Understanding In-Video Dropouts and Interaction Peaks in Online Lecture Videos

Juho Kim (MIT CSAIL)
Philip J. Guo (MIT CSAIL, U of Rochester)
Daniel T. Seaton (MIT Office of Digital Learning)
Piotr Mitros (edX)
Krzysztof Z. Gajos (Harvard EECS)
Robert C. Miller (MIT CSAIL)
Video Lectures in MOOCs
Classrooms: rich, natural interaction data
it's like i'm talking to a wall...
How do learners use videos?

Data-Driven Approach:
Analyze learners’ interaction with the video player
Why does data matter?

• detailed understanding of video usage
• design implications for
  – Instructors
  – Video editors
  – Platform designers
• new video interfaces and formats

**Improved video learning experience**
How do learners use videos?

• Watch sequentially
• Pause
• Re-watch
• Skip / Skim
Challenge for instructors/editors

• Don’t know how students use lecture videos
  – Confusion
  – “Aha” moments
  – Bored
  – Re-watching important parts

• MOOC-scale video interaction data
  – Clickstream (play, pause, scrub)
Collective Interaction Traces

Student #7888
Student #7887

......

Student #4
Student #3
Student #2
Student #1

video time
Collective Interaction Traces into Interaction Patterns

second-by-second activity tracking
~40M video interaction events from 4 edX courses

<table>
<thead>
<tr>
<th>Learners</th>
<th>Videos</th>
<th>Mean Video Length</th>
<th>Processed Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>127,839</td>
<td>862</td>
<td>7:46</td>
<td>39.3M</td>
</tr>
</tbody>
</table>

Courses: Computer science, Statistics, Chemistry
Analyzing Clickstream

• Events: play / pause
• In-video time and absolute time
• Learner ID: first-time or re-watching

<table>
<thead>
<tr>
<th>Clickstream interaction log</th>
<th>Per-learner watching segments</th>
<th>Per-second stats for views, re-watches, plays, &amp; pauses</th>
</tr>
</thead>
<tbody>
<tr>
<td>learner xxx</td>
<td>learner xxx</td>
<td></td>
</tr>
<tr>
<td>0:00 play</td>
<td>Segment 1 - 0:00-0:34</td>
<td></td>
</tr>
<tr>
<td>0:34 pause</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0:57 play</td>
<td>Segment 2 - 0:57-1:47</td>
<td></td>
</tr>
<tr>
<td>1:47 pause</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Collective Interaction Patterns

1. In-video dropout

![Graph showing viewership over video time with a drop-off in viewership at a certain point.]

2. Interaction peaks

![Graph showing interaction events over video time with peaks at specific points.]

Collective Interaction Patterns

1. In-video dropout

viewership

video time

video time

2. Interaction peaks

interaction events

video time
1. In-video dropout

2. Interaction peaks
In-video Dropout

% watching sessions that end before the video finishes.
36%: dropout within first few seconds

Why?

- Auto-play playing unwanted videos
- Misleading video titles / interfaces
36% of all sessions were dropouts within the first few seconds.

- 55% dropout sessions
- 36% first few seconds
- 19% rest of the video
- 45% complete sessions

Why?

- Auto-play playing unwanted videos
- Misleading video titles / interfaces
55% of all sessions were dropouts.

- 45% are complete watching sessions
- 19% rest of the video
- 36% first few seconds

66% of dropouts occur within the first few seconds.

- 34% of dropouts occur during the rest of the video.

Why?

- Auto-play playing unwanted videos
- Misleading video titles / interfaces
Longer videos lead to more dropouts.
Re-watching sessions lead to more dropouts than first-time sessions.
**Tutorial videos** lead to more dropouts than **lecture videos**.

**Lecture**
- introduction to concepts
- continuous flow

**Tutorial**
- supplementary examples
- step-by-step demos
1. In-video dropout

2. Interaction peaks
Interaction Peaks

Temporal peaks in the number of interaction events, where a significant number of learners show similar interaction patterns
Two Types of Interaction Peaks

- **Re-watching peak**
  - re-watching session counts

- **Play peak**
  - play button clicks
Anatomy of an Interaction Peak

- Maximum height
- Peak point
- Start point
- Width
- End point
- Area
- Height
- # events vs video time
Analytic Workflow

1. Bin data into per-second segments
2. Apply a kernel smoother
3. Detect peaks

Graph showing time in video (minutes) and number of play events.
Re-watching sessions show stronger and more peaks than first-time sessions.
**Tutorial videos** show stronger and more peaks than **lecture videos**.
What causes an interaction peak?
Observation: **Visual transitions** in the video often coincide with a peak.
Some code

```python
x = int(input('Enter an integer: '))
ans = 0
while ans**3 < x:
    ans = ans + 1
if ans**3 != x:
```

So we'll keep doing it until we find the first one where answer cubed is

number of re-watching sessions

lecture video
Step 1. Visual Analysis

- second-by-second pixel differences between adjacent frames
Analytic Workflow

Step 2. Peak Categorization

• Manually inspected 80 videos
• Interaction peak <-> Visual transition
Comparing Multiple Data Streams
Five Explanations for an Interaction Peak

Type 1. Beginning of new material
Type 2. Returning to content
Type 3. Tutorial step
Type 4. Replaying a segment
Type 5. Non-visual explanation
Type 1. Beginning of new material

before transition

Idea: Admissibility

Admissible (optimistic) heuristics slow down bad plans but never outweigh true costs

Admissible (optimistic) heuristics break optimality by trapping good plans on the fringe

after transition

Admissible Heuristics

A heuristic $h$ is admissible (optimistic) if:

$$0 \leq h(n) \leq h^*(n)$$

where $h^*(n)$ is the true cost to a nearest goal

interaction peak

visual transition
Type 2. Returning to content

before transition

```
def fact(n):
    """assumes that n is an int > 0
    returns n!""
    res = 1
    while n > 1:
        res = res * n
        n = n - 1
    return res

def factR(n):
    """assumes that n is an int > 0
    returns n!""
    if n == 1:
        return n
    return n*factR(n-1)
```

interaction peak

after transition

visual transition
Type 3. Tutorial step

before transition

after transition

interaction peak

visual transition
Type 4. Replaying a segment

before peak

during peak

after peak

interaction peak

visual transition

visual transition
Type 5. Non-visual explanation

**before peak**

**How to Explore?**

- Several schemes for forcing exploration
  - Simplest: random actions ($\epsilon$-greedy)
  - Every time step, flip a coin
  - With (small) probability $\epsilon$, act randomly
  - With (large) probability $1-\epsilon$, act on current policy
- Problems with random actions?
  - You do eventually explore the space, but keep thrashing around once learning is done
  - One solution: lower $\epsilon$ over time
  - Another solution: exploration functions

**after peak**

**How to Explore?**

- Several schemes for forcing exploration
  - Simplest: random actions ($\epsilon$-greedy)
  - Every time step, flip a coin
  - With (small) probability $\epsilon$, act randomly
  - With (large) probability $1-\epsilon$, act on current policy
- Problems with random actions?
  - You do eventually explore the space, but keep thrashing around once learning is done
  - One solution: lower $\epsilon$ over time
  - Another solution: exploration functions

interaction peak

no visual transition
61% of interaction peaks involved a visual transition.

<table>
<thead>
<tr>
<th>Peak Category</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 1. Beginning of new material</td>
<td>25%</td>
</tr>
<tr>
<td>Type 2. Returning to content</td>
<td>23%</td>
</tr>
<tr>
<td>Type 3. Tutorial step</td>
<td>7%</td>
</tr>
<tr>
<td>Type 4. Replaying a segment</td>
<td>6%</td>
</tr>
<tr>
<td>Type 5. Non-visual explanation</td>
<td>39%</td>
</tr>
</tbody>
</table>
Can interaction data improve the video learning experience?
Lessons for instructors, video editors, and platform designers

1. Make shorter videos.
2. Add informative titles and easy navigation.
3. Avoid abrupt visual transitions.
Lessons for instructors, video editors, and platform designers

4. Make interaction peaks more accessible.

5. Enable one-click access for tutorial steps.
Next Steps: More Data Streams

- What would transcript / text add to the analysis? How about acoustic data?

So n length equals 0.
So now, while s length does not equal and then, backslash 0.
So remember, this backslash 0, it is an actual character, and it indicates the end of the string.
Just like, also, backslash n is an actual character.
Backslash 0 is going to indicate the end of our string.
I don't want to put that there.
And while s indexed by length is not equal to the null terminator, then we're just going to increment length.
So then, at the end of our program, length is eventually going to
Next Steps: **Scalability**

- Reliably & automatically detect peak types?
- How much data is needed until we see patterns?

![Graph showing viewership over video time for 5 learners and 5,000 learners](image-url)
For instructors & editors: Video Analytics

Video Heatmap

views  unique viewers  replay  skip  play  pause

0 1,000 2,000 3,000 4,000 5,000 6,000 7,000
0 1:40 3:20 5:00 6:40

Video Duration: 2:52 / 7:57
For learners: **Data-Driven Video UI**

---

**Video**

**Highlights (4)**

1:21 Iterative algorithms
- looping constructs (e.g., while or for loops) lead naturally to iterative algorithms
- Can conceptualize as capturing computation in a set of "state variables" which update on each iteration through the loop

2:56 Iterative multiplication by successive additions
- Imagine we want to perform multiplication by successive additions:
  - Inner loop is a, and so on in rows
  - State variables:
    - multiplication (partial product)
  - updates:
    - result = current value of computation + a
  - loop:
    - result = 0
    - result = result + a

3:25 code snippet-

4:28 Transcript

---

1:22 And those variables update or change their values on each
1:25 iteration through the loop.
1:27 Let's look at an example.
1:29 So here's a simple little example.
1:31 Imagine that we want to do multiplication, and we're
1:34 going to do it just using successive addition.
1:37 Our computer only comes with addition.
1:38 It doesn't come with a built-in multiply.
1:40 I know that's dumb, but it gives us a simple
1:42 example to deal with.
1:44 If we wanted to do that, to do multiplication by successive
1:47 additions, then a simple way to think about it is to say if
1:50 we want to multiply a by b, we're just going to add a to
1:53 itself b times.
1:55 And iteratively, what does that mean?
1:57 Literally, we could say computation's going to be
2:01 captured by two variables.
2:03 One is the iteration number.
2:05 Which time through the loop am I at?
Data-Driven Video UI

Highlights (4)

- My highlights
- Other learners
- Visual transitions
- Topic transitions

1:21 Iterative algorithms

Iterative algorithms

- Loops (e.g., while or for loops) lead naturally to iterative algorithms.
- Can conceptualize as capturing computation in a set of "state variables" which update on each iteration through the loop.

2:56 Iterative multiplication

Iterative multiplication by successive additions

- Imagine we want to perform multiplication by successive additions.
- To multiply a by b, add a to itself b times.
- State variables:
  - $i$: iteration number.
  - $\text{result}_i$: current value of computation.
- Update rules:
  - $i 
  - $\text{result}_{i+1} = \text{result}_i + a$

3:25 Iterative multiplication

def iterMul(a, b):
    result = 0
    while b > 0:
        result += a
        b -= 1
    return result

4:28 Running iterative multipli...
Contributions

• A first MOOC-scale in-video dropout rate analysis
• A first MOOC-scale in-video interaction peak analysis
• Categorization of learner activities responsible for an interaction peak
• Data-driven design implications for video authoring, editing, and interface design
Understanding In-Video Dropouts and Interaction Peaks in Online Lecture Videos

Juho Kim
MIT CSAIL
juhokim@mit.edu
juhokim.com
Backup slides
Domain
- educational videos
  - How-to videos
  - MOOC videos

Theory
- interactive learning
  - Role of interactivity
  - Learner control
  - Subgoal labeling

Method
- scalable data collection to realize theory
  - Crowdsourcing
  - Learnersourcing
Vision in learnersourcing

• Feedback loop between
  – Learners: natural, pedagogically useful activities
  – System: improve interaction using learner data

• Visualize and analyze large-scale video learning activities

• Use data to inform learning platform design
How can we design online video learning platforms that are as effective as in-person classrooms?
Problem 1.
Video doesn’t adapt to learners

• Fixed video & interface once published

• What if video & interface adapt to collective learner behaviors?
Problem 2. Video interaction is limited

• Proven interaction patterns: do they apply to video interaction?
  – Skimming
  – Control+F
  – Copy/Paste, Bookmarking
  – Working with multiple videos, collage
  – Hierarchical structure (h1, h2, div, p, …)
  – Overview + Context
  – Linking to external resources