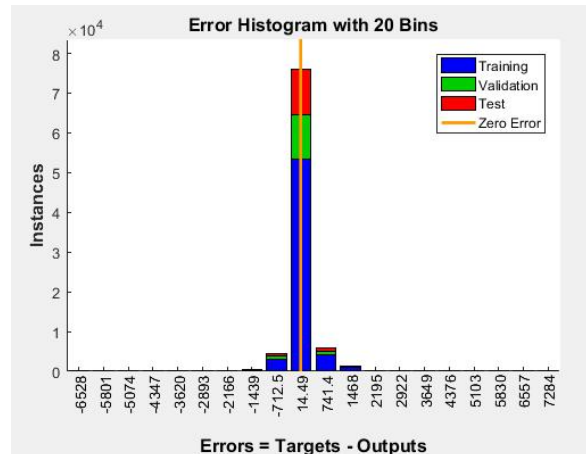


图14 初始缓冲时延的误差分布直方图

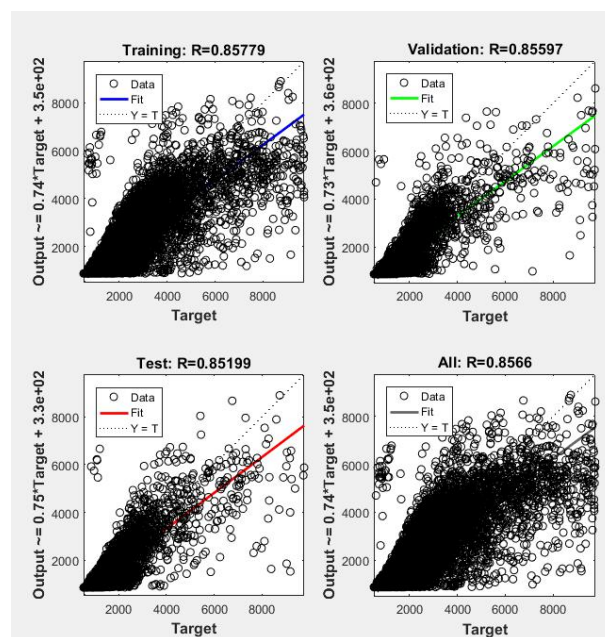


图15 初始缓冲时延的线性回归效果

### 3.3.2 关于卡顿时长占比的神经网络的训练结果：

Results			
	Samples	MSE	R
Training:	1460	3.66140e-3	6.32418e-1
Validation:	313	2.70906e-3	7.09256e-1
Testing:	313	3.75801e-3	6.44808e-1

图16 卡顿时长占比的神经网络训练结果



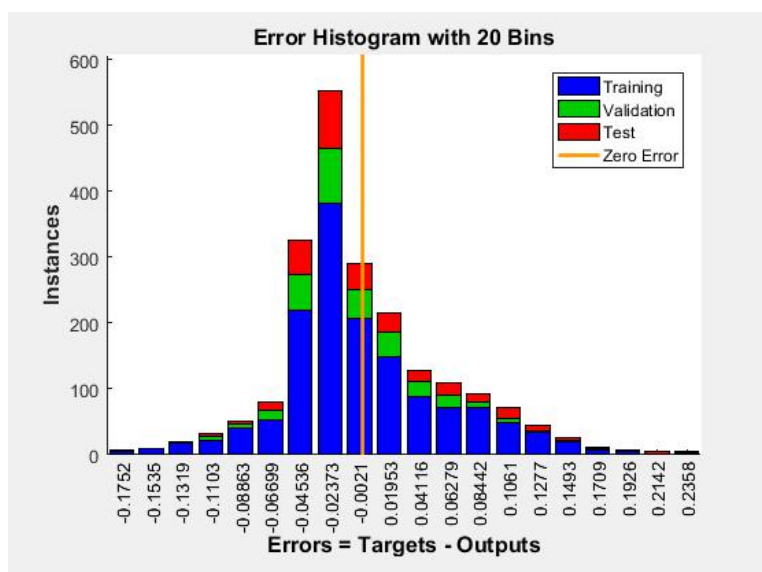


图17 卡顿时长占比的误差分布直方图

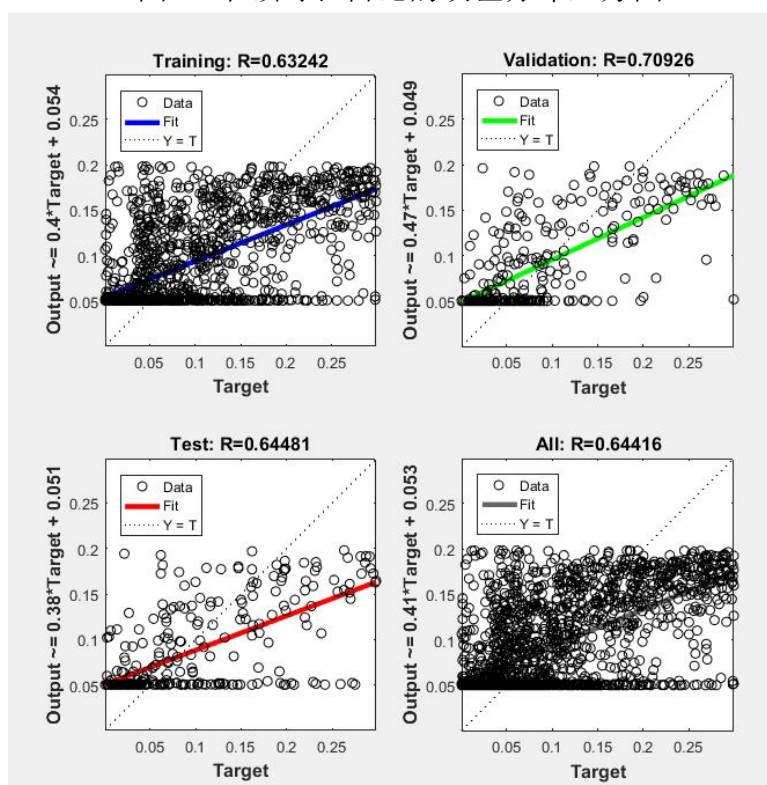


图18 卡顿时长占比的线性回归效果

这说明数据集的确在一定程度上是相互关联的（否则R会非常小），而相比之下，初始缓冲时延与三者的关系更大一些，而卡顿时间占比与三者的关系相对小一些（R值相对小了很多）。但是由于神经网络不能提供出具体的表达式，因此我们又采用了别的方案。

#### （四） 回归分析

我们根据附件中所给出的数据，来逐步探索不同变量之间存在的关系，主要运用了曲线拟合及线性回归的方法<sup>[1]</sup>。

### 3.4.1 各项用户评价体验变量与网络侧变量之间的关系

下面我们来具体研究各项用户体验评价变量与网络侧变量之间的关系。以下我们主要分为播放阶段和缓冲阶段两个阶段来研究。首先我们先利用所给的数据画出图像并做定性分析。

#### 3.4.1.1 播放阶段

首先，与（一）中相同，根据之前决策树的部分，我们已经对边缘数据有了一定的分类，于是我们在对播放阶段作回归分析时，将边缘数据剔除，仅保留余下的部分，这样既方便处理，又可以得到相对更有价值的结果。

以下我们用两种方式进行回归分析。

##### 3.4.1.1.1 分段线性回归

###### case1 两个线性部分可以不连续

有了之前的定性分析，我们要对一些关系较强的数据做定量的回归分析。由于播放阶段的平均下载速率（D列）会影响到此时缓冲区的数据量，于是我们考虑先考虑这两个变量之间的关系。

之前我们画出的散点图如下：

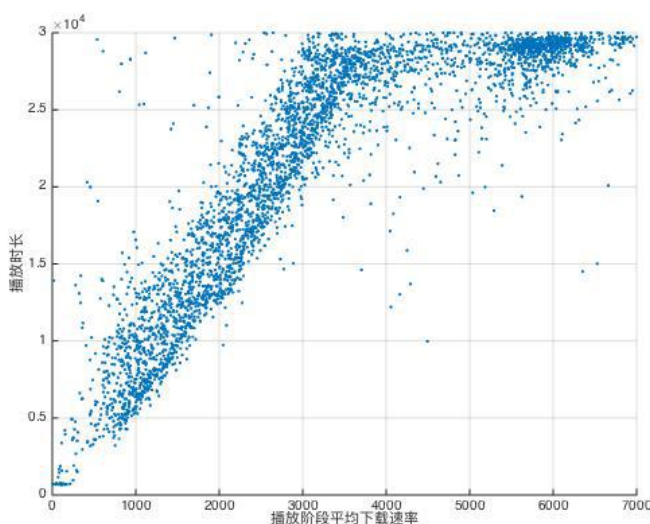


图19 播放时长  $w$  和播放阶段平均下载速率  $z$  之间的散点图

直观的从图像来看，当播放阶段下载速率较小时，小于3000时，播放时长和下载速率大致有线性关系。而当播放阶段下载速率较大时，大于3000时，播放时长大多数都在25000ms-30000ms之间，即卡顿时长是比较小的。

因此为了方便研究，首先按下载速率是否小于3000，把以上数据分为两段。

使用matlab中的曲线拟合工具，分别对它们做线性回归。结果如下：

对于下载速率小于3000的部分，有：

```
f =

Linear model Poly1:
f(x) = p1*x + p2
Coefficients (with 95% confidence bounds):
    p1 =          7.29   (7.112, 7.469)
    p2 =         1808   (1458, 2158)
```

用  $z$  来表示平均下载速率， $w$  来表示播放时长，即有如下的线性关系：

$$w = 7.29z + 1808$$

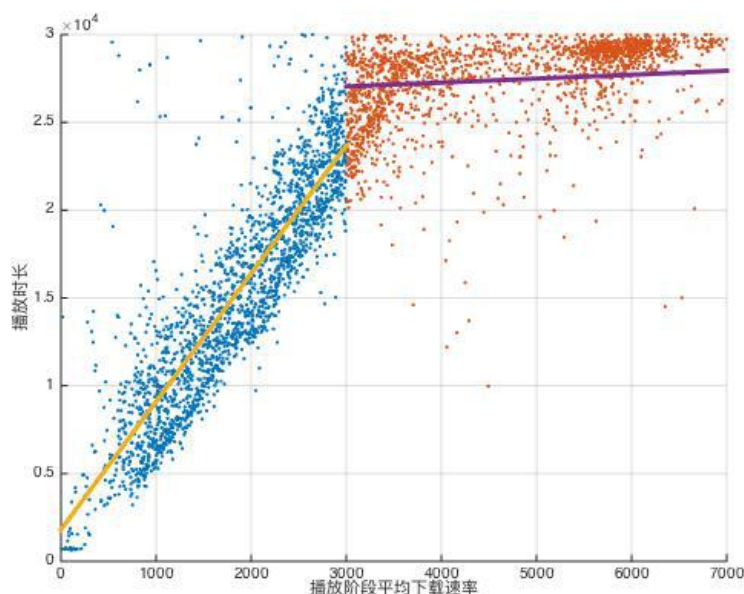
对于下载速率大于3000的部分，有：

```
f =

Linear model Poly1:
f(x) = p1*x + p2
Coefficients (with 95% confidence bounds):
    p1 =         0.2246   (0.173, 0.2761)
    p2 =    2.637e+04   (2.607e+04, 2.666e+04)
```

表达式为： $w = 0.2246z + 26370$

画出拟合的效果图如下：

图20  $w-z$  拟合效果图

总体来看效果还是不错的，我们对回归的系数做  $t$  检验，检验结果如下：

在下载速率小于3000时， $t$  值为80.1458；在下载速率大于3000时， $t$  值为8.5426。

通过查表比较，知有95%以上的把握认为回归效果是显著的。

因此我们得到的最终结果为

$$w = \begin{cases} 7.29z + 1808, & z \leq 3000 \\ 0.2246z + 26370, & z > 3000 \end{cases}$$

总均方根误差为2917.4ms。

从检验结果我们可以发现，在下载速率大于3000时， $t$  值要比下载速率小于3000的部分小的多，我们根据实际情况推测原因如下：由于在这里大多数点下载速率分布在3000-6000，超出视频码率（值为2934kbps）并不多，我们推测了一下产生这种现象可能的原因：由于网络并不稳定，下载速率随时间可能会有较大波动，一旦下载速率低于视频码率（2934kbps），缓冲区内的数据可能会迅速消耗，造成短期的卡顿。因此卡顿时长会很大程度上依赖于这种随机性的波动，导致随机性相对强一些。

## Case2 两个线性部分需要连续

公式推导：

分段线性函数的表达式为：

$$f(x) = a(x - x_0) + y_0 (x \leq x_0)$$

$$f(x) = b(x - x_0) + y_0 (x > x_0)$$

可以将数据分为两个部分，第一部分的横坐标不超过  $x_0$ ，记为  $(s_i, t_i)$ ；第二部分的横坐标大于  $x_0$ ，记为  $(p_i, q_i)$ ，将其代入表达式中，则只要求

$$\sum_{i=1}^n (f(s_i) - t_i)^2 + \sum_{j=1}^m (f(p_j) - q_j)^2$$

的最小值。

我们注意到，表达式展开后并非线性。因此对变量  $x_0$  在范围  $[3000, 4000]$  内做搜索。

先以一百为跨度，再在最优范围内不断缩小跨度。因此在对每个  $x_0$  而言，上面的式子可以展开为如下的形式：

$$g(a, b, y_0) = \sum_{i=1}^n [a(s_i - x_0) + y_0 - t_i]^2 + \sum_{j=1}^m [b(p_j - x_0) + y_0 - q_j]^2$$

分别求目标函数关于变量的偏导数，有：

$$\frac{\partial g}{\partial a} = \sum [a \text{ 的一次项} + y_0 \text{ 的一次项}] (s_i - x_0) = 0$$

$$\frac{\partial g}{\partial b} = \sum [b \text{ 的一次项} + y_0 \text{ 的一次项}] (p_i - x_0) = 0$$

$$\frac{\partial g}{\partial y_0} = \sum [a \text{ 的一次项} + y_0 \text{ 的一次项}] + \sum [b \text{ 的一次项} + c \text{ 的一次项}] = 0$$

解上述方程组，可求得目标函数的极小值。拟合结果如下。

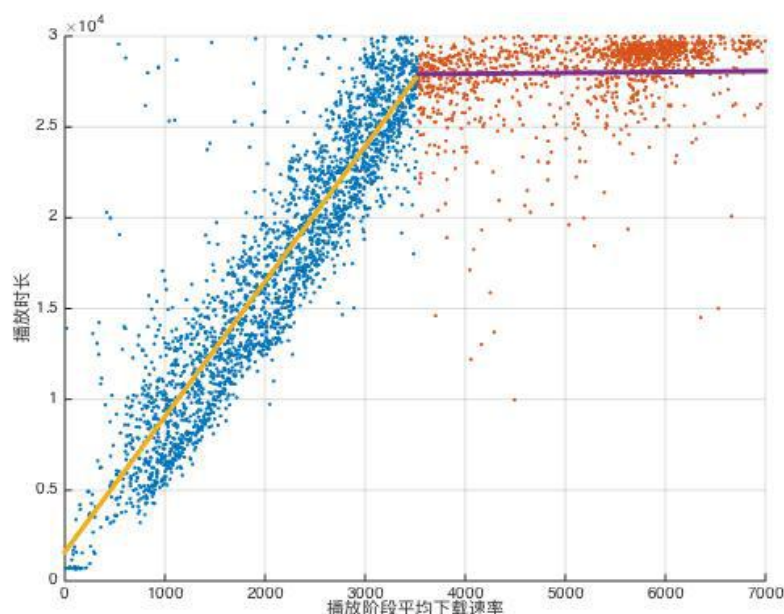


图21  $w-z$  连续的拟合效果图

$$w = 7.4445(z - 3534.3) + 27910 (z \leq 3534.3)$$

$$w = 0.0492(z - 3534.3) + 27910 (z > 3534.3)$$

我们对回归结果做  $t$  检验，当下载速率小于3534.3时， $t$  值为113.3681。当下载速率大于3534.3时， $t$  值为1.62，通过查表比较，并求得  $p$  值为0.10，因此有90%的把握认为回归效果是显著的。但在  $t$  检验中，我们一般要求超过95%的把握才认为通过  $t$  检验，在这里  $t$  检验并没有通过。

之后为了探索  $w$  是否和  $z$  的高次项有关系，我们又对  $w$  和  $z$  的五次多项式做回归，但发现各项系数的  $t$  检验均不能通过，于是我们又分别求了  $w$  与  $z^2, z^3, z^4, z^5$  的相关系数  $R$ ，看是否有一定的关系。求得相关系数分别为0.0074, -0.0014, 0.0013, 0.0058，都很小，因此几乎可以认为在此时  $w$  与  $z$  没有关系。因此可取  $w$  为一常数，为保证连续性，当  $z > 3534.3$  时，取  $w = 27910$  即可。事实上之前得到的系数为0.0492，很小，用常数代替与之前的结果也较为接近。

当已知后一段我们用常数来表示，我们还可以适当调整  $x_0$  的值，使得总残差平均进一步变小。调整后的结果如下：

$$w = \begin{cases} 7.4270(z - 3556.6) + 28040 & (z \leq 3556.6) \\ 28040 & (z > 3556.6) \end{cases}$$

最后我们用补充数据来对我们得到的结果进行验证，计算播放时长的残差平均为3450.3ms，即3.45秒左右。可见效果还是不错的。

### 3.4.1.1.2 logistic回归分析

根据散点图的分布情况，以及数据的特性（所有数据有上确界）我们考虑到用logistic函数来拟合也许能得到更好的效果。根据题意，我们用如下形式的函数：

$$w = 30000 \cdot \frac{1 - e^{az+b}}{1 + e^{az+b}}$$

其中  $a, b$  为待定的系数。

对上式做变换，有

$$\ln\left(\frac{1 + w/30000}{1 - w/30000}\right) = az - b$$



我们令

$$s = \ln\left(\frac{1 + w/30000}{1 - w/30000}\right)$$

那么  $s$  和  $z$  呈线性关系。于是可以用一元线性回归的方法来确定  $a$  和  $b$  的值。

利用matlab的线性回归功能，我们求得  $a = 7.4578 \times 10^{-4}$ ， $b = 0.1627$ 。

画出图像如下图所示：

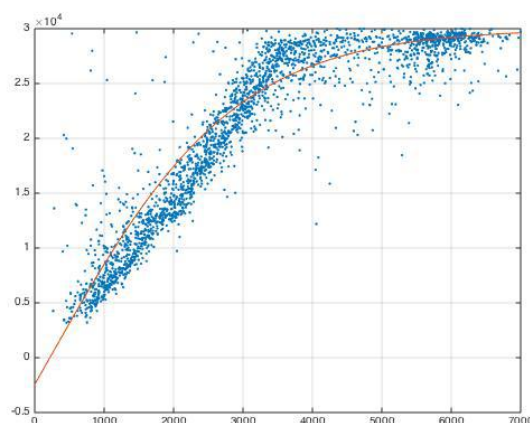


图22 logistic回归分析效果图

我们对回归进行了检验：其中  $R^2$  值为0.7713，F值为8159.7，p值为0。总体效果还是不错的。

用logistic回归方法求得的最终表达式为：

$$w = 30000 \cdot \frac{1 - e^{7.4578 \times 10^{-4} z + 0.1627}}{1 + e^{7.4578 \times 10^{-4} z + 0.1627}}$$

补充说明：事实上，这里的30000只是适用这道题目而言的，因为对于一般的情况，播放时长上限可能不是30s，因此应该换成

$$v = kT \frac{1 - e^{az+b}}{1 + e^{az+b}}$$



其中,  $k$  表示比例系数,  $T$  表示播放总时长,  $a, b$  是待定参数。由于题目中只给出了  $T=30s$  的数据, 因此无法具体求出  $k$  的值, 就这个题目而言, 之前给出的公式已经够用了。

### 3.4.2 缓冲阶段

同样的, 为了简化起见, 我们也只保留初始缓冲下载数据量( $N$ 列)在1500000到1550000之间的数据。通过查找一些相应的资料<sup>[2]</sup>, 我们得到了视频整个播放的过程示意图如下:

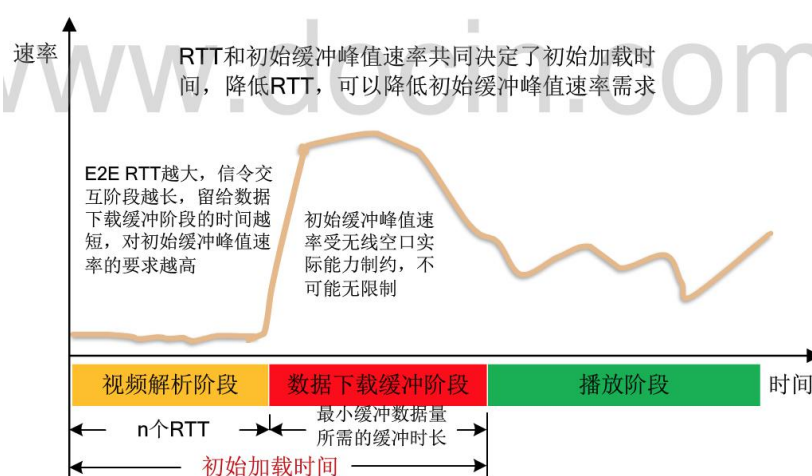


图23 视频播放过程示意图

因此, 我们知道初始缓冲时延( $E$ 列)与初始缓冲峰值速率( $B$ 列)即E2E RTT( $C$ 列)有关。我们用  $u$  来表示初始缓冲时延( $E$ 列), 用  $x, y$  分别表示初始缓冲峰值速率( $B$ 列)即E2E RTT( $C$ 列)。

在之前的决策树部分, 我们已经分析清楚了在什么情况下缓冲得分为1分, 即初始缓冲时延大于10000ms。因此在此处我们做进一步的筛选, 只保留缓冲时延不超过10000ms的数据来做回归分析。

下面我们来研究它们之间的关系。根据之前的定性分析, 我们知道  $u$  和  $\frac{1}{x}$  有较强的正相关性, 而与  $y$  有很弱的正相关性。

因此我们考虑用  $\frac{1}{x}$  和  $y$  的线性函数来做回归分析。为了方便, 我们记  $x' = \frac{10^6}{x}$ 。利用Matlab中的曲线拟合工具箱, 可得如下结果:

Linear model Poly11:  
 $f(x,y) = p00 + p10*x + p01*y$   
 Coefficients (with 95% confidence bounds):  
 $p00 = -230.8 \quad (-249.8, -211.9)$   
 $p10 = 22.04 \quad (21.87, 22.21)$   
 $p01 = 14.39 \quad (14.07, 14.72)$

即：

$$u = f(x,y) = -230.8 + 22.04x + 14.39y$$

拟合的效果图片如下：

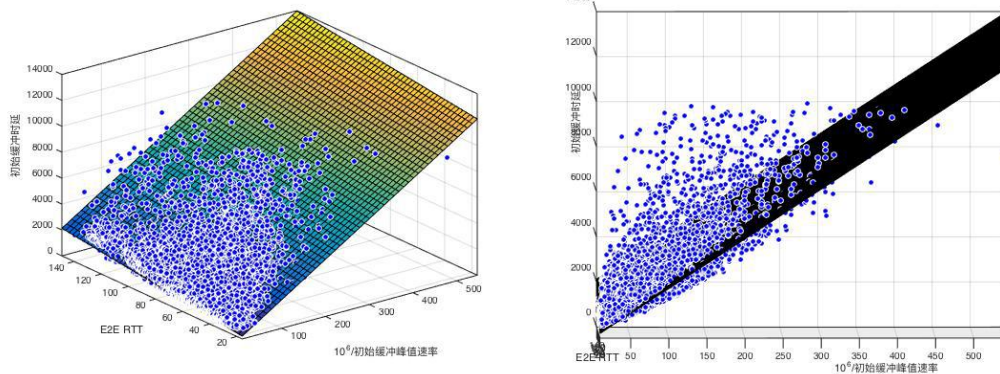


图24 缓冲阶段回归分析效果图

检验结果如下：

$g =$

sse: 6.2035e+09  
 rsquare: 0.8047  
 dfe: 21908  
 adjrsquare: 0.8047  
 rmse: 532.1273

3个系数对应的  $t$  值分别为：

$t =$

-23.8915  
 254.5477  
 86.5232

从  $t$  值来看，通过查表比较，可知回归的结果是显著的。

其中 $R^2$ 值为0.8047，还是有很强的相关性。并计算出F值为45147，p值为0。均方根误差为532.1273ms。

从检验结果看，总体的效果还是不错的，大多数点与拟合出的曲面较为靠近，但还是有一部分点与曲面距离较远。推测其可能的原因如下：因为题目中仅给出初始缓冲阶段的峰值下载速率，但峰值并不能完全反映下载速率的平均水平。由于网络不稳定等原因，可能导致下载速率的波动很大，可能会出现峰值速率很高，但是平均速率很低的情况，会对我们的预测结果造成较大的误差。

但我们的模型在大多数情况下还是可以得到较为正确的结果。因此还是有一定的价值。

### 3.4.3 其他的工作：各项视频得分与其他变量的关系

首先考虑3项得分值（O，P，Q三列）与哪些变量有关。

**3.4.3.1** 不难发现视频质量得分（O列）只有4.33和4.34这两个值，发现它仅与视频码率有关。视频码率只有3个取值，当码率为2903或2934时，视频质量得分为4.33，当码率为2966时，视频质量为4.34。

**3.4.3.2** 再考虑初始缓冲得分（P列），从数据中可以看出，它的取值范围是1-5，我们推测它可能与初始缓冲时延（E列）有关。于是我们做出了所有记录中，这两个变量的二维散点图如下：

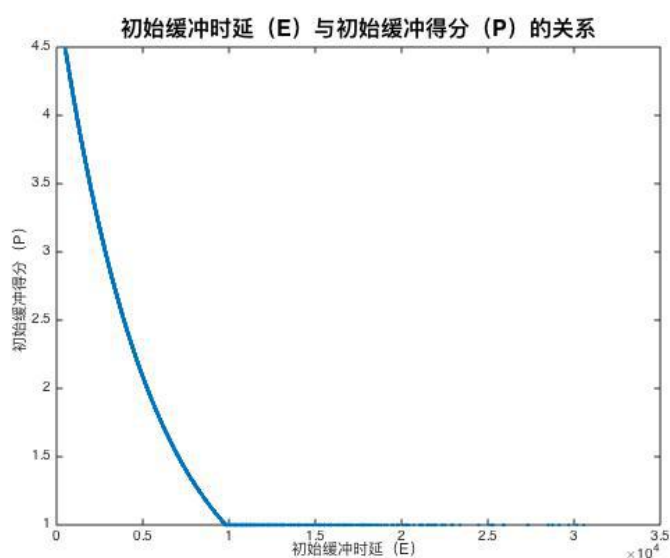


图25 初始缓冲得分和初始缓冲时延关系

从图中可以看出，散点图几乎连成了线，说明这两个变量有很强的相关性，并且很明显可以分为两段，当缓冲时延大于 $10^4$ 时，得分取常值，为1。

我们用 $x_1$ 来表示初始缓冲时延，单位为毫秒（ms），用 $x_2$ 来表示缓冲得分。使用matlab中的曲线拟合工具箱，对 $x_1 < 10^4$ 的部分做曲线拟合，发现用3次函数可以得到较不错的效果，结果如下：

f =

```
Linear model Poly3:
f(x) = p1*x^3 + p2*x^2 + p3*x + p4
Coefficients (with 95% confidence bounds):
p1 = -2.146e-12 (-2.155e-12, -2.136e-12)
p2 = 6.566e-08 (6.552e-08, 6.579e-08)
p3 = -0.0008359 (-0.0008364, -0.0008354)
p4 = 4.89 (4.889, 4.891)
```

为了方便表示，我们令 $w = \frac{x_1}{10^4}$ ，这样得到的拟合结果为

$$y = -2.146 w^3 + 6.566 w^2 - 8.359 w + 4.89$$

拟合效果如下图：

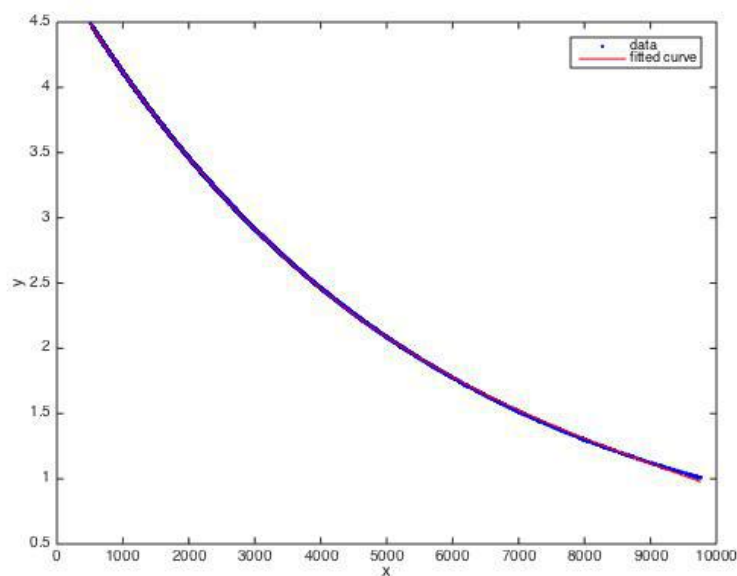


图26 得分拟合效果图

从图中看效果是非常不错的，检验结果如下：

g =

```
sse: 0.0747
rsquare: 1.0000
dfe: 4525
adjrsquare: 1.0000
rmse: 0.0041
```

其中 $R^2$ 值为1，说明相关性极强，方均根误差为0.0041，并计算F值为 $7.1 \times 10^7$ 。拟合效果确实很好。

**3.4.3.3** 再考虑卡顿得分（第Q列），推测它可能与卡顿占比（F列）有关。我们同样画出这两个变量之间的二维散点图如下：

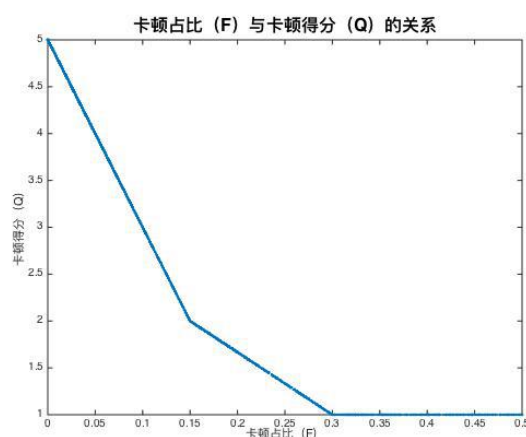


图27 卡顿占比和卡顿得分关系图

正如图中所示，是明显的分段线性关系，总共分为3段。

设  $x$  为卡顿占比， $y_2$  为卡顿得分。 $y_2$  的取值同样为1-5之间。简单计算可知，卡顿得分与卡顿占比的函数关系式如下：

$$y_2 = \begin{cases} -20x + 5, & x < 0.15 \\ -\frac{20x}{3} + 3, & 0.15 \leq x < 0.3 \\ 1, & x \geq 0.3 \end{cases}$$

**3.4.3.4** 通过查找一些资料我们知道VMOS（G列），是视频观看体验的综合指标，它由视频质量得分（O列）、初始缓冲得分（P列）、卡顿得分（Q列）确定。取值范围也是1-5分，分数越高表示视频体验越好。

但由于在所给数据中，视频质量得分几乎是个常数，因此我们主要考虑另外两个因素对VMOS的影响。

首先我们画出初始缓冲得分，卡顿得分，与VMOS的三维散点图，方便我们大致看一下趋势。图像如下：

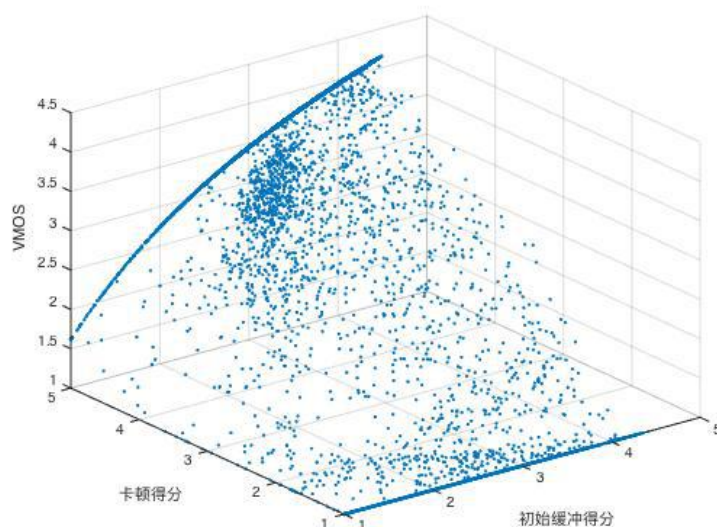


图28 VMOS和初始缓冲得分、卡顿得分的关系图

我们发现其看起来是一张较光滑的曲面。但VMOS的最小值为1，因此我们去掉VMOS取值为1的点，对大于1的部分来做拟合。在这里，我们用  $x$  来表示初始缓冲得分， $y$  表示卡顿得分， $z$  表示VMOS。我们使用Matlab中的曲面拟合工具，用  $x$  的3次函数， $y$  的2次函数做拟合，可以得到较好的效果。

得到的结果  $z = f(x, y)$  如下：

```
Linear model Poly32:
f(x,y) = p00 + p10*x + p01*y + p20*x^2 + p11*x*y + p02*y^2 + p30*x^3 + p21*x^2*y
        + p12*x*y^2
Coefficients (with 95% confidence bounds):
p00 =    -2.989    (-3.014, -2.965)
p10 =     1.417    (1.404, 1.429)
p01 =     0.8668   (0.857, 0.8766)
p20 =    -0.2777   (-0.2795, -0.2759)
p11 =     0.1604   (0.1565, 0.1643)
p02 =    -0.05116  (-0.05223, -0.05009)
p30 =     0.02254   (0.02243, 0.02266)
p21 =    -0.007472 (-0.00782, -0.007123)
p12 =    -0.01399  (-0.01432, -0.01366)
```

即：



$$z = f(x, y) = -2.989 + 1.417x + 0.8668y - 0.2777x^2 + 0.1604xy - 0.05116y^2 + 0.02254x^3 - 0.007472x^2y - 0.01399xy^2$$

拟合的效果如下图所示：

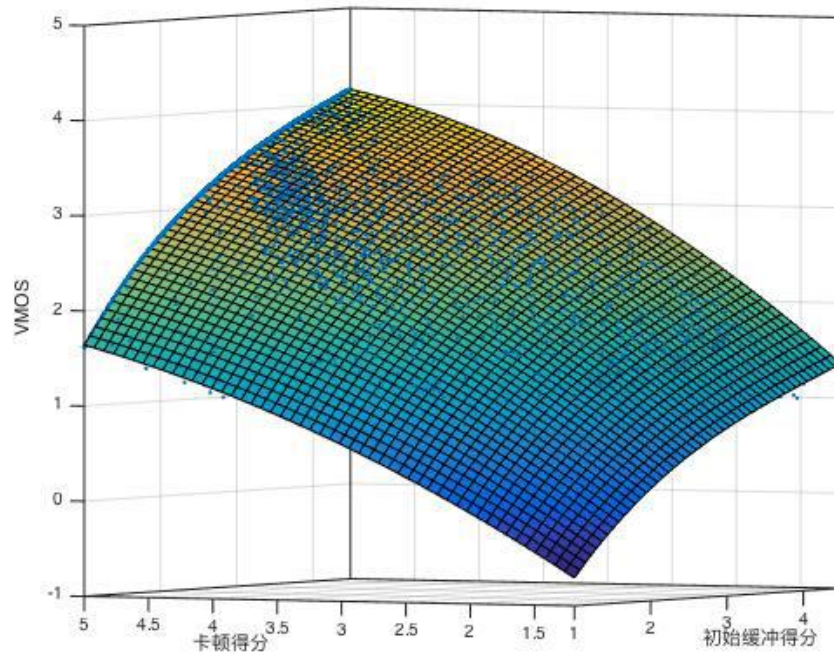


图29 VMOS拟合效果图

从图中看出效果还是不错的，检验结果如下：

$g =$

```
sse: 2.2884|
rsquare: 0.9996
dfe: 86975
adjrsquare: 0.9996
rmse: 0.0051
```

并求得对应系数的  $t$  值如下：

$t =$

```
-238.4083
221.1235
172.8149
-300.1094
81.5421
-93.7369
382.2540
-42.0078
-82.8462
```



$R^2$ 值很高，为0.9996接近于1，均方根误差很小，只有0.005左右，求得F值为 $3.0 \times 10^7$ 。并通过查表将  $t$  值加以比较，回归的效果是显著的。

由于VMOS的取值不低于1，因此最终得到的函数关系式如下：

$$Z = \max \{ f(x, y), 1 \}$$

通过以上的的工作，我们就得到了各项评分指标与哪些因素有关，以及具体的关系式。

### 3.4.4 播放过程模拟分析

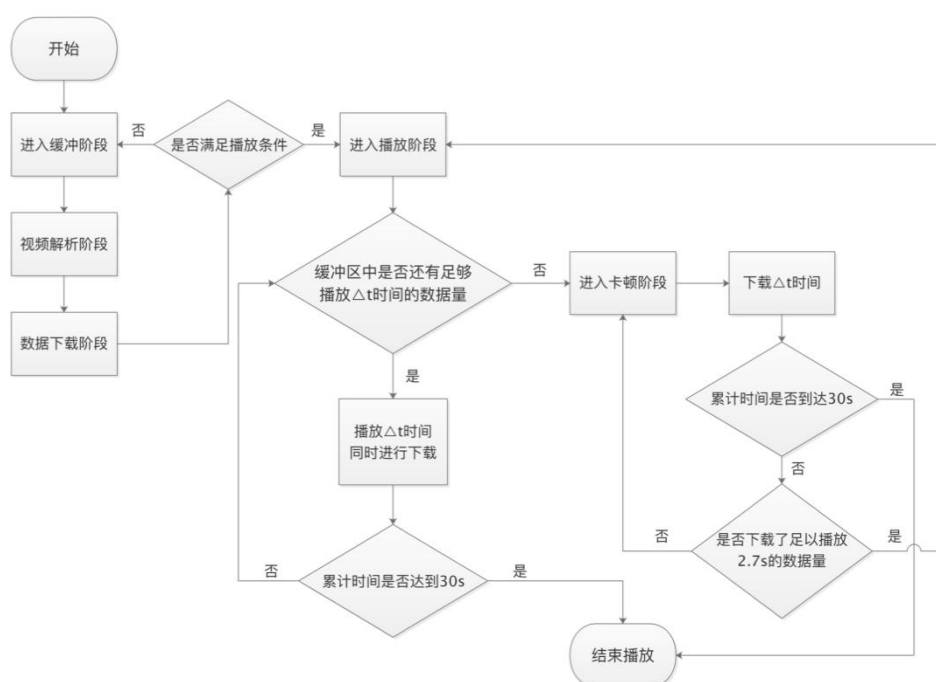


图30 视频播放流程图

该过程与文中开始叙述的“水槽模型”是一致的。前者是一个现实中形象化的描述。事实上我们更希望能寻找出下载速率的一种特殊的分布，如果分布确定，那么播放过程就确定。但是，事实上，我们无法通过已知速率得出这种分布，因为并没有实时下载速率。

为了定量描述播放阶段卡顿的情况，我们可做如下分析：

以播放开始的时刻为0时刻, 设 $x(t)$ 表示在 $t$ 时刻缓冲区的数据量, 那么 $x(0)$ 为初始缓冲阶段已下载好的数据量;  $v(t)$ 为 $t$ 时刻的下载速率,  $v_0$ 表示视频的播放速率, 即认为是视频码率, 在本题中认为是个常数, 于是有以下微分方程成立:

$$\text{在播放时: } x'(t) = v(t) - v_0$$

$$\text{在卡顿时: } x'(t) = v(t)$$

根据假设中卡顿门限与重播放门限的取值, 可知, 在播放过程中, 一旦 $x(t) < 0$ , 则进入卡顿状态; 在卡顿过程中, 若 $x(t) > 2.7a$ , (其中 $a$ 为视频码率, 取2934kbps), 则重新回到播放状态。

于是, 问题转化为了一个微分方程模型。当给定了下载速率随时间的函数, 我们便可以模拟整个播放过程, 并确定其卡顿的情况。

求解微分方程我们可以利用数值方法, 对时间做足够细的分划, 比如取分隔区间为10ms, 在每个小区间段认 $v(t)$ 为是个常数, 这样借助计算机很容易得到缓冲区数据量随时间变化的图像, 及一些卡顿的参数, 结果还是比较精确的。

例如: 如果我们取速率 $v = 1500 + 500\sin(t/2)$ , 初始数据量 $x(0) = 12000\text{kb}$ , 运行程序后我们便可以得到如下结果:

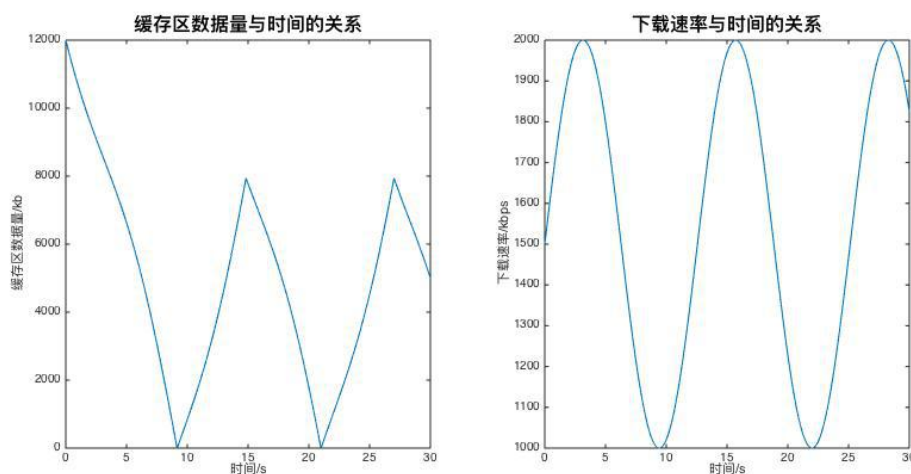


图 31 模拟过程中数据量和下载速率随时间的关系

并得到如下参数:

卡顿次数: 2

播放时长: 18.32 s

卡顿时长: 11.68 s

模拟的结果还是比较可靠的。但是实际的下载速率可能变化很剧烈，并不一定会这样规则，所以在实际情况中，卡顿状况可能也会变得十分复杂。

### case 1 均匀分布

我们假设 $v(t)$ 服从均匀分布，那么给定取值范围可以对播放过程进行随机模拟，我们取0.3s为一个时间间隔，初始缓存区数据量根据假设取4s的数据量，范围分别取1500–2500和2500–3500来进行随机模拟，可以反映出有卡顿和没有卡顿两种情况的效果。得到的效果图如下：

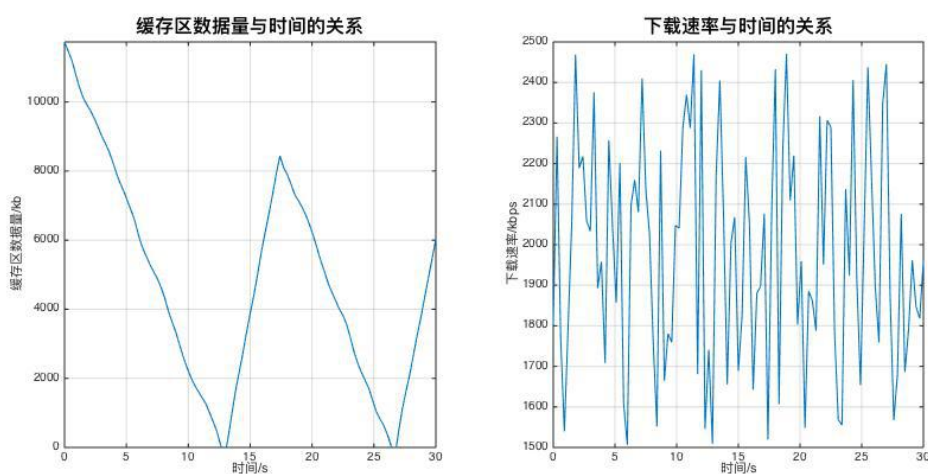


图32 下载速率范围在1500–2500模拟图像

得到如下结果：

卡顿次数：2

播放时长：22.50s

卡顿时长：7.50s

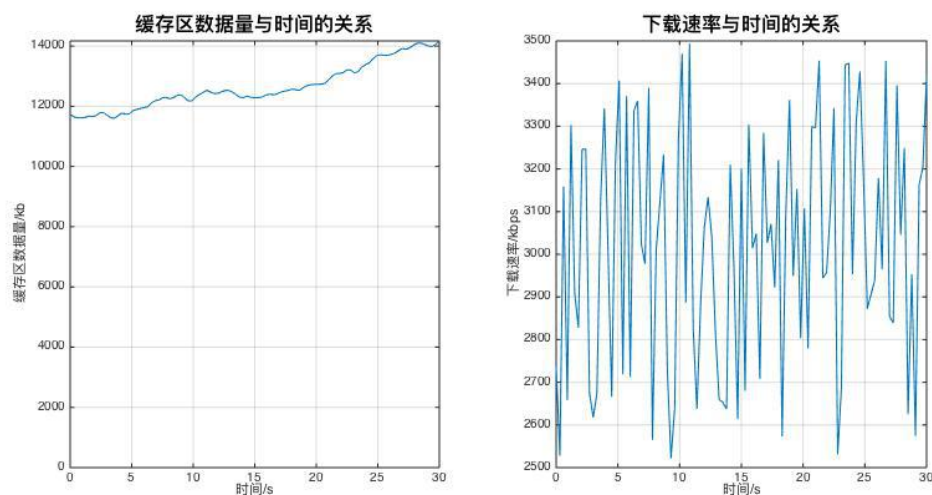


图33 下载速率范围在2500-3500模拟图像

得到如下结果：

卡顿次数：0

播放时长：30s

卡顿时长：0s

即此时无卡顿。

## case 2 正态分布

我们再假设  $v(t)$  服从正态分布，那么我们可以给定均值和标准差来进行随机模拟，同样每0.3s为一个时间间隔，初始缓存区数据量根据假设取4s的数据量，取我们分别取了均值为1500和3000，对应标准差为均值的一半（即分别为750和1500）来进行随机模拟（因为位于  $\mu-2\sigma$  到  $\mu+2\sigma$  的概率已达到95%以上，随机速率很小可能会出现负值，即使出现我们也可以人为剔除），这样可以反映出有卡顿和没有卡顿两种情况的效果。得到的效果图如下：

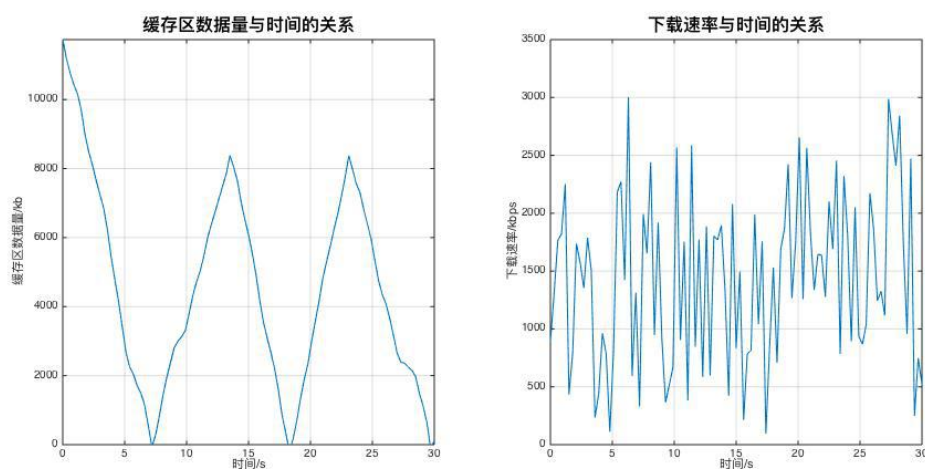


图34 平均下载速率为1500模拟图像

有如下结果：

卡顿次数：3

播放时长：18.90s

卡顿时长：11.10s

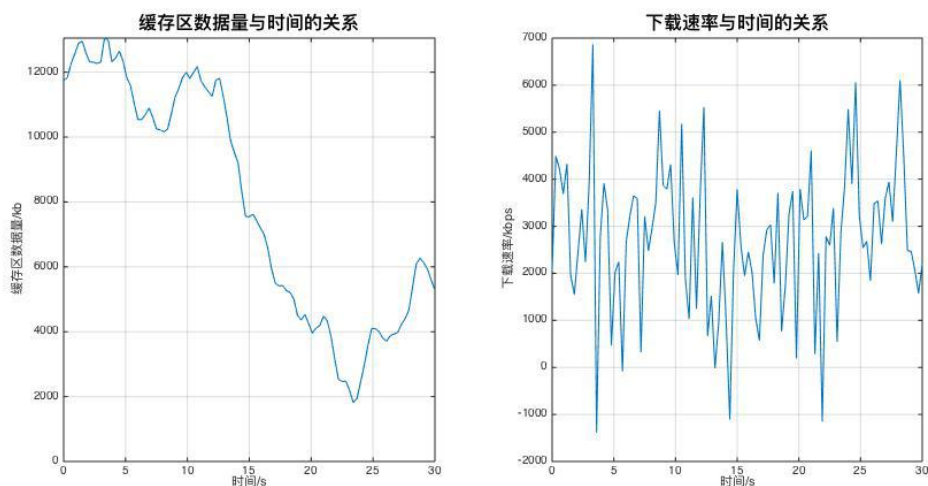


图35 下载速率为3000模拟图像

有如下结果：

卡顿次数：0

播放时长：30.00s

卡顿时长：0.00s

此时无卡顿。

我们可以进行大量的多次模拟，取下载速率均值从500起一直到5000，中间间隔1。

记录每一次对应的播放时长，并作出散点图如下：

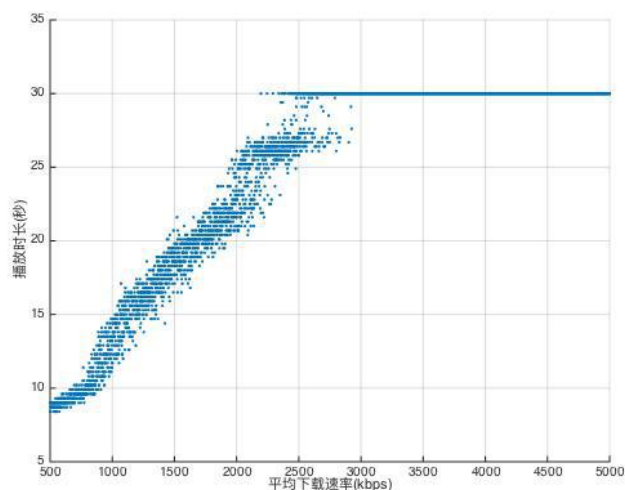


图36 多次模拟数据散点图

大体趋势和真实数据较为相像。我们和我们之前分段连续的拟合结果相比较，残差为4905.8ms，效果还算可以，但比我们对实际数据做线性回归的残差要大。

为了减少误差，我们对数据进行调整，例如，减少初始的数据量为0.5s的数据量，其他不变，作出图像如下：

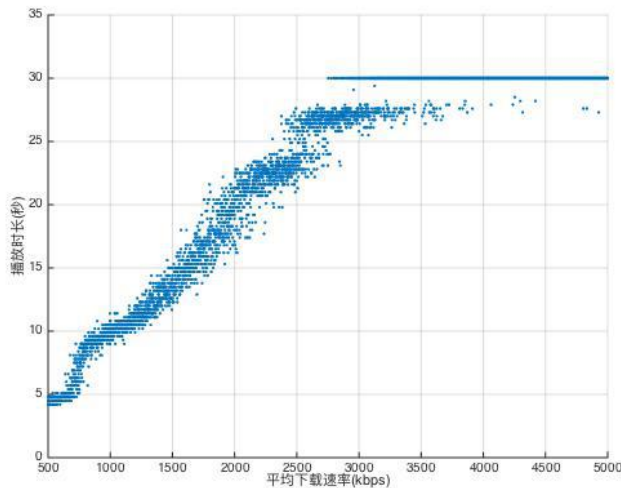


图37 调整初始量 多次模拟散点图

此时残差为2862.8ms，与对实际数据做回归分析的误差较为接近。

推测原因：在播放阶段刚开始时，下载速率可能很慢，并不服从正态分布，此时初始数据量一旦耗尽，便会产生一定的卡顿。而后恢复正常，下载速率会在均值附近做小范围的上下浮动，根据我们的模拟效果，推测可能为正态分布。

## 四、创新点与特色

我们的模型有如下优点：

1. 文章从多个角度思考，用多种方法进行研究、模拟，由浅入深的逐步解决问题。
2. 决策树很好得解决了边界数据带来的数据混乱情况，为之后的研究带来了便利。
3. 神经网络可以较好的说明各变量之间的相关性，排除变量无关因而做无用功的可能，并能对输入的结果给出预测。
4. 使用回归分析直观、清晰的给出了和各变量之间的函数关系式，并且整体效果不错。

我们的模型有如下特色：

1. 全文并没有使用高级的拟合方式，提供的方法易懂却又很好的拟合了选出的数据。并且给出了各个变量之间的散点图，方便直观地看出变量之间的简单联系。
2. 简化假设较好地概括了模型的特点，又不偏离问题的本质，为问题的进展做了良好的保证。

## 五、进一步研究可能需要的数据

1. 从文献综述中看，通常的视频质量评分体系都与视频流传输的丢包率有关。
2. 从文章的3.4.4的分析中，我们希望能得到视频在播放阶段下载速率随时间的关系。
3. 根据缓冲得分看，得分都不到5分，希望有更多的数据能遍布在1~5之间。

## 参考文献：

- [1] 盛骤 概率论与数理统计 高等教育出版社 2008年6月
- [2] 华为 基于移动视频的移动承载网络要求白皮 2016年9月  
<http://www.docin.com/p-1727526773.html>
- [3] 周志华 机器学习 清华大学出版社
- [4] 于新 无线网络中端到端视频流业务的用户体验质量预测及优化技术 浙江大学
- [5] Ripley, B. D Pattern Recognition and Neural Network 1996年
- [6] 陆文秀 神经网络逼近中的几个问题 2013年5月