User Generated Content (UGC)

Any form of content created by users of a system or service and made available publicly on that system

Type:

Websites: Entertainment media publications include Reddit, 9Gag, 4chan, Upworthy

Video Games: mods, fan patches, etc.
    Some games involved user-generated world: mincraft

Retailers: bargain hunting websites: eBay

Education: Wikipedia.

Live-video streaming: Youtube, Twitch, Steam, etc

Etc
What is it about?

Karine Pires
Gwendal Simon

trace

Two User-Generated Streaming System Services

2014.01-2014.04

Data

Observation:
1. Both generate traffic with frequent peaks at more than 1 Tbps.
2. Popularity of channels is more heterogeneous than other platforms
## DATASET

**Original Data**

Fetch the following information at every 5 minutes

<table>
<thead>
<tr>
<th></th>
<th>Twitch</th>
<th>YouTube</th>
</tr>
</thead>
<tbody>
<tr>
<td>channel id</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>session id</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>nb. of viewers</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>video bitrate</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>video resolution</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>uploader country</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>

**session**: time when the uploader is online, broadcasting live video

**inter-session**: uploader is offline
Data Visualization

show the data in figure or table for analyzing

nb. of viewers

Figure 1: A life in a channel
Pre-Processing
Filtering Out the “Actual Uploaders” vs “Testers”
1. Uploader who launched a channel for only one session with less than ten minutes
2. Channels from uploaders have no viewers at all

<table>
<thead>
<tr>
<th></th>
<th>Twitch</th>
<th>YouTube</th>
</tr>
</thead>
<tbody>
<tr>
<td>total nb. of channels</td>
<td>1,536,492</td>
<td>236,957</td>
</tr>
<tr>
<td>total nb. of sessions</td>
<td>6,242,609</td>
<td>737,233</td>
</tr>
<tr>
<td>10 min. channels</td>
<td>25%</td>
<td>27%</td>
</tr>
<tr>
<td>no viewers</td>
<td>11%</td>
<td>40%</td>
</tr>
<tr>
<td>filtered nb. of channels</td>
<td>1,068,138 (69%)</td>
<td>120,097 (51%)</td>
</tr>
<tr>
<td>filtered nb. of sessions</td>
<td>5,221,208 (83%)</td>
<td>527,677 (71%)</td>
</tr>
</tbody>
</table>

Table 2: Filtering testers from the traces

Testers may do harm to delivery infrastructure on the uplink. UGC live system should be able to detect and prevent testers from harming the serverce.
Size of these systems

- Overall bandwidth
- Number of concurrent sessions
- Number of different channels
Overall bandwidth

- Both peaks more than 1Tbps
- Twitch has higher peaks (1.6 Tbps)

Figure 2: Bandwidth usage for live video delivery
Number of Concurrent Sessions

- Youtube: 300-700
- Twitch: more than 6000

Figure 3: Number of simultaneous online channels
Number of Different Channels

- Twitch has far more living channels
- Seems the numbers are still increasing
Diurnal & weekly patterns

Figure 5: Average number and confidence interval of simultaneous online channels by hour and weekday
Twitch is less sensitive

- Service starts earlier, uploaders from the whole world
- Twitch is more related to e-sport, which is popular in Asia, so it is more balanced during the day & week
Zipf's law

\[ f(k; s, N) = \frac{1/k^s}{\sum_{n=1}^{N} (1/n^s)} \]

Larger \( s \) (\( \alpha \)), fewer channels have high popularity.

Probability mass function on log-log scale
Both systems follow zipf’s law

- Both youtube and twitch follows zipf’s law, NRMSD < 5%.

- Youtube has larger $\alpha$ values, and not that stable.

Figure 6: Zipf $\alpha$ coefficient evolution over time
Dataset Usage

1. Forecast of popular Channels

Goal: identify the most popular channels as early as possible.

Selected characteristics of Channels: the length of their sessions, the interval between sessions and the # of sessions.

Preliminary Method: compare the distribution with existing top 1% popular channels
Dataset Usage: Forecast of popular Channels (some results)

(a) YouTube

(b) Twitch
Dataset Usage

2. study of delivery methods on live streaming services

Goal: to reduce the delivery bandwidth cost and increase QoE (quality of experience) of viewers

Method:
- Treat as management problem
- Design strategies for deciding which online channels should be delivered by adaptive

Resulting paper:
Dataset Usage (other work)

3. evaluation of scenarios that are based on different service providers competitors. In other word check the content delivery network (CDN)’s fairness.

--------------------------------

4. ??? What’s more in data mining

<table>
<thead>
<tr>
<th></th>
<th>Twitch</th>
<th>YouTube</th>
</tr>
</thead>
<tbody>
<tr>
<td>channel id</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>session id</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>nb. of viewers</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>video bitrate</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>video resolution</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>uploader country</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>
Discussion and suggestion

What is a good dataset and a dataset paper.
Discussion

1. Better have more the attributes for channels:
   For example:
   - The key word or content for the video
   - Viewers’ information (with privacy protection)
   - Comments under stream

2. It’s unfair to compare Youtube live stream and Twich stream. (different domain)

3.
4.
5.
6. More
Thank you!
Project Update

1. Finish sampling by using Steam API
   Sampling strategies using currently

   a. Start with a list of seed steamID selected from different country (famous steam user). Get their friends list and expanding until the networks size reach the maximize value(10,000 now)

   b. Start with a list of seed steamID selected from US (famous steam user, but playing different types of games). Get their friends list and expanding until the networks size reach the maximize value(10,000 now)
# Project Update

2. finish building attribute network

<table>
<thead>
<tr>
<th>ID</th>
<th>Recently_played_games</th>
<th>recently played time</th>
<th>User Countrycode</th>
<th>User StateCode</th>
</tr>
</thead>
</table>
| 123234843   | {"total_count": 12, "games": [
|             | "appid": 256620, "name": "SteamVR", "playtime_seconds": 1095, "playtime_forever": 1641, "leg_icon_url": "7f286d0c2990501602b0e0a505e21c1f7500729", "leg_logo_url": "d7f7e18ecc3c87f0d6de0f77cc2b9f8e6f60" |
|             | "appid": 60579, "name": "Slay the Spire", "playtime_seconds": 652, "playtime_forever": 1141, "leg_icon_url": "38e811d20c0b8a0c70515490b348a9f51413ca", "leg_logo_url": "d0bf74cbb2e4fe7d7d090bda3e34f3e3e36e05103" |
|             | "appid": 777169, "name": "Fallout 4", "playtime_seconds": 108, "playtime_forever": 113, "leg_icon_url": "77f8dadeba3a2af267c0f6dbb503dfb72c07f", "leg_logo_url": "8977ae0ffbab3c0ef005ebf7db6e245555e" |
|             | ]}                                                                                   | 9802 sec             | US               | WA             |
Project Update

3. Next week

a. run community detection algorithm

b. analize attribute distribution

c. Add weight to the network