A Bandits Approach to Intelligent Tutoring Systems using Concept Evolution

Sudha S[+], Arun Rajkumar[+]
[+]Indian Institute of Technology Madras

Abstract

With the huge number of learning resources available online today, the Intelligent Tutoring Systems (ITS) are of great need more than ever. An ITS is a system that personalizes the course contents to each learner. In this paper, we address the problem of suggesting an effective & efficient learning sequences to learners based on their knowledge levels. We take a multi-armed bandits approach to action choosing where we suggest that action which has the highest estimated learning outcome at each step. We model the actions as Beta distributions & the learners' knowledge level as concept vectors. We also learn the prerequisite relationships that can exist among the concepts automatically. We propose a novel algorithm that achieves the goal efficiently. Our experimental results show that our algorithm's performance is comparable to that of the optimal algorithm.

1 Introduction

Traditional teaching methods utilize a uniform approach for all learners, disregarding individual abilities and needs. Intelligent Tutoring Systems (ITS) adapt teaching strategies according to learner's unique parameters. This paper presents an ITS framework for devising tailored learning actions sequences for each learner, optimizing concept learning. We model this problem as a Multi-Armed Bandits setting, viewing learning actions as arms and the learning level gained as rewards. The model also considers prerequisite relationships between concepts.

Our approach allows a learner's knowledge level to range between 0 and 1, a shift from the conventional binary (0, 1) states. This accounts for varying mastery levels of a concept. Our framework permits each learning action to contribute variably to multiple concepts. We also incorporate prerequisite relationships between concepts with varying intensity levels. The algorithm autonomously learns these prerequisite relationships, negating the need for expert input.

2 Related Work

[1] suggests a Zone of Proximal Development (ZPD)-based action sequence selection, incorporating multi-armed bandits to maximize rewards. Their method relies heavily on time-consuming ZPD graph creation by an expert, a dependency absent in our approach.

[2] applies a POMDP approach to ITS in a question-and-answer context, limiting learner concept understanding to binary (0,1) values. Our method allows continuous values in [0,1] for knowledge levels, uses practical learning actions like videos, and doesn't require prerequisite information.[3] also applies a POMDP approach to ITS, but solving a POMDP is generally challenging due to the polynomial number of states.

[4] embeds Personalised Learning Action (PLA) between fixed assessment sequences to boost immediate assessment performance using the CLUB & ACLUB algorithms. Unlike them, our goal is efficient concept learning, not immediate assessment performance.

[5] proposes a Thompson Sampling & Knowledge Gradient variation for PLAs to improve immediate assessment performance, but doesn't address prerequisite dependencies. Our focus is on concept learning. [6] merges automatic curriculum generation with ZPDES bandits approach, framing curriculum generation as a graph coloring problem. This approach requires intensive ZPD graph initialization.

3 Problem Setting & Modelling Assumptions

N denotes the count of learners in an ITS system aiming to learn K concepts. Each learner i's knowledge state is indicated by vector $C_i \in [0,1]^K$, with C_{ij} signifying learner i's mastery of concept j (e.g., $C_{23} = 0.7$ means learner

2 has 70% grasp of concept 3). ITS system's objective is to teach all N learners all K concepts to a threshold θ level of mastery.

ITS possesses a set of actions A (e.g., videos, lectures) affecting the learner's knowledge level. The system learns the impact of these actions over time. Concept relationships are considered in two cases: one assumes independence, and the other considers prerequisite relationships affecting the impact of an action on a concept.

Learner-specific parameters determine individual learning rates, accommodating variations between fast and slow learners. The ITS must deduce these rates. We assume learner knowledge evolves Markovianly, and knowledge level estimates are assumed to be noisy.

Independent Concepts:

For the independent concepts, the effect of action a on concept i at round t is given as follows:

$$c_i^{t+1} = c_i^t + \text{Beta}(\alpha_a, \beta_a, c_i^t) \cdot (1 - c_i^t) \tag{1}$$

where a is the action chosen at time step t and $\text{Beta}(\alpha_a, \beta_a, c_i^t)$ is the CDF value of the action a at value c_i^t.

Dependent Concepts:

The value update for the dependent concepts is as follows:

$$c_i^{t+1} = c_i^t + \sum_{j=1}^{D} c_j^t \lambda_{j->i} \cdot \text{Beta}(\alpha_a, \beta_a, c_i^t) \cdot (1 - c_i^t) \tag{2}$$

where D is the number of prerequisite concepts to c_i and $\sum_{j=1}^{D} \lambda_{j->i} = 1$

Here again, $\text{Beta}(\alpha_a, \beta_a, c_i^t)$ is the value of the Beta CDF at value c_i^t.

Learner Specific Parameter:To model the learner's unique abilities, we use a user specific parameter $\gamma_i \in [0, 1]$. The effect of an action on a learner then will depend on the action, the specific learner & the current knowledge state of the learner. This is made formal below:

$$c_i^{t+1} = c_i^t + \gamma_i \cdot \text{Beta}(\alpha_a, \beta_a, c_i^t) \cdot (1 - c_i^t) \tag{3}$$

Parameters to Estimate: The ITS system is completely specified using $2 * K$ action parameters that govern the Beta CDFs, and $K * N$ parameters that describe the learner's knowledge state and N learner specific parameters.

4 ITS-BPECE - Bandits Based Parameter Estimtation for Concept Evolution

This section gives an overview of the parameter estimation for the independent & dependent concepts. The parameters that need to be estimated for the independent and the dependent concepts are different. Hence, the estimation approaches vary as well. The subsequent subsections give an overview of the algorithm we propose which we call Bandits based Parameter Estimation for Concept Evolution (BPECE) and the section ends with a pseudo code of the BPECE in Algorithm 1.

Algorithm Overview:

We start off by choosing an action uniformly at random till each action has been chosen a minimum of (a small value) A_{min} times. We observe the data thus generated which looks as:

$$\{..., (C_{i1}^t, C_{i1}^{t+1}), (C_{i2}^{t'}, C_{i2}^{t'+1}), ...\} \tag{4}$$

If it is an independent concept in question, we use the Zeroth-Order(ZO) optimization to estimate the action parameters. The objective function for the ZO in the case of independent concepts is:

$$f(\alpha_a, \beta_a) = \left(\frac{c_i^{t+1} - c_i^t}{1 - c_i^t} \right) - \text{Beta}(\alpha_a, \beta_a, c_i^t) \tag{5}$$

We use the ZO estimation after every D_{min} number of data samples we collect and we increase the value of D_{min} over time.

In the dependent concepts case, not only do we have to estimate the action parameters, but also the $\lambda_{j->i}$ parameters for all dependency pairs (i, j). We start off by fixing the values of $\lambda_{j->i} = \frac{1}{K-1}$ for all (i, j). We estimate the Beta parameters using the ZO optimization. To estimate the $\lambda_{j->i}$ parameters, we fix the Beta parameters thus obtained. We train a Neural Network (NN) for each dependent concept with the concept vector being the input and the objective value being the output.

We alternatively fix $\lambda_{j->i}$ and estimate Beta parameters and fix Beta parameters and estimate $\lambda_{j->i}$ till the values of the parameter converge. Algorithm 1 presents the pseudo code of the algorithm.

We incorporate the MAB idea of choosing the arms that have the highest reward by picking those actions that push the learner concept vectors the farthest. We use a version of ϵ-greedy where we pick the best action with probability $(1 - \epsilon)$ and an action uniformly at random with probability ϵ. While we use an ϵ-Greedy strategy, more sophisticated bandit strategies can also be used in the framework.

Algorithm 1: BPECE

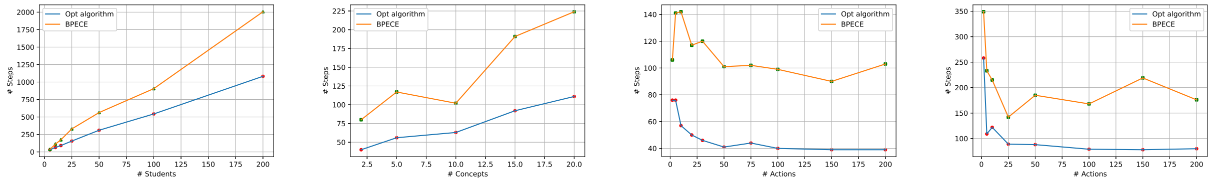
```
Input: A set of learner concept vector estimates, C_j, j = 1, 2, ...N
       Parameters A_min, D_min, ε
Output: Next action a_j for each learner j
for j ← 1 to N do
    if ∃a ∈ A where count(a) < A_min then
        a_j ← a
    end
    else
        for c_ji ∈ C_j do
            if c_ji is Independent then
                Estimate the (α_a, β_a)∀a ∈ A using Zeroth-Order Optimization on 5
            end
            if c_ji is Dependent then
                Initialize λ_{k->ji} values uniformly ∀(k, ij)
                while λ_{k->ji} AND (α_a, β_a)∀a ∈ A are not converged do
                    Fix λ_{k->ji}
                    Estimate (α_a, β_a)∀a ∈ A using Zeroth-Order Optimization on 5
                    Fix (α_a, β_a)∀a ∈ A
                    Estimate λ_{k->ji} using the Neural Nets
                end
            end
        end
        Update C'_{j_a} using Equation 3 & 2 ∀a ∈ A
        With probability 1 − ε
        a_j ← arg max_{a∈A} ||C'_{j_a} − C_j||_2

        With probability ε
        a_j ← choose an action a ∈ A uniformly at random
    end
end
```

5 Experiments

Setup: We use as performance metric the number of steps/rounds it takes for *all* concept values to go beyond 0.9. We compare our algorithm results against an optimal algorithm. The optimal algorithm we consider is an algorithm that has all the true parameter values of the actions and the dependencies and uses those to pick the best action for the learners greedily.

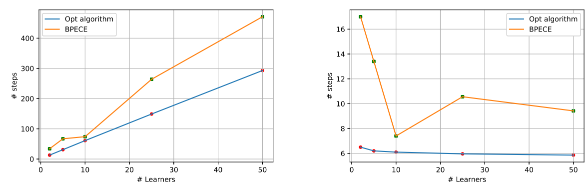
Results for Independent Concepts

Figure 1 depicts the results for the Independent case where we vary different parameters.



(a) # learners vs # Steps (b) # Concepts vs # Steps (c) # Actions vs # Steps, # Stds = 10 (d) # Actions vs # Steps, # Stds = 20

Figure 1: Number of Steps for the Independent Concepts with varying parameters



(a) # learners vs # Steps (b) # learners vs # Avg Steps

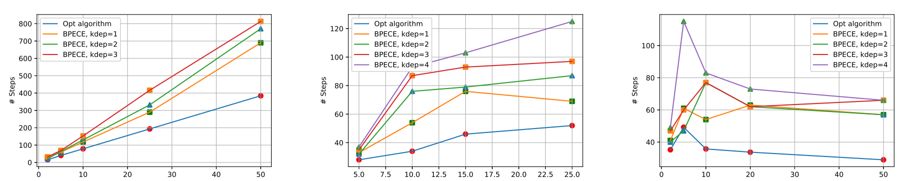
Figure 2: Total & Average Number of Steps for Independent Concepts for varying number of learners with the learner-specific parameter

Results for Independent Concepts with Student-Specific Parameter

Figure 2 shows the results for the case where we include a learner-specific parameter γ that accounts for each learner's learning rate. We vary the number of learners from 2 through 50 while fixing the number of actions and concepts.

Results for Dependent Concepts

Figure 3 shows the results for the number of steps taken with for dependent concepts, while the Figure **??** shows the average number of steps taken per learner. We vary the number of dependent concepts from 1 to 4 to show how the algorithm performs in each case.



(a) # learners vs # Steps (b) # Concepts vs # Steps (c) # Actions vs # Steps

Figure 3: Number of Steps for Dependent Concepts with Varying No of Dependent Concepts

6 Conclusion & Future Work

We proposed a novel bandits based parameter estimation approach to suggest learning actions to learners based on each learner's knowledge level. We considered the cases where the concepts are independent and dependent. In the dependent case, we took into consideration the prerequisite relationships between various concepts. We modeled each learning action's effect on a concept as a function of Beta distribution. For the prerequisite relationships, we trained NNs to estimate the degree of dependence. Finally, we used an ϵ-greedy approach to choose the best action for the learners. We back our proposed method with extensive experimental results.

As a future work, we can extend the learner-specific parameters to account for the different learning rate each learner for the dependent concepts as well.

References

[1] Benjamin Clement, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes. Multi-armed bandits for intelligent tutoring systems. *arXiv preprint arXiv:1310.3174*, 2013.

[2] Fangju Wang. Pomdp framework for building an intelligent tutoring system. In *CSEDU (1)*, pages 233–240, 2014.

[3] Jeremiah T Folsom-Kovarik, Gita Sukthankar, and Sae Schatz. Tractable pomdp representations for intelligent tutoring systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(2):1–22, 2013.

[4] Andrew S Lan and Richard G Baraniuk. A contextual bandits framework for personalized learning action selection. In *EDM*, pages 424–429, 2016.

[5] Indu Manickam, Andrew S Lan, and Richard G Baraniuk. Contextual multi-armed bandit algorithms for personalized learning action selection. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6344–6348. IEEE, 2017.

[6] Tong Mu, Karan Goel, and Emma Brunskill. Program2tutor: Combining automatic curriculum generation with multi-armed bandits for intelligent tutoring systems. In *Conference on Neural Information Processing Systems*, 2017.