Predicting Popularity of YouTube Video
Midway Report

Anonymous Author(s)
Affiliation
Address
email

1 Revised Plans

As a result of our feedback on the proposal, as well as a meeting with Anthony, we have changed the scope of our project significantlythough we ensured that we could still use our web-crawler data, since the crawler has been running for weeks, now.

Our previous plan was to generate a pandora-radio-like playlist based on a user input seed video, where the list was supposed to appeal to a viewer who enjoyed the seed video. However, since viewer opinion is highly subjective and since evaluating our results would be extremely expensive, we found that this was no practical. As well, were we to merely cluster videos by similarity and evaluate our clustering without considering user enjoyment, the only easily available labeling we could use to evaluate is the set of YouTube-recommended links for each videoand if that is our standard of success, we would at best be mimicking an existing functionality, but with far less data to help us.

Our new goal is to make predictions concerning the popularity (number of views, percentage of likes, and percentage of dislikes) of a youtube video given the rest of its metadata this is both simpler and more easily evaluated than our old goal, and yet should still involve plenty of machine learning and make use of topics covered during the course.

2 Evaluation Metric

2.1 Two Types of Predictors

We will learn two types of predictors for each of our three outputs (number of views, percentage of likes, and percentage of dislikes): The first will predict these values directly, and the second will take a pair of videos and predict which will be more popular. The comparing of videos could be done indirectly by comparing the predicted values, but we are interested to see whether directly training a comparison predictor will yield more accurate results or not.

2.2 Comapring Order-of-Magnitude

It is important to decide what we will consider as being ”close to correct”. 0-1 loss is sufficient for our comparison-based version, and for our predictors of the percentage of likes and dislikes, we can simply consider our loss in terms of the square of the difference between our prediction and the true value. When predicting the number of views, however, we must deal with the gigantic variance in our observed data.

Ideally, we wish to consider orders of magnitude rather than direct counts, and for this we will set our loss function equal to the square of the difference between the log of our prediction and the log of the observed value. The motivation is that we wish to reflect the human intuition that there is more difference between the popularities of two videos with 10 and 1,000 views (respectively) than between two videos with 1,000,000 and 1,001,000. This will prevent petty among between the most popular videos from drowning out the differences in all others.

The decision to use a log-number-of-views-based loss function may be reversed at a later time if we find better results without it.

Currently we are using a linear regression to predict the number of views, and we deal with the number of views in log scale for the regression. One anticipated effect of this is that features will be expected to contribute multiplicitively, rather than additively, to the popularity of a video. While this certainly seems interesting, it is something that we may ultimately change our minds about as we hone our results to finer detail. As our model grows more sophisticated than mere linear regression, it may be possible to achieve multiplicitive effects when needed even while working directly with the number of views rather than its log.

2.3 Stretch Goals

We certainly hope to attempt increasingly sophisticated learning techniques to reduce our loss as much as possible. Time allowing, there are also other interesting results we can persue. Chief among these in our mind is to make more time-dependent predictions. It may be, for example, that video A is very popular at first, but that video B maintains it's popularity better over time, eventually overtaking video A in terms of the numbers of views and of likes.

3 Data Set

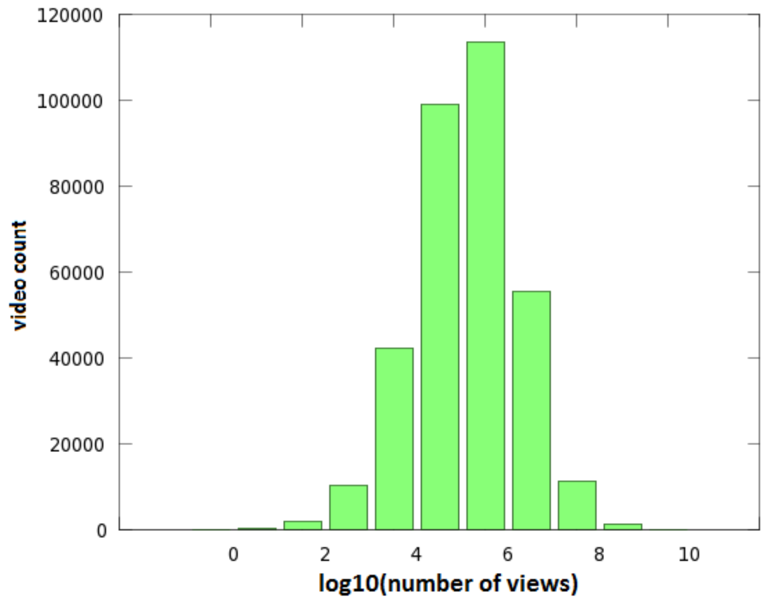
We have, for several weeks, been collecting data by crawling YouTube. We initialize the crawler with a (hopefully) random ”seed” video, and it recursively explores all other videos that YouTube suggests as being related to that vide. Periodically, we restart the crawler over with a new seed video, to ensure a broader sampling. (We began that practice relatively late, so some video categories are more fully explored than others). For each video, we grab the title, uploader, description, upload date, number of views/likes/dislikes, video length, and a number of other attributes, as well as the list of the first 40 videos YouTube recommends as being similar.

There is one additional piece of information that we decided our crawler should collect, which still needs to be added: the number of subscribers for each user. Since this was not needed for our original proposal, we will need to go back and gather this information, which may take some time considering the vast quantity of videos crawled.

3.1 Preliminary Data Statistics

Our crawler continues to work, but thus far we have gathered a decent amount of data:

- Number of videos crawled: 335,373
- Number of uploaders: 146,655
- Most viewed video: 2,107,560,304 views.
- Most ”liked” video: 8,647,905 likes.
- Most ”disliked” video: 4,184,459 dislikes.
- Size of ”bag of words” dictionary produced for the titles: 129,553 entries.



4 First Steps Taken

The data gathering required significant time, and is a big part of the project, but apart from this we had to change most of our plans regarding first steps. Therefore, what we will present in this report is the results of our first attempt to predict the popularity of a video. For this, we have considered the data from just five days of crawling, and we have reduced the number of fields considered to the title, uploader, video length, and upload date. We consider only the number-predicting version, not the direct-comparison-prediction version.

4.1 Technique Details

The first step is to build a dictionary mapping the uploader to the number of videos they have uploaded and the total number of views there videos have. We also take care to prevent ”cheating”: In order to ensure that our predictor has only such information as would be abailiable before the video’s publishing is ever used, we temporarily reduce thse number of video-views and the total number of video uploads for the uploader according to the publish date of the video under current consideration.

We train a linear regression model for each of our three outputs on the following features:

- Many features extracted via a bag-of-words model on the title, using Tf-idf.
- The # of videos uploaded by the uploader prior to the current video’s upload date.
- The total # of views for an uploader due to videos released prior to the current video’s upload date.
- The fraction of the previous two features (the average number of views per video for videos uploaded by the same uploader prior to the current video’s upload date).
- The runtime of the video, in seconds.
- The age of the video at the time of crawling, in days.

We then randomly select 80

5 Results

6 Next Steps

7 Related work