

Leveraging Video Interaction and Content to Improve Video Learning

Juho Kim (MIT CSAIL)

Shang-Wen (Daniel) Li (MIT CSAIL)

Carrie J. Cai (MIT CSAIL)

Krzysztof Z. Gajos (Harvard EECS)

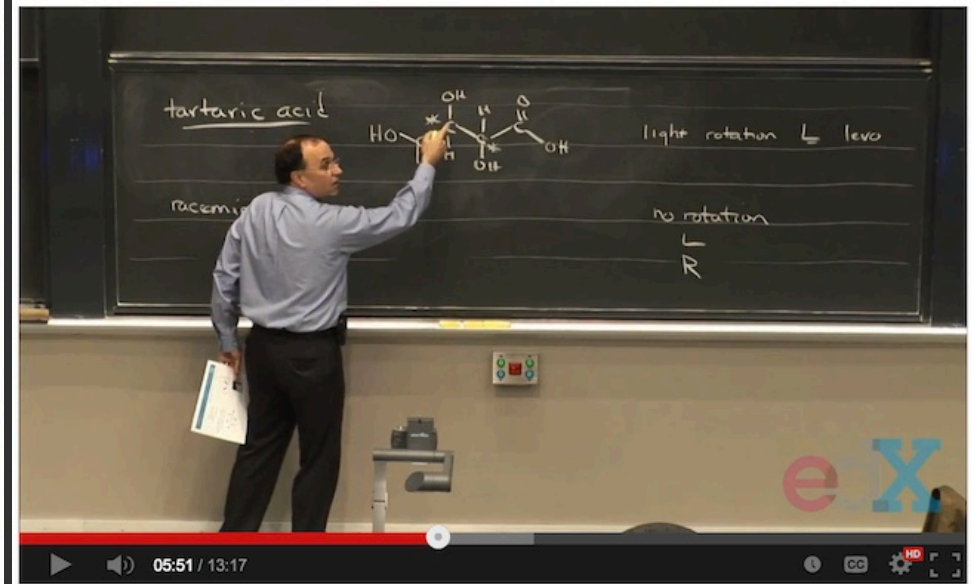
Robert C. Miller (MIT CSAIL)

2014-04-27

CHI 2014 | Workshop on
Learning Innovation at Scale



Video Lectures in MOOCs



```
def findPayment(loan, r, m):
    """Assumes: loan and r are floats, m an int
    Returns the monthly payment for a mortgage of size
    loan at a monthly rate of r for m months"""
    return loan*((r*(1+r)**m)/((1+r)**m - 1))

class Mortgage(object):
    """Abstract class for building different kinds of mortgages"""
    def __init__(self, loan, annRate, months):
        """Create a new mortgage"""
        self.loan = loan
        self.rate = annRate/12.0
        self.months = months
        self.paid = [0.0]
        self.owed = [loan]
        self.payment = findPayment(loan, self.rate, months)
        self.legend = None #description of mortgage

    def makePayment(self):
        """Make a payment"""
        self.paid.append(self.payment)
        reduction = self.payment - self.owed[-1]*self.rate
        self.owed.append(self.owed[-1] - reduction)

    def getTotalPaid(self):
        """Return the total amount paid so far"""
        return sum(self.paid)

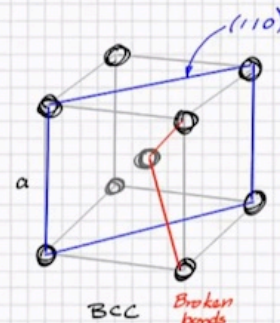
    def __str__(self):
        return self.legend
```

```
Python 2.7.3 [EPD 7.3-2 (32-bit)] (default, Apr 12 2012,
11:28:34)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more
>>>
>>>
>>> ===== RESTART =====
>>>
>>>
```



3.091 Screencast

CALCULATE THE SURFACE ENERGY OF THE (110) SURFACE OF POTASSIUM CRYSTALS



Potassium heat of atomization $89540 \frac{\text{J}}{\text{mole}}$

$$H_a = \frac{89540}{6.02 \times 10^{23}} = 1.49 \times 10^{-19} \frac{\text{J}}{\text{atom}}$$

$$W_{K-K} = \frac{H_a}{8} = \frac{1.49 \times 10^{-19}}{8} = 1.86 \times 10^{-20} \frac{\text{J}}{\text{bond}}$$

$$\text{bond energy required for each atom} = (2 \frac{\text{bonds}}{\text{atom}}) (1.86 \times 10^{-20} \frac{\text{J}}{\text{bond}}) = 3.72 \times 10^{-20} \text{ J}$$



2:27 / 8:40

2:27 / 6:00

Classrooms: rich, natural interaction data



*Maria Fleischmann / Worldbank on Flickr | CC by-nc-nd
Love Krittaya | public domain*

*armgov on Flickr | CC by-nc-sa
unknown author | from pc4all.co.kr*

its like im
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

~40M video interaction events
from 4 edX courses

Learners	Videos	Mean Video Length	Processed Events
127,839	862	7:46	39.3M

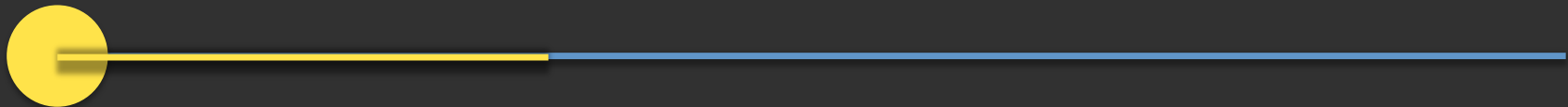
Courses: Computer science, Statistics, Chemistry

How do learners use videos?

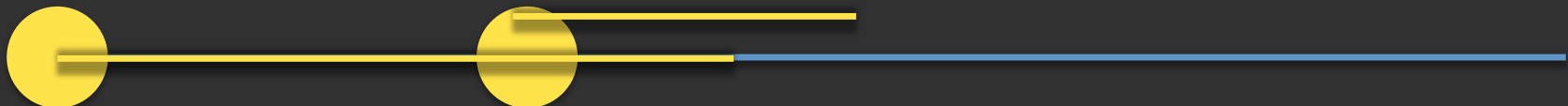
- Watch sequentially



- Pause



- Re-watch



- Skip / Skim

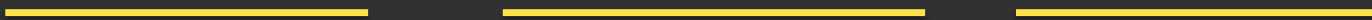


Collective Interaction Traces

Learner #7888



Learner #7887



.....

Learner #4



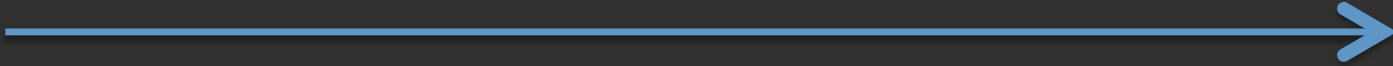
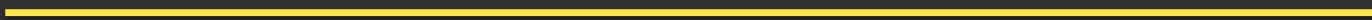
Learner #3



Learner #2

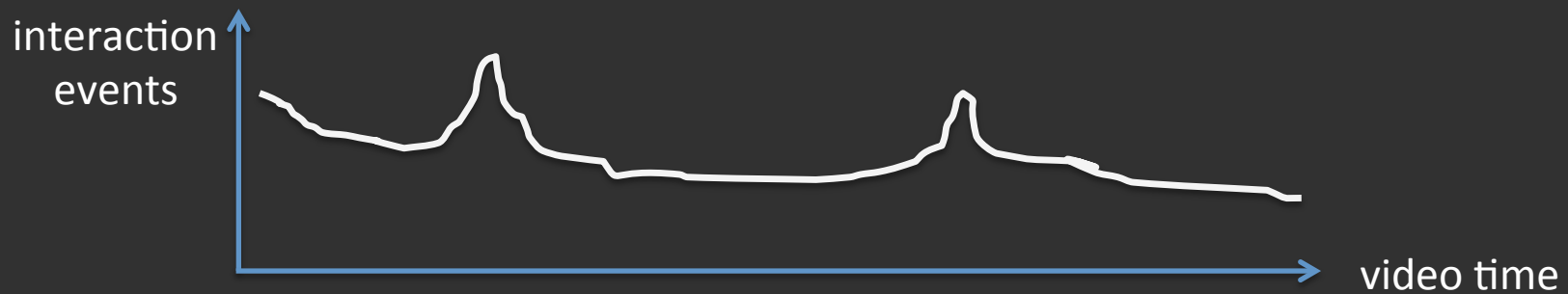


Learner #1



video time

Collective Interaction Traces into **Interaction Patterns**



second-by-second in-video activity

Data-Driven Analysis and Design for Educational Videos

Research Directions: Data-Driven Analysis and Design for Educational Videos

1. Analyze interaction patterns

*scalable and automatic methods to
interpret interaction data*

2. Improve video learning

*video interfaces that adapt to
collective learner interaction patterns*

Research Directions: Data-Driven Analysis and Design for Educational Videos

1. Analyze interaction patterns

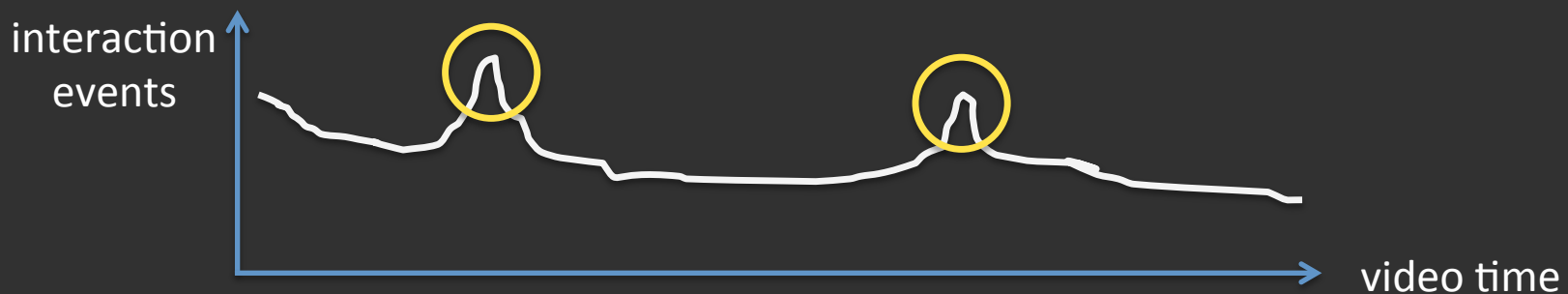
*scalable and automatic methods to
interpret interaction data*

2. Improve video learning

*video interfaces that adapt to
collective learner interaction patterns*

Interaction Peaks

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



Understanding In-Video Dropouts and Interaction Peaks in Online Lecture Videos.
Juho Kim, Philip J. Guo, Daniel T. Seaton, Piotr Mitros, Krzysztof Z. Gajos, Robert C. Miller.
Learning at Scale 2014.

What causes an interaction peak?

Video interaction log data



Video content analysis

- Visual content
- Text from transcript
- Speech & acoustic stream

Observation: **Visual transitions** in the video often coincide with a peak.

```
Python 2.7.3 (v2.7.3:70274d3e1dd, Apr 9 2012, 20:52:43)
[GCC 4.2.1 (Apple Inc. build 5464) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
>>>
>>> iterMul(3, 8)
11
>>> iterMul(26, 35)
910
>>> |
```

edX

And it does it pretty quickly.

4:50 / 5:04



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

edX

We haven't shown out the intermediate stages of the


4:51 / 5:04

Type 1. Returning to content

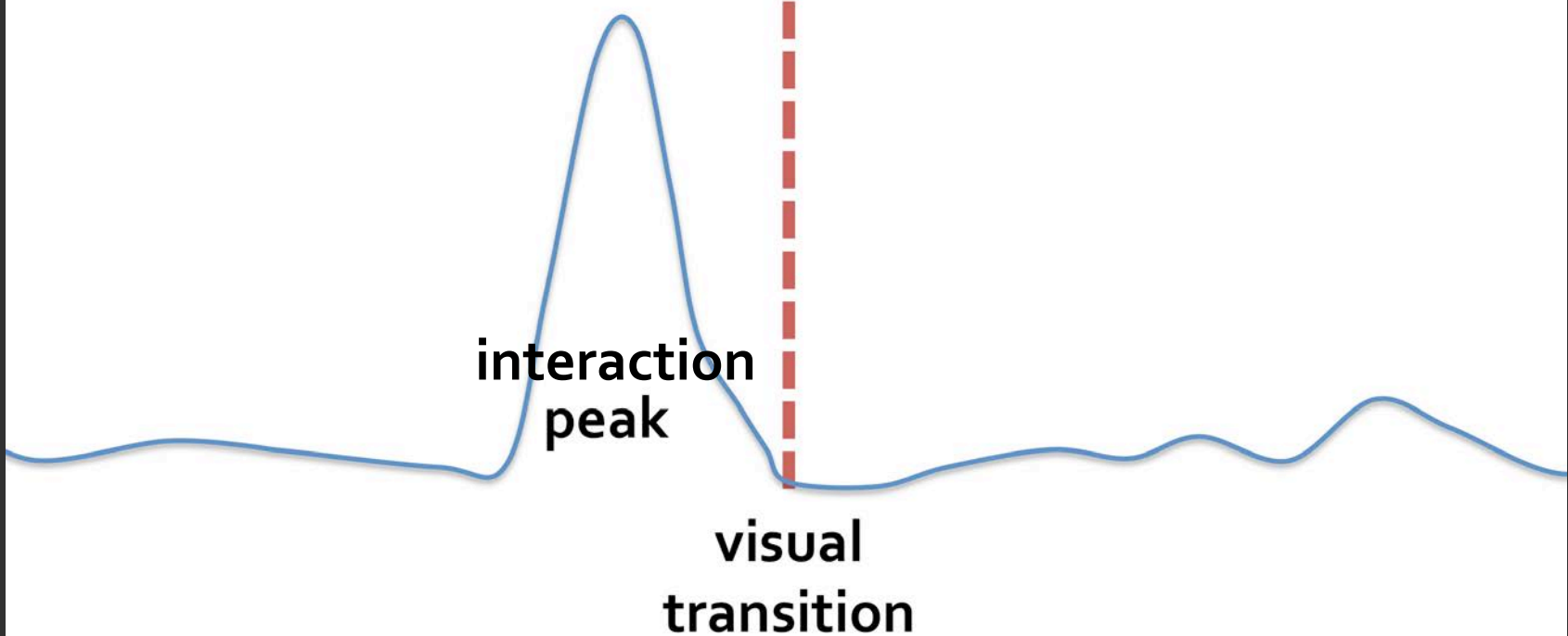
before transition

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

*Handwritten notes: A red circle around 'def fact(n)', a red circle around 'res = 1', a red circle around 'return res', and a red arrow pointing from 'res = res * n' to 'res'.*



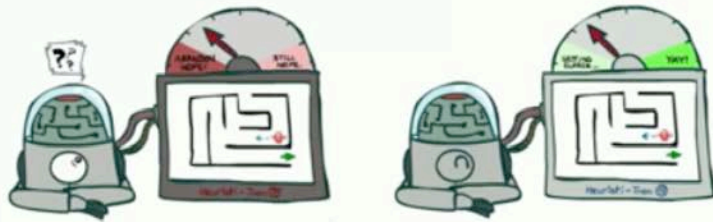
after transition



Type 2. Beginning of new material

before transition

Idea: Admissibility



Inadmissible (pessimistic) heuristics break optimality by trapping good plans on the fringe

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

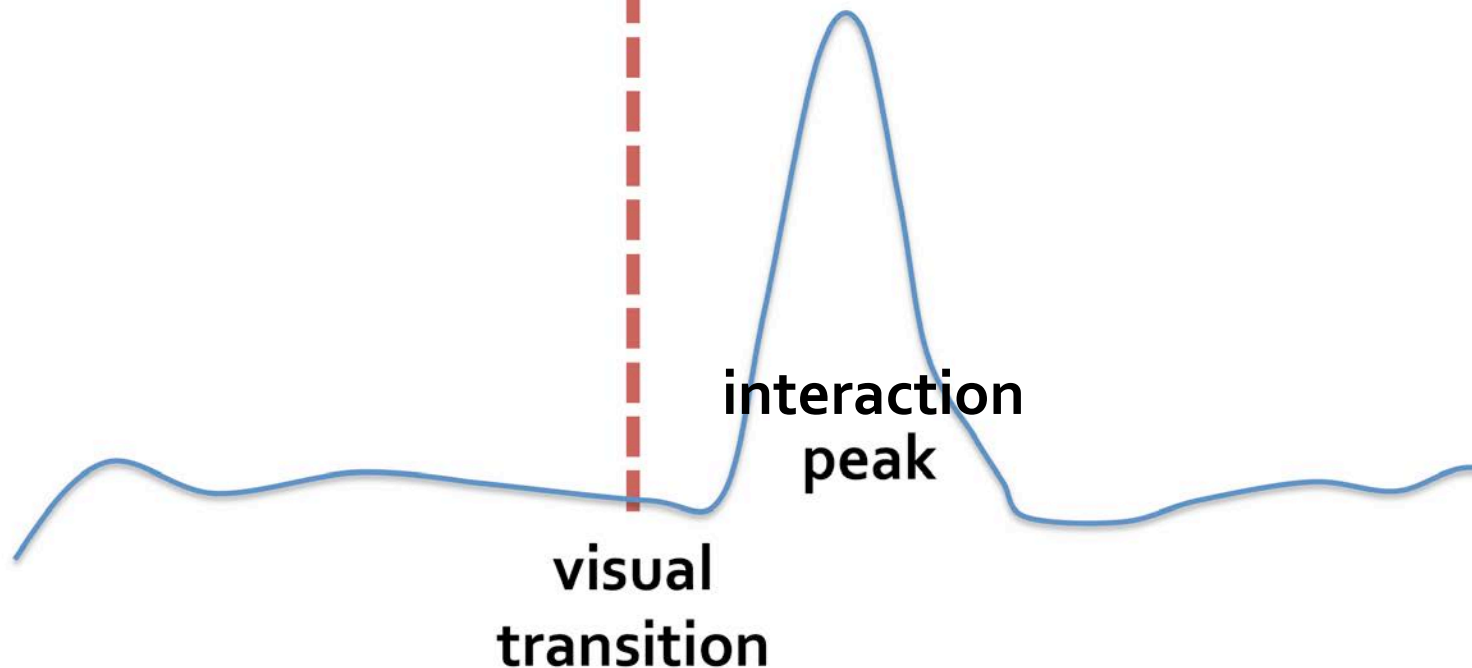
after transition

Admissible Heuristics

- A heuristic h is *admissible* (optimistic) if:

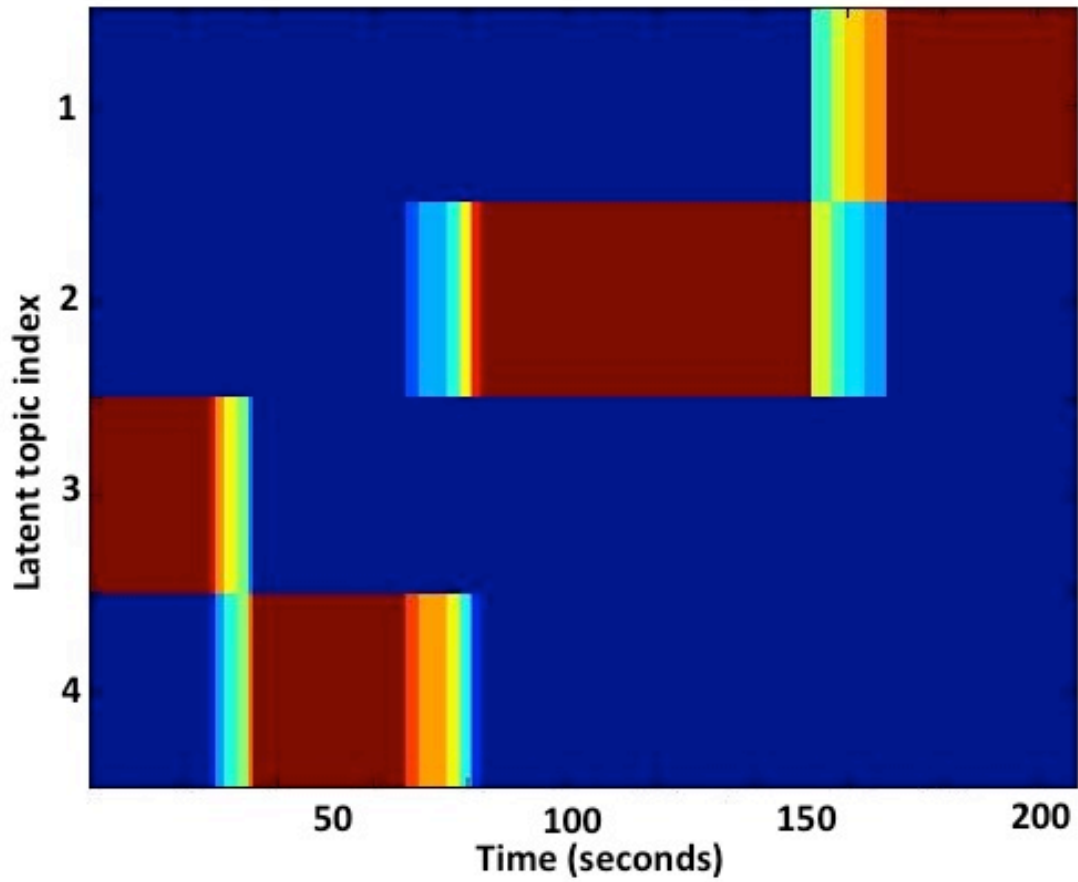
$$0 \leq h(n) \leq h^*(n)$$

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



Text Analysis

- Topic transitions & interaction peaks
 - Topic modeling
- Linguistic patterns & interaction peaks
 - N-gram analysis

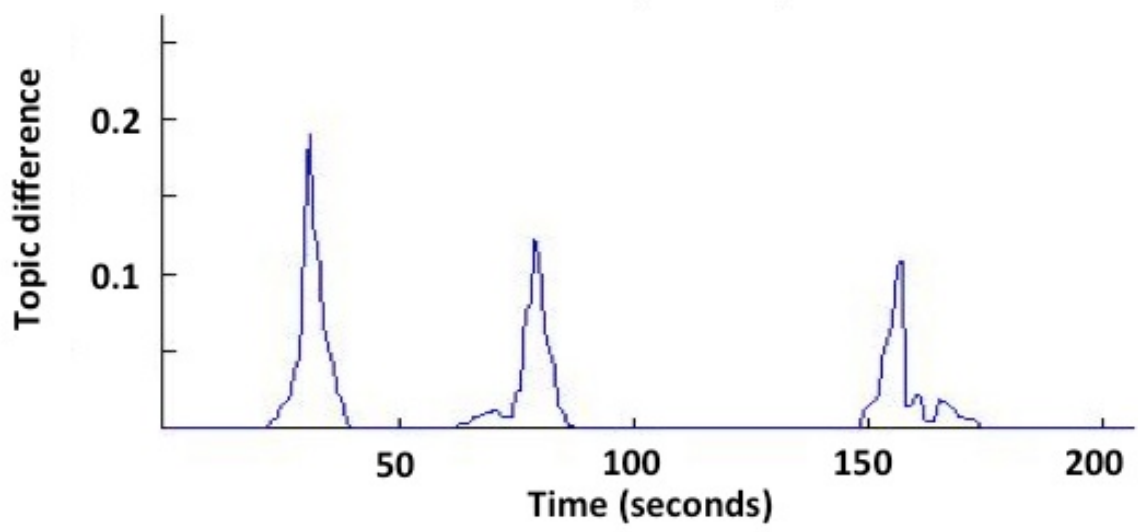


mean

cost, equal, path

node, optimal, goal

expand



topic transition
likelihood over time

N-gram Analysis

“<start-of-sentence> So”

- initiating an explanation

“So let me spend a second on that,” “So that means,”

- arriving at the take-home message

“So we can get lots of information just from these five number summaries.”

N-gram Analysis

“this is”

- visual explanation

“this is the double bonds here on this oxygen”

- naming of a particular concept

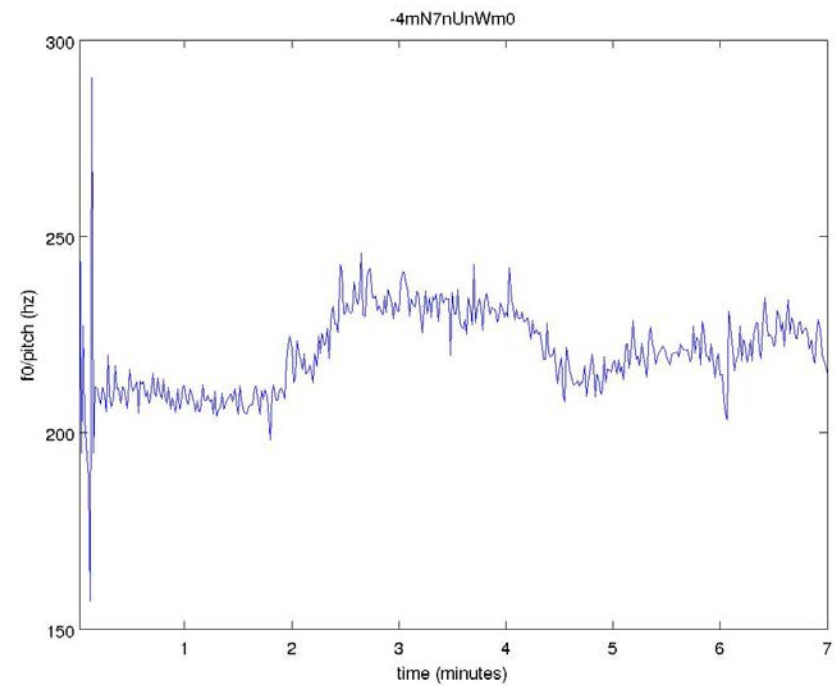
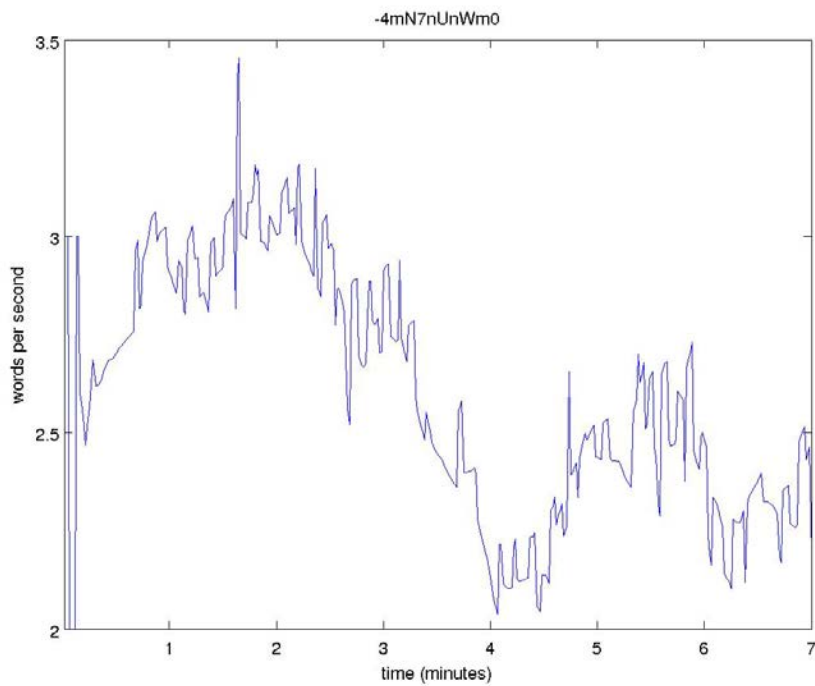
“and this is called a dislocation”,

“so this is sometimes called the first quartile”

Acoustic Analysis

Speaking rate

Pitch

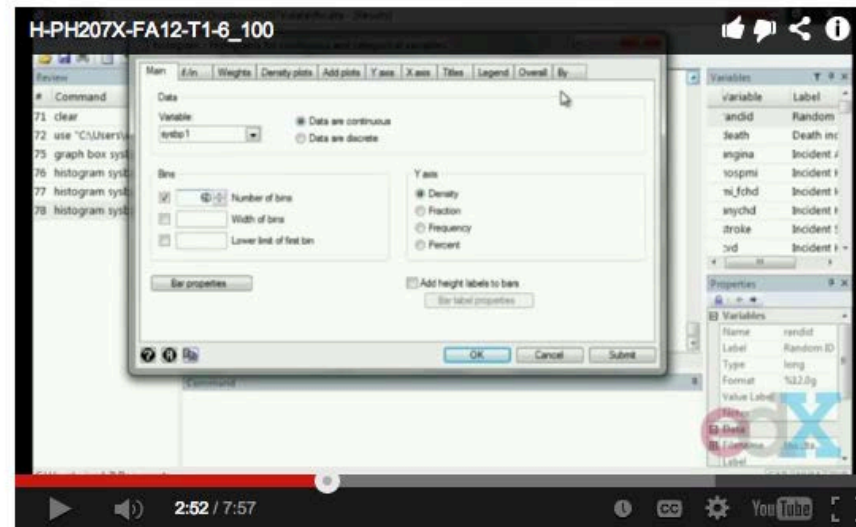


**Automatic, multi-channel, scalable
peak detection and classification**

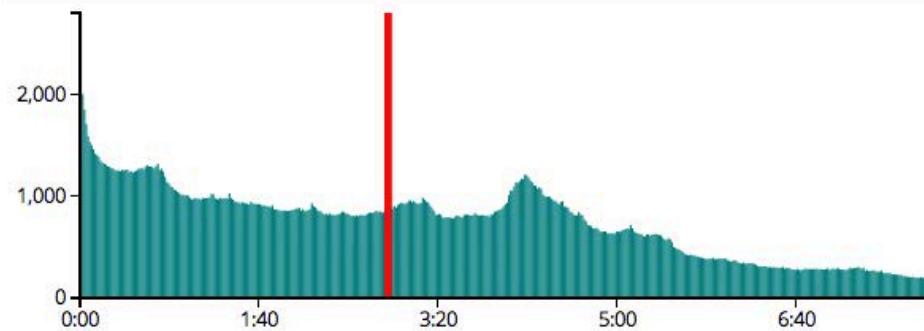
Video Analytics:

“debugging” interface for instructors & editors

Video Heatmap



views unique viewers replay skip play pause



Research Directions: Data-Driven Analysis and Design for Educational Videos

1. Analyze interaction patterns

*scalable and automatic methods to
interpret interaction data*

2. Improve video learning

*video interfaces that adapt to
collective learner interaction patterns*

Data-Driven Interaction Techniques to Support Video Navigation



palindrome
left onward draw
characters
string
palindrome
ispalindrome
isPalindrome

How to solve this recursively?

- First, convert the string to just characters, by stripping out punctuation, and converting upper case to lower case
- Then
 - Base case: a string of length 0 or 1 is a palindrome

[4:06-4:25] You watched this segment.

▶ 1:44 / 7:29 🔊 🔇 Add Bookmark

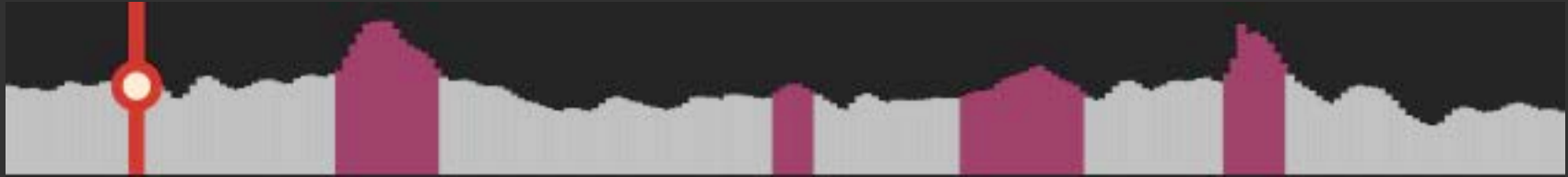
So how do we solve it?
 We first convert the string to just characters.
 We'll look to that in a second.
 And solving it recursively is actually pretty easy.
If I have a string that's either of length zero or of length one, it's a
 palindrome.
 So length one is just one character.
 Otherwise, to solve this, what I'm going to do is take the string and ask the following question.
 If the first and last character are the same, then they satisfy the condition.
 And let me then simply look at the remaining string, throwing away the first and last character, and ask is that a palindrome.
 Wonderful.
 There's that recursive property again.
 If I can break it down into that problem, I'm set.
 So I could write code to do that.
 Just to give you the example again, this says I'm going to take something like "Able was I ere I saw Elba" and reduce it to just that string of characters without the spaces or any punctuation.
 And then, to test whether that string is a palindrome, that's the same as asking are the first and last characters the same?

my bookmarks frequently rewatched by others

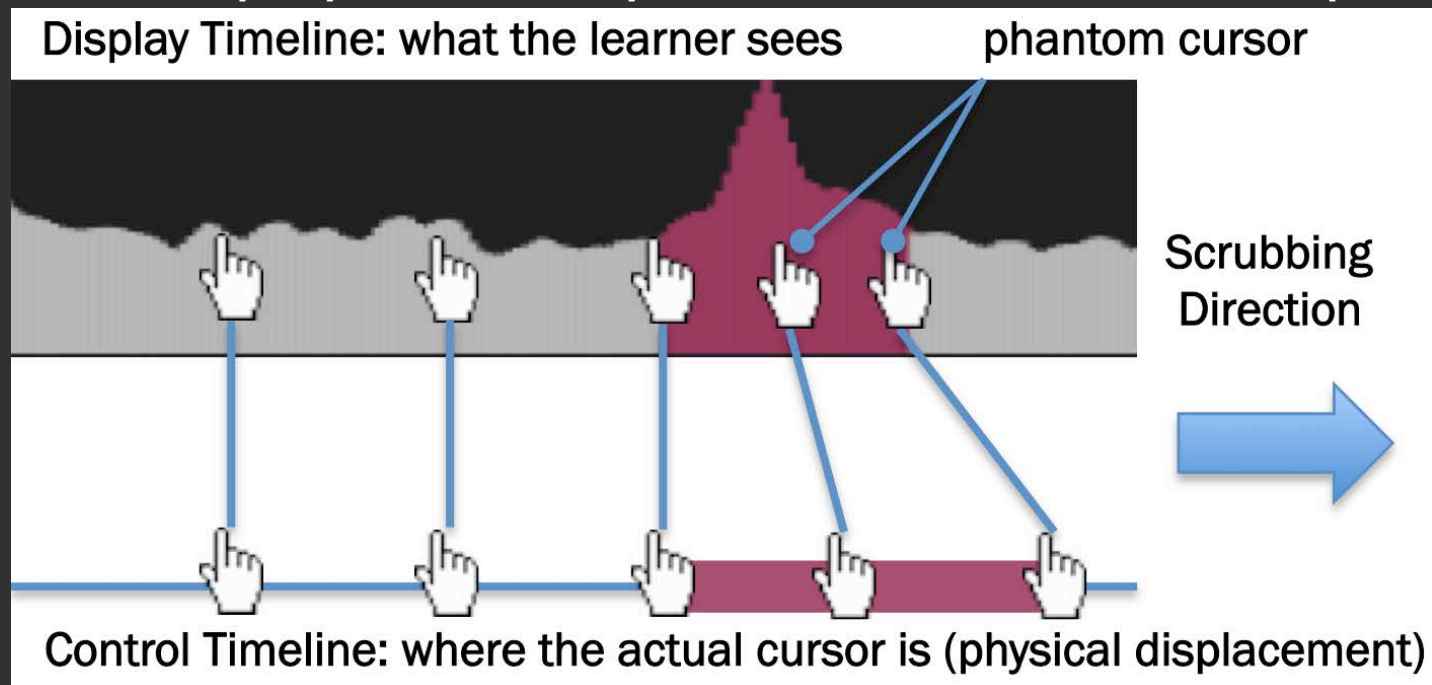
Other learners rewatched this part.	Other learners rewatched this part.	Other learners rewatched this part.	Other learners rewatched this part.
<p style="text-align: center; margin: 0;">Recursion on non-numerics</p> <ul style="list-style-type: none"> • How could we check whether a string of characters is a palindrome, i.e., reads the same forwards and backwards <ul style="list-style-type: none"> – "Able was I ere I saw Elba" – attributed to Napoleon – "Are we not drawn onward, we few, drawn onward to new era?" <div style="text-align: right; margin-top: 10px;"></div>	<p style="text-align: center; margin: 0;">How to solve this recursively?</p> <ul style="list-style-type: none"> • First, convert the string to just characters, by stripping out punctuation, and converting upper case to lower case • Then <ul style="list-style-type: none"> – Base case: a string of length 0 or 1 is a palindrome – Recursive case: <ul style="list-style-type: none"> • if first character matches last character, then is a palindrome if middle section is a palindrome <div style="text-align: right; margin-top: 10px;"></div>	<pre style="font-family: monospace; font-size: small; margin: 0;"> isPalindrome(s) if (s.length() == 0 s.length() == 1) return true; else return (s.charAt(0) == s.charAt(s.length() - 1) && isPalindrome(s.substring(1, s.length() - 1))); </pre> <div style="text-align: right; margin-top: 10px;"></div>	<pre style="font-family: monospace; font-size: small; margin: 0;"> isPalindrome(s) if (s.length() == 0 s.length() == 1) return true; else return (s.charAt(0) == s.charAt(s.length() - 1) && isPalindrome(s.substring(1, s.length() - 1))); </pre> <p style="color: red; font-style: italic; margin-top: 10px;">Internal recursion</p> <div style="text-align: right; margin-top: 10px;"></div>

Rollercoaster Timeline

- Embedded visualization of collective interactions



- Visual & physical emphasis on interaction peaks



Automatic Summarization

- Visual summary with captured highlights

✓ my bookmarks ✓ frequently rewatched by others

Other learners rewatched this part.

Recursion on non-numeric

- How could we check whether a string of characters is a palindrome, i.e., reads the same forwards and backwards
 - “Able was I ere I saw Elba” – attributed to Napoleon
 - “Are we not drawn onward, we few, drawn onward to new era?”



Other learners rewatched this part.

How to solve this recursively?

- First, convert the string to just characters, by stripping out punctuation, and converting upper case to lower case
- Then
 - Base case: a string of length 0 or 1 is a palindrome
 - Recursive case:
 - If first character matches last character, then is a palindrome if middle section is a palindrome



Other learners rewatched this part.

```
def isPalindrome(s):  
    def toChars(s):  
        s = s.lower()  
        ans = ""  
        for c in s:  
            if c in "abcdefghijklmnopqrstuvwxyz":  
                ans = ans + c  
        return ans  
    def isPal(s):  
        if len(s) <= 1:  
            return True  
        else:  
            return s[0] == s[-1] and isPal(s[1:-1])  
    return isPal(toChars(s))
```



Other learners

```
def isPalindrome(s):  
    def toChars(s):  
        s = s.lower()  
        ans = ""  
        for c in s:  
            if c in "abcdefghijklmnopqrstuvwxyz":  
                ans = ans + c  
        return ans  
    def isPal(s):  
        if len(s) <= 1:  
            return True  
        else:  
            return s[0] == s[-1] and isPal(s[1:-1])  
    return isPal(toChars(s))
```

- Keyword summary with word cloud

Iterative
ability

tackling
algorithm

state

variables

property

Iteration
stop

condition

Automatic Side-by-Side View

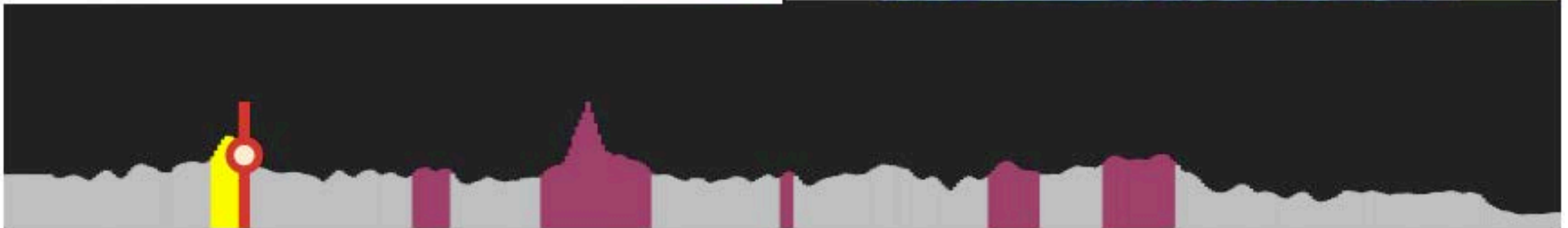
Pinned slide

Video stream



Recursion on non-numeric

- How could we check whether a string of characters is a palindrome, i.e., reads the same forwards and backwards
 - “Able was I ere I saw Elba” – attributed to Napoleon
 - “Are we not drawn onward, we few, drawn onward to new era?”



Lab Study: edX & On-Campus Students

*"It's not like cold-watching.
It feels like watching with other students."*

*"[interaction data] makes it seem more classroom-y,
as in you can compare yourself to
what how other students are learning
and what they need to repeat."*

Rethinking Educational Videos

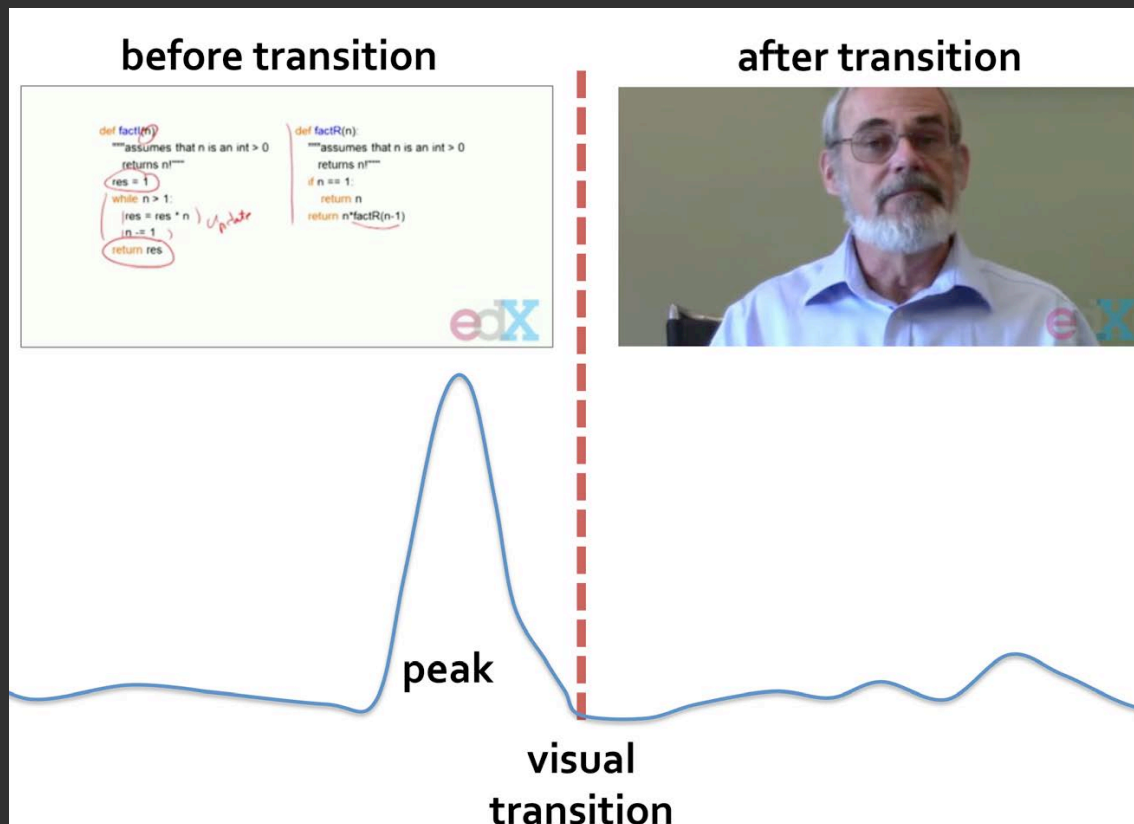
Are behind-the-encoding-wall videos the best format?

- Hard to edit once published
- Only a single stream is published
- Lack of useful metadata (concepts, difficulty...)
- Hard to comment on, point to specific parts

Toward More Direct & Social Interaction for Video Learning

- Alternative explanations from learners
- Synchronous watching with other learners
- Linking relevant resources with different levels of scaffolding
- Experimenting with in-video examples & data

Leveraging Video Interaction and Content to Improve Video Learning



Juho Kim
MIT CSAIL

juhokim@mit.edu

juhokim.com