

Good afternoon, ladies and gentlemen, we are team triple-W from CUHK(SZ). These are my teammates, Xie Guochao and Mo ZiHeng. My name is Hao Senyue. Now, please let me introduce our algorithm, which is called Deep-Greedy-Decision. My today's speech has three parts.

Firstly, the solution framework is inspired by the basic reinforcement learning. For the last state as input, through training, we may predict the reward under some bitrate and target buffer for present state, and also the condition for following states. Then, we choose the maximum value of the present reward plus the following reward among all the choices for bitrates and target buffer.

To get those rewards before online test, we need to first train off-line. By generating over  $10^8$  pieces of data, giving all the parameters for  $i$  state, and choose all choices for bitrates and target buffer, we are able to calculate the corresponding QoE and  $i + 1$  state parameters. Then we use linear regression to fit the input and the output.

When our algorithm are tested, we assume for bitrates, we have  $n$  choices and  $m$  choices for target buffer. As a result, for the input, we will have  $n * m$  models for current state. Meanwhile, we will also have  $n * m$  models related to the next state under corresponding choice for bitrate and target buffer. Then, we implement recursion to achieve the max score after  $k$  steps decision. We take *current\_score* plus sigma  $k$  from 1 to infinite  $max\_future\_score(k) * \alpha^k$  as the total QoE.

There are three specific parts of our algorithm.

The reasons we chose discrete values for target buffer during the final are because, first, for our training part, it needs definite bitrates and target buffer to calculate QoE. Second, our decision time is limited. As the number of target buffer increases, our algorithm needs more time to return the decision value. As a result, instead of uniformly choosing many values between 0~4 seconds, we decided to choose only six values. And by counting the times of different target buffer occur, it helped us to determine which value we wanted.

Second, for our algorithm, from a theoretical view, when  $k$ , the depth of recursion, goes to infinite, we may get the global optimal solution for this problem. But the costing time is not acceptable. So, we should decrease the value of  $k$ . Although it will save time, the algorithm now is only able to optimize for limited future states, and increase the switching frequency.

The reasons we chose linear regression to fit our model are because, first we tried our best to implement Neural Network or Support Vector Regression. However, neither of them works well. Meanwhile, by the property of linear regression, not like the other two, it will avoid overfitting the data. Most important, the regression model works well. The rate for fitting are all over about 0.8.

Second part is about performance analysis.

We had tried different numbers of choices for bitrate and target buffer to test. Here are two diagram under one day dataset tested offline. From the left one, we conclude that 10 choices will get higher QoE during our experiments. And from the right one, we can see the running time of our algorithm goes exponentially as the number of choices increases. Therefore, as the time limited, we chose 10 choices as final decision.

The final part is about the future work.

For our algorithm, several ways can be improved further. For example, branch and bound. Set a upper bound for each model to calculate the maximum QoE under some choice, if it is already smaller than known max value, then dismiss this model. Also, use parallel computation techniques in practical system to improve the efficiency of the algorithm. And with more information for the coming video, the model-based solution can perform better. Some API may be established in practice, as well.

That's all about our algorithm. Thanks Altrans give us a chance to do this speech. And especially thanks Prof. Cui and Dr. Yang. Now is for Q&A part.