July 2021

Importance Sampling for Evaluation of Video Transcoder Performance

Yao-Chung Lin

Jeremy Dorfman

Neil Birkbeck

Importance Sampling for Evaluation of Video Transcoder Performance

ABSTRACT

Optimization of a video transcoder is performed, e.g., by fine-tuning their parameters, based on evaluation of the performance of a transcoder over a small, fixed video dataset. The use of a small, fixed video dataset enables reproducibility, fast evaluation, and regression testing. However, transcoders that are fine-tuned based on a small, fixed dataset can often deliver suboptimal transcoding performance when utilized to transcode videos from a much larger dataset, e.g., videos served by a video hosting and sharing service. This is because a small, fixed set of videos is not sufficiently representative of the total corpus of videos hosted by a video sharing service and does not cover the scale and diversity of such videos. This disclosure describes the use of importance sampling in the evaluation of video transcoders using a small dataset of videos. The techniques can deliver a high-performance transcoder even when the transcoder is optimized using a small dataset that is insufficiently representative of a large-scale video corpus.

KEYWORDS

- Importance sampling
- Video transcoder
- Transcoder performance
- Compression performance
- Video streaming
- Video quality
- Video bitrate

BACKGROUND

Video hosting and sharing services compress and/or transcode videos such that they can be streamed to different client devices such as mobile devices, high-resolution TVs, desktop computers, etc. at resolutions, frame rates, and other attributes that are appropriate to the screens and the network connection quality of each device. Video compression has a substantial impact both on user experience and infrastructure costs for such services. Better video compression enables delivery of higher quality video at lower streaming bitrates.

Video transcoders are optimized, e.g., their parameters fine-tuned, by evaluating their compression performance over a small, fixed video dataset. The use of a small, fixed video dataset enables reproducibility, fast evaluation, and regression testing. Using the entire corpus of uploaded videos for evaluation purposes is infeasible. Also, use of such corpus may not be feasible due to security and/or privacy-related aspects.

Unfortunately, transcoders that are fine-tuned based on a small, fixed dataset can often deliver suboptimal transcoding performance. This is because a small, fixed set of videos is not sufficiently representative of the total corpus of videos hosted by a video sharing service and does not cover the scale and diversity of such videos. The characteristic distributions of the dataset and the corpus inevitably differ. For example, the dataset may have a preponderance of slow-moving, slide-sharing videos while the corpus may be dominated by fast-moving music videos. Furthermore, as more videos are uploaded to the video hosting service, the corpus changes in size and composition. Thus, a sampled dataset that was once representative might not remain so with the passage of time. Optimizing transcoders over a small dataset with improper aggregation yields suboptimal results when the transcoder is utilized for videos in the production corpus.

A common way to aggregate performance results - averaging performance over the small dataset- may not be a good estimate of the true transcoder performance. When optimizing video encoders using metrics averaged over a small dataset, over-tuning can take place for extreme cases, yielding suboptimal results on the actual corpus.

Importance sampling

Importance sampling is a technique by which statistical measures (e.g., average performance) assessed over a small dataset with a certain probability density function (PDF) can be generalized to a larger dataset with a different PDF. Briefly, importance sampling works as follows.

Let the small dataset have a PDF $q(x)$ over some (multi-dimensional) random variable x. For example, x can be a two-dimensional vector with entries representing spatial texture and temporal correlation of the videos in the dataset. In general, x is a property of the video that can impact a transcoder performance metric such as encoded bitrate, reconstructed quality, etc. Let the larger corpus of videos have a PDF $p(x)$ over the random variable x. Let f be a performance metric of the transcoder, e.g., bitrate, video quality, etc. As mentioned before, f depends on x; this is denoted as $f(x)$: f is a function of x. Per the principles of importance sampling, an estimate of the average (expected) performance over the larger corpus can be obtained by using n observations of f over the smaller dataset as follows.

$$\text{Expected value of } f \text{ over the } \textit{larger} \text{ corpus} = E(f) \cong \frac{1}{n} \sum_i \quad f(x_i) \frac{p(x_i)}{q(x_i)},$$

where the samples x_i are drawn from the PDF $q(x)$ of the *smaller* dataset. In this manner, assuming the availability of the ratio $p(x_i)/q(x_i)$ for each x_i, average performance over the *larger*

corpus can be obtained by sampling and averaging over the *smaller* dataset. The ratios $p(x_i)/q(x_i)$ are known as importance weights, denoted w_i.

DESCRIPTION

This disclosure describes the evaluation of video transcoders using a small dataset of videos in such a manner that the evaluation remains true to a large-scale video corpus of which the small dataset is an imperfect representative. Such evaluations of transcoder performance can be used to fine-tune the transcoder using the small dataset, such that the transcoder provides sufficient performance when utilized for the large-scale video corpus.

Content representation, e.g., modeling x and determining p(x) and q(x)

To use the importance sampling framework to obtain the importance weights w_i, the random variable x is to be modeled, and estimates of the PDFs $p(x)$ and $q(x)$ are to be obtained. The characteristics (or representation) of x are optimally rich enough to define the content space for the application at hand (e.g., transcoding); good content characteristics are likely good predictors of the quality or the bitrate obtained by transcoding a particular piece of content.

Using content category to model content

To model content using category, the content, x, is modeled as a discrete variable, e.g., a content category such as $x \in \{\text{gaming, sports, music, ...}\}$. The PDFs $p(x)$ and $q(x)$ are respectively the empirical probability of a particular content category in the larger corpus and the smaller dataset.

Example: Consider that there are three possible content categories: gaming, sports, and music. The production (corpus) probability distribution is measured as follows.

P(gaming) = 0.3;

P(sports) = 0.1; and

P(music) = 0.6.

The smaller dataset is evenly sampled, e.g., it has 100 gaming clips, 100 sports clips, and 100 music clips. The weights for gaming, sports, and music are respectively proportional to

$w_{\text{gaming}} = 0.3 / (100 / 300) = 0.9$;

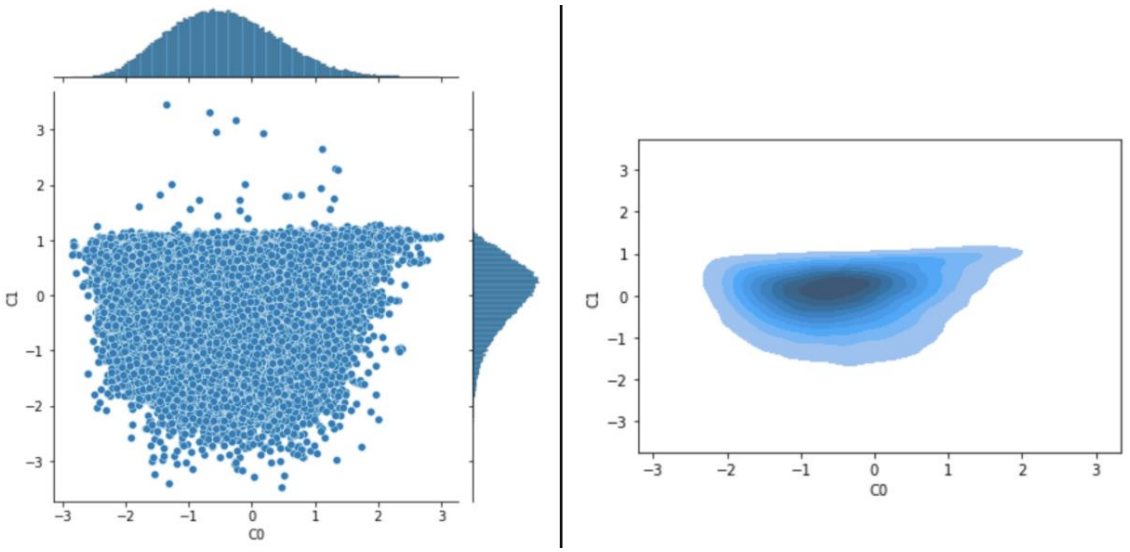
$w_{\text{sports}} = 0.1 / (100 / 300) = 0.3$; and

$w_{\text{music}} = 0.6 / (100 / 300) = 1.8$.

Essentially, since music content is relatively more prevalent in the corpus than in the smaller dataset, it receives a higher weighting.

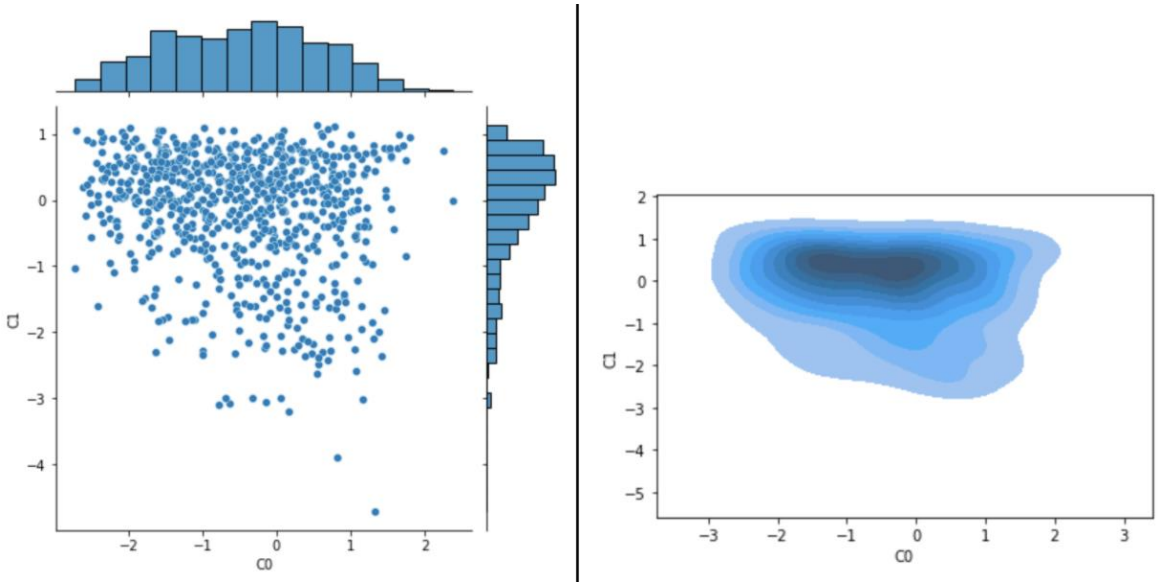
Using content characteristics to model content

Content can be represented in an *m*-dimensional ($m \geq 2$) content-complexity space, the dimensions of which comprise, e.g., spatial complexity (texture), temporal complexity (time-correlation); two-dimensional complexity space derived from rate-distortion curves; etc. The PDFs $p(x)$ and $q(x)$ can be approximated from samples, e.g., a scatter plot, using a kernel density estimator.



(a) **(b)**

Fig. 1 (a) Content complexity of a large video corpus represented as a scatter plot of spatial complexity (X-axis, C0) and temporal complexity (Y-axis, C1). (b) The corresponding PDF $p(x)$, where x is a two-dimensional vector with indices C0 and C1, generated using kernel density estimation



(a) **(b)**

Fig. 2 (a) Content complexity of a small dataset represented as a scatter plot of spatial complexity (X-axis, C0) and temporal complexity (Y-axis, C1). (b) The corresponding PDF $q(x)$, where x is a two-dimensional vector with indices C0 and C1, generated using kernel density estimation

For example, Figures 1 and 2 illustrate the use of a kernel density estimator to estimate the PDFs $p(x)$ and $q(x)$ from samples, e.g., scatter plots, of their respective two-dimensional content representation space. As mentioned before, the PDF $q(x)$ of the smaller dataset can be relatively stationary, whereas the PDF $p(x)$ of the large corpus can vary with uploaded content, e.g., the PDF $p(x)$ of Fig. 1(b) is in effect a snapshot at a particular time of a changing distribution.

While content in a given category can have some similarity in content characteristics, different content items can in fact have differing weights (densities) in differing regions of content-complexity space. Further, perceptual studies indicate that content with similar content characteristics is rated similarly by human raters, *even if the content is from different categories*. For this reason, it can be better to represent content in an m-dimensional content-complexity space than by category.

Aside from spatial and temporal correlation of the video and categorical classification of the video, other examples of x, the content representation of the video, include the input quality in bits per frame, a video encoder parameter of interest, a deep-learned embedding of the video, or in general, any feature of the video that can impact a performance metric of interest. For example, higher spatial and/or temporal complexity costs more bits to encode and is hence correlated with the performance metric of encoded bitrate.

Example Workflow

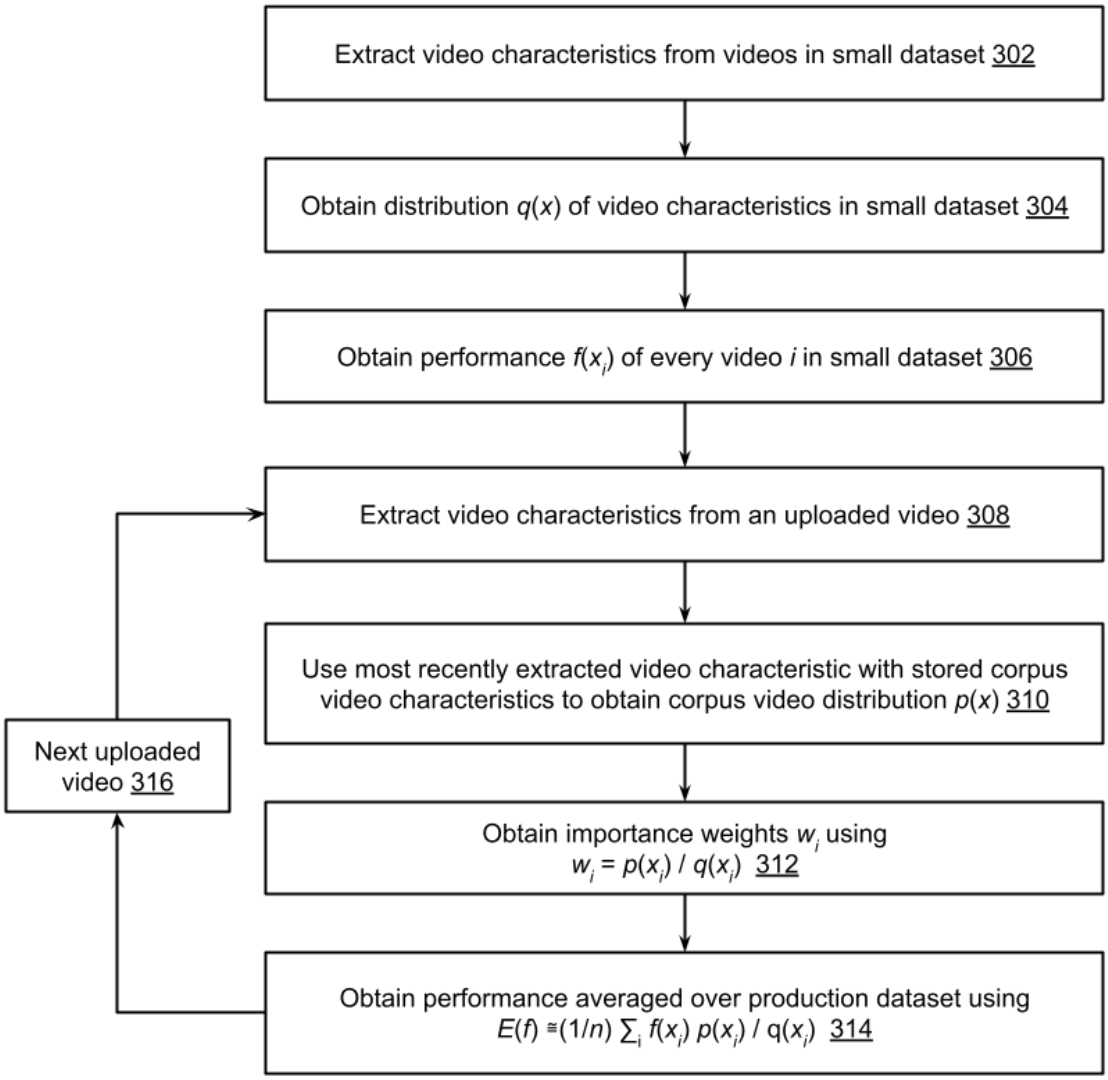
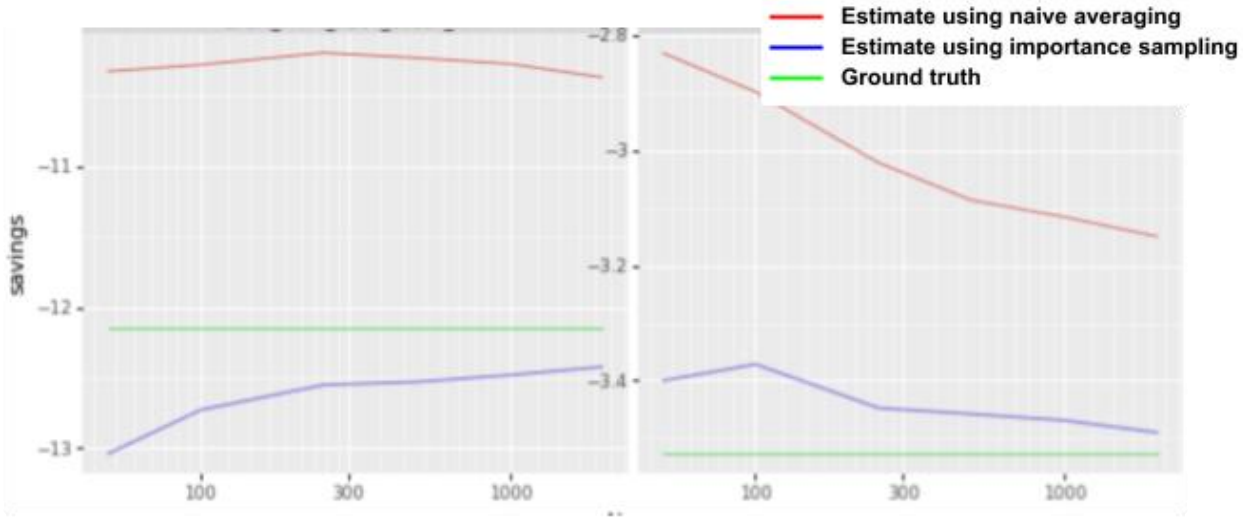


Fig. 3: An example workflow to determine average performance over a large-scale video corpus from a smaller dataset

Fig. 3 illustrates an example workflow to determine average performance of a transcoder over a large-scale video corpus, e.g., of videos uploaded by users of a video hosting and sharing service, from a smaller dataset. Video characteristics are extracted for videos in a small dataset (302). A distribution $q(x)$ of video characteristics in the small dataset is obtained (304) using, e.g., kernel density estimation as explained above.

The performance $f(x_i)$ of the transcoder under evaluation is obtained of every video i in the small dataset (306). Some example performance metrics include encoded bitrates, reconstructed quality measurements such as peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), video multimethod assessment fusion (VMAF), etc. A service in the video pipeline extracts video characteristics from every (or randomly sampled) uploaded video in the corpus (308).

The distribution $p(x)$ of video characteristics over the corpus is estimated using both the recently extracted video characteristic and the thus-far obtained and stored corpus video characteristics (310). For each video i in the small dataset, we estimate its importance weight $w_i = p(x_i) / q(x_i)$ using the distributions $p(x)$ of the corpus and $q(x)$ of the small dataset (312). Weighted averages of evaluation metrics over the small dataset are the estimated metrics against the video corpus (314). The procedure is repeated for the next uploaded video (316) which may be selected by randomization.



(a) **(b)**

Fig. 4: Estimate of average bitrate savings (percentage) for two transcoders

Fig. 4 illustrates that for two transcoders, the estimate of a certain performance metric, the average bitrate savings, converges faster with increasing sample size to its ground truth when computed using importance sampling rather than naive averaging.

In this manner, the techniques of this disclosure leverage importance sampling to assign a weight for each video in a small video dataset. The performance of a transcoder for a larger corpus of uploaded videos is estimated using a weighted average of the performance of the transcoder for videos in the small video dataset. As compared to naive averaging over the small dataset, performance metrics estimated using the described weighted averaging are closer to the true performance when the transcoder is utilized for the larger video corpus.

Performance evaluation can be tailored towards storage or streaming costs. While the described techniques apply directly to storage, for streaming, each video may have different contributions to the network traffic. For example, popular videos may be viewed millions of times, while some videos may have a very small number of views. In this case, the video characteristics distribution can be sampled from network traffic to yield another set of weights.

The techniques can be applied to automatic performance tracking, in which weightings are sampled and updated to trigger re-optimization of encoders, either periodically or when the change to the video corpus meets a threshold.

CONCLUSION

This disclosure describes the use of importance sampling in the evaluation of video transcoders using a small dataset of videos. The techniques can deliver a high-performance transcoder even when the transcoder is optimized using a small dataset that is insufficiently representative of a large-scale video corpus.

REFERENCES

[1] “Importance sampling” available online at https://en.wikipedia.org/wiki/Importance_sampling accessed on June 29, 2021.

[2] https://statweb.stanford.edu/~owen/mc/Ch-var-is.pdf accessed on June 29, 2021.

[3] “An introduction to importance sampling” available online at https://www.youtube.com/watch?v=V8f8ueBc9sY&t=385s accessed on June 29, 2021.

[4] “P.910 : Subjective video quality assessment methods for multimedia applications” available online at https://www.itu.int/rec/T-REC-P.910-200804-I accessed on June 29, 2021.