Prerequisite Skill Structures in ASSISTments

Michel Desmarais Sameer Bhatnagar

July 24, 2015

Abstract

At the 2015 Artificial Intelligence in Education conference, Seth Adjei and Neil Heffernan presented their work on scrutinizing expert-defined prerequisite skill graphs. Using randomized controlled trials in PLACEments, the computer-adaptive-testing feature in the ASSISTments learning environment, they were able to identify some prerequisite skill arcs that were not supported by data.
This proposal outlines two alternative techniques that could be used to achieve the goal. We will start with a basic introduction to Partial Order Knowledge Structures (POKS), then try to mathematically formalize what a prerequisite skill might mean in this framework. The first proposed technique will be graph based, while the second will be built on interpreting probabilities during POKS structural learning.

1 POKS

Partial Order Knowledge Structures (POKS) determine which items in a test are prerequisite to others. POKS are derived from the theory of Knowledge Spaces, where such prerequisite relationships are written as $A \rightarrow B$, which means if a student got item A correct, they likely will get item B also correct; said differently, item B is a prerequisite of item A.

1.1 Working Example

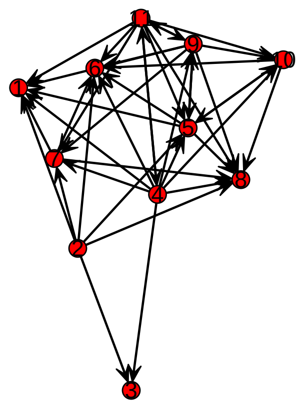
Let us look at a classical dataset from Tatsuoka, which is well studied in de la Torre.

```
## 'data.frame': 536 obs. of  11 variables:
##  $ H01: int  1 1 1 1 1 0 0 1 0 0 ...
##  $ H02: int  1 1 1 0 0 0 0 0 0 0 ...
##  $ H03: int  0 1 0 0 0 1 1 0 1 1 ...
##  $ H04: int  1 1 0 1 0 0 0 0 0 0 ...
##  $ H05: int  1 1 0 1 0 0 0 0 0 0 ...
##  $ H06: int  1 1 0 1 1 0 0 0 1 1 ...
##  $ H08: int  1 1 1 0 0 0 0 0 0 0 ...
##  $ H09: int  1 1 1 1 0 0 0 0 0 0 ...
##  $ H10: int  0 1 0 0 0 0 0 0 0 0 ...
##  $ H11: int  1 1 0 1 0 0 0 0 0 0 ...
##  $ H13: int  1 1 0 1 0 0 0 0 0 0 ...
```

If we run the POKS code on this data, we get the following adjacency matrix.

	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	0	1	1	1	1	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	1	0	1	0	1	1	1	1	1	1	1
5	1	0	0	0	0	1	0	1	1	1	0
6	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	1	0	1	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	1	0	0	0	1	1	1	1	0	1	1
10	0	0	0	0	1	1	0	1	0	0	0
11	1	0	0	1	1	1	1	1	1	1	0

This structure can be visualized as follows:



What stands out is item three, which has no prerequisites (no edges leaving node 3). Let us use this as a place to work out some numbers to demonstrate how the POKS structure is induced. We see that mastery of item 2 implies mastery of item 3 from the table and the graph. Let us check the contingency table for these two items.

```
##       item3
## item2   0   1
##     0 149 173
##     1  40 174
```

Now we can calculate $P(item2|item3)$ which should be greater than some threshold, as should $P(\neg item3|\neg item2)$. We must also ensure that the distributions over the two items are actually interacting with one another using a chi-square test.

```
##
##  Pearson's Chi-squared test with Yates' continuity
##  correction
```

```
##
## data:  item2 and item3
## X-squared = 40, df = 1, p-value = 1e-10
```

2 Prerequisite Skills

It is important to recognize that to go from item-item structures, to skill-skill structure, we need an item-skill mapping. This has been coined the Q-matrix, where each item may require one or more skills. Here is an example of a expert defined q-matrix from our dataset:

	QH1	QH2	QH3
H01	1	1	0
H02	1	0	1
H03	1	0	1
H04	1	0	0
H05	1	1	0
H06	1	1	0
H08	1	0	1
H09	1	0	1
H10	1	0	0
H11	1	0	0
H13	1	1	0

3 Deriving Prequisite skills - proposal 1

4 Deriving Prequisite skills - proposal 2

if an item is succeeded, then all the skills involved should be considered succeeded. If it is failed, then we need to determine how confident we are that skills involved are not mastered. And with the accumulated numbers, we simply build the POKS structure over skills!