Reconstructing proprietary video streaming algorithms

Maximilian Grüner, Melissa Licciardello, Ankit Singla
Roadmap

• Introduction
Roadmap

• Introduction
• Reconstruction
Roadmap

• Introduction
• Reconstruction
• Experiments
Roadmap

- Introduction
- Reconstruction
- Experiments
- Limitations and future work
Roadmap

• Introduction

• Reconstruction

• Experiments

• Limitations and future work

• Takeaways
Streaming in Industry

Share of global internet traffic %

Categories

[Sandvine 2019]
Streaming in Industry

Share of global internet traffic %

Categories

Sandvine 2019
Streaming in Industry

23.8% Various, 23.1% Netflix, 12.7% YouTube, 4.2% Twitch…

[Sandvine 2019]
Streaming in academia

# Google Scholar citations

Year


0 5K 10K 15K 20K 25K
Streaming in academia

# Google Scholar citations

Year

- 1998
- 2000
- 2001
- 2002
- 2003
- 2004
- 2005
- 2006
- 2007
- 2008
- 2009
- 2010
- 2011
- 2012
- 2013
- 2014
- 2015
- 2016
- 2017
- 2018
- 2019
- 2020

# Google Scholar citations
Applications of reconstruction
Applications of reconstruction
Applications of reconstruction
Applications of reconstruction
Introduction into ABRs

Played

Resolution

Throughput

[Video from YouTube]
Introduction into ABRs

Played

Resolution

Throughput

[Video from YouTube]
Introduction into ABRs

Played

Resolution

Throughput

[Video from YouTube]
Introduction into ABRs

- Played
- Resolution
- Throughput

[Video from YouTube]
Introduction into ABRs

- Played
- Resolution
- Throughput

[Video from YouTube]
Introduction into ABRs

- Played
- Resolution
- Throughput

[Video from YouTube]
def choose_next_quality(throughput_history):
    throughput_estimate = throughput_history[-1]
    if throughput_estimate == low:
        next_quality = ↓↓
    else:  # throughput_estimate == high
        next_quality = ↑↑

    return next_quality
def choose_next_quality(throughput_history):
    throughput_estimate = throughput_history[-1]
    if throughput_estimate == low:
        next_quality = ↓↓
    else:  # throughput_estimate == high
        next_quality = ↑↑
    return next_quality
def choose_next_quality(throughput_history):
    throughput_estimate = throughput_history[-1]
    if throughput_estimate == low:
        next_quality = ↓↓
    else:  # throughput_estimate == high
        next_quality = ↑↑
    return next_quality
def choose_next_quality(throughput_history):
    throughput_estimate = throughput_history[-1]
    if throughput_estimate == low:
        next_quality = ↓↓
    else:  # throughput_estimate == high
        next_quality = ↑↑
    return next_quality
def choose_next_quality(throughput_history):
    throughput_estimate = throughput_history[-1]
    if throughput_estimate == low:
        next_quality = ⊶状
    else: # throughput_estimate == high
        next_quality = ♂♂
    return next_quality
Simple ABR as a tree

`throughput_estimate == low`

- **False**
  - `next_quality = ↓↓`

- **True**
  - `next_quality = ↑↑`
def choose_next_quality(throughput_history):
    throughput_estimate = np.mean(throughput_history[-2:]):
    if throughput_estimate == low:
        next_quality = ↓↓
    else:  # throughput_estimate == high
        next_quality = ↑↑
    return next_quality
(Slightly more) complex ABR in code

```python
def choose_next_quality(throughput_history):
    throughput_estimate = np.mean(throughput_history[-2:])
    if throughput_estimate == low:
        next_quality = ↓↓
    else:
        next_quality = ↑↑
    return next_quality
```
Why do we need primitives
Why do we need primitives
Why do we need primitives

2 Rules

$\text{throughput}_{t-1}$

$\text{throughput}_t$
Why do we need primitives

5 Rules

\[ \text{throughput}_{t-1} \]

\[ \text{throughput}_t \]
Why do we need primitives

throughput_{t-1}

throughput_t

25 Rules
Why do we need primitives

$\text{throughput}_{t-1}$

$\text{throughput}_t$

100 Rules
def choose_next_quality(throughput_history, buffer_size):
    throughput_estimate = np.mean(throughput_history[-2:])
    if throughput_estimate == low:
        download_time = chunk_size_high / throughput_estimate
        if download_time < buffer_size:
            next_quality = ↓↓
    else:
        next_quality = ↑↑
    else:
        next_quality = ↑↑
    return next_quality
def choose_next_quality(throughput_history, buffer_size):
    throughput_estimate = np.mean(throughput_history[-2:])
    if throughput_estimate == low:
        download_time = chunk_size_high / throughput_estimate
        if download_time < buffer_size:
            next_quality = ↓↓
    else:
        next_quality = ↑↑
    else:  # throughput_estimate == high
        next_quality = ↑↑
    return next_quality
def choose_next_quality(throughput_history, buffer_size):
    throughput_estimate = np.mean(throughput_history[2:])
    if throughput_estimate == low:
        download_time = chunk_size_high / throughput_estimate
        if download_time < buffer_size:
            next_quality = ↓↓
    else:
        next_quality = ↑↑
    else:  # throughput_estimate == high
        next_quality = ↑↑
    return next_quality
Number of primitives
Number of primitives

Features
Number of primitives

Features × Lookback
Number of primitives

Features × Lookback × Actions
Number of primitives

Features $\times$ Lookback $\times$ Actions

$\sim 1300$ Primitives
## Services of interest

<table>
<thead>
<tr>
<th>Provider</th>
<th>Description</th>
<th>Alexa Rank</th>
</tr>
</thead>
</table>


## Services of interest

<table>
<thead>
<tr>
<th>Provider</th>
<th>Description</th>
<th>Alexa Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>YouTube</td>
<td>Broad coverage</td>
<td>2, Global</td>
</tr>
</tbody>
</table>
## Services of interest

<table>
<thead>
<tr>
<th>Provider</th>
<th>Description</th>
<th>Alexa Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>YouTube</td>
<td>Broad coverage</td>
<td>2, Global</td>
</tr>
<tr>
<td>ZDF</td>
<td>German Public Service</td>
<td>47, Germany</td>
</tr>
</tbody>
</table>
## Services of interest

<table>
<thead>
<tr>
<th>Provider</th>
<th>Description</th>
<th>Alexa Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>YouTube</td>
<td>Broad coverage</td>
<td>2, Global</td>
</tr>
<tr>
<td>ZDF</td>
<td>German Public Service</td>
<td>47, Germany</td>
</tr>
<tr>
<td>Pornhub</td>
<td>Pornographic video sharing website</td>
<td>46, Global</td>
</tr>
</tbody>
</table>
# Services of interest

<table>
<thead>
<tr>
<th>Provider</th>
<th>Description</th>
<th>Alexa Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>YouTube</td>
<td>Broad coverage</td>
<td>2, Global</td>
</tr>
<tr>
<td>ZDF</td>
<td>German Public Service</td>
<td>47, Germany</td>
</tr>
<tr>
<td>Pornhub</td>
<td>Pornographic video sharing website</td>
<td>46, Global</td>
</tr>
<tr>
<td>Arte</td>
<td>French-German, cultural</td>
<td>270, France</td>
</tr>
<tr>
<td>Fandom</td>
<td>Gaming, pop-culture</td>
<td>91, Global</td>
</tr>
<tr>
<td>SRF</td>
<td>Swiss Public Service</td>
<td>45, Switzerland</td>
</tr>
<tr>
<td>TubiTV</td>
<td>Movies and series of all genres</td>
<td>1330, USA</td>
</tr>
<tr>
<td>Twitch</td>
<td>Live and VoD streaming service, gaming</td>
<td>39, Global</td>
</tr>
<tr>
<td>Vimeo</td>
<td>Artistic content</td>
<td>188, Global</td>
</tr>
<tr>
<td>XVideos</td>
<td>Pornographic video sharing website</td>
<td>67, Global</td>
</tr>
</tbody>
</table>
Inference features
Inference features

Providers
Inference features

Providers $\times$ Net-traces
Inference features

Providers $\times$ Net-traces $\times$ Videos
Inference features

Providers $\times$ Net-traces $\times$ Videos $\times$ Length
Inference features

Providers × Net-traces × Videos × Length

∼ 9 hours of training/testing streaming time
Why is agreement not sufficient
Why is agreement not sufficient

Played

Resolution

Throughput

[Video from YouTube]
Why is agreement not sufficient

Played

Resolution

Throughput

[Video from YouTube]
Why is agreement not sufficient

[Video from YouTube]
Why is agreement not sufficient

[Video from YouTube]
Why is agreement not sufficient

[Video from YouTube]
Agreement
Agreement

- **true positives**
- **false negatives**
- **true negatives**
- **false positives**

**selected elements**
Agreement

relevent elements

false negatives  true negatives

true positives  false positives

[Select elements]
Agreement
Agreement

relevant elements

false negatives
true negatives

ture positives
false positives

selected elements
Agreement
Agreement

relevant elements

false negatives
true negatives

true positives
false positives

selected elements

How many selected items are relevant?
How many relevant items are selected?

Precision =
Recall =
Agreement

relevant elements

false negatives  true negatives

true positives  false positives

selected elements

How many selected items are relevant?

How many relevant items are selected?

Precision = 

Recall = 

[Wikipedia]
Agreement

\[ F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \]
Similarity
Similarity

Scikit-learn
Similarity
Similarity

[Scikit-learn]
$\text{Score}_{\text{Similarity}} = \mathcal{P}(\text{Decision from reference distribution})$
General metric analysis
General metric analysis

- Black Box

- Rate

- Rule-Set

$F^1_{macro}$

Score$_{similarity}$

Score$_{similarity}$

Score$_{similarity}$
General metric analysis

$F_{macro}^1$ vs. $Score_{similarity}$ for Black Box, Rate, and Rule-Set categories.
General metric analysis

\[ F^{1}_{macro} \]

\begin{align*}
\text{Worst} & \quad 0.0 & \quad 0.2 & \quad 0.4 & \quad 0.6 & \quad 0.8 & \quad 1.0 \\
\text{Best} & \quad 0.5 & \quad 1.0 \\
\text{Score}_{Similarity} & \quad 0.5 & \quad 1.0 \\
\text{Rate} & \quad 0.5 & \quad 1.0 \\
\text{Rule-Set} & \quad 0.5 & \quad 1.0 \\
\end{align*}
General metric analysis

![Graph showing metric analysis for different video platforms.](image)
General metric analysis
General metric analysis

![Graph showing metric analysis for different platforms including Arte, Fandom, Pornhub, MPC, NN, TubiTV, XVideos, YouTube, Twitch, Vimeo, SRF, ZDF. The graphs show the relationship between similarity scores and macro F1 scores for Black Box, Rate, and Rule-Set scenarios.]
General metric analysis
General metric analysis

![Graph showing F₁ macro vs Score Similarity for different platforms: Arte, Fandom, Pornhub, MPC, NN, Pornhub, SRF, TubiTV, Vimeo, XVideos, YouTube, ZDF. The graphs are labeled as Black Box, Rate, and Rule-Set.](image-url)
Perceptual metric analysis
Perceptual metric analysis

\[ \mu_{\text{VMAF switches per second}} \]

\[ \mu_{\text{Rebuffer per second}} \]
Perceptual metric analysis
Perceptual metric analysis

μ\text{VMAF switches per second} vs μ\text{Rebuffer per second}

μ\text{VMAF per second} vs μ\text{VMAF per second}
Perceptual metric analysis

\[ \mu_{VMAF \text{ switches per second}} \]

\[ \mu_{\text{Rebuffer per second}} \]
Perceptual metric analysis

\[ \mu_{VMAF \text{ switches per second}} \]

\[ \mu_{Rebuffer \text{ per second}} \]
Perceptual metric analysis

\[ \mu_{VMAF\ \text{switches per second}} \]

\[ \mu_{Rebuffer\ per\ second} \]
Perceptual metric analysis

\[ \mu_{\text{VMAF switches per second}} \]

\[ \mu_{\text{Rebuffer per second}} \]
Perceptual metric analysis
Perceptual metric analysis

μ_{VMAF switches per second} vs. μ_{Rebuffer per second}
Perceptual metric analysis

\[ \mu_{VMAF \text{ switches per second}} \]

\[ \mu_{VMAF \text{ per second}} \]

\[ \mu_{Rebuffer \text{ per second}} \]
Comparing industry with academia
Comparing industry with academia
Comparing industry with academia
Analysing the simplified SRF tree
Analysing the simplified SRF tree
Analysing the simplified SRF tree

\[ R_{\text{relative}} (\bullet, \mathbb{I}^+, \mathbb{P}^+, \mathbb{W}^-) < 0.6 \]
Analysing the simplified SRF tree

\[ R_{\text{relative}} (\bullet, T^+, P^+, W^-) < 0.6 \]

Startup Completed

Buffer \((\bullet, T^-) < 0.847\)

True

False

Startup Completed

Buffer \((\bullet, T^-) < 0.847\)
Analysing the simplified SRF tree

\[ R_{\text{relative}} (\bullet, T^+, P^+, W^-) < 0.6 \]

\[ R_{\text{absolute}} (\bullet, T^+, P^+, W^-) < -0.049 \]

Buffer \( (\bullet, T^-) < 0.847 \)

Startup Completed
- False
- True

Startup beyond 50%
Analysing the simplified SRF tree

\( R_{\text{relative}} (\bullet, \mathbb{T}^+, \mathcal{P}^+, \mathcal{W}^\sim) < 0.6 \)

\( R_{\text{absolute}} (\bullet, \mathbb{T}^+, \mathcal{P}^+, \mathcal{W}^\sim) < -0.049 \)

Buffer \( (\bullet, \mathbb{T}^-) < 0.847 \)

Startup Completed

False

True

Startup beyond 50%

Startup beyond 25%

Fetch-time \( (\bullet, \mathbb{T}^+) < 4.5 \)

Startup Completed

Buffer

Startup beyond 50%

Startup beyond 25%

Fetch-time
What usage has interpretability
What usage has interpretability

• Identify potential issues
What usage has interpretability

• Identify potential issues
• Tracing the input-output mapping
Limitations & Future Work
Limitations & Future Work

• Primitives
Limitations & Future Work

• Primitives
• Decision space
Limitations & Future Work

• Primitives
• Decision space
• Interpretability is subjective
Limitations & Future Work

• Primitives
• Decision space
• Interpretability is subjective
• Region
Takeaways
Takeaways

• Decision trees are a great versatile tool
Takeaways

- Decision trees are a great versatile tool
- Interpretability is useful
Takeaways

• Decision trees are a great versatile tool
• Interpretability is useful
• Interpretability is subjective
Takeaways

- Decision trees are a great versatile tool
- Interpretability is useful
- Interpretability is subjective
- Limited by domain knowledge
Melissa Licciardello
melissa.licciardello@inf.ethz.ch
Maximilian Grüner
mgruener@inf.ethz.ch

Melissa Licciardello
melissa.licciardello@inf.ethz.ch

Ankit Singla
ankit.singla@inf.ethz.ch
Ankit Singla
ankit.singla@inf.ethz.ch

Melissa Licciardello
melissa.licciardello@inf.ethz.ch

Maximilian Grüner
mgruener@inf.ethz.ch

https://github.com/magruener/reconstructing-proprietary-video-streaming-algorithms