PPT

1. Abstract: 根据后面的综合一下

2. Intuition:

2.1 基本思路：受到RL启发，对于传入参数（state），作为input，我们可以预测它在某个bitrate和buffer下的该步得分（reward），同时可以预测下一个state；然后，我们选取 该步得分+预计的后续得分 作为决策的依据，直接选取分数最高的决策作为当前决策。

2.2 与RL区别：RL是end-to-end的网络，通过深度学习得到各决策的概率并进行决策。但是，由于深度学习训练周期长，且容易出现过拟合的情况，我们采用人为设定的深度-贪心-决策结构。

3. Algorithm:

3.1 整体算法

假定bitrate有n种选择，buffer有m种选择。

对于input state，我们训练出n\*m个score模型，分别得到使用这些决策的预估得分，作为该步得分current\_score。同时，我们也训练出n\*m个Next\_state模型，分别得到使用这些决策下一步的预估状态。然后，使用递归可以得到后k步决策的最大值Max\_future\_score(k)。我们把current\_score + max\_future\_score(k) \* alpha 的总分（其意义为当前步及后k步内总分）作为当前使用该决策的总分，然后贪心地取最大的总分决策即可得到最优决策。

3.2 对整体算法的解释

1. 为什么使用离散值作为决策的选择：

主要原因是因为初赛时规则就是离散值……此外，如果使用连续值，难以得到最优的buffer，因为buffer对总分的影响并不是线性的，无法使用二分等方法得到。基于我们算法在初赛的优秀表现，我们决定继续使用原算法。

2.（不知道怎么写题目）：

我们可以看到，当k趋向无穷的时候，我们的算法理论上可以得到”global maximum”。但也会出现一些问题：耗时过长；因为预测next\_state会产生误差，所以步数越多对next\_state的误差越大；若k很小，我们就只能看到最近的几步，可能会造成决策的频繁变化。

3. state的选取（基于abr.run的input）：

预赛：(S\_time\_interval[-1], S\_send\_data\_size[-1], S\_chunk\_len[-1], S\_rebuf[-1], S\_buffer\_size[-1], S\_play\_time\_len[-1], S\_end\_delay[-1], S\_decision\_flag[-1], S\_buffer\_flag[-1], S\_cdn\_flag[-1], end\_of\_video, bitrate\_change(与上一个bitrate的差值), buffer\_change(与上一个buffer的差值)

决赛：(S\_time\_interval[-1], S\_send\_data\_size[-1], rebuf\_time, S\_buffer\_size[-1], S\_end\_delay[-1], buffer\_flag, cdn\_flag, bitrate\_change, buffer\_change)

为什么选取[-1]：因为通过最后一帧的变化，我们可以得到准确的对当前video和network状态的预测。

3.3 训练方式

我们使用的是离线训练的方法。具体的实现为：我们预设好bitrate和buffer的取值范围，然后在每一次决策时随机选择一对组合，并记录下所有的state和这一次决策的总得分（reward\_sum – last reward\_sum）。我们通过随机决策的方法生成了大量数据（>10^8条，以csv格式存储），并通过这些数据进行以下的训练。

3.4 Current\_score模型的具体实现

我们使用LinearRegression 进行score的拟合。具体的实现方法为：对离线训练得到的数据进行预处理后，把state和score提取出来，用sklearn进行Regression。

3.5 Next\_state 模型的具体实现

同样是LinearRegression，预处理把state作为input，下一帧state作为output，进行Regression。

3.6 为什么使用LinearRegression（而不用神经网络/SVR）

1. 神经网络/SVR不容易收敛，且容易过拟合。

2. LinearRegression有较好的拟合程度，且不会过拟合。（这里放一个score的图表）

3. 对于一些不可预测的值，如send\_data\_size等，LinearRegression会用平均值代替，相当于减弱该因素的影响，考虑其他因素，因此对于未知的环境表现得更好（不需要重新训练）。

4. Parameters

具体的参数为：

bitrate: [0, 1]

buffer: [0.2, 0.3, 0.5, 0.8, 1.5, 3.0] (暂定，明天再改，会删一些，不然超时)

k：1（层数过多会超时，若服务器配置更高的情况下，提高k可以提高performance）

5. Proposals

阈值（剪枝优化）

并行化处理