MEDIA STREAMING IBM AND CLOUD VIDEO STREAMING

BATCH MEMBERS

B.K.VIBHU VISHWA DEEP

K.VINOTH KUMAR

T.VIGNESHWARAN

S.SRIHARI SUNDHAR

INTRODUCTION:

One of the common questions we get asked is: “ Why do we need machine learning to improve streaming quality?” This is a really important question, especially given the recent hype around machine learning and AI which can lead to instances where we have a “ solution in search of a problem.” In this blog post, we describe some of the technical challenges we face for video streaming at Netflix and how statistical models and machine learning techniques can help overcome these challenges.

Netflix streams to over 117M members worldwide. Well over half of those members live outside the United States, where there is a great opportunity to grow and bring Netflix to more consumers. Providing a quality streaming experience for this global audience is an immense technical challenge. A large portion of this is engineering effort required to install and maintain servers throughout the world, as well as algorithms for streaming content from those servers to our subscribers' devices. As we expand rapidly to audiences with diverse viewing behavior, operating on networks and devices with widely varying capabilities, a " one size fits all" solution for streaming video becomes increasingly suboptimal. For example:

- Viewing/browsing behavior on mobile devices is different than on Smart TVs
- Cellular networks may be more volatile and unstable than fixed broadband networks
- Networks in some markets may experience higher degrees of congestion
- Different device groups have different capabilities and fidelities of internet connection due to hardware differences

We need to adapt our methods for these different, often fluctuating conditions to provide a high-quality experience for existing members as well as to expand in new markets. At Netflix, we observe network and device conditions as well as aspects of the user experience (e.g., video quality) we were able to deliver for every session, allowing us to leverage statistical modeling and machine learning in this space. A previous post described how data science is leveraged for distributing content on our servers worldwide. In this post we describe some technical challenges we face on the device side.

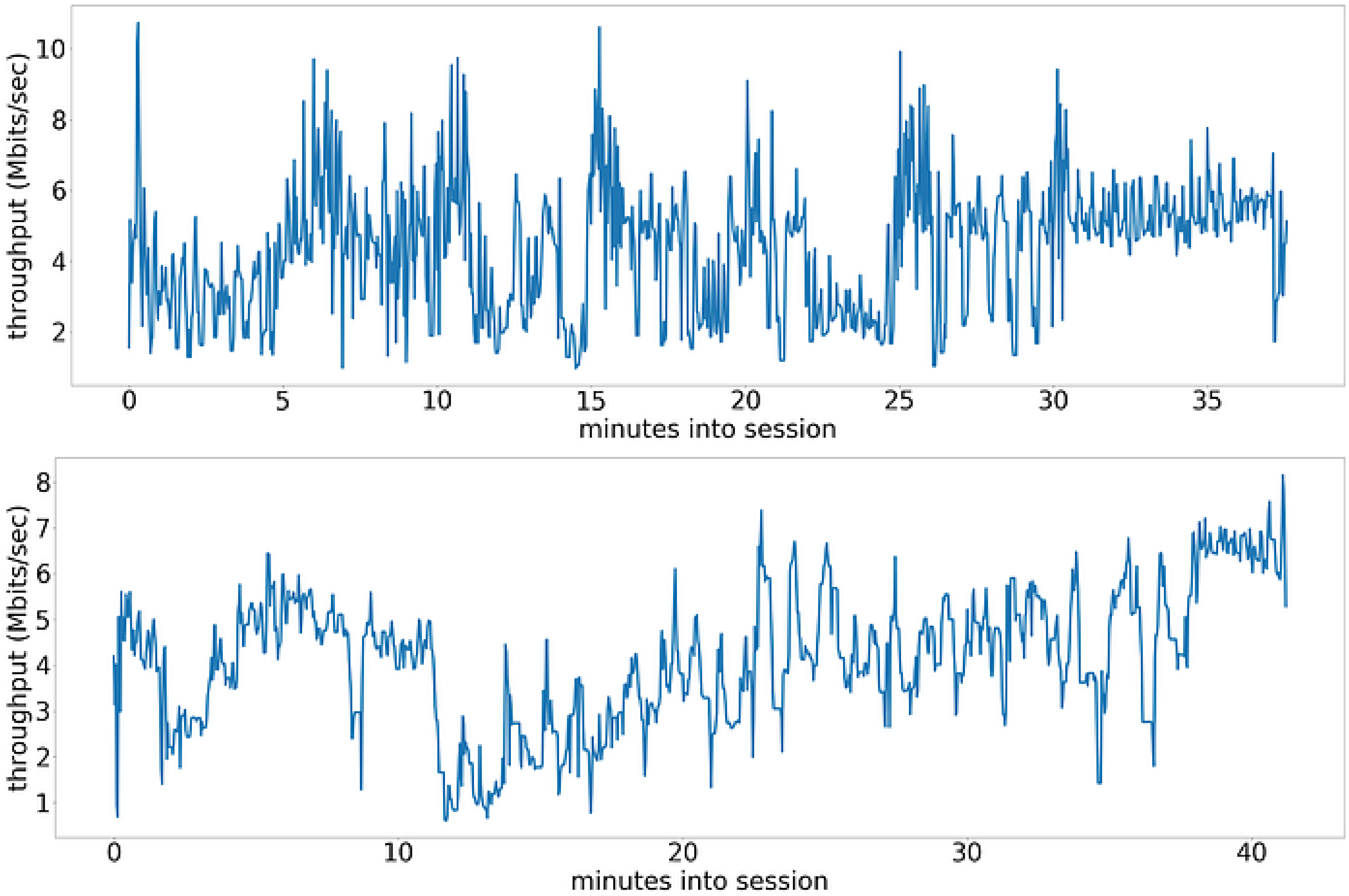
Network quality characterization and prediction

Network quality is difficult to characterize and predict. While the average bandwidth and round trip time supported by a network are well-known indicators of network quality, other characteristics such as stability and predictability make a big difference when it comes to video streaming. A richer characterization of network quality would prove useful for analyzing networks (for targeting/analyzing product improvements), determining initial video quality and/or adapting video quality throughout playback (more on that below).

Below are a few examples of network throughput measured during real viewing sessions. You can see they are quite noisy and fluctuate within a wide range. Can we predict what throughput will look like in the next 15 minutes given the last 15 minutes of data? How can we incorporate longer-term historical information about the network and device? What kind of data can we provide from the server that would allow the device to adapt optimally? Even if we cannot predict exactly when a network drop will happen (this could be due to all kinds of things, e.g. a microwave turning on or going through a tunnel while

streaming from a vehicle), can we at least characterize the *distribution* of throughput that we expect to see given historical data?

Since we are observing these traces at scale, there is opportunity to bring to bear more complex models that combine temporal pattern recognition with various contextual indicators to make more accurate predictions of network quality.



Examples of network throughput traces measured from real viewing sessions.

One useful application of network prediction is to adapt video quality during playback, which we describe in the following section.

Video quality adaptation during playback

Movies and shows are often encoded at different video qualities to support different network and device capabilities. Adaptive streaming algorithms are responsible for adapting which video quality is streamed throughout playback based on the current network and device conditions (see here for an example of our colleagues' research in this area). The figure below illustrates the setup for video quality adaptation. Can we leverage data to determine the video quality that will optimize the quality of experience. The quality of experience can be measured in several ways, including the initial amount of time spent waiting for video to play, the overall video quality experienced by the user, the number of times playback paused to load more video into the buffer ("rebuffer"), and the amount of perceptible fluctuation in quality during playback.

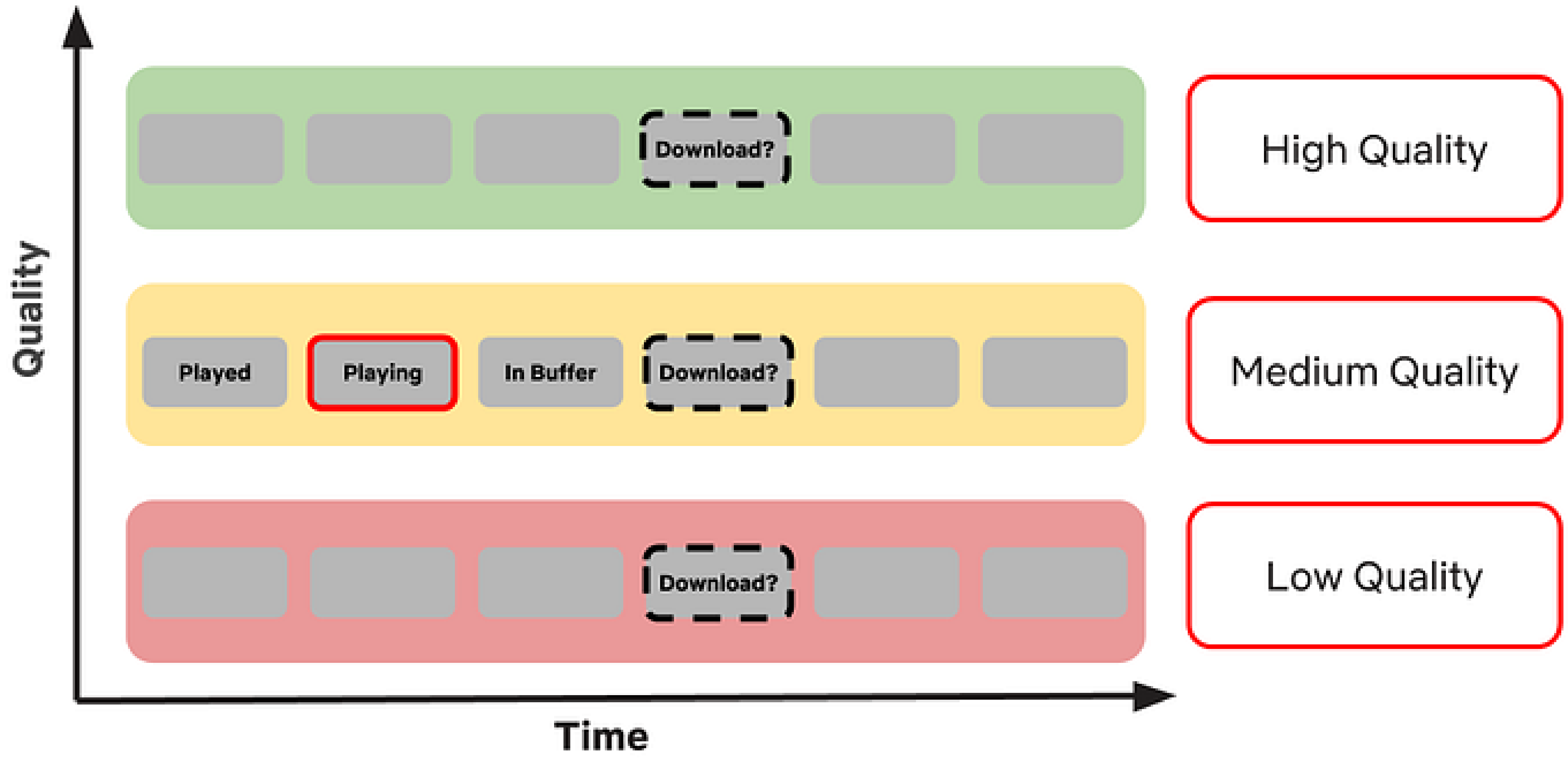


Illustration of the video quality adaptation problem. The video is encoded at different qualities (in this case 3 qualities: high in green, medium in yellow, low in red). Each quality version of the video is divided up into chunks of a fixed duration (grey boxes). A decision is made about which quality to choose for each chunk that is downloaded.

These metrics can trade off with one another: we can choose to be aggressive and stream very high-quality video but increase the risk of a rebuffer. Or we can choose to download more video up front and reduce the rebuffer risk at the cost of increased wait time. The feedback signal of a given decision is delayed and sparse. For example, an aggressive switch to higher quality may not have immediate repercussions, but could gradually deplete the buffer and eventually lead to a rebuffer event on some occasions. This **"credit assignment" problem** is a well-known challenge when learning optimal control algorithms, and machine learning techniques (e.g., recent advances in reinforcement learning) have great potential to tackle these issues.

Predictive caching

Another area in which statistical models can improve the streaming experience is by predicting what a user will play in order to cache (part of) it on the device before the user hits play, enabling the video to start faster and/or at a higher quality. For example, we can exploit the fact that a user who has been watching a particular series is very likely to play the next unwatched episode. By combining various aspects of their viewing history together with recent user interactions and other contextual variables, one can formulate this as a supervised learning problem where we want to maximize the model's likelihood of caching what the user actually ended up playing, while respecting constraints around resource usage coming from the cache size and available bandwidth. We have seen substantial reductions in the time spent waiting for video to start when employing predictive caching models.

Device anomaly detection

Netflix operates on over a thousand different types of devices, ranging from laptops to tablets to Smart TVs to mobile phones to streaming sticks. New devices are constantly entering into this ecosystem, and existing devices often undergo updates to their firmware or interact with changes on our Netflix application. These often go without a hitch but at this scale it is not uncommon to cause a problem for the user experience — e.g., the app will not start up properly, or playback will be inhibited or degraded in some way. In addition, there are gradual trends in device quality that can accumulate over time. For example, a chain of successive UI changes may slowly degrade performance on a particular device such that it was not immediately noticeable after any individual change.

Detecting these changes is a challenging and manually intensive process. Alerting frameworks are a useful tool for surfacing potential issues but oftentimes it is tricky to determine the right criteria for labeling something as an actual problem. A “liberal” trigger will end up with too many false positives, resulting in a large amount of unnecessary manual investigation by our device reliability team, whereas a very strict trigger may miss out on the real problems. Fortunately, we have history on alerts that were triggered as well as the ultimate determination (made by a human) of whether or not the issue was in fact real and actionable. We can then use this to train a model that can predict the likelihood that a given set of measured conditions constitutes a real problem.

Even when we’re confident we’re observing a problematic issue, it is often challenging to determine the root cause. Was it due to a fluctuation in network quality on a particular ISP or in a particular region? An internal A/B experiment or change that was rolled out. A firmware update issued by the device manufacturer.Is the change localized to a particular device group or specific models within a group? Statistical modeling can also help us determine root cause by controlling for various covariates.

By employing predictive modeling to prioritize device reliability issues, we’ve already seen large reductions in overall alert volume while maintaining an acceptably low false negative rate, which we expect to drive substantial efficiency gains for Netflix’s device reliability team.

CONCLUSION:

The aforementioned problems are a sampling of the technical challenges where we believe statistical modeling and machine learning methods can improve the state of the art:

- there is sufficient data (over 117M members worldwide)
- the data is high-dimensional and it is difficult to hand-craft the minimal set of informative variables for a particular problem
- there is rich structure inherent in the data due to complex underlying phenomena (e.g., collective network usage, human preferences, device hardware capabilities)

Solving these problems is central to Netflix’s strategy as we stream video under increasingly diverse network and device conditions.