SCTE·ISBE CABLE-TEC EXPO 2019 NEW ORLEANS, LA SEPT. 30-OCT. 3

2019 Fall Technical Forum SCTE·ISBE • NCTA • CABLELABS

Optimizing Video Customer Experience with Machine Learning

A Technical Paper prepared for SCTE•ISBE by

Mariela Fiorenzo
Senior Data Scientist
Telecom Argentina S.A.
Agüero 2392, Buenos Aires, Argentina
Phone: +5411 5330 6946
mafiorenzo@teco.com.ar

Claudio Righetti
Chief Data Scientist
Telecom Argentina S.A.
crighetti@teco.com.ar

María Cecilia Raggio
Data Scientist
Telecom Argentina S.A.
mcraggio@teco.com.ar

Fernando Ochoa
Data Scientist
Telecom Argentina S.A.
fochoa@teco.com.ar

Gabriel Carro
VP Engineer and R&D
Telecom Argentina S.A.
gcarro@teco.com.ar

Table of Contents

List of Figures

List of Tables

Title Page **Number**

Abstract

In the recent years Artificial Intelligence (AI) and Machine Learning (ML) are transforming our industry in many different areas. The main application of AI in the telecommunications industry is in the network management area. But there are also multiple initiatives to improve customer experience using AI and ML. But, do we really need ML to do this? In this paper we give this answer and an approach of some technical challenges we are facing in order to optimize the customer experience on video. We have two main objectives: First, to optimize some statistical model and ML techniques that already exist and second, to use them to equalize the video quality of the different video platforms that we have at our Company.

Content

1. Introduction

The videos have several characteristics that determine their quality: resolution, frame rate, aspect ratio, color model, etc. On the other hand, the videos are digitized, compressed and transported through a network before reaching the final device. For this, there are also a diversity of methods with different characteristics that affect the final perception of the users.

For video service providers it is important to evaluate the processes that affect the quality of videos considering the perception of customers. The subjective evaluation of the quality of the videos measuring the opinion of human users is expensive and slow, although there are public databases with standardized results. To automate the evaluation of quality of videos objective models that try to approach the subjective evaluation human are used. These objective models can be applied on a large scale because they are executed by a computer. There is a great variety of classic models that vary from simple formulas that compare the original image with the final pixel by pixel, to complex processes that involve different metrics.

Recently, objective models emerged using Machine Learning (ML) algorithms which are trained using databases with subjective evaluations, to combine a variety of classical metrics. Classical metrics are much simpler to implement and at a lower cost but, they produce worse results that do not always fit the human perception. On the contrary, the metrics based on ML produces results very close to the subjective opinion of the customers, but they are more complex to implement and provide development opportunities.

Within the objective metrics based on ML there are two methods that produce similar results: Video Multimethod Assessment Fusion (VMAF) and Video Quality Model with Variable Frame Delay (VQM-VFD). VMAF it's an open-source method proposed by Netflix in 2016 and VQM-VFD was standardized by ITU in 2003.

2. What is Machine Learning?

While there are many ways to define what is Machine Learning, the following two definitions are the most important within data science.

- Arthur Samuel (1959): *"Field of study that gives computers the ability to **learn** without being explicitly programmed".*

- Tom Mitchell (1997): *"A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E".*

The field of machine learning addresses the question of how to build computer programs that automatically improve with experience via the training and testing datasets. In the last decades, many successful applications have been developed and they are developing more and more, year by year. From data mining programs that learn to detect fraudulent bank transactions, to predict churn, recommendation systems based in user preferences, even autonomous vehicles that learn to drive on public highways, face and voice recognition.

According to Mitchell [1], ML algorithms have proven to be especially useful in data mining problems where large databases may contain valuable implicit regularities that can be discovered automatically; poorly understood domains where humans might not have the knowledge needed to develop effective algorithms; and domains where the program must dynamically adapt to changing conditions.

The ML algorithms can be divided into three major categories: supervised learning, unsupervised learning and reinforcement learning. Supervised learning is useful in cases where a label is available for a given data set, but it must be predicted for other instances. Unsupervised learning is useful in cases where the challenge is to discover patterns in an untagged data set. Reinforcement learning falls between these two extremes: there is some form of feedback available for each step or predictive action, but there is no precise label. These are the most important and well-known algorithms: Decision Trees, Logistic and Linear Regression, Association Rules, Support Vector Machines (SVM), Clustering, Artificial Neural Networks (ANN), Deep Learning.

For any of those algorithms there is a process to follow from getting de data to obtain results and improve them as we can see in Figure 1.

In the following section we are going to study more deeply one of the most robust ML algorithms which is applied to VMAF technique: Support Vector Machines.

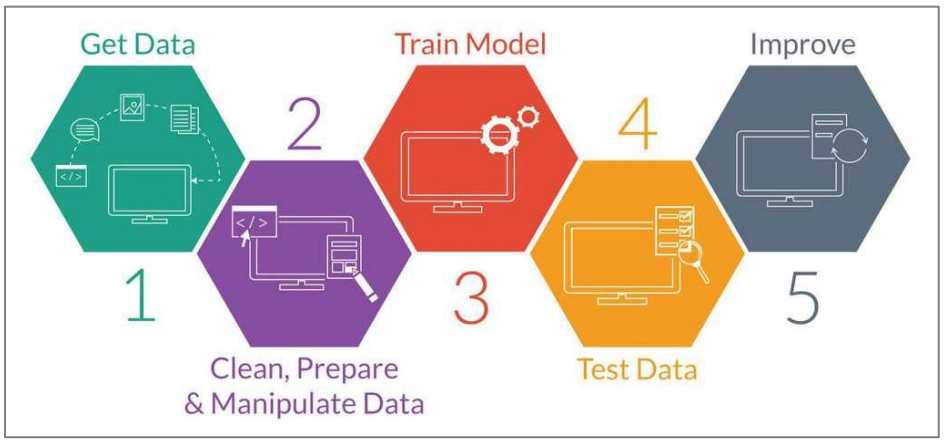

Figure 1 – Machine Learning process.

2.1. Support Vector Machines

Support Vector Machines (SVM) is a technique that works with labeled training data, it is highly preferred by many as it produces significant accuracy with less computation power and it can be used for both regression and classification tasks. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space where N is the number of features that distinctly classifies the data points. In 2-dimensional space this hyperplane is a line dividing a plane in two parts where each class lay in either side. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. The objective is to find a plane that has the maximum margin. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Figure 2 illustrates the key idea of an SVM:

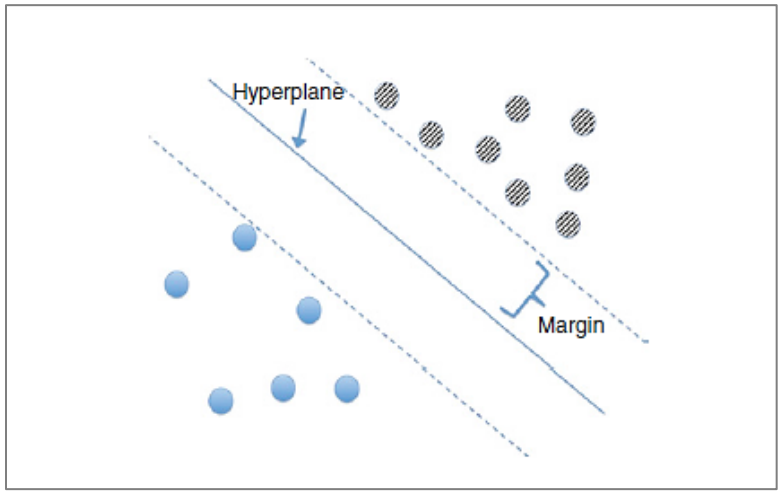

Figure 2 – Example of data points, hyperplane and margin using SVM in a 2-dimensional space.

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane.

The assumption that there actually is such a hyperplane is called "linear separability". Training the SVM involves finding the hyperplane that (1) separates the datasets and (2) is "in the middle" of the gap between the two classes [2] [3] [4].

In the simple case of a linear classifier, our goal is to estimate a linear decision function:

$$f(x) = \theta_0 + \theta^T x$$

In 2-dimensional space, the hyperplane is specified by the equation:

$$f(x) = b + w \cdot x = 0$$

where w is a vector perpendicular to the hyperplane and b measures how far offset it is from the origin. To classify a point x, simply calculate $f(x)$ and see whether it is positive or negative. Training the classifier consists of finding the w and b that separates the dataset while having the largest margin.

2.1.1. Optimization Objective

To obtain a large margin classifier we must estimate θ by minimizing the following expression [5] [6]:

$$J(\theta) = C \sum_{i=1}^{m} [y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)})] + \frac{1}{2} \sum_{i=1}^{n} \theta_i^2$$

Where, m is the number of examples, n is the number of features and $y^{(i)} \in \{0,1\}$ a binary response. Let's call $z = \theta^T x$. To obtain $\min_\theta J(\theta)$ we have two options:

- If $y = 1$, we want $z \geq 1$
- If $y = 0$, we want $z \leq -1$

The cost functions are represented in Figure 3.

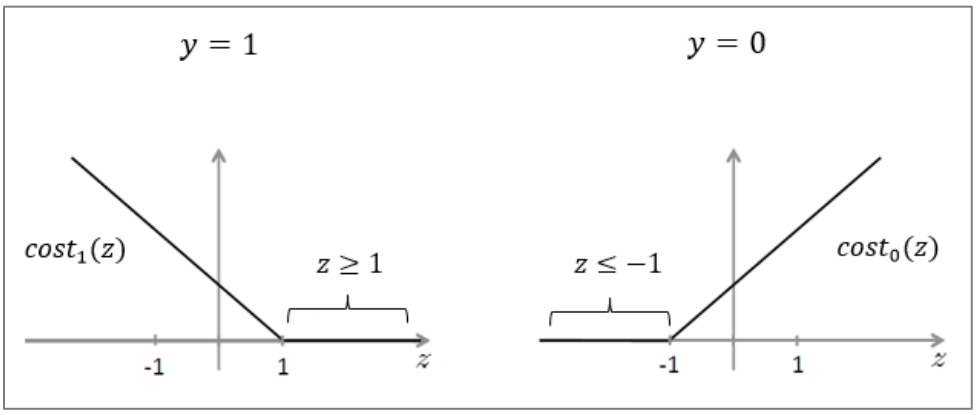

Figure 3 – Cost functions for the SVM optimization problem.

2.1.2. Kernels

In practice, there often is no hyperplane that completely separates the two classes in the training data. Intuitively, what we want to do is find the best hyperplane that almost separates the data, by penalizing any points that are on the wrong side of hyperplane.

An SVM will fail utterly on a dataset as we can see in Figure 4:

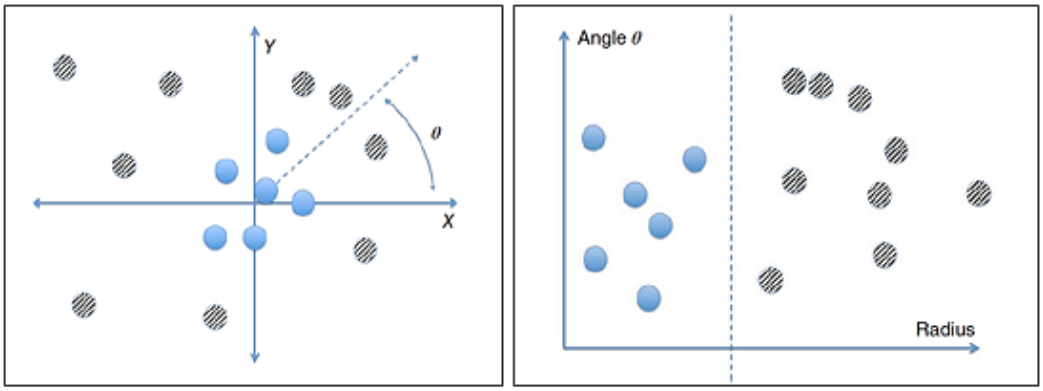

Figure 4 – Non-linear separability problem and a kernel SVM solution.

There is no line between the two classes of points. The pattern is clear if you just look at it – one class is near the origin, and the other is far from it – but an SVM can't tell. The solution to this problem is a very powerful generalization of SVM called "kernel SVM." The idea of kernel SVM is to first map our points into some other space in which the decision boundary is linear, and then construct a support vector machine that operates in that space.

Generally, a kernel function is noted as $k(x, y)$ and some of the most popular are as follows:

- Polynomial kernel: $k(x, y) = (x \cdot y + c)^n$

- RBF or Gaussian kernel: $k(x, y) = exp[-\frac{1}{2\sigma^2} \cdot ||x - y||^2]$
- Sigmoid kernel: $k(x, y) = \tanh(x \cdot y + r)$

And in this case our goal is to estimate a decision function:

$$f(x) = \theta_0 + \sum_{i=1}^{m} \theta_i k(x, x_i)$$

The optimization objective is to obtain $\min_{\theta} J(\theta)$, where

$$J(\theta) = C \sum_{i=1}^{m} [y^{(i)} cost_1(\theta^T k^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T k^{(i)})] + \frac{1}{2} \sum_{i=1}^{n} \theta_i^2$$

2.1.3. SVM Parameters

When using an SVM, one of the things you need to choose is the parameter C which was in the optimization objective. With a large C value you tend to have a lower bias and high variance, and with a small C value, a higher bias and a low variance. It's very important to control the variance and the bias because a high bias can cause underfitting and a high variance can cause overfitting as we can see in Figure 5.

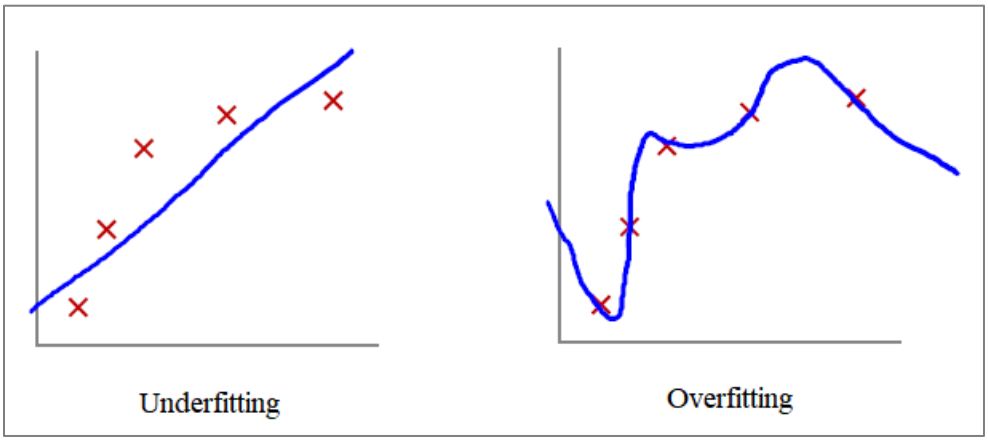

Figure 5 – Examples of Underfitting and Overfitting caused by high bias and high variance.

Particularly, overfitting is when we have too many features and the learned model may fit the training set very well but fail to generalize to new examples.

Another parameter we have to choose is σ^2 which appear in the Gaussian kernel. If σ^2 is large, then the features will vary more smoothly and so we have higher bias and lower variance. On the other side, if σ^2 is small, then the features will vary less smoothly and we have lower bias and higher variance as we can see in Figure 6.

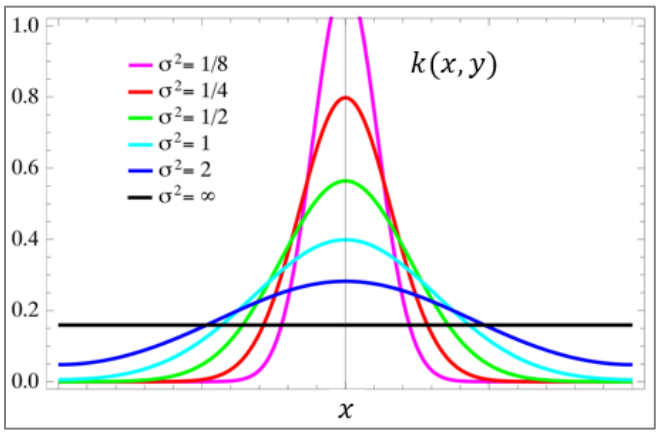

Figure 6 – Variance and bias of the features depending on σ^2 value.

2.2. Support Vector Regression

Support Vector Regression (SVR) works on similar principles as Support Vector Machine (SVM) classification. One can say that SVR is the adapted form of SVM when the dependent variable is numerical rather than categorical. A major benefit of using SVR is that it is a non-parametric technique, the output model from SVR does not depend on distributions of the underlying dependent and independent variables. Instead the SVR technique depends on kernel functions. Another advantage of SVR is that it permits for construction of a non-linear model without changing the explanatory variables, helping in better interpretation of the resultant model. The basic idea behind SVR is not to care about the prediction as long as the error is less than certain value, because of the principle of maximal margin. The regression can also be penalized using the cost parameter, which becomes handy to avoid over-fit.

3. Video Quality Metrics

There are a lot of variables that affect the quality of digital video, from its digitalization, compression, transmission and exhibition. This is why there are many techniques to evaluate the video quality, both in the different stages individually and in the final perception. The techniques for evaluating the quality of a digital video can be divided into two large groups: subjective evaluations and objective evaluations.

Subjective evaluations are linked to the perception of the individuals who watch the video. Every subject indicates his opinion regarding particular videos. Although these evaluations reflect directly the perception of the user, since they involve human interaction, are usually expensive and demand a lot of time. There are several standards of how to carry out these tests in a repeatable way, for example Degradation Category Rating (DCR), Double Stimulus Rating (DSR), Double Stimulus Impairment Scale (DSIS), Pair Comparison (PC), Double Stimulus Continuous Quality Scale (DSCQS), among others.

Objective evaluations are mathematical models that attempt to approximate the results of subjective evaluations but are based on metrics that can be measured objectively by a computer. In this case the evaluations are quick, inexpensive and can be automated. However, they do not

always reflect the perception of the human being. Some of the most commonly used classic metrics are:

- Mean Square Error (MSE): This is the simplest method, which compares the original frames against the modified pixel by pixel calculated its mean square error. MSE it can be defined according to the following formula:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(y_{ij} - x_{ij})^2$$

- Peak signal-to-noise ratio (PSNR): This metric also compares pixel by pixel and measures the relationship between the maximum value of a signal (a frame in this case) and noise. Sound expressed in decibels and calculated with the following formula:

$$PSNR = 20 \cdot \log_{10}(MAX) - 10 \cdot \log_{10}(MSE)$$

  PSNR does not consistently reflect human perception.

- Structural Similarity (SSIM): This method compares the similarity between different frames using regions of the image. It also incorporates perceptual concepts of human vision such as luminance, contrast and structure (the distortions are less obvious in bright regions or with many textures) [7]

Recently new metrics have emerged that use machine learning algorithms to improve the measurement of video quality. While the classic metrics listed previously are widely used given their relatively simple implementation, not necessarily reflect user's opinion: the metric can be very good, however the perception of most of the users are bad (or vice versa). Each of the classic metrics has its advantages and disadvantages, being able to represent some characteristics of the user's opinion but not others.
The techniques based on machine learning use databases with the opinion of real users about the quality of a lot of videos to do the training. The videos are usually cataloged with different criteria (for example: action, drama, documentary, etc.). Among the modern ML-based metrics we can highlight the following:

- Video Quality Model with Variable Frame Delay (VQM-VFD): model proposed by the National Telecommunications and Information Administration (NTIA) agency and accepted as standard by the International Telecommunication Union (ITU). VQM-VFD is an algorithm that uses a neural network model to fuse low-level features, such as spatial and temporal gradients, into a final metric.

- Video Multimethod Assessment Fusion (VMAF): open-source model proposed by Netflix in 2016. It predicts subjective quality by combining multiple elementary metrics using a ML algorithm. We will go deeper into this technique in the next section [8] [9].

4. VMAF

As we mention in the previous section VMAF is a model proposed by Netflix in 2016 and is a video quality metric that combines human vision modeling with ML in order to provide a great viewing experience to their members.

First, they innovate in the area of video encoding. When videos are compressed too much or improperly, these techniques introduce quality impairments, known as compression artifacts. There also exist scaling artifacts - for lower bitrates, video is downsampled before compression, and later upsampled on the viewer's device.

4.1. The Dataset

To evaluate video quality, they work with a dataset of 34 source clips, each 6 seconds long of different genres and content characteristics. Using these source clips, they encoded H.264/AVC video streams at resolutions ranging from 384x288 to 1920x1080 and bitrates from 375 kbps to 20,000 kbps, resulting in about 300 distorted videos.

4.2. The Algorithm

VMAF is a fusion of elementary metrics into a final metric using a ML algorithm. Each elementary metric may have its own strengths and weaknesses with respect to the source content characteristics, type of artifacts and degree of distortion. The ML algorithm used is Support Vector Machines regressor, which assigns weights to each elementary metric, the final metric could preserve all the strengths of the individual metrics and deliver a more accurate final score. Netflix trained and tested the model using the previously mention dataset and the opinion scores obtained through a subjective experiment [10].

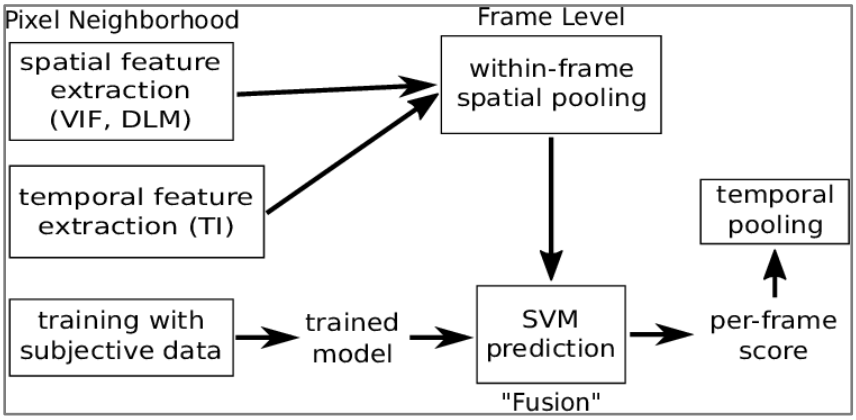

Figure 7 – How VMAF works.

The elementary metrics fused by SVM regression are:

- Visual Information Fidelity (VIF): VIF is a well-adopted image quality metric based on the premise that quality is complementary to the measure of information fidelity loss. It was developed by Hamid R Sheikh and Alan Bovik at the Laboratory for Image and Video

Engineering (LIVE) at the University of Texas at Austin in 2006 and shown to correlate very well with human judgments of visual quality. In its original form, the VIF score is measured as a loss of fidelity combining four spatial scales. In VMAF, a modified version of VIF has been adopted where the loss of fidelity in each scale is included as an elementary metric yielding in four VIF features [11]. VIF goes from 0 to 1.

- Detail Loss Metric (DLM): DLM is an image quality metric based on the rationale of separately measuring the loss of details which affects the content visibility, and the redundant impairment which distracts viewer attention. In VMAF, DLM is only adopted as an elementary metric. DLM goes from 0 to 1.

VIF and DLM are both quality metrics, so another simple feature has been introduced to account for the temporal characteristics of video:

- Motion: This is a simple measure of the temporal difference between adjacent frames. This is accomplished by calculating the average absolute pixel difference for the luminance component. It is also called Temporal Information (TI). Motion goes from 0 to 20.

Each of these six features is extracted as a feature map of size equal to the corresponding scale. Next, the average value of each feature map is calculated, to produce one feature value per video frame and feature type. For training purposes, VMAF aggregates the per frame features over the entire video sequence, yielding one feature value per training video. These six feature values are fed, together with the corresponding subjective ground truth, to an SVR model. For testing purposes, VMAF predicts one value per video frame and calculates the arithmetic mean over all per frame predictions to predict the overall video quality.

To avoid overfitting to the dataset, this has to be divided in two subsets: TRAIN and TEST, with no overlapped clips. The SVM regressor is then trained with the TRAIN subset and tested on the TEST subset.

4.3. Subjective Experiment and Scoring

The subjective experiment is used to determine how non-expert observers would score the impairments of an encoded video with respect to the source clip. This methodology is the Double Stimulus Impairment Scale (DSIS) method: The viewer sits in front of a 1080p display in a living room-like environment with a viewing distance of 3x the screen height (3H) and sees an unimpaired reference video, then the same video impaired, and after that they are asked to vote on the second video using the Absolute Category Rating (ACR) scale of "bad", "poor", "fair", "good" and "excellent", then they are translated to the values 1, 2, 3, 4 and 5 when calculating the MOS [12]. Hence, VMAF scores range from 0 to 100, with 0 indicating the lowest quality, and 100 the highest. Table 1 shows how to map ACR scale with VMAF scores.

ACR scale	VMAF Scores
"bad"	0-20
"poor"	20-40
"fair"	40-60
"good"	60-80
"excellent"	80-100

Table 1 – Relationship between ACR scale and VMAF score.

4.4. Prediction Uncertainty

As we mentioned before, VMAF is trained on a set of representative video genres and distortions. Due to limitations in the size of subjective experiments, the selection of video clips does not cover the entire space of perceptual video quality. Thus, VMAF predictions should be associated with a confidence interval (CI) that expresses the inherent uncertainty of the learning process. This CI is established through bootstrapping on the prediction residuals [13] using the full training data. It trains multiple models, using resampling with replacement, on the residuals of prediction. The variability of these predictions quantifies the level of confidence. In this case, the CI is 95%. The bootstrapping technique will not necessarily improve the accuracy of the trained model, but will give a statistical meaning to its predictions.

4.5. VMAF Development Kit (VDK) – Open Source Package

Netflix open sourced a VMAF Development Kit package on Github [14] because they think that with the contribution of the industry it can evolve over time and improve its performance.

The feature extraction portion in the VDK core is written in C for efficiency because it is computationally-intensive. And the control code is written in Python for fast prototyping. The package comes with a command-line interface or in batch mode.

VDK offers a number of trained VMAF models to be used in different scenarios. Besides the default VMAF model which predicts the quality of a video displayed on a HDTV in a living-room viewing condition, VDK also includes a number of additional models, covering mobile phone and 4KTV viewing conditions.

4.6. YUV

The video format used in VMAF is YUV. YUV is a color encoding system typically used as part of a color image pipeline. It encodes a color image or video taking human perception into account, allowing reduced bandwidth for chrominance components, thereby typically enabling transmission errors or compression artifacts to be more efficiently masked by the human perception than using a "direct" RGB-representation. Other color encodings have similar properties, and the main reason to implement or investigate properties of Y′UV would be for interfacing with analog or digital television or photographic equipment that conforms to certain Y′UV standards. The Y′UV model defines a color space in terms of one luma component (Y′) and two chrominance (UV) components. Y′ stands for the luma component (the brightness) and U and V are the chrominance (color) components; luminance is denoted by Y and luma by Y′ – the prime symbols (′) denote

gamma compression, with "luminance" meaning physical linear-space brightness, while "luma" is (nonlinear) perceptual brightness.

The higher (or the lower when negative) the U and V values are, the more saturated (colorful) the pixel gets. The closer the U and V values get to zero, the lesser it shifts the color meaning that the red, green and blue lights will be more equally bright, producing a greyer pixel. In turn this meant that when the U and V signals would be zero or absent, it would just display a greyscale image [15].

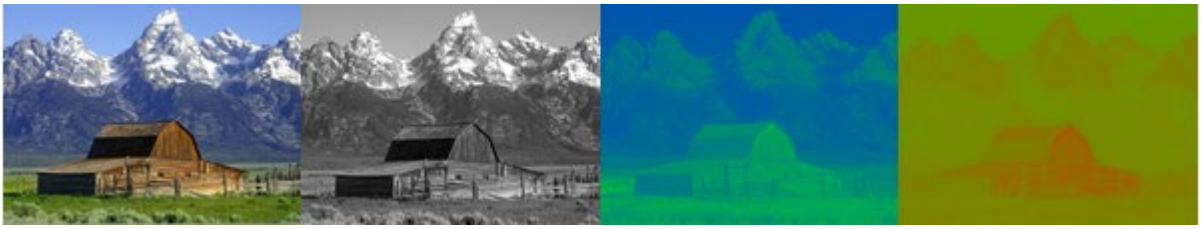

Figure 8 - An image along with its Y′, U, and V components respectively.

4.7. FFmpeg

FFmpeg is the leading multimedia framework, able to decode, encode, transcode, mux, demux, stream, filter and play pretty much anything that humans and machines have created. It supports the most obscure ancient formats up to the cutting edge. No matter if they were designed by some standards committee, the community or a corporation. It is also highly portable: FFmpeg compiles, runs, and passes our testing infrastructure FATE across Linux, Mac OS X, Microsoft Windows, the BSDs, Solaris, etc. under a wide variety of build environments, machine architectures, and configurations [16].

With help from the FFmpeg community, they packaged VMAF into a C library called libvmaf. VMAF is now included as a filter in FFmpeg. Using FFmpeg with libvmaf is very powerful, as you can create complex filters to calculate VMAF directly on videos of different encoding formats and resolutions.

5. Our Use Case

Telecom has its own IPTV platform, called FLOW, in order to deliver the best entertainment service to its subscribers, to increase market penetration and to gain competitive advantage. This deployment is based on unmanaged (second screens) and also on managed devices (set top boxes). It provides different types of advance video services, including Linear TV, various flavors of On Demand services (VoD, CuTV, Reverse EPG, StartOver, network DVR, Pause Live TV and Trick Modes) using different streaming technologies, Search and Recommendations. Thus, it's very important for the Company to develop VMAF as a tool for optimizing FLOW customer experience and to equalize video quality with the other existing video platforms.

5.1. Measuring the Performance

After training and testing the model we need to measure their performance in terms of the correlation between MOS and VMAF values for each video in order to quantify the prediction

accuracy and also the prediction error. To do so, we calculated three statistical measurements, their results will be shown later but in this section we give a brief description of them:

- Spearman's rank correlation coefficient (SRCC): is a nonparametric measure of the statistical dependence between the rankings of two variables. It assesses how well the relationship between two variables can be described using a monotonic function, whether linear or not. A perfect Spearman correlation of +1 (positive correlation) or −1 (negative correlation) occurs when each of the variables is a perfect monotone function of the other [17]. In our use case, a value closer to 1 is desirable.

- Pearson correlation coefficient (PCC): is a measure of the linear correlation between two variables and it is widely used in the sciences. It has a value between +1 and −1, where 1 is total positive linear correlation, 0 is no linear correlation, and −1 is total negative linear correlation. The interpretation of a correlation coefficient depends on the context and purposes. A correlation of 0.8 may be very low if one is verifying a physical law using high-quality instruments, but may be regarded as very high in other sciences where there may be a greater contribution from complicating factors [18]. A value closer to 1 is desirable, like SRCC.

- Root mean squared error (RMSE): is a frequently used measure of the differences between values predicted by a model or an estimator and the values observed. It represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences. These deviations are called residuals when the calculations are performed over the data sample that was used for estimation and are called errors when computed out-of-sample. It is always non-negative, and a value of 0 would indicate a perfect fit to the data. In general, a lower RMSE is better than a higher one. However, comparisons across different types of data would be invalid because the measure is dependent on the scale of the numbers used [19]. For this measurement, a value closer to 0 is desirable.

5.2. Training

5.2.1. FLOW Dataset

To train VMAF for optimizing FLOW customer experience we defined a dataset following Netflix suggestions regarding the type of content. We selected 35 videos, each 10-sec long from FLOW catalog. To make the distortions, each source video is encoded with 6 resolutions up to 1080p. Each resolution has associated a bitrate as it is shown in Table 2. In Figure 9 we can see the encoding complexity in terms of bitrate (kbps) across FLOW dataset contents. The characteristics of the videos are variable, we selected videos with fire, water, nature, animation, close-up, action, crowd, among others.

#	Codec	Resolution	FPS	Bitrate (kbps)
1	H.264	424x240	29.97	678
2	H.264	640x360	29.97	1344
3	H.264	854x480	29.97	1972
4	H.264	1024x576	29.97	2740
5	H.264	1280x720	29.97	3900
6	H.264	1920x1080	29.97	7580

Table 2 – Resolutions and bitrates used to perform the distortions for each source video.

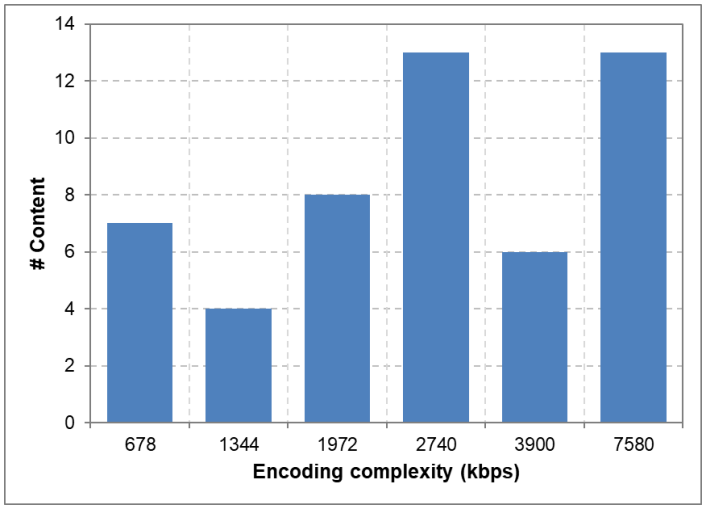

Figure 9 – Encoding complexity across FLOW dataset, expressed as the bitrate (kbps).

5.2.2. Subjective Test for 1080p model

We ran a subjective test through 6 different focus groups. Each group of about 15 subjects. The methodology was the same we mentioned in section 4.3. Each subject sits in a living room-like environment and is instructed to watch an unimpaired reference video, then the same video impaired and give a rating on a continuous scale from "bad" to "excellent" (ACR methodology), then we translated the scale to a range from 1 to 5 and calculate the MOS. Not all videos were viewed by each subject. After calculating the MOS for each video, we translated its value in a scale from 0 to 100 according to Table 1. Then, we defined the training and the testing datasets using 70% and 30% of FLOW Dataset, respectively.

After performing the subjective test, we trained VMAF model with its results and the elementary metrics calculated to the videos of our training dataset. To do the training we run the algorithms provided in the VMAF Development Kit using as elementary metrics the four VIF scales, DLM and Motion. Then, we trained several models with different sets of parameters for the SVR and we found that the best are those presented in Table 3, because by choosing them we avoid the underfitting and the overfitting.

Parameter C was the one set in the algorithm and parameter σ^2 was chosen with the following criteria: $2\sigma^2 = \#\, features$.

Parameters	Values
C	4
σ^2	3

Table 3 – Selected parameters for our VMAF model.

After training the models we calculated their performance metrics described in Section 5.1. In Table 4 we show the metrics for the final model. We can see that the correlation metrics are not near 1, but as it was mentioned before, their values are not low taking into account the type of experiment we are measuring where complicated factors affects the metrics. In our case, the complicated factors are related to the results of the subjective experiment. So, we decided that a value of 0.7 was acceptable.

Metrics	Values
SRCC	0.7
PCC	0.7
RMSE	19.9

Table 4 – Performance metrics for the training dataset.

5.3. Testing

Once we had our model trained, we tested it with the testing dataset previously defined. After the testing we calculated the performance metrics in order to measure the prediction accuracy. The obtained values are shown in Table 5. As expected, SRCC and PCC have lower values for the testing than for the training and RMSE a higher value. These results are indicating that we have to continue improving them by training the model with a larger dataset. In this way, the algorithm could learn from more examples and be able to predict values for future instances with higher accuracy. Furthermore, training with other elementary metrics combinations and finding more appropriate parameters for the SVR could improve our model.

Metrics	Values
SRCC	0.6
PCC	0.6
RMSE	25.3

Table 5 – Performance metrics for the testing dataset.

Conclusion

In this work we presented a new way to measure customer experience in video using Machine Learning techniques that also gives us the possibility to make an optimization in terms of encoding decision. To do so, this technique – called Video Multimethod Assessment Fusion (VMAF) - fuses elementary metrics and subjective evaluations into a Support Vector Regressor to obtain a predicted value of subject opinions about videos.

We made an introduction about Machine Learning giving some academic definitions and explaining its typical process. Furthermore, we described three classes of ML algorithms, how they work and we also mentioned the most used of them. After that, we focused on Support Vector Machines, the algorithm in which VMAF is based on.

After mentioning some classical video quality metrics, we concentrated on how VMAF works. We explained how to build the dataset (type of contents, distortions, videos length, amongst others), the elementary metrics used to train the model, how to perform the subjective experiments with different focus groups and its scoring and we gave details related to the VMAF Development Kit which contains the algorithms and tools to run the models.

Finally, we described our use case for FLOW, Telecom IPTV platform. Particularly, how we constructed our video dataset, the development and scoring of our subjective tests through 6 different focus groups, the selection of the SVR's parameters and the performance metrics that allow us to conclude the best model. Based on the results we obtained, we understand that we have to continue improving the model with larger training and testing datasets, other elementary metrics combination and maybe more appropriate parameters for the SVR. Once we have our definitive model for 1080p, we are going to explore models for second screens and 4K to continue optimizing customer experience in other devices.

Abbreviations

ACR	absolute category rating
AI	artificial intelligence
ANN	artificial neural network
CI	confidence interval
CUTV	catch up television
DCR	degradation category rating
DLM	detail loss metric
DMOS	differential mean opinion score
DSCQS	double stimulus continuous quality scale
DSIS	double stimulus impairment scale
DSR	double stimulus rating
DVR	digital video recorder
EPG	electronic program guide
HDTV	high definition television
IPTV	Internet Protocol television
ITU	International Telecommunication Union
Kbps	kilobits per second
LIVE	Laboratory for Image and Video Engineering
ML	machine learning
MSE	mean square error
MOS	mean opinion score
NTIA	National Telecommunications and Information Administration
PC	pair comparison
PCC	Pearson correlation coefficient
PSNR	peak signal to noise ratio
RBF	radial basis function
RMSE	root mean squared error
SRCC	Spearman's rank correlation coefficient
SSIM	structural similarity
SVM	support vector machines
SVR	support vector regressor
TI	temporal information
VDK	VMAF development kit
VIF	visual information fidelity
VMAF	video multimethod assessment fusion
VOD	video on demand
VQM-VFD	video quality model with variable frame delay

Bibliography & References

[1] T. Mitchell, Machine Learning, USA: WCB/McGraw-Hill, 1997.

[2] F. Cady, The Data Science Handbook, Hoboken, New Jersey: Wiley, 2017.

[3] R. Gandhi, "Towards Data Science," 7 June 2018. [Online]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

[4] C.-W. Hsu, C.-C. Chang and C.-J. Lin, "A Practical Guide to Support Vector Classification," Department of Computer Science - National Taiwan University, Taipei, Taiwan, 2016.

[5] A. Ng, "Coursera," [Online]. Available: https://www.coursera.org/learn/machine-learning/home/welcome.

[6] T. Hastie, S. Rosset, R. Tibshirani and J. Zhu, "The Entire Regularization Path for the Support Vector Machine," *Journal of Machine Learning Research,* no. 5, pp. 1391-1415, 2004.

[7] "Wikipedia, the free encyclopedia," 4 July 2019. [Online]. Available: https://en.wikipedia.org/wiki/Structural_similarity.

[8] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy and M. Manohara, "The Netflix Tech Blog," 6 June 2016. [Online]. Available: https://medium.com/netflix-techblog/toward-a-practical-perceptual-video-quality-metric-653f208b9652.

[9] Z. Li, C. Bampis, J. Novak, A. Aaron, K. Swanson, A. Moorthy and J. De Cock, "The Netflix Tech Blog," 25 October 2018. [Online]. Available: https://medium.com/netflix-techblog/vmaf-the-journey-continues-44b51ee9ed12.

[10] C. G. Bampis, Z. Li and A. C. Bovik, "SpatioTemporal Feature Integration and Model Fusion," *IEEE Transactions on Circuits and Systems for Video Technology,* 2018.

[11] H. Sheikh and A. Bovik, "Image Information and Visual Quality," *IEEE Transactions on Image Processing,* vol. 15, no. 2, pp. 430-444, 2006.

[12] "Wikipedia, the free encyclopedia," 11 July 2019. [Online]. Available: https://en.wikipedia.org/wiki/Subjective_video_quality.

[13] "Wikipedia, the free encyclopedia," 9 July 2019. [Online]. Available: https://en.wikipedia.org/wiki/Bootstrapping_%28statistics%29#Resampling_residuals.

[14] "Netflix Github," [Online]. Available: https://github.com/Netflix/vmaf.

[15] "Wikipedia, the free encyclopedia," [Online]. Available: https://en.wikipedia.org/wiki/YUV.

[16] "FFmpeg," [Online]. Available: https://ffmpeg.org/about.html.

[17] "Wikipedia, the free encyclopedia," August 2019. [Online]. Available: https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient.

[18] "Wikipedia, the free encyclopedia," August 2019. [Online]. Available: https://en.wikipedia.org/wiki/Pearson_correlation_coefficient.

[19] "Wikipedia, the free encyclopedia," April 2019. [Online]. Available: https://en.wikipedia.org/wiki/Root-mean-square_deviation.

