Identifying High-Level Student Behavior Using Sequence-based Motif Discovery

David H. Shanabrook¹, David G. Cooper², Beverly Park Woolf², and Ivon Arroyo²
dhshanab@acad.umass.edu, dcooper@cs.umass.edu
¹Department of Education, University of Massachusetts, Amherst
²Computer Science Department, University of Massachusetts, Amherst
Motifs are approximate periodic patterns

Motif discovery has been used in such diverse fields as gene sequencing, stock price prediction.
We want to find patterns of reoccurring behavior in students when using Wayang Geometry tutoring system.

That way, we can implement tutor interventions to address these behaviors
These students are engaged
Detecting Behaviors: using correlation

record actions  \[\rightarrow\]  find correlations  \[\rightarrow\]  ranking based on performance

measure performance
Detecting Behaviors: using motif discovery

- record actions
- symbolize metrics (binning) by hand
- discover motifs (algorithm)
- determine behaviors (meaning groups) by hand
- tutor interventions (real time meaning group match)
example problem and metrics:

In the figure above, the large square is divided into two smaller squares and two shaded rectangles. If the perimeters of the two smaller squares are 8 and 20, what is the sum of the perimeters of the two shaded rectangles?

A 14
B 18
C 20
D 24
E 28

secToFirstAttempt= 0
incorrectAttempt= 0
hintsSeen= 0
secOtherAttempt= NA
example problem and metrics:

In the figure above, the large square is divided into two smaller squares and two shaded rectangles. If the perimeters of the two smaller squares are 8 and 20, what is the sum of the perimeters of the two shaded rectangles?

- A 14
- B 18
- C 20
- D 24
- E 28

- [Image of a puzzle interface with options: Go Away!, Mute, Formulas, new problem, help, village]
example problem and metrics:

In the figure above, the large square is divided into two smaller squares and two shaded rectangles. If the perimeters of the two smaller squares are 8 and 20, what is the sum of the perimeters of the two shaded rectangles?

- A: 14
- B: 18
- C: 20
- D: 24
- E: 28

problem_022

secToFirstAttempt = 10
incorrectAttempt = 1
hintsSeen = 1
secOtherAttempt = NA
example problem and metrics:

```
secToFirstAttempt= 10
incorrectAttempt= 4
hintsSeen= 2
secOtherAttempt= 3.5
last hint solves= FALSE
skip=FALSE
```

```
In the figure above, the large square is divided into two smaller squares and two shaded rectangles. If the perimeters of the two smaller squares are 8 and 20, what is the sum of the perimeters of the two shaded rectangles?
```

```
Perimeter rectangle = 2 HEIGHT + 2 WIDTH
```

```
2 8 2
2
5
5 20 5
5
14
18
20
24
28
```
binning and symbolizing

Seconds to
First Attempt

Seconds to First Attempt Frequency

Seconds Binned

< 5 5 to 30 > 30

d e f
binning and symbolizing

Seconds to
First Attempt

guessing  typical  extended

Seconds Binned

d  e  f

Frequency

Seconds
binning and symbolizing

Hints Seen

Frequency of hints per problem

Hints Seen

0

5000

10000

15000

20000

25000

30000

35000

hints per problem
binning and symbolizing

Hints Seen

Frequency

hints per problem

not using
using
using to solve
binning and symbolizing

Seconds
other Attempts

Mean Seconds

Frequency

0 5 10 15 20 25 30

0 5000 10000 15000 20000 25000 30000

< 1.2 1.2-2.9 > 2.9

0 0 15

< 1.2 1.2-2.9 > 3
binning and symbolizing

Seconds
other Attempts

Mean Seconds

Frequency

freq

0 5 10 15 20 25 30

0 5000 10000 15000 20000 25000 30000

skip sof guess typical extended

16
bining and symbolizing

Incorrect Attempts

Attempts per Problem

Frequency

0
1
2
3
4

0
1 or 2
3 or 4

17
binning and symbolizing

Incorrect Attempts

Frequency

Attempts per Problem

18

0 1 2 3 4

skip, solved attempts default
## Problem Binning

<table>
<thead>
<tr>
<th>hints seen</th>
<th>a</th>
<th>using</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>b</td>
<td>not using</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>using to solve</td>
</tr>
<tr>
<td>seconds to first attempt</td>
<td>d</td>
<td>guess</td>
</tr>
<tr>
<td></td>
<td>e</td>
<td>typical</td>
</tr>
<tr>
<td></td>
<td>f</td>
<td>extended</td>
</tr>
<tr>
<td>seconds for other attempts</td>
<td>g</td>
<td>skip</td>
</tr>
<tr>
<td></td>
<td>h</td>
<td>sof</td>
</tr>
<tr>
<td>Other attempts</td>
<td>i</td>
<td>guess</td>
</tr>
<tr>
<td></td>
<td>j</td>
<td>typical</td>
</tr>
<tr>
<td></td>
<td>k</td>
<td>extended</td>
</tr>
<tr>
<td>incorrect attempts</td>
<td>p</td>
<td>attempts</td>
</tr>
<tr>
<td></td>
<td>q</td>
<td>default</td>
</tr>
</tbody>
</table>
resulting string:
479 students, 37,712 problems, 150,850 characters
random projection multivariate motif discovery algorithm

Based on Projection Algorithm - Buhler, Tompa (2001)
Extended to find projections in time series
Chiu, Keogh, Lonardi (2003),

Representing a time series symbolically, then discovering motifs in this series
Allows for space and time efficient, probabilistic discovery of motifs

Modified for 10 problem window, sliding 4 characters at a time.
Random projection multivariate motif discovery algorithm

10 problem window

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| a ff k q b f h o c e k p | b f h o a e h o c e k q | b f q i | b f k q | b f i p | a e h o |
| b f h o c e k p b f h o a e h o c e k q b f i q b f k q b f i p a e h o a e h o |
| c e k p b f h o a e h o c e k q b f i q b f k q b f i p a e h o a e h o a e i p |
| b f h o a e h o c e k q b f i q b f k q b f i p a e h o a e h o a e i p |
| a e h o c e k q b f i q b f k q b f i p a e h o a e h o a e i p |
| c e k q b f i q b f k q b f i p a e h o a e h o a e i p |

37,672

Random Selection

10 columns

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>h o e o a e b b f a</td>
<td>k p f o c e b a e a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h o e q b f a a e a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h o e q b f a a e a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>k q f q b f a a e a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i q f p a e a e a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Collision matrix
Defining Meaning Groups

30 Discovered Motifs:

6 meaning groups

• on-task
• not challenged
• game-like
• frustration
• too difficult
• skipping
Defining Meaning Groups

On-task (n)

aeiq aeho aeho aekp aeho aeiq aeho aeip aeho aeip
Defining Meaning Groups

On-task (n)

aeiq  aeho  aeho  aekp  aeho  aeip  aeho  aeho  aeip

default  sof  sof  solved  sof  default  sof  solved  sof  solved
Defining Meaning Groups

6 meaning groups symbolized

(n) • on-task mixture
(k) • not challenged sof, no hints
(g) • game-like not reading problems
(f) • frustration hint|default, then skipping
(r) • too difficult reading, hints solve
(z) • skipping no effort, skip-5, sof-5
Application of meaning groups on student strings
Application of meaning groups on student strings
Future work

• Implementation
• Validation
• Refinement
Future work

• Implementation - Meaning motifs will be implemented in Wayang and detect behaviors in realtime.
• Validation - On behavior detection, verification by notifying a monitor, or by querying the student.
• Refinement -
  Variations in problem window.
  Problem specific binning based on statistics: Arroyo et. al. (2010) at EDM Conference