Weekly update – Wednesday 29 June

**Definition of problem outcomes:** Previously, for a given problem, we defined four outcomes:

1. No submission (0)
2. Problem run with errors
3. Problem submitted with errors (1)
4. Problem submitted successfully (2)

For this task, we only consider three outcomes (1, 3, 4) as they can be directly extracted from the events data. For a given pair of problems, there are 9 (3 x 3) possible outcomes, given that there are three possible outcomes for one problem.

**Modules and slide breakdowns:** Within each module (e.g. w5p1), there are two sets of two problems. One set occurs at the half-way point of the module, and the other set occurs at the end of the module. For instance, consider the following slide breakdown:

0 1 2 **3 4** | 5 6 **7 8**

Where slides 0-2 and 5-6 are interactive slides (i.e. slides that can be completed by the student and when completed, are marked by a slide\_steps\_complete event in the event data). Slides 3-4 and 7-8 are problem slides (i.e. slides that ask the students to solve a problem, and can be marked by problem\_passed and problem\_failed events).

**Note:** Slide numbers start with 0.

**Representing student slide behaviour**: We encode the student’s interaction with the slides in each module with an interaction string. For interactive slides, we use 1 to represent that the student successfully completed the slide (i.e. recorded a slide\_steps\_complete event) and 0 to represent otherwise. For problem slides, we use 0 to indicate that no submission was made, 1 to indicate that the student failed in their submission, and 2 to indicate that the student successfully passed.

For instance, assume that there is a problem with 10 slides, and that slides 3-4 and 8-9 are problem slides and all others are interactive slides. Then, consider the following interaction strings below:

1. 1 1 0 2 2 – indicates that the student completed slides 0 & 1, but not slide 2, and successfully completed slides 3 & 4.
2. 1 1 0 1 1 | 1 1 1 2 2 – indicates that initially, the student completed slides 0 & 1, but not slide 2, and as a result, failed their submission on slides 3-4. In their next attempt, the student completed slide 2 (in addition to already completing slides 0 & 1), and as a result, successfully passed slides 3-4.

**Implementation:**

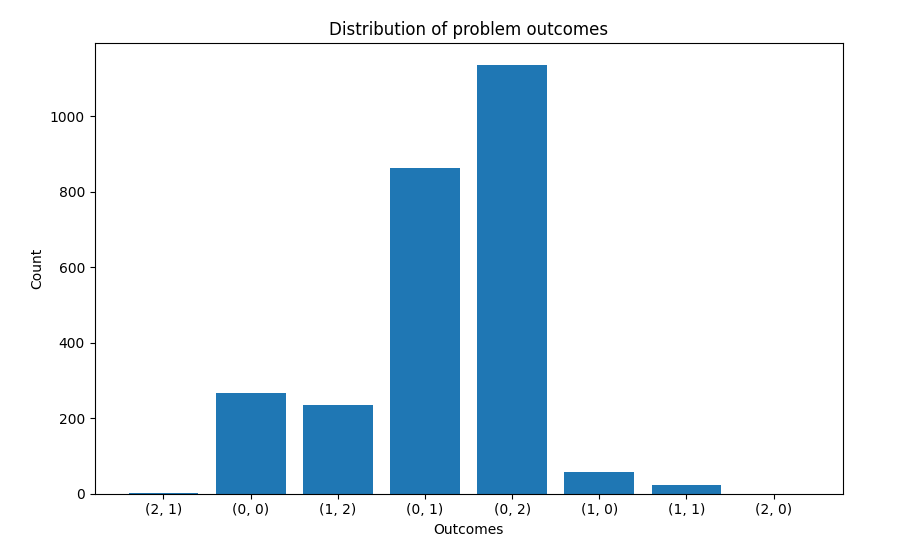
To test the procedure, I focused on a specific module **“challenge-newbies-2018-w1p1”,** and focused on the first five slides of that problem. There are a total of 4571 students, of whom 2583 made a submission to the problem.

Load the dataset and filter the dataset to match the specified problem.

* 1. To reduce time complexity and filtering the dataset again based on student ID, sort the dataset first by user ID.

1. Determine the interaction sequences for all students attempting the specified problem.
   1. Create a dictionary that has student IDs as keys, and a list of interaction strings for the student.
   2. Iterate through the events for a given student and build the interaction string for the student as you go.
   3. Append the interaction string to the list after the student has attempted a problem (i.e. resulted in a problem\_passed or problem\_failed).
2. From the interaction sequences, extract the outcomes and plot the distribution of outcomes.

**Results:**



In this figure, we can see that most students did not complete the first problem, and either failed or passed the second problem. No students completed both problems successfully, and a negligible number completed the first problem.

In addition, we can also work out the most-occurring interaction strings.

* (0, 0, 0, 0, 2): 1123 (43%)
  + Most students did not complete any interactive slides and were able to complete the last problem successfully.
* (0, 0, 0, 0, 1): 859 (33%)
  + A considerable number of students did not complete any slides, and failed the last problem (did not successfully complete it).
* (0, 1, 0, 0, 0): 190
* (0, 0, 1, 1, 2): 132
* All other interaction sequences had negligible results (less than 100).

# Thesis Ideas

**Thesis workflow**

Overarching pedagogical question

* Pedagogical question: Does engaging with slides improve outcome of the module? Is there a change
* Students might whiz through material at first when it’s easier (e.g. 0 0 0 0 2)
* Does this change afterwards
* How consistent is this – do they start off low engagement and then increase engagement – and then does that persist later?

**There are two phases to the project**

1. **Analytical** 
   * Given some **record/encoding of a student’s interaction with the slides** (e.g. an interaction string), create **several numerical metrics** that quantify the student’s progression through the course.
     1. Interactions : Encoding : Progression metric
   * **Why is this useful:** It can be used to predict whether a student will likely progress through the course.
     1. **One example:** We might want to measure the **linearity/completion** of someone’s progress. Ideally, the most diligent students will complete the slides in order (e.g. Slide 0, 1, 2, 3, 4). Note, this is different to the order in which slides may be accessed/viewed.
        1. **Encoding**
           1. We could do a binary encoding for each of the different interactive slides indicating slide completion.

**Strengths:** Easy and simple to implement

**Weakness:** Lose information about the order in which slides are completed. Binary representation is a less rich representation of student slide interactions.

* + - 1. **Numeric metrics** 
         1. We may want a metric that award people for progressing sequentially (e.g. for each given slide completed in order, award one point 1 + 1 + 1 + 1 + 1).
         2. **Look at other approaches**

Association rule mining

Linear feature extraction

* + 1. For each of these encodings, talk about the strengths and weaknesses of them, and under what conditions you might choose one encoding over another.

1. Predictive

Scaling up:

1. Compare different course outcomes
   1. Biggest cohort is beginners then intermediate
2. Achieve two things
   1. Find something that we can say about the data using the analysis technique that I have done so far. If we can write something that is publishable at this point, then that would be good.
   2. Think about the next step of analysis that we could do and try to meet some of the machine learning goals that I’d like to do in this project.
   3. Ways of mining these interaction strings

How would you explain this to other weeks, how would you extend this?

Longer sequences

Sequences corresponding to successful students

Could we reduce the data a little bit? Rather than having the individual slides – regardless of number of interactive slides – regardless of structure of each model, say something about the interactivity of the material.

Potentially we could transform the data to represent different behaviours – derive a property from the sequence (e.g. 1 before 2). If there is a sequential behaviour etc.

IDEA: Classify the different behaviours from these sequences of numbers and turn that into something more descriptive at a higher level about the technique.

RN: Sequencing of the order of the slides. One thing we lose is the order that the student accessed the slides – people do jump to the problem as we have worked out before. We want to capture not only slide completion, but the order of slide completion.

Irena: We could try to predict (0, 1, 2) based on the previous slide, and the previous interaction sequences.   
  
TWO PHASES OF PROJECT:

1. Analytical: do something analytical of the data so far
   * 3. 1. We could award a point for completion
           1. Point for doing some of the activities
           2. Point for completing the activities
           3. Do a score per slide and then normalise against available points for the module
           4. 1 – complete attempt at all the material in the module 0 – not attempting any slides
           5. Could plot per problem and per module

Two dimensions: Completion vs. linearity

Could encode colours

And then plot how students move

Could do that across modules of the course and then make some observations at that level

Fewer dimensions – the easier it is to cluster

1. Predictive: do something predictive – ML

Simple:

* + - 1. Build simple classifier given the previous sequence and predict the three outcomes [0, 1, 2].
      2. Do it per module or do it using all the available data – compare the differences and look at the success rate.
      3. Can take the score available to the system – (ask about this)
         1. They get points if they pass all the validation tests
         2. Points available to them drop if they make too many submissions (in some ways they are already motivated by that)
         3. We could use that as a metric that they’re aiming for – a lot of students want to maximise their score which is different to their learning

Features we talk about extracting:

1. They may not be the best ones, but we can certainly start with that
   1. If we see data is grouped on one end

Given previous data, predict the outcome of the task

We are at middle of semester, we would like to know how students would perform at the end.



If people who did engage with it, did well – that’s evidence that we should get people to engage

If we could identify different groups, then we could look at the ratio of passed to fail on those different activities – hypothesis: when people don’t engage with interactive slides, there’s probably a big spread – e.g. people who already know it skip slides and do it, and then there are people who didn’t know a lot and underestimated. Then, for the people who do the slides, there is more linear relationship.

Let’s get all five weeks (get a whole sequence of all 5 weeks/whole course/half course – due to high attrition halfway through the course), we have a score associated with the final performance

* Try to understand why they’re not successful
* Compare with other courses

Or you could do data per module and then aggregate afterwards

Sophia was able to extract the score – there may be a score associated with each problem – submission stats table, have a look.

Prediction:

1. Early behaviour --> later behaviour (look-ahead windows)
2. At-risk behaviour --> abandoning the course

Continue to end of the course the analysis, work out different sequences over the whole course, and try to make sense of this data, and compare with other weeks.

TO DO: Find assessment outcomes

Have a look at conferences in this area – AIED, EDM – last 5 years – student behaviour using ML, see if we can find something

Back in Aug

Update by email

ML course: Tutor – send some examples. Monday evening (8-9 online) – 6 in parallel – need 6 tutors.2 assignments – first automarked, 2nd marked manually – quick marking

Going through tutorial notes which explain how we use python to implement BNN – more like a demonstration, you have the code and solutions – you go through code, time to time ask questions – can you modify this code and do this question

Send email address

Making good progress but need to clarify the task. We need to narrow the task. In next few weeks.

Personal preference: Finish September