Context-aware encoding and dynamic encoding ladders

Nasim Jamshidi Avanaki, Navid Shahbazi, Mohammad Al-Diabat

Abstract— Video streaming has gained considerable popularity, with video-on-demand (VOD) distributors such as Netflix and YouTube responsible for half of the Internet traffic in the US in 2014 [mm]. User satisfaction is the ultimate goal of any service provider to keep their customer in their service. Therefore, Over-the-top (OTT) media service providers are trying to offer the optimal video quality based on the end device and network constraint (e.g., bandwidth limit). Traditional video streaming recommendations is to use fixed ladder resolution-bitrate pairs. For example, the ladder recommends to use a resolution of 720p for bandwidth (approx.) lower than 2Mbps. However, video contents are different in terms of video complexity, thus, using a fixed ladder is not an optimal solution as it may result in wasting bandwidth (in case of very low complex video) or facing any video artifacts such as Blockiness (in case of highly complex video).

This report investigates the possibility of using machine learning techniques to predict the best resolution for a specific bitrate level based on the content complexity.

In the present document, we describe briefly some objective measurement for video quality, accompanied by the findings of Netflix's new Video Multi-Method Analysis Fusion (VMAF) metric. Since the current quality metrics lie on the signal-based category and mostly use the Full Reference method, we decided to employ the parametric-based method to define what features can be used to predict the VMAF, with no need to have access to the reference videos and encoded video for predicting the video quality.

Furthermore, four regression models, such as Linear Regression, Support Vector Regression (radial basis function kernel), Random Forest, and XGBoost, were used. As a result, the Random Forest with the RMSE as 7.3 and the Linear regression with the RMSE as 11.97 had the best and the worst prediction (based on their similarity to the actual model), respectively.

I. Introduction

In late 2010, video engineers developed encoding recipes that worked the best across videos at that time. It was called the one-size-fits-all fixed bitrate ladder.

For high complex videos, such as videos with high camera noise or film grain noise, even the highest bitrate would still face video artifacts like Blockiness. On the other side, for some low complex videos, such as animations, using high bitrate for a certain resolution would result in wasting bandwidth.

Online streaming offers distributor companies such as Netflix, the opportunity to customize their streams to the bandwidth and viewing devices available to viewers. The contents in Netflix's video collection have a very high diversity in signal characteristics. In figure 1, the Rate-Distortion curve of 100 movies has been shown [bb], the X-axis is the bitrate, and Y-axis is PSNR, which is the most commonly used video quality metric. This visualization could also be shown with other objective quality metrics such as VMAF (Video Multi-Method Assessment Fusion), which is a perceptual video quality metric developed by Netflix in collaboration with the University of Southern California.

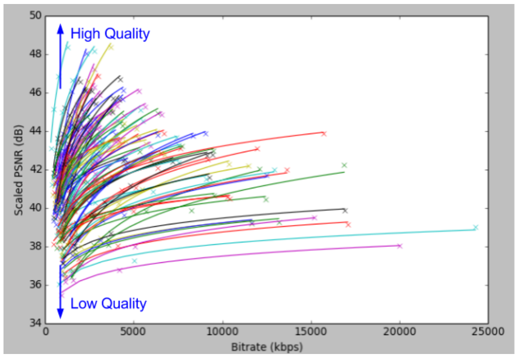


Fig. 1. RD-curves of 100 movies. The X-axis is Bitrate by kbps, and Y-axis is PSNR which is a common quality metric [bb]

The first per-title encoding strategy established around 2012 and Netflix is heavily technology driven that was then used efficiently from 2015. During that time, researchers were able to significantly reduce the computing power required to analyze the quality of the video content. Per-title encoding was proven to be an easy solution to achieve massive savings in bandwidth and reduce the needed bitrate while preserving nearly optimum performance for DASH streaming.

II. Elaboration of analysis per-title

To model the best per-title bitrate ladder, according to several realistic constraints the maximum number of quality-rates and the bitrate quality pairs for each qualitative level were chosen; thus the selection of the resolution restricted to a limited set – for example, 1920x1080, 1280x720, 720x480, 512x384, 384x288, and 320x240. The bitrate range, therefore, restricted to a limited set with an increase of approximately 5 percent in the neighboring bitrate [xx]. The bitrate-resolution pairs should be selected carefully in which we can find the switching points. Optimally, two similar bitrates should have a perceptual gap just below one JND.[xx]

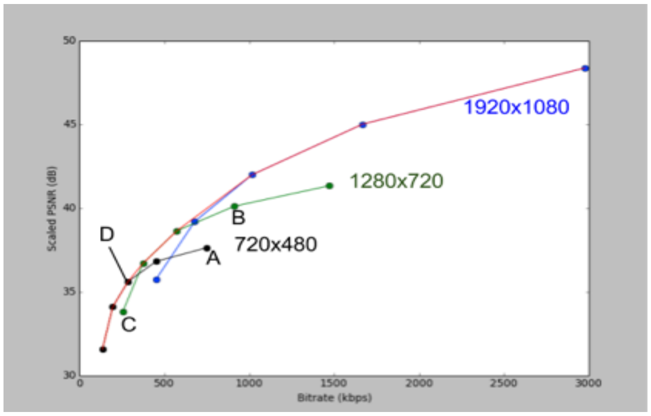


Fig. 2. Encoding at three resolutions and various bitrates. The blue marker depicts the encoding point, and the red curve indicates the PSNR-bitrate convex hull.

The video quality increases monotonously as bitrate increases, but when the bitrate reaches a certain threshold, the curve begins flattening out (A and B). On the other hand, a high-resolution encode might create a quality lower than that generated by encoding at the same bitrate but at a lower resolution (see C and D) [gg]. The explanation is that encoding additional pixels with less precision will create the worst image, rather than encoding fewer pixels with higher accuracy and sampling and interpolation. However, at very low bitrates, the overhead encoding of each fixed-size block begins to control the bitrate consumption, leaving very few bits to the real signals encoded. We can see that there is a bitrate area in each resolution where other resolutions outperform. If all these areas gather from all standard resolutions, they form a convex hull together. We prefer to operate precisely on the convex hull, but we do want to select bitrate-resolution pairs as close to the convex hull as possible because of practical constraints (for example, we can choose only a limited number of standard resolutions). It is impossible to create full bitrate equality graphs spanning the entire quality area for each title in the video-on-demand (VOD) catalog. Figure 2 is an example of where encoded videos with different levels of bitrates different levels of resolutions together with their PSNR labels are shown [xx]. The switching points and flattening areas can be seen in Figure 2.

Practically, a video sequence consists of scenes with different levels of complexity. That is why the segments of the video having a high complexity end of the title employed in order to generate the optimal bitrate ladder. This ensures the best quality in the highly complex scenes but can over-allocate bits for simple video segments.

III. Encoding Process

A. *Per-title Encoding*

The encoding procedure is initiated once the title complexity has been analyzed and a per-title-bitrate ladder built. A video encoding is generated in the cloud-based video encoding pipeline for each resolution-bitrate pair that the video source is broken up into fixed chunks, and each chunk is separately encoded in the parallel encoding pipeline. Upon completion of all the fragments, a video assembler stitches together the bitstream to create the entire bitstream of the title. The objective bitrate is achieved with the same average target bit rate for all the chunks, using two-pass bitrate control on each of the chunks.[xx]

B. *Per-chunk bitrate setting and encoding*

Another way of encoding can be Per-chunk setting and encoding. In this encoding method, Netflix developers divided an encode video into multiple chunks as for each video chunk; they selected the bitrate such that it aligns with the complexity of the video for that particular chunk. The complexity analysis results in optimal resolution-bitrate pairs for that title, as described above. This encoding method is out of the scope of this project [xx].

IV. Quality Assessment

Usually, a typical and most reliable way to assess the quality is using subjective tests, which we ask participants to rate the quality of a video sequence that is encoded with different bitrate-resolution pairs in a standard manner [yy]. However, subjective tests are expensive and not always available. Therefore, objective metrics have become an alternative. Since the objective metrics used in the prediction of quality, errors are an inseparable part of the results. It has to be noted that a functional consequence of this method is that it can be applied simply to every encoded video.

The objective signal-based quality assessment, depending on the availability of an initial reference image that is presumed to have perfect quality (signal based models), may be classified into full-reference (FR), reduced-reference (RR), and no-reference (NR) methods. FR measures require full access to the reference image, while NR methods assume entirely no access to the reference [kk]. RR methods provide a compromise in-between, where only partial information in the form of RR features extracted from the reference image is available in assessing the quality of the distorted image [kk]. The challenge of this approach is that the video has to be encoded first.

In the second approach, which named parametric based models [cc], specific parameters need to be determined by network and encoding parameters to yield the best results. Unlike the signal-based model, there is no need to encode the videos beforehand. However, as less information is available, they are typically less accurate compared to signal-based models.

Moreover, the combination of these two approaches can use as Hybrid models to enhance their performance and overcome the weaknesses.

Currently, the most commonly used objective metrics as full reference metrics are PSNR (Peak Signal-To-Noise Ratio) and VMAF (Video Multi-Method Assessment Fusion). VMAF is a perceptual video quality assessment algorithm developed by Netflix which outperforms other metrics. While, VMAF is shown to be accurate for video quality prediction, VMAF is still expensive procedure as the videos needed to be encoded in several bitrate-resolution pairs to get RD curves. Therefore, in this project, we decided to replace them with parametric based models.

V. Convex Hull

The convex hull is a ubiquitous structure in computational geometry. Convexity A set *S* is convex if x ∈ S and y ∈ S implies that the segment xy ⊆ S. meaning, Given any two points in the polygon, the line segment between them stays inside the polygon, in other words, the convex hull or convex envelope of a set X of points in the Euclidean plane or a Euclidean space (or, more generally, in an affine space over the reals) is the smallest convex set that contains X. For instance, when X is a bounded subset of the plane, the convex hull may be visualized as the shape enclosed by a rubber band stretched around X (Figure 3).

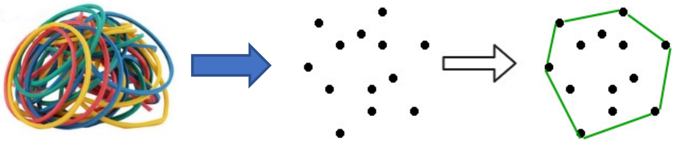


Fig. 3. Convex hull for a set of points in a two-dimensional space.

By having a closer look at the rate-distortion plots of each of the video clips, it can observe that in each specific resolution, by increasing the bitrate, the quality enhances to a certain point, after passing that point the quality becomes flattened, thus jumping to a higher resolution is convenient.

By knowing the definition of the convex hull, to obtain the resolution switching points are not out of access.

Figure 4, the convex hull, the red line, was drawn for a rate-distortion diagram:

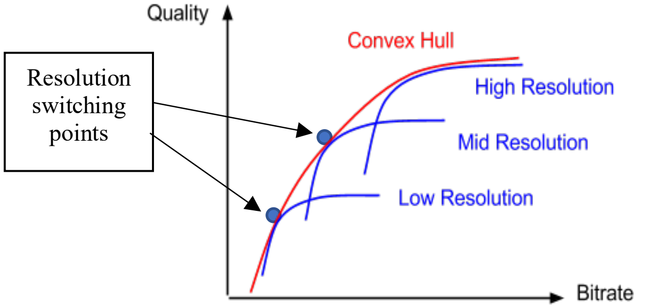


Fig. 4. Using Convex hull, the red line, instead of resolution curves, blue lines.

In the figure5, It is shown, by having the convex hull, the intersecting points between the convex hull and RD-curves are the resolution switching points, and the resolution of the clip after those points are flattening and switching to a better resolution is convenient.

The convex hull set can use as the best quality the distributer can use and provide at each bit rate for each clip.

In the figure 6, the distortion diagram of one of the existing video clips and its convex hull set based on PSNR and VMAF in different resolutions present.

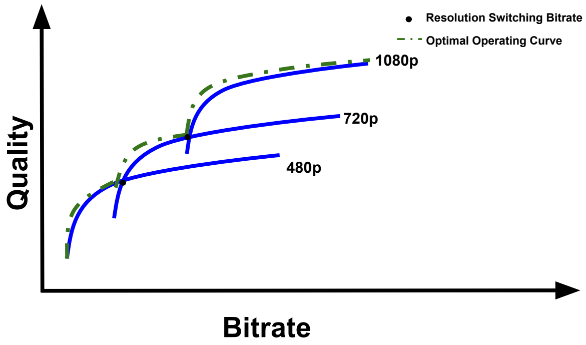


Fig. 5. By increasing the bitrate, quality increases, but after passing a point, it starts to flatten.

VI. Problem Definition

The main challenge of this project is to predict VMAF based on the encoding parameters as well as video information. However, the prediction strongly affected by content complexity, such as temporal and spatial complexity.

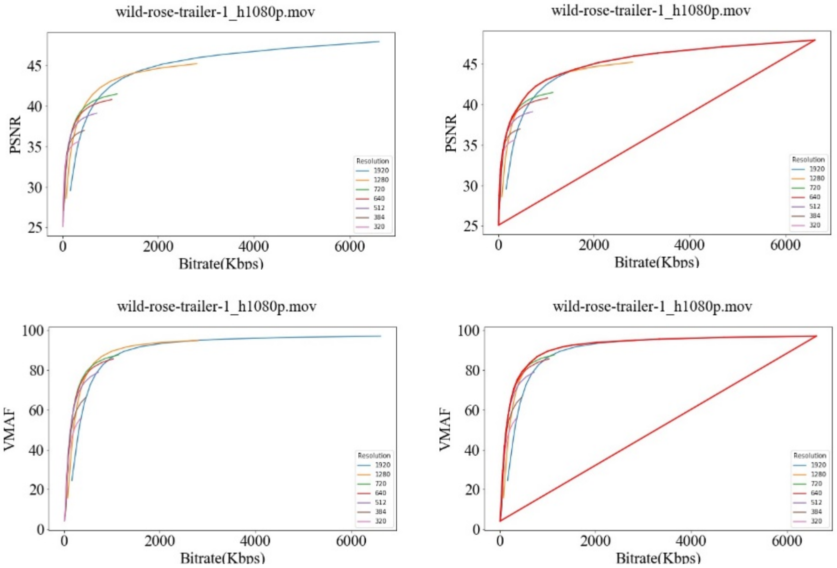


Fig. 6. One of the Source videos based on the quality metrics such as PSNR and VMAF, and fitted Convex Hull over them.

Our dataset consisted of four tables as below:

- **Clip**: includes the attributes of a clip.
- **Encode**: describes a single encode test with used encoding parameters (e.g., CRF, Resolution, etc.) and derived quality metrics (VMAF, PSNR) of an encode.
- **Scene Change**: scene changes indicate a change of the setting/scene in a video (complex content has a lot of scene changes).
- **Label**: consisted of labels for each source video which are tagged using machine learning classifiers like Tensorflow Mobilenet to define labels and categories for the content.

A. *Data preprocessing*

In order to make the dataset ready for further analysis, some preprocessing techniques were applied. For instance, the samples having the missing value were removed entirely. In order to remove the extreme outliers, the technique ITU-T Recommendation P.1401 was followed, which is basically for subjective rating, but we found it suitable for our data as well. In this method, unlike the general Box and whiskers graphs [aa] in which data out of the range 1.5 times the Interquartile Range (IQR) considering as outliers, 3 times the IQR used to find and remove the extreme outliers [yy].

As an example of data preparation, the format of data for the framerate from 30000/1001 changed to 30 fps. Also, in the feature Clip-Frame-Rate, as the most values are in the range of 20 to 30, the data out of this range were removed as extreme outliers.

The categorical data transformed into numerical data that can be used for the regression task using the dummy coding method. Since there was only one categorical feature in the given dataset, video profile, it was classified as zero means high and one as main profile. However, in the project, since this feature did not play an important role in the result was removed for the final model.

Additionally, since the features had different ranges, normalizing the data was required in order to make all the numeric columns in the dataset lie on a common scale.

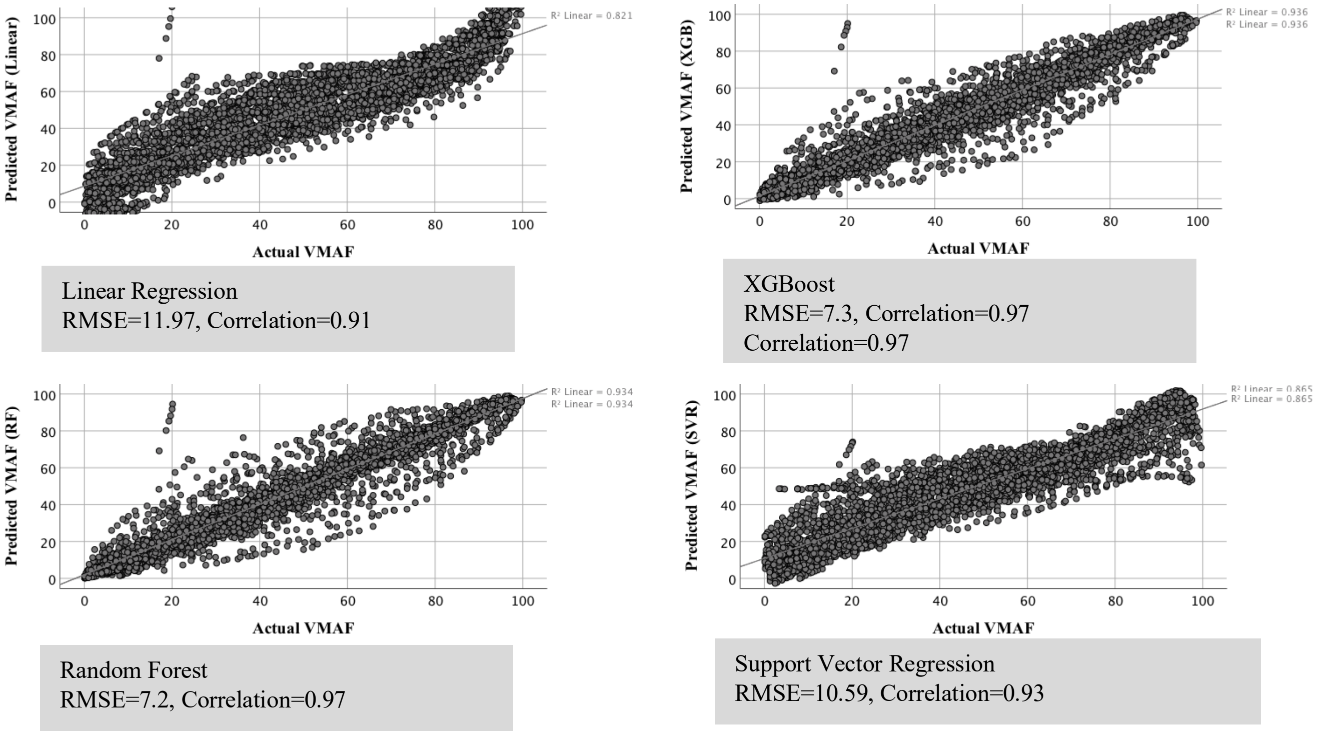
B. *Training data*

In this phase of the project, three different methods were used. The first approach was Blind Cross-Validation, where we used 5-fold cross-validation on all data points. The result looks excellent which made us suspicious about using simple cross-validation. We realized that using simple cross-validation would not be fair/correct as some encoded sequences of one source video sequence could be included in the training set while some others from the same source would present in the test set.

Some of the encoded videos of each source video may place in train data and the rest in the test set. In this way, it looks unfair because the results probably biased to the training set, and the trend of each source video would be predictable easily; This is why this approach had the best result.

reference video. Approximately 25% of source videos (together with all encoded of source videos) are chosen to be the test data, and 75% remaining were using in the training set. This method ensures that we are not biased to the training set.

The second approach in the project was splitting the dataset into a fixed training and test sets, so based on the results would be more reliable.



Linear Regression
RMSE=11.97, Correlation=0.91

XGBoost
RMSE=7.3, Correlation=0.97
Correlation=0.97

Random Forest
RMSE=7.2, Correlation=0.97

Support Vector Regression
RMSE=10.59, Correlation=0.93

Fig. 7. The result of applying the method One-Leave-Out on all used regression models.

However, there was still a problem that data divided randomly; it means that might happen the low complexity videos place in training set and high complexity videos in the test set and vice versa.

The third approach, as leave-one-out cross-validation, took benefits of the two other methods. It means it worked like typical cross-validation with the difference that K=n (n is the number of source videos), then all source videos can take into account as test and train test. Besides, the benefit of the second approach is that instead of working with encoded videos, we could divide data based on source videos.

Furthermore, in order to have the best comparison, four regression models as Linear regression, Support Vector Regression (radial basis function kernel), random forest, and XGBoost were used.

C. Predicting VMAF

During the first phase of the project, to see the first result, only two tables, Clip and Encode, were merged with the assumption that all videos have the same complexity level. The results would not be that precise as the video complexity (Temporal and spatial complexity) is not considered.

The features used for training consist of as below:

- Clip table: clip_duration, clip_frame_rate, clip_height, clip_size
- Encode table: encode_WidthHeight, encode_bitrate_video , encode_crf

Then, in the second phase of the project, we merged the table scene change in order to take into account the temporal video complexity. In order to have a standard parameter for scene change (as the size of clips varies across clip), scene change percentage in one second is calculated for each clip. Therefore,
the sum of scene change percentage was divided to the clip size in a second for each clip.
It has to be noted that while temporal complexity could be estimated using scene change ratio in a second, there is still the weakness of missing the spatial complexity information. For the spatial complexity the Mobile-net labels could help, which was not investigated in detail in this project due to high number of labels compared to sample.

The final result, which is a comparison between four employed regression models implemented on merged three tables Clip, Encode and Scene change using Leave-one-out cross-validation method, can be seen in the figure7.

D. Importance of features for training

As the importance and influencing level of all features in a dataset are not the same, there are many ways to find out what features play the leading role in the prediction. We utilize the XGBoost library, which gives possibility for the XGBoost regression to rank the features contribute the most. It is shown in the figure8 that features based on their F-score are prioritized. Since the number of features in our dataset was not that much, the result after removing three low ranked feature did not have any significant change on the final results.

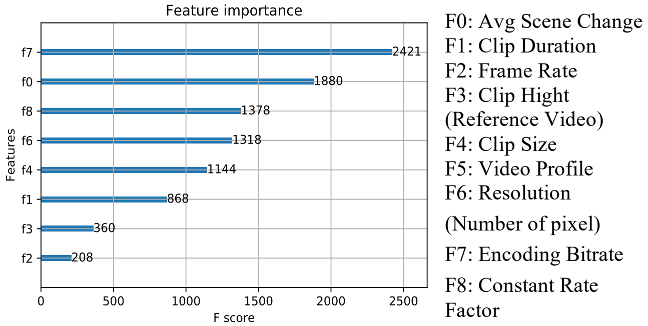


Fig. 8. The result of applying XGBoost method to have feature importance.

E. Improving the results by adding content labels (Mobilenet labels)

As we supposed to involve both temporal and spatial complexity features such as motion and texture, during the last phase of the project in order to take the spatial complexity into account, the table Label merged with other tables. The goal was to use labels to determine the level of spatial (and termporal) complexity. Due to very high number of labels in total, we first tried to only keep the most frequent labels in which we can analysis the labels subjectively. Then the dummy coding method could have been employed to allow categorical data be used in the regression. Therefore, frequency histogram of labels was drawn which a long list of the labels with various frequencies appeared in the plot. Since, we could not find a small set of labels that cover a wide range of clips, we did not apply it on the final model. So, it remains an item for future works.

F. Predicting Convex hull using our trained model

At the end, the convex hull for all twenty source videos based on the actual VMAF and four regression models such as Linear regression, SVR (RBF kernel), random forest and XGBoost applied and as a result Random Forest, and Linear regression had the best and the worst prediction (based on their similarity to the actual model) respectively. Figure 7 illustrates the convex hull for Random Forest model.

Conclusion

In this project, we build a parametric based model that can predict the convex hull using encoding parameters as well as content information such as scene change information. In addition, we compared different types of regression models to find the best model that predict VMAF. In this project, we first processed the raw data by removing outliers, handle the missing values and change the non-standardized range of feature values. Next, we aim at predicting VMAF based on encoding parameters using different cross-validation strategies. Once the best model was found out, the convex hull prediction was predicted based on the regression model.

In this project, we did not use the mobile-net labels due to high number of labels compared to samples. However, the mobile-net labels could semantically be reduced to a few limited labels (or classified into a few groups) which remains as future works.