Adaptive Engine for MOOC

Ilia Rushkin
Cambridge, USA

ABSTRACT

UPDATED—April 5, 2017. Description of an adaptive recommendation engine for serving assessment questions in a MOOC. Prototype in R exists.

1. INTRODUCTION

We would like to create a simple adaptive recommendation engine for a MOOC, capable of deciding what url to serve to a user next based on the user's history and the information about the url's content, provided by an SME. Primarily, we are interested in serving assessment items (below we call them questions), but instructional materials are also possible. These can be instructional webpages or videos intended to be mixed with assessment items ("if a student has trouble with that question, let them read this" etc.)

We use a variety of Bayesian Knowledge Tracing (BKT) model to estimate the students' state. What makes our situation special is that, as we learned from the adaptive pilot and just generally from seeing MOOCs,

1) Questions in the course differ widely in nature, and in particular in difficulty. Thus, we cannot assign the same values of guess, slip and transit probabilities to them, even if they are all tagged with the same learning objective.

2) Tagging is complicated: often a question is tagged with several learning objectives (aka "skills", "knowledge components"), possibly with varying degree of relevance.

3) In a self-paced MOOC environment, there is a need for a causal structure in the learning objectives: we should not serve to a user items tagged with a learning objective, if the user has shown lack of knowledge of other learning objectives that are pre-requisite to that one. In the simplest case, it can be dictated by a simple ordered list (the natural order of learning the content of the course), but it could also be a detailed graph of pre-requisite relationships among learning objectives.

4) In a MOOC, the number of students is high, so we can afford to define a model with a large number of parameters and optimize them based on the student interaction data.

2. RELATED LITERATURE

TBD

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

3. MODEL DESCRIPTION

Recommendation part

The content knowledge of the course is represented as a list of N LOs, short for "learning objectives" with pre-requisite relationships traced among them. The pre-requisite relationships are naturally visualized as a directed acyclic graph, and is stored as an $N \times N$ matrix w of pre-requisite strengths, $w_i j$ representing the strength of the graph edge from LO j to LO i (j is a pre-requisite for i). We define this strength to be on the scale from 0 to 1. If there are no connections, w a zero matrix.

For each question q ($q = 1, 2, ...Q$) of the course, we define the relevance of each LO i, denoted k_{qi} and on the scale from 0 to 1. Thus, we form a $Q \times N$ matrix k, each row of which is for a question and each column is for an LO. This matrix is the result of tagging questions with LOs.

We assume that the mastery of each LO by each course user is a binary latent variable – the user either has learned it or not – and we update the mastery matrix p, where the element $p_u i$ is the currently estimated probability that the user u has the mastery of the LO i. We define the mastery threshold $p^* \in [0,1]$, and if $p_{ui} \geq p^*$, we say that the mastery of i by the user u is sufficiently certain and no longer needs verification.

For each LO and for each user, we can define the pre-requisite readiness:

$$r_{ui} = \prod_{j=1}^{N} \min\left(1, \frac{p_{uj}}{p^*}\right)^{w_{ij}}. \quad (1)$$

In terms of matrix multiplication, we form a matrix $X_{uj} = \log\left(\min\left(1, p_{uj}/p^*\right)\right)$ and $r_{ui} = \exp\left((X \cdot w)_{ui}\right)$. If we denote $\circ\exp$ the element-wise (Hadamard-style) exponential of a matrix, $r = \circ\exp(X \cdot w)$.

An element r_{ui} has value 1 if the user has sufficiently mastered all LOs pre-requisite for the LO i, and less than 1 if the matery probabilities for some pre-requisites are not yet certain. If the pre-requisite strength w_{ij} is weaker, it enters r_{ui} with a smaller weight, in a sense allowing less certain mastery of less important pre-requisites. Introducing a forgiveness parameter $r^* \in [0,1]$, we assume that a user u is sufficiently ready for learning an LO i if $r_{ui} \geq r^*$. If $r^* < 1$, it means that we "forgive" some degree of uncertain knowledge of pre-requisites.

To recommend the next question for a student, we subset the matrix k_{qi} to only those questions (matrix rows) that belong to the current homework and that a user u has not seen yet. Thus, we obtain a user specific matrix $k_{qi}^{(u)}$. We define the non-negative user-specific vectors "question readiness", "question demand", and "question appropriateness" (in terms of difficulty

level of the problem $d_q \in [0,1]$):

$$R_q^{(u)} = \sum_{i=1}^{N} k_{qi}^{(u)} \max(r_{ui} - r^*, 0) \tag{2}$$

$$D_q^{(u)} = \sum_{i=1}^{N} k_{qi}^{(u)} \max\left(1 - \frac{p_{ui}}{p^*}, 0\right) \tag{3}$$

$$A_q^{(u)} = \sum_{i=1}^{N} k_{qi}^{(u)} (1 - |p_{ui} - d_q|) \tag{4}$$

In terms of matrix multiplication,

$$R^{(u)} = k^{(u)} \cdot \max(r_u - r^*, 0)^T \tag{5}$$

$$D^{(u)} = k^{(u)} \cdot \max\left(1 - \frac{p_u}{p^*}, 0\right)^T \tag{6}$$

A large readiness of a question means that the user has overall high chance of knowing all the pre-requisites for this question's LOs. A larger demand for a question means that this question focuses on those LOs for which the user's probability of mastery is currently low. A sensible recommendation policy should combine ideas: serving questions with high readiness so the user has all enough pre-requisite knowledge, with high demand so the user benefits, and with high appropriateness so the users with higher proficiency get more difficult items. We introduce a vector of importance weights $V = (V_r, V_d, V_a)$ (defined up to normalization) which measures the relative importance of each: the next item q to serve should be the one which maximizes the combination $V_r R_q^{(u)} + V_d D_q^{(u)} + V_a A_q^{(u)}$. The serving stops in two cases:

1) if we exhausted the available questions (the matrix $k^{(u)}$ has no rows)

2) if $D_q^{(u)} = 0$ for all q, which means that the user has reached the mastery threshold p^* on all LOs relevant for the available questions.

Knowledge tracing part

We initialize the mastery probability matrix $p = p^{(0)}$ (students' prior knowledge), after which, when the user has been served a question, we record that the user has now seen it (so next time this question's row will be removed from $k^{(u)}$). When the user submits an answer to the question, it gets a correctness value (score) $C_q^{(u)} \in [0,1]$ and we update the mastery probability of each LO (i.e. this user's row of the matrix p) as:

$$x_0 = \frac{p_{ui} p_{qi}^{slip}}{p_{ui} p_{qi}^{slip} + (1 - p_{ui})(1 - p_{qi}^{guess})} \tag{7}$$

$$x_1 = \frac{p_{ui}(1 - p_{qi}^{slip})}{p_{ui}(1 - p_{qi}^{slip}) + (1 - p_{ui}) p_{qi}^{guess}}$$

$$x = x_0 + C_q^{(u)}(x_1 - x_0)$$

$$p_{ui} \to p_{ui} + k_{qi}\left(x + (1 - x) p_{qi}^{trans} - p_{ui}\right)$$

This is a type of Bayesian Knowledge Tracing, with several modifications. The main one is that we allow for the question to be tagged with multiple LOs, so that the parameters carry the index i. To control for which LOs need to be updated and by how much, we include multiplication by the relevance k_{qi} in the last line. We also adapted the possibility that the scores are not binary. Because of this, the definitions of p_{qi}^{trans}, p_{qi}^{guess}, p_{qi}^{slip} become a bit convoluted, but no matter: we can just think of the matrix p^{slip} as some parameters to be optimized later (see section 5).

For terminological simplicity we referred to the content items as questions. However, the model can accomodate instructional items as well, e.g. videos or text. We can adopt a rule that, if an item q is instructional, the outcome of user's interaction with it is always "correct". A way to think of it is to imagine that q includes an assessment part of trivial difficulty. The slip probabilities $p_{qi}^{slip} = 0$, the guess probabilities now have the meaning of the probability of not learning an LO from the item, and so we set them to $p_{qi}^{guess} = 1 - p_{qi}^{trans}$. The update procedure of eq. (7) reduces to

$$p_{ui} \to p_{ui} + k_{qi}\left(\frac{p_{ui} + (1 - p_{ui})(1 - p_{qi}^{trans}) p_{qi}^{trans}}{p_{ui} + (1 - p_{ui})(1 - p_{qi}^{trans})} - p_{ui}\right). \tag{8}$$

Also, in case of an instructional item, unlike an assessment item, we may or may not want to mark it as "seen", because it is allowed to serve it to a user multiple times (or perhaps we should impose a limit on how many times it can be served).

If the matrix p contains zeros it is possible to encounter indeterminacies of the 0^0 and $0/0$ types in the expressions (1) and (7). The holistic way to preclude these is to adopt a small cutoff, e.g. we can set $\varepsilon = 10^{-10}$, and append the updating calculation (7) with the line $p \to \max(p, \varepsilon)$. Evidently, if the initial values of the matrix p are all non-zero, the only way zeros can appear is if users work on questions with the slip probability 0 or 1 and give correct/incorrect results contrary to these.

4. MODEL PARAMETERS

As described in section 3, the model is governed by the following constants: the mastery threshold p^* (maybe 0.95), the pre-requisite forgiveness parameter r^* (maybe 0.95), the vector of relative importances of readiness/demand/appropriate difficulty V (maybe $V_r = 3, V_d = 1, V_a = 2$) and the regularization cutoff ε (maybe 10^{-10}). In addition, as will be described in section 5, there is a threshold for updating BKT parameters η (maybe 5).

To form the matrix w, we require an SME to produce the pre-requisite relationship graph of learning objectives, indicating the strength of the relations. Realistically, we expect the SMEs to use three distinct values of strength: "none" (the absence of the edge in the graph), "weak" and "strong", and we will then convert these to numeric values adopting some convention (e.g. "none"=0,"weak"=0.5, "strong"=1). How complex a graph the SME produces will vary strongly. But at a minimum,

we should have a chain of learning objectives indicating the order in which they should be learned.

To form the matrix k, we require an SME to tag each content item with relevant LOs, indicating the level of relevance of each. Realistically, we expect the SME to tag each item with only a few (possibly just one) LOs, describing their relevance as "weak" or "strong", which we will then convert to numeric values. We ask the SME to indicate the difficulty level d_q ("easy", "regular", "hard"), which the SME will likely tie to the maximum number of points for a problem.

Furthermore, we offer the SME to indicate the likelihood of guessing the correct answer without knowing each of the tagged LOs, and the learning value for each of the tagged LOs (all of that simplified, e.g. a choice of "none", "strong", and "weak"). We set the value of the slip probabilities based on difficulty (higher for more difficult items), of the guess probabilities based on likelihood of guessing and of the transit probabilities based on learning value. If any of this information is not given to us by the SME, we resort to simple initial estimates (e.g. $p^{trans} \approx 0.1$ and $p^{slip} \approx 0.1$ for a problem, higher for more difficult items, lower for less difficult items; $p^{guess} = 1/K$ for a multiple-choice question with K answer options), to be updated later.

Finally, we need to initialize the mastery probability matrix p. At its simplest, we can settle at $p^{(0)} = 0$ (to be immediately converted to $p = \varepsilon$).

Thus, to form the data matrices, we rely partly on SMEs judgement, partly on simple estimates, partly on the choice of the constants used to translate SMEs terms ("weak", "easy", etc) into numbers. For the generalized BKT parameters (matrices p^{slip}, p^{trans}, p^{guess} and $p^{(0)}$) this matters only initially: once we have a substantial amount of student data, we will optimize them (section 5).

5. OPTIMIZATION OF BKT PARAMETERS

We will rely on a way to optimize our BKT parameters, inspired by the "empirical probabilities" method of [1].

At some point in time, when we decide to run the optimization, suppose that the items submitted by a user u are $\{q_j^{(u)}\}$ ($j = 1,...J^{(u)}$), indexed in chronological order. Let the correctness of answers be $C_j^{(u)} \in [0,1]$. We denote $K_{ij}^{(u)}$ this student's knowledge of an LO i just before submitting the item $q_j^{(u)}$. Assuming that there is no forgetting, the knowledge is a non-decreasing function with values 0 and 1, so it is chracterized simply by the position of the unit step: for j from 1 to some n_i knowledge is 0 and from $n_i + 1$ onward it is 1. We find which n_i gives the highest accuracy of predicting correctness from knowledge:

$$n_i = \text{argmin} \left(\sum_{j=1}^{n_i} k_{q_j^{(u)},i} C_j^{(u)} + \sum_{j=n_i+1}^{J^{(u)}} k_{q_j^{(u)},i} (1 - C_j^{(u)}) \right), \quad (9)$$

where we try all values $n_i \in [0, J^{(u)}]$ and adopt the convention that if the lower limit of a sum is greater than the upper limit, the sum is 0. We construct the step function $K_{ij}^{(u)}$ using the obtained location of the step n_i. If there are multiple equal minima, and so multiple n_i, we construct the step function for each and take the average of the step functions. The resulting $K_{ij}^{(u)}$ is our empirical estimate of the knowledge of all LOs by the user u. Repeat the procedure for each user.

Since the order of items was chronological, $K_{i,1}^{(u')}$ estimates the prior knowledge of concepts by that user. Moreover, we can identify the occasions when slips, guesses or transfers of knowledge occured. Averaging these over the users (average weighted by the total relevances $R_i^{(u)}$) we get the empirical matrices of prior knowledge, transit, guess and slip probabilities as ratios:

$$P_{u'i}^{(0)} = \frac{\sum_u \left(K_{i,1}^{(u)} \sum_{j=1}^{J^{(u)}} k_{q_j^{(u)},i} \right)}{\sum_u \left(\sum_{j=1}^{J^{(u)}} k_{q_j^{(u)},i} \right)} \quad (10)$$

(same priors for all users u', i.e. all rows of $P^{(0)}$ are identical)

$$P_{qi}^{trans} = \frac{\sum_u \left(\sum_{j=1}^{J^{(u)}-1} k_{q_j^{(u)},i} (1 - K_{ij}^{(u)}) K_{i,j+1}^{(u)} \mathbf{1}(q_j^{(u)} = q) \right)}{\sum_u \left(\sum_{j=1}^{J^{(u)}-1} k_{q_j^{(u)},i} (1 - K_{ij}^{(u)}) \mathbf{1}(q_j^{(u)} = q) \right)} \quad (11)$$

$$P_{qi}^{guess} = \frac{\sum_u \left(\sum_{j=1}^{J^{(u)}} k_{q_j^{(u)},i} (1 - K_{ij}^{(u)}) C_j^{(u)} \mathbf{1}(q_j^{(u)} = q) \right)}{\sum_u \left(\sum_{j=1}^{J^{(u)}} k_{q_j^{(u)},i} (1 - K_{ij}^{(u)}) \mathbf{1}(q_j^{(u)} = q) \right)} \quad (12)$$

$$P_{qi}^{slip} = \frac{\sum_u \left(\sum_{j=1}^{J^{(u)}} k_{q_j^{(u)},i} K_{ij}^{(u)} (1 - C_j^{(u)}) \mathbf{1}(q_j^{(u)} = q) \right)}{\sum_u \left(\sum_{j=1}^{J^{(u)}} k_{q_j^{(u)},i} K_{ij}^{(u)} \mathbf{1}(q_j^{(u)} = q) \right)} \quad (13)$$

Here again, we adopt the convention that if the lower limit of a sum is greater than the upper limit, the sum is 0 (this happens when $J^{(u)}$ is 0 or 1). The value of the denominator in each of these expressions is a measure of how much student information we have for estimating the probability. In case there is no data for estimating an empirical probability, any of these expressions becomes a 0/0 indeterminancy. We should not want to update a probability in this case. Moreover, we impose a threshold η (e.g. 3) and say that we will not update a particular probability if the denominator in the corresponding equation is less than η. This is simple to enforce in practice: add to eqq. (10, 11,12,13) the rule that in each of them the denominator less than η should be replaced by 0. After that, any non-numeric (indeterminate or infinite) elements of the matrices $P^{(0)}$, P^{trans}, P^{guess}, P^{slip} should be replaced by the corresponding elements of the matrices $p^{(0)}$, p^{trans}, p^{guess}, p^{slip}. Now we can update the BKT parameter matrices with the estimates: $p^{(0)} = P^{(0)}$, $p^{trans} = P^{trans}$, $p^{guess} = P^{guess}$, $p^{slip} = P^{slip}$. But in addition to that, we should also update some elements of the current mastery matrix p. Namely, let us call "pristine" those elements of p, which have never been updated with a non-zero shift via (7). All the pristine elements of p

should be replaced with the corresponding elements of $P^{(0)}$. This way, the optimized prior LO knowledge values will be used for all users yet to come to the course, but also for the existing users for those learning objectives that they have not yet explored.

CONCLUSION

REFERENCES

1. William J Hawkins, Neil T Heffernan, and Ryan SJD Baker. 2014. Learning bayesian knowledge tracing parameters with a knowledge heuristic and empirical probabilities. In *International Conference on Intelligent Tutoring Systems*. Springer, 150–155.