April 2020

MONETIZATION ABUSE DETECTION BASED ON CO-WATCH SIMILARITY

Luca Chiarandini

Lukasz Heldt

MONETIZATION ABUSE DETECTION BASED ON CO-WATCH SIMILARITY

Content sharing platforms allow users to access and consume media content, as well as allow users to store and share media content with other users. The media content may include video content, audio content, other content, or a combination thereof. The content may include content from professional content creators (e.g., movies, television clips, music, etc.) as well as content from amateur content creators (e.g., video blogging, short original videos, etc.).

Content sharing platforms may provide users with one or more channels. A channel owner can customize their channel(s) by uploading or linking media content (e.g., videos) to the channel. Other users can visit the channel of the channel owner, view videos hosted by the channel, and express opinions on the videos via, for example, likes, dislikes, comments, etc. Content sharing platforms can also track how many users view each video, which are expressed as "views" hereafter.

Content sharing platforms may provide channel owners with monetization opportunities based on videos uploaded to their channel. For example, the channel owner may be allowed to enable ads on their videos and generate income via ad-based revenue streams. These monetization opportunities may be associated with one or more requirements and/or policies, such as, for example, displaying appropriate content, subscriber count, popularity, etc. If a channel of a channel owner satisfies predefined requirements and/or policies of the content sharing platform, the channel owner may receive access to monetization features (ad-based revenue streams, subscription fees, sale of merchandise, etc.) for the respective channel. A

channel that is associated with enabled monetization features is referred to herein as a "monetized channel."

However, some channel owners may exploit the policies related to a monetized channel. For example, a policy related to a monetized channel may require a user not to upload inappropriate content. Inappropriate content may include indecent/sexually suggestive videos, inappropriate children's content, abuse videos, etc. To exploit this policy, a user may create a channel, upload appropriate content until the channel meets the requirements and policies to become a monetized channel, and then upload inappropriate content that no longer meets said policies. As a consequence, advertisers are unintentionally sponsoring content that may be deemed inappropriate by the standards of the content sharing platform.

Aspects of the present disclosure address the above and other deficiencies by providing a system capable of detecting monetization abuse by channels of content sharing platforms. The system disclosed herein may use machine learning methods to generate a predictive model capable of being applied to channels to determine a likelihood that the channel is engaged in monetization abuse. Specifically, the disclosed system may train a machine learning model using a plurality of training videos. The training may be based on generating clusters of videos based on a likelihood of videos being watched together, and identifying which clusters are risky. The machine learning model may then be applied to existing and new channels to detect monetization abuse (e.g., channels uploading inappropriate content) and flag potentially abusive channels for administrative review.

Figure 1 illustrates a flow diagram of a method for monetization abuse detection. The method may be performed on a computer system, hereafter "system." At block 102, the system generates a plurality of video clusters. Specifically, the system first receives, as input, a plurality of training videos. The plurality of training videos may be videos previously uploaded by users of a content sharing platform and stored in a repository, or a database, etc. hosted by the content sharing platform. Each of the plurality of training videos may include related video data. The video data may include metadata related to each training video, (such as a title, a description, entity extraction data, etc.), metadata related to a channel associated with the training video (e.g., channel name, channel identification, etc.), data related to user viewing history (e.g., average duration the video was viewed (a watch-time), an amount of times video was viewed, which user(s) viewed the video, etc.), and other such data.

The system may assign each training video to the one or more clusters. The clusters may be based on any type of elements shared by two or more videos (e.g., viewed by the same user(s), a type of content, etc.). A likelihood of similarity related to the elements can be estimated using video metadata and video viewing history. The clusters may be generated using any type of clustering method, such as, for example, *k*-means clustering. It should be understood that a single training video can be assigned to multiple clusters.

k-means clustering is a method of vector quantization which partitions *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

Voronoi cells are regions which, for each object in a set, consist of all points on a plane closer to the object than to any other object.

The system may further assign a relevance weight to each training video in each cluster, where the relevance weight relates to a relevance of the training video to its assigned cluster. For example, the relevance weight can indicate how similar a particular video is to other videos in the cluster the particular video is assigned to.

At block 104, the system trains a prediction model using clustered training videos and cluster labels. Cluster labels may, for example, identify some clusters as risky, and may be defined using viewing histories and data relating to an administrative review. The administrative review can include a human generated labeling regarding whether a channel meets a monetization policy, e.g., whether the channel is eligible to generate income via ad-based revenue streams. Specifically, the administrative process may indicate channels that were rejected due to violation of the monetization policy. By way of example, the system can determine whether a cluster is risky using Equation 1, expressed below:

$$risk(cluster) = \sum_{channel \in C} I(channel) \sum_{video \in channel} I(video \mid channel)\, I(cluster \mid video)$$

Equation 1

where: I (channel) = an importance of the channel per-se;

I (video | channel) = an importance of the video to the channel;

I (cluster | video) = an importance of the cluster to the video.

In particular, the system may first determine an importance of a cluster to a training video (i.e., *I* (cluster | video)) by using the relevance weight of the training video (e.g., a weight indicating how similar the video is to other videos in the cluster) to generate an importance score between 0 and 1. For example, the system may use the importance score of the video to determine whether the video is closely associated with a cluster, far removed from the cluster, or within any range in between.

Next, the system determines an importance of the training video to the channel (i.e., *I* (video | channel)). For example, the system may determine the significance of the training video to the channel for revenue earning (e.g., most watched video, etc.) In a first example, the importance of the training video to the channel is determined based on formula 1/N, where N is the amount of videos uploaded to a channel prior to the channel being labeled as in violation of a monetization policy. If the channel was not labeled as being in violation of the monetization policy, *I* (video | channel) is set to 0. It should be understood that N can be different for different channels, and videos uploaded prior to the channel being labeled as in violation of a monetization policy may be heterogeneous and not all violative.

In a second example, the importance of the training video to the channel is determined based on whether the training video was the last video uploaded to the channel prior to the channel being labeled as in violation of a monetization policy. If the channel was not the last video uploaded prior to the channel being labeled as in violation of a monetization policy, *I* (video | channel) is set to 0. It should be noted that with this example, although the last video has a high likelihood of being a “policy abusive” video, using one video per channel may contain too

little information to train an adequate mathematical model (e.g., a predictive model) for supervised learning.

In a third example, the importance of the training video to the channel is determined based on a weighted watch-time of the video. An advantage of the method of this example is that videos with higher watch-times have a higher chance of being "abusive" videos. Another advantage of the method of this example is that the watch-time of a video adapts to changes in the video consumption from users. However, it should be noted that the method of this example can experience a cold start, where it may take time to catch new trends since users need to watch the videos to accumulate watch-times.

In a fourth example, the importance of the training video to the channel is determined based on a weighted view amount for each video. Similar to the third example, an advantage of the method of this example is that videos with the higher weighted view amounts have a higher chance of being "abusive" videos, and that the weighted view amount of a video adapts to changes in the video consumption from users. However, the method of this example can also experience a cold start.

Lastly, the system determines an importance of the channel per se (i.e., I (channel)). The importance of the channel per se can be related to the channel's popularity, collective watch-time, subscriber count, etc. Accordingly, each training video votes on risky clusters based on their importance. The system then weighs the votes of each training video of a cluster by the importance of the training video to the channel. Lastly, the contributions of each of the channels

are weighted by their importance. By calculating the weighted votes, the system can build a model from the ensemble of training videos to make predictions.

At block 106, the system flags possibly abusive channels for review using the prediction model. Abusive channels may be those that upload risky content. For example, the system can determine a risk score of a channel using Equation 2, expressed below:

$$risk(channel) = \sum_{video \in channel} I(video \mid channel) \sum_{cluster \in video} I(cluster \mid video)\, risk(cluster)$$

Equation 2

where: *risk* (channel) = a risk score of a channel..

The system generates a risk score (i.e., *risk* (channel)) for each of a plurality of channels on a content sharing platforms. The risk score indicates a likelihood of the channel being "risky." The channel can be a new channel, or an existing channel. The system can apply the predictive model to data associated with the channel and obtain output of the prediction model, which indicates whether the video belongs to a cluster that is considered risky, and which can be used in Equation 2 to determine a risk score of the channel. The system then sorts the channels based on their respective risk score. The channels with a risk score above a threshold, within a top percentage of the plurality of channels, or any combination thereof, can be enqueued for administrative review. Alternatively, the channels with a risk score above a threshold or within a top percentage of the plurality of channels can be automatically labeled as in violation of a policy, and reprimanded.

ABSTRACT

A system for detecting monetization abuse by channels of content sharing networks is disclosed. The proposed system uses machine learning methods to generate a predictive model capable of being applied to channels to determine a likelihood that the channel is engaged in monetization abuse. Specifically, the proposed system may train a machine learning classifier using a plurality of training videos. The training may be based on generating clusters of videos and identifying which clusters are risky. The machine learning classifier may then be applied to existing and new channels to detect monetization abuse (e.g., channels uploading inappropriate content) and flag the potentially abusive channels for administrative review.

Keywords: machine learning, monetization abuse, clusters, channels, content sharing networks, training, classifier, k-means clustering, predictive model, policy violation, administrative review, inappropriate content, co-watch similarity

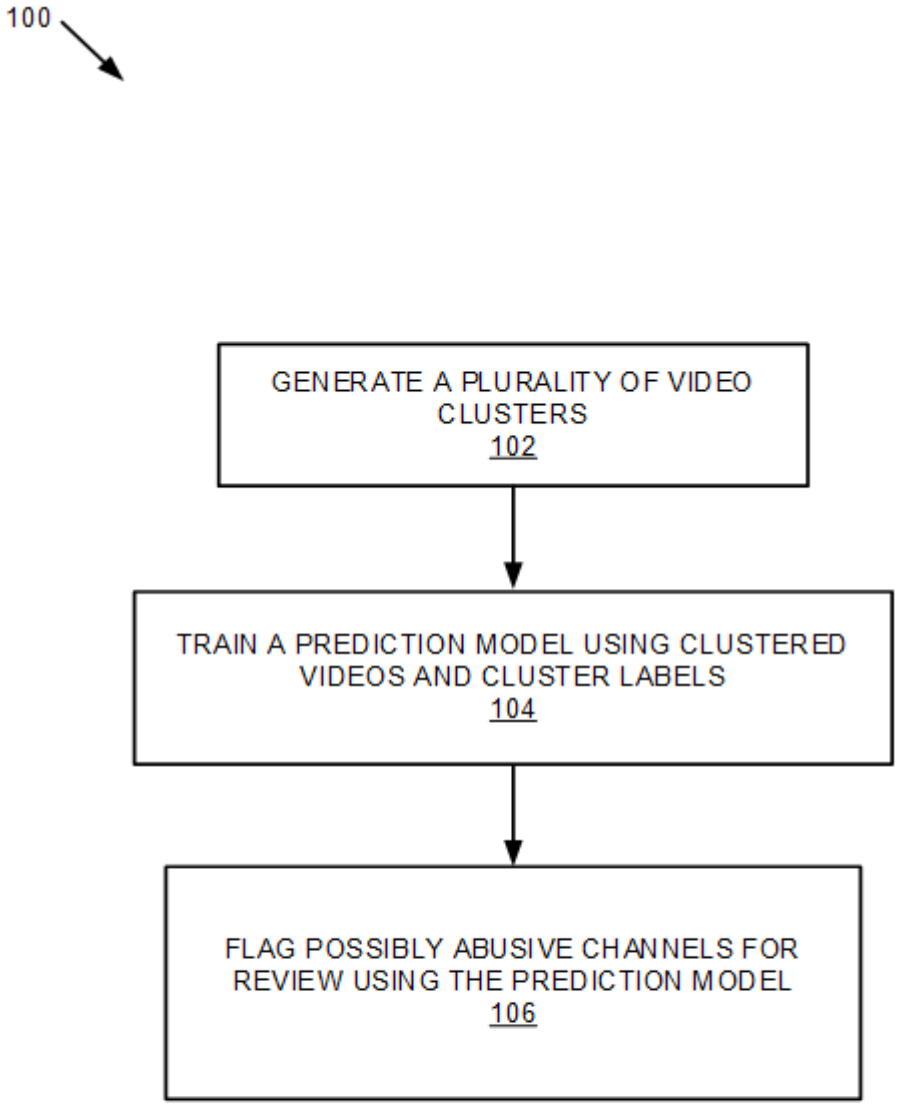
100
GENERATE A PLURALITY OF VIDEO CLUSTERS
102
TRAIN A PREDICTION MODEL USING CLUSTERED VIDEOS AND CLUSTER LABELS
104
FLAG POSSIBLY ABUSIVE CHANNELS FOR REVIEW USING THE PREDICTION MODEL
106

FIG. 1