



# Adaptive Remediation with Multi-modal Content

Yuwei Tu<sup>1(✉)</sup>, Christopher G. Brinton<sup>1,2</sup>, Andrew S. Lan<sup>3</sup>, and Mung Chiang<sup>4</sup>

<sup>1</sup> Zoomi, Inc., Chesterbrook, USA

`yuwei.tu@zoomi.ai`

<sup>2</sup> Princeton University, Princeton, USA

<sup>3</sup> University of Massachusetts Amherst, Amherst, USA

<sup>4</sup> Purdue University, West Lafayette, USA

**Abstract.** Remediation is an integral part of adaptive instructional systems that provide a supplement to lectures in case the delivered content proves too difficult for a user to fully grasp in a single class session. To extend the delivery of current remediation methods from single type of sources to combinations of different material types, we propose an adaptive remediation system with multi-modal remediation content. The system operates in four main phases: ingesting a library of multi-modal content files into bite-sized chunks, linking them based on topical and contextual relevance, then modeling users' real-time knowledge state when they interact with the delivered course through the system and determining whether remediation is needed, and finally identifying a set of remediation segments addressing the current knowledge weakness with the relevance links. We conducted two studies to test our developed adaptive remediation system in an advanced engineering course taught at an undergraduate institution in the US and evaluated our system on productivity. Both studies show that our system is effective in increasing the productivity by at least 50%.

## 1 Introduction

There are a lot of online learning platforms have emerged in recent years, servicing learning scenarios from corporate training programs to Massive Open Online Courses (MOOCs). Originally developed as supplements to in-class delivery, Adaptive Instructional Systems (AISs) have received recent attention in online learning [7, 8, 17, 18]. The typical AISs today will analyze question responses submitted by users to maintain individual knowledge state models, and then adjust the delivery of future modules by re-ordering, augmenting, and/or skipping over content according to a set of rules and possibly alternate content files created by the instructor [16].

Remediation is an integral part of the learning process of AISs: it provides a supplement to lectures where the delivered content proves too difficult for a user to fully grasp in a single class session, due to many possible factors (e.g., weakness in prerequisites, disengagement in class, or unclear explanations by the

instructor). The traditional “one-size-fits-all” approach to remediation, in which the instructor creates a single set of remediation content for the entire class, is undesirable due to the well-documented heterogeneity in user backgrounds, abilities, and strategies. Creating and managing personalized remediation for each user, on the other hand, would be difficult for an instructor to scale to even medium-sized courses. As a result, it is desirable to build systems that can automatically create, select and deliver individualized remediation content based on inferences of a user’s weaknesses in particular topic areas.

To date, many methods for adaptive learning in education have been proposed, with results demonstrating varying levels of effectiveness [4, 11, 15, 23]. One aspect that remains largely unstudied, however, is the systematic delivery of multi-modal remediation content, i.e., combinations of different material types like textbooks, lecture videos, web pages, practice questions, and interactive simulations. Doing so is increasingly relevant today as users are becoming accustomed to visiting multiple sources of content outside of the standard lecture material (e.g., through a search engine). It is thus desirable to develop methods that select pieces of multi-modal content for a particular user’s remediation and integrate them together into a single application for delivery to the user.

In this paper, we develop and evaluate a system that provides adaptive multi-modal remediation to users as they progress through a course. The operation of our system can be divided into four main phases:

1. Ingesting a library of multi-modal content files forming the baseline course and segmenting the materials into bite-sized chunks.
2. Linking those content chunks based on topical and contextual relevance.
3. Modeling users’ knowledge state when they interact with the delivered course through the system, and determining whether remediation is needed at this point.
4. Identifying a set of remediation segments addressing the current knowledge weakness and integrating them into single module of content to users.

In Sect. 5, we present the results of a trial we conducted to evaluate our remediation model in an advanced engineering mathematics course taught at an undergraduate institution in the US. In particular, we evaluate our method on a productivity metric defined as the total score on the questions divided by the total time spent. Using a series of statistical tests, we show that our individualization system outperforms one-size-fits-all course delivery significantly, increasing productivity by at least 50%, especially when content is varied at the segment level.

## 2 Related Work and Contributions

The long history of thoughts and admonitions about adapting instruction to individual student’s needs has been documented by many researchers (e.g., [19, 20, 24]). There exist several possibilities how the instruction is adapted: the *macro-adaptive* approach, the *aptitude-treatment interaction* approach, the

*micro-adaptive* approach and the *constructivistic-collaborative* approach, listed in chronological order beginning with the oldest approach [12]. Among those theoretical approaches, the most popular one is the *micro-adaptive* approach where adaptive instruction is conducted on a micro-level by diagnosing the user's specific learning needs during instruction [17]. It has been widely used in modern adaptive instructional systems like Intelligent Tutoring Systems (ITSs), which applies a variety of artificial intelligence techniques to represent the learning and teaching process.

Inspired by ITS, and combined with hypermedia-based systems, another family of AISs is developed as Adaptive Hypermedia Systems (AHSs). There are three main criteria for AHSs: the system is based on hypertext or hypermedia; a user model is applied and third; the system is able to adapt the hypermedia by using this user model [5]. There are also two different types of AHSs: adaptive presentation will directly provides an adaptation of the content which presented in different ways or orders. The content can be adapted to various details, difficulty, and media usage to satisfy users with different needs, background knowledge, and knowledge state [6]. Another type is adaptive navigation support, which only present an adaption of navigation by direct guidance, adaptive hiding or re-ordering of links, link annotation, map adaptation, link disabling and link removal [22].

As users are becoming accustomed to visiting multiple sources of content outside of the original lecture materials, it is necessary for AHSs to combine different material types (like textbooks, lecture videos, web pages, external links) and unfortunately, most existing adaptive systems do not meet this requirements [7, 15]. One of our main contribution is to propose an adaptive remediation system which is capable of systematic delivery of multi-modal re-remediation content. Our method identifies pieces of multi-modal content for a particular user's remediation without the limitation of standard course content and integrate them together into a single application for delivery for the user's convenient. It greatly enhances the extensibility and efficiency of regular adaptive remediation systems.

In addition, even though a variety of AHSs have been designed and implemented, it is still claimed that here was little empirical evidence for the effectiveness of AHSs [2, 4, 17]. Besides, most of the current works only compare the difference between users' grade with and without AHSs, but unfortunately, having higher scores doesn't means learning more efficiently [21]. Another main contribution of our paper, is that we propose a new evaluation metric - productivity, which considers the trade-off between points earned and time spent. This can be further extended to an uniform adaptive system evaluation criteria [9]. Besides, we also conduct trials in real classrooms and our experiment results provide strong evidence of the effectiveness of our proposed multi-modal remediation system.

### 3 System Methodology

In this section, we present the system we have developed to support multi-modal remediation. A high-level block diagram of its major components is shown in Fig. 1, with the individualization itself consisting of three parts: Content Tagging (Sect. 3.2), User Modeling (Sect. 3.3), and Path Switching (Sect. 3.4). In describing these components, we will overview several variants implemented for different use cases. Subsequently, in Sect. 4, we will formalize the specific algorithms implemented for the user trials in Sect. 5.

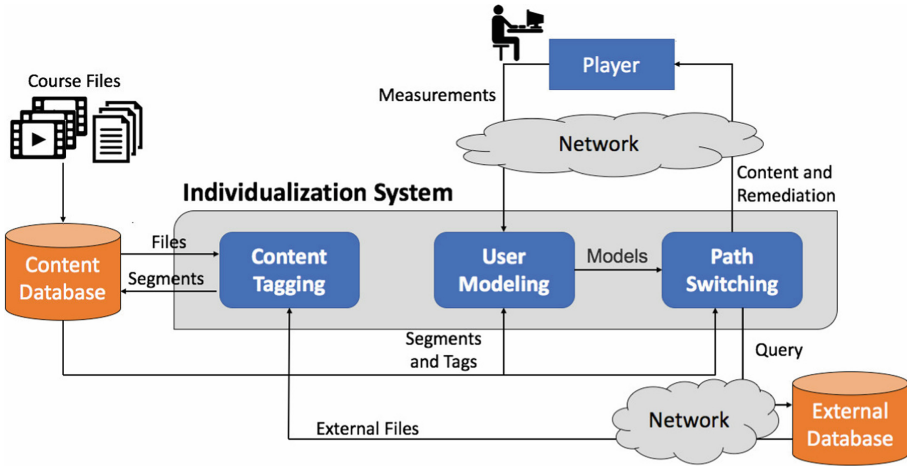


Fig. 1. Overview of our adaptive remediation system and its key components.

#### 3.1 Inputs

The Individualization System in Fig. 1 receives two types of inputs: measurements on user behavior, and the course content itself.

**Course Content.** The course content is stored in the Content Database. It consists of a series of modules ordered for delivery to users according to the instructor’s syllabus for the course, with each module containing a set of content files. Importantly, these files can be of different formats corresponding to different learning modes, including videos (.mp4), PDFs (.pdf), and slideshow presentations (.pptx). The modules can also include quiz questions, which are delivered to the user upon finishing the content in the module by the Player application.

**User Behaviors.** As the user interacts with the course application, the Player collects four types of data: responses to assessment questions, clickstream measurements generated from navigation through content files [3], posts on discussion forums comprising the Social Learning Network (SLN) [25], and annotations consisting of notes, bookmarks, and highlights. Each assessment response is tied to a particular question, while each clickstream measurement is recorded on a particular segment: segments are partitions of content files, as determined by Content Tagging described next. If necessary, performance prediction [13] can also be used to estimate a user's score on assessments she/he did not take.

### 3.2 Content Tagging

The purpose of Content Tagging is to associate the learning materials with the topics they contain. To do this, course files are first broken down into segments, where each segment is a 20 s chunk of video or a page/slide in a document/slideshow. In this process, a textual representation of each segment is obtained, where any audible components are passed through speech-to-text conversion and optical character recognition (OCR) is applied to any images. Quizzes are also included in this process, with each question comprising a quiz being treated as a segment.

The collection of segments in the course are then passed through natural language processing (NLP) algorithms for topic inference, with each segment modeled as a bag-of-words. Several NLP techniques are possible here, the important requirement being that they infer segment-topic and topic-word distributions, by either (i) treating topics as latent dimensions of the model or (ii) treating an outline/syllabus of the course as the topics and associating each segment with these topics by e.g., frequency of occurrence. The output of Content Tagging is then an association of each segment with its constituent topics: this association is specified as a probability distribution for each segment, with each part of the distribution expressing the amount of a particular topic comprising the segment.

### 3.3 User Modeling

The purpose of User Modeling is to estimate a user's knowledge state and/or content preferences with respect to each topic as the user proceeds through the course, so that Path Switching can adapt accordingly. The user model is updated through analysis of all or a subset of the input measurements collected by the Player. As a simple example, answering a test question correctly signifies an increase in content knowledge on the tested topics [4]. As another example, exhibiting high engagement in a certain file is interpreted as an increase in preference for this content mode [9]. More generally, our AIS invokes specific behavioral sequences called "motifs" to update the user model: these motifs are recurring subsequences of actions identified through sequential pattern mining algorithms that have been a-priori associated with increases/decreases in knowledge state and/or increases/decreases in content preferences [3].

When a user exhibits a motif on a set of segments, the dimensions of the user model corresponding to the topics covered in this set are updated based on the motif's association with knowledge state changes. In this way, the user's knowledge state is tracked as they progress through the course, indicating topics needing further instruction/remediation and presentation preferences [10].

### 3.4 Path Switching

Upon completion of each module, Path Switching analyzes the user model to determine whether remediation is needed, and if so, synthesizes the alternate content for rendering in the Player application. The decision to remediate is based on a comparison of the current knowledge state to an expected state of each topic from the module coverage. If the knowledge state is insufficient, the combination of topics for which the user needs assistance is identified, and then Path Switching searches for the segments that have the highest scoring match to the needed remediation, described further below. These segments are drawn primarily from the course files in the Content Database, but may also come from alternate files available in an External Database, e.g., additional courses provided by the instructor.

In general, the match score between a segment and the needed remediation is determined based on three factors: topic relevance, contextual relevance, and historical utility. Topic relevance measures the variation between the topic distributions (i.e., lower variation segments are more useful), and contextual relevance measures how far away the segment is from the current module (i.e., closer segments are more relevant to what is being taught). Historical utility quantifies how effective this segment has been for remediation in the past, updating over time as the segment is chosen and subsequent changes in knowledge state are observed. With the list of segments in hand, they are then split into different sets for each content type, and remediation files are created containing these segments. Finally, the files are instantiated in a remediation module and delivered to the user, after which point the user is returned to the original path.

## 4 Algorithms and Implementation

With an understanding of our AIS from Sect. 3, we now detail the specific algorithms used to test multi-modal content remediation in the trials in Sect. 5. Here, we restrict the input of the system to only quiz responses, which is most popular on other adaptive systems.

We denote the course library as a set of segments  $\mathcal{S}$  and the linear version of the course is delivered as a sequence of modules  $\mathcal{M} = \{1, 2, \dots\}$ .  $\mathcal{S}_m \subset \mathcal{S}$  is the subset of segments making up module  $m$ , and  $\mu(s)$  is the module (or set of modules) where  $s$  appears.  $\mathcal{Q} \subset \mathcal{S}$  is the set of segments that are quiz questions, with  $\mathcal{Q}_m \subset \mathcal{Q}$  being those set of questions that are asked in  $m$ .  $\mathcal{C}$  is the set of content material (non-quiz) segments,  $\mathcal{S} = \mathcal{C} \cup \mathcal{Q}$ , and  $\mathcal{C}_m$  is the set of content segments in module  $m$ ,  $\mathcal{S}_m = \mathcal{C}_m \cup \mathcal{Q}_m$ .

## 4.1 Content Topic Modeling

With each file in the course broken down into segments and ordered accordingly, our system extracts key words from a collection of documents and then create a bag-of-words representation for each segment  $s$ ; concretely, the bag-of-words over the dictionary  $\mathcal{X} = \{w_1, w_2, \dots\}$  of non-stopwords appearing in the course is  $\mathbf{x}_s$ , where  $\mathbf{x}_s(k)$  is the number of times word  $w_k \in \mathcal{X}$  appears in  $s$ . We then infer a topic distribution for each segment through the Latent Dirichlet Allocation (LDA) algorithm [1]. LDA extracts document-topic and topic-word distributions from a corpus of documents; here, segments are treated as separate documents. With the number of topics  $T$  extracted by the model chosen to minimize the coherence value, the resulting topic distributions  $\theta_{s=1\dots S}$  are  $T$ -dimensional probability vectors, forming the matrix  $\Theta = [\theta_s]$ . Each segment's topic distribution can then be taken as its content tag.

As a simple example, suppose a course consists of six topics A, B, C, D, E, and F. The distribution  $[0, 0.2, 0, 0.5, 0.3]$  then specifies a segment comprised of 50% of words from topic D, 30% of words topic E, and 20% of words topic B, which would occur if the breakdown of words in this segment followed these particular topic proportions (a distribution must sum to 1). The frequencies of the topic terms is significant here because intuitively, the more frequently a term appears, the more important that particular term is likely to be to the particular segment. Note also that stop words (for example, I, and, the, and so forth) need to be excluded prior to applying the NLP techniques. Also, if a syllabus, outline, or related material is available, that material is usable as a guidepost to better understand topics.

The distributions, particularly the frequency of the topic terms, are used later to calculate similarities between content files and are used relative to syllabus topics. With the segment-topic distribution matrix, we construct a segment-to-content similarity matrix  $\mathbf{D}$ . It has dimensions  $|\mathcal{S}| \times |\mathcal{C}|$ . More specifically,  $\mathbf{D} = [d_{s,c}]$ , where  $d_{s,c} = \cos(\Theta_s, \Theta_c)$  is a number between 0 and 1 that measures how similar segment  $s$  (either content material or quiz file) is to content segment  $c$  with respect to their topic proportions. Higher is more similar.

## 4.2 User Behavior Modeling

In this individualization trial, we only consider quiz performance to be triggers and here we construct the sequence of quiz questions where the users answered wrong in each module.  $Q_w \subseteq Q_m$  is the subset of these questions that this user answered incorrectly.

Then for each question  $q$  in  $Q_w$ , we search for a remediation set  $R = \mathcal{N}_1(q), \mathcal{N}_2(q), \dots, \mathcal{N}_L(q)$ . These are  $L$  neighborhoods for question  $q$  where  $L$  is the maximum number of times the individualization review can be triggered for a single module, with each  $l = 1, \dots, L$  corresponding to a different review iteration. Each neighborhood  $\mathcal{N}_l$  is of length  $K$ ,  $|\mathcal{N}_l(q)| = K$ , where  $K$  is the maximum number of segments shown per question per iteration.  $\mathcal{N}_l(q) \subset \mathcal{C} = \arg \max_{c=c_1, \dots, c_K} d_{q,c}$

is chosen such that  $u \neq \mu(c_1) \neq \mu(c_2) \neq \dots \neq \mu(c_K)$  and  $c_k \notin \mathcal{N}_{1:l-1}(q) \forall k$ . In other words, the segments in the  $l$ th neighborhood for question  $q$  are the  $K$  segments with highest similarity to  $q$  such that (i) none of the segments appear in the same module with each other, (ii) none of the segments already appear in a previous neighborhood, and (iii) none of the segments are in the current module  $m$ . Realistically, this can be built by sorting  $\mathcal{C}$  in descending order based on  $d_{q,c}$ , removing all the files from the current module, and then setting  $\mathcal{N}_1(q)$  to the first  $K$  items from this list that are not in the same module,  $\mathcal{N}_2(q)$  to the second  $K$  items not from the same module, and so on.

### 4.3 Individualization

At a high level, the path switching selects a sequence of segments that have a high likelihood of making the learning process more efficient based on the User Model. In our individualization trial, the path switching will be triggered when a user has just finished the quiz  $Q_m$  at the end of module  $m$ .  $Q_w \subseteq Q_m$  is the subset of these questions that this user answered incorrectly. Starting with  $l = 1$ , the following is the logic to determine the individualization the user receives at this point:

1. If  $Q_w = \emptyset$  (all questions correct) or  $l > L$  (maximum iterations reached), go to step 5.
2. Set  $\mathcal{N}_l = \cup_{q \in Q_w} \mathcal{N}_l(q)$ , the collection of unique segments in the  $l$ th neighborhoods of the questions answered incorrectly.
3. For each segment  $s \in \mathcal{N}_l$ :
  - (a) Obtain the similarities between  $s$  and all other content segments that appear in the same module as  $s$ , i.e.,  $d_{s,c} \forall c \in \mathcal{C}_{\mu(s)}$ .  $\mathcal{S}_s \subseteq \mathcal{C}_{\mu(s)}$  is the subset of these segments for which  $d_{s,c} \geq \delta$ , those for which the similarity is at least  $\delta$ .
  - (b) Generate a module  $r$  containing the segments  $s \in \{s, \mathcal{S}_s\}$  for which  $e(o(s)) > E$  (the segments in the set for which the user's engagement on that mode is at least  $E$ ). If there are no such segments, then let this module just consist of segment  $s$  as a standalone document.
  - (c) Show the user the module  $r$ .
4. Let the user take the set of questions  $Q_w$  he/she answered wrong again. Update  $Q_w$  based on the result (the subset of questions answered incorrectly again). Increment  $l$  and return to step 1.
5. Unlock the explanations of the questions in module  $m$ , and allow the user to proceed to module  $m + 1$ .

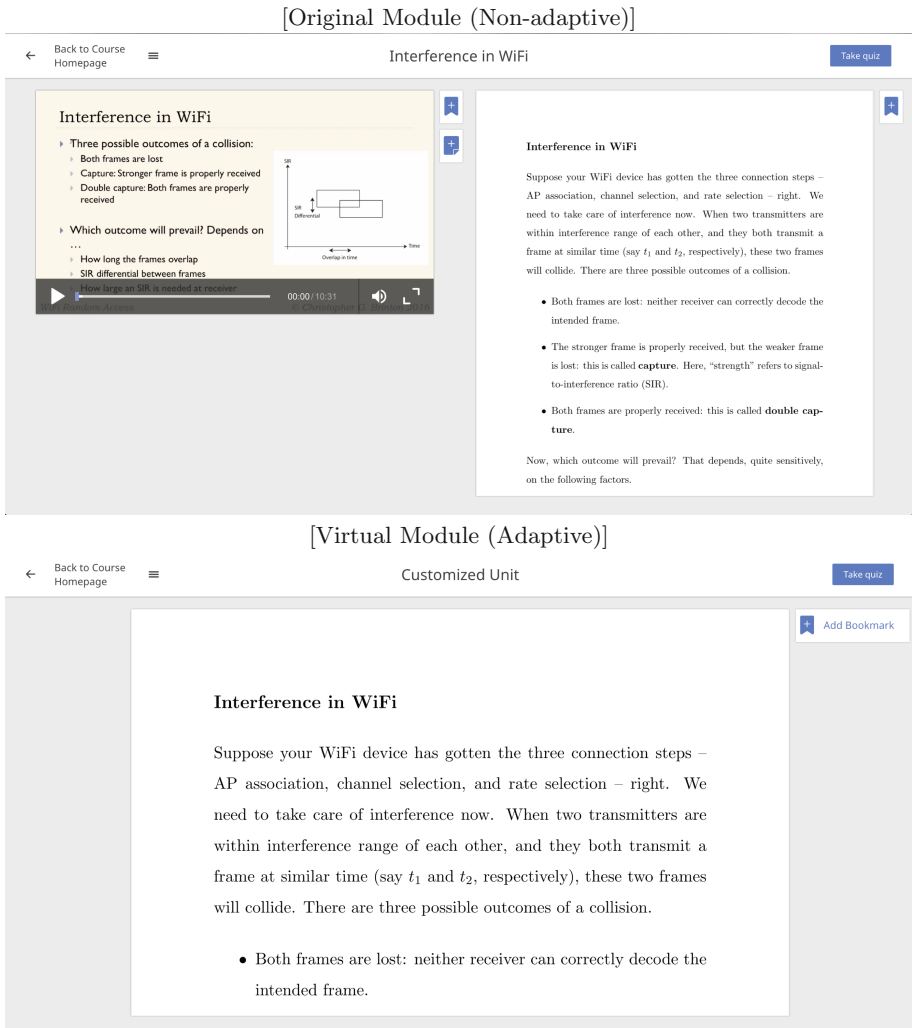
## 5 Experiments

To test the efficacy of our adaptive remediation system, we conducted randomized control trials with one course offered to upper-class engineering students at an undergraduate institution in the US. We performed two trials at different levels of the course: the module level, and the segment level, which will be



described in details in the following section. And for each trial, we randomly divided the users into two groups, one experimental group using our adaptive system and one control group without the adaptive system.

By considering the trade-off between points gained for each question answered correctly and its relation to time spent on each module of material, we produce an overall score measurement called productivity. Higher productivity represents higher learning efficiency and it was hypothesized that the experimental group of users will have significantly higher productivity when compared to that of the control group.



**Fig. 2.** A comparison of the original module and the virtual module.

## 5.1 Experimental Setting

46 students enrolled in the engineering course participated in this study. The course content was on WiFi and had no relation to material that users were learning in class. Furthermore, the course used for this study was taught completely online. The baseline content consisted of fourteen modules(chapters), each being a combination of a lecture video and a PDF of lecture notes. Twelve of these fourteen modules were followed by a set of questions. Since the test subjects were split in half, with half in the experimental group and the other half in the control group, users were treated differently based on whether their responses to these question sets were correct or incorrect. For the control group, no matter the response to the questions, users were automatically pushed forward to the next module. However, for the experiment group, the adaptive remediation algorithm was implemented if any question was answered incorrectly, searching for material in the database that most directly corresponded to that specific question. The adaptive remediation system also consists of two types, individually conducted in two separate trials. The first trial provides support material on a module level, whereas the second trial provides support material on a segment level (segment extracted from individual files and put together as another file). This supporting material, shown in Fig. 2 was presented to the user with the hope that the learned knowledge would be reinforced and the user would be able to answer the question correctly before moving onto the next module.

## 5.2 Evaluation Metric

It is clear that the adaptive version would have users gain more points due to multiple chances at incorrect questions. However, this is at the expense of spending more time to get those points due to users sitting through a virtual reinforcement segment. Therefore, the benefit is analyzed statistically by considering the ratio of points gained per minute of time spent on the respective modules. Our productivity measure for a user at a given point in the course is:

$$p_{\alpha}(s, t) = s/t^{\alpha}$$

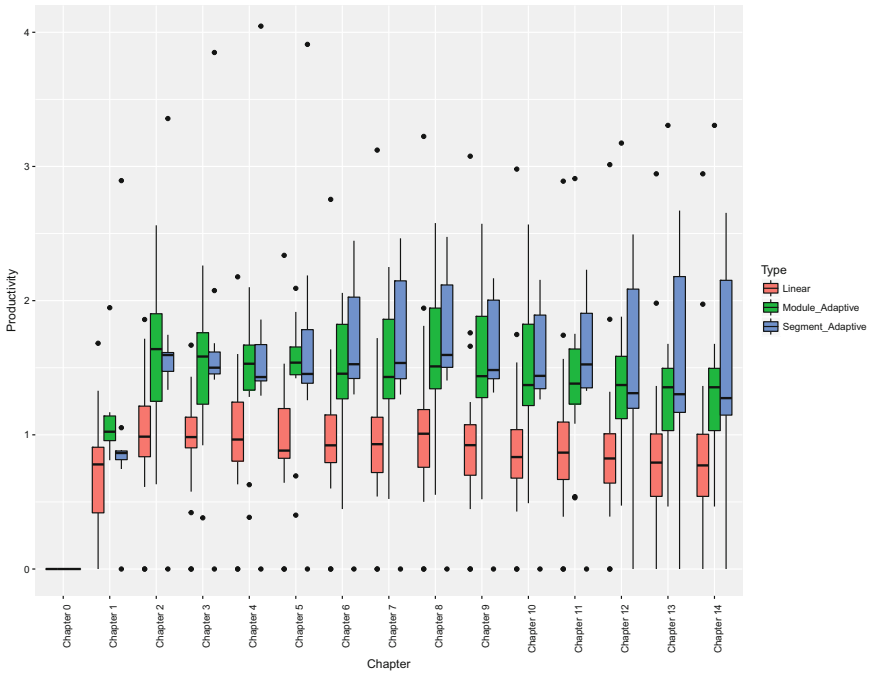
where  $s$  is the total points cumulatively obtained to that time,  $t$  is the cumulative time spent, and  $\alpha$  is a parameter controlling the importance of  $s$  versus  $t$  to the productivity measure. A higher  $\alpha$  value places more importance on time spent, and thus a lower overall score, while a lower alpha value places more importance on points gained.

## 5.3 Results

We start by describing some basic statistics of the course are as follows. Table 1 shows summary comparison between the aggregated productivity  $p$  between adaptive participants and non-adaptive participants, with different choices of  $\alpha$ . Clearly, participants in the adaptive course demonstrated higher productivity

**Table 1.** Summary statistics

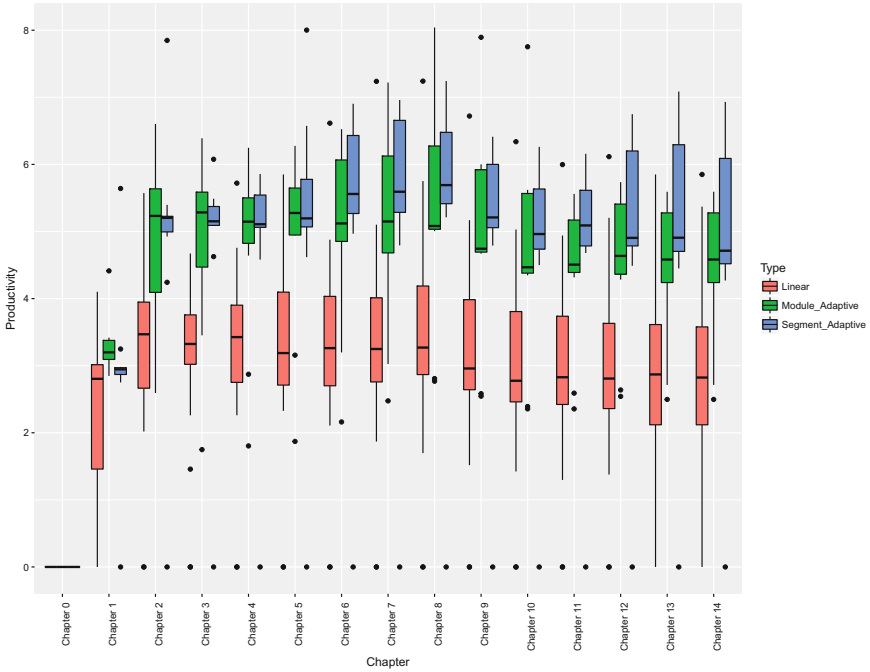
Method	Mean	SD	Median	Min	Max	Kurtosis	$\alpha$
Adaptive (Segment level)	1.433	0.805	1.454	0	4.046	0.816	1
Adaptive (Module level)	1.329	0.646	1.389	0	3.306	0.579	1
Non-adaptive	0.878	0.582	0.887	0	3.223	3.010	1
Adaptive (Segment level)	4.585	2.172	5.108	0	8.552	0.390	0.5
Adaptive (Module level)	4.491	1.826	4.834	0	9.350	0.998	0.5
Non-adaptive	2.843	1.559	2.977	0	7.241	-0.021	0.5



**Fig. 3.** Productivity  $p$  by Chapter for  $\alpha = 1$

with the both  $\alpha$  choices. In particular, users gain relatively higher productivity points on average with segment level adaptation. Overall, our adaption system increases the productivity by 50%–60%.

Now we plot the accumulative productivity score chapter by chapter in Figs. 3 and 4, grouped by segment level adaptation, module level adaptation, and no adaptation. In Figs. 3 and 4, consistently across all of the chapters in the course, users always gain higher productivity score in adaptive courses. In particular, in 8 out of the 11 chapters, adaptation on module level demonstrates a higher median while 8 out of 11 adaptation on segment level demonstrates higher 75th percentile values. Intuitively, this observation can be explained by adaptation at



**Fig. 4.** Productivity  $p$  by Chapter for  $\alpha = 0.5$

a segment level providing supporting materials at a detailed level with the exact information. However, supporting materials consists of file segments may lack the coherence of information that is in an adaptation at a module level.

**Table 2.** Statistical differences between groups with choice of  $\alpha$ .

	Non-adaptive versus MLA	Non-adaptive versus SLA	SLA versus MLA
$\alpha = 0.5$	1.522e-25	1.894e-27	0.025
$\alpha = 1$	1.933e-18	1.038e-21	0.113

In additional to the visual comparison in Figs. 3 and 4, we conduct Wilcoxon Rank-sum test, also called Mann-Whitney U test [14], to test if there is a statistical difference between the productivity in segment level adaptation (SLA), module level adaptation (MLA), and the non-adaptive version of the course. Above in Table 2 we record the p-values for each test and they show a statistically significant difference between the groups, except for SLA versus MLA when  $\alpha = 1$ . It demonstrates that users in the experimental group achieve higher productivity than those in the control group with significance, which proves our

method is effective at individualization. Moreover, individualizing at the segment level shows relatively higher productivity overall than individualizing at the module level, which directs our future research on individualization with finer granularity.

## 6 Conclusion

In this paper, we propose an adaptive remediation system with multi-modal remediation content. The system consists of four main phases: ingesting a library of multi-modal content files into bite-sized chunks, linking them based on topical and contextual relevance, then modeling users' real-time knowledge state when they interact with the delivered through the system, and finally identifying a set of remediation segments addressing the current knowledge weakness with the relevance lines. We conducted two studies to test our developed adaptive remediation system in an engineering mathematics course taught at an undergraduate institution and evaluated our system on productivity. Using a series of statistical tests, we show that users in the experimental group achieve higher productivity than those in the control group with significance. Moreover, individualizing at the segment level shows relatively higher productivity overall than individualizing at the module level. These results show that our method is effective at individualization, increasing the overall productivity by 50%–60%.

In the future, we will conduct trials using additional inputs to user modeling, like viewing behaviors and social learning networks. More user features will lead to more sophisticated user modeling techniques. Besides, for multi-modal content remediation, rather than ingesting the course materials by natural characteristics, like splitting videos with equal duration, we can further develop our content digesting methods based on the semantic meaning of course content with advanced text/language segmentation techniques. Another direction we are investigating is a comparison to an augmented version of our adaptive remediation system with self-learning reinforcement learning techniques. The system integrated with reinforcement learning can provide remediation options to users, collect their responses, and subsequently self-adjust the adaption agent for each user based on their responses to our remediation content.

## References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
2. Bra, P.D.: Pros and cons of adaptive hypermedia in web-based education. *Cyberpsychology Behav.* **3**(1), 71–77 (2000)
3. Brinton, C.G., Buccapatnam, S., Chiang, M., Poor, H.V.: Mining MOOC click-streams: video-watching behavior vs. in-video quiz performance. *IEEE Trans. Signal Process.* **64**(14), 3677–3692 (2016)
4. Brinton, C.G., Rill, R., Ha, S., Chiang, M., Smith, R., Ju, W.: Individualization for education at scale: MIIC design and preliminary evaluation. *IEEE Trans. Learn. Technol.* **8**(1), 136–148 (2015)

5. Brusilovsky, P., Kobsa, A., Vassileva, J.: Adaptive Hypertext and Hypermedia. Springer, Dordrecht (1998). <https://doi.org/10.1007/978-94-017-0617-9>
6. Brusilovsky, P., Millán, E.: User models for adaptive hypermedia and adaptive educational systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) *The Adaptive Web*. LNCS, vol. 4321, pp. 3–53. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-72079-9\\_1](https://doi.org/10.1007/978-3-540-72079-9_1)
7. Brusilovsky, P., Peylo, C.: Adaptive and intelligent web-based educational systems. *Int. J. Artif. Intell. Educ. (IJAIED)* **13**, 159–172 (2003)
8. Brusilovsky, P., et al.: Adaptive and intelligent technologies for web-based education. *Ki* **13**(4), 19–25 (1999)
9. Chen, W., Joe-Wong, C., Brinton, C.G., Zheng, L., Cao, D.: Principles for assessing adaptive online courses
10. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. *User Model. User-Adapted Interact.* **4**(4), 253–278 (1994)
11. De Bra, P., Calvi, L.: AHA: a generic adaptive hypermedia system. In: *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia*, pp. 5–12 (1998)
12. Froschl, C.: User modeling and user profiling in adaptive e-learning systems. Master Thesis, Graz, Austria (2005)
13. Lan, A.S., Waters, A.E., Studer, C., Baraniuk, R.G.: Sparse factor analysis for learning and content analytics. *J. Mach. Learn. Res.* **15**(1), 1959–2008 (2014)
14. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Stat.* **18**, 50–60 (1947)
15. Melis, E., et al.: ActiveMath: a generic and adaptive web-based learning environment. *Int. J. Artif. Intell. Educ. (IJAIED)* **12**, 385–407 (2001)
16. Paramythis, A., Loidl-Reisinger, S.: Adaptive learning environments and e-learning standards. In: *Second European Conference on e-learning*, vol. 1, pp. 369–379 (2003)
17. Park, O.C., Lee, J.: Adaptive instructional systems. *Educ. Technol. Res. Dev.* **25**, 651–684 (2003)
18. Park, O.C., Tennyson, R.D.: Computer-based instructional systems for adaptive education: a review. *Contemp. Educ. Rev.* **2**(2), 121–135 (1983)
19. Reiser, R.A.: Instructional technology: a history. In: *Instructional Technology: Foundations*, pp. 11–48 (1987)
20. Ross, E.W.: Adapting teaching to individual differences. *Soc. Sci. Res.* **26**(1), 27–29 (1989)
21. Shepard, L.A.: The role of assessment in a learning culture. *Educ. Res.* **29**(7), 4–14 (2000)
22. Sonwalkar, N.: Adaptive individualization: the next generation of online education. In: *EdMedia: World Conference on Educational Media and Technology*, pp. 3056–3063. Association for the Advancement of Computing in Education (AACE) (2007)
23. Tadlaoui, M.A., Aammou, S., Khaldi, M., Carvalho, R.N.: Learner modeling in adaptive educational systems: a comparative study. *Int. J. Mod. Educ. Comput. Sci.* **8**(3), 1 (2016)
24. Tobias, S.: When do instructional methods. *Educ. Res.* **11**(4), 4–9 (1982)
25. Yang, T.Y., Brinton, C.G., Joe-Wong, C.: Predicting learner interactions in social learning networks. In: *IEEE INFOCOM 2018 IEEE Conference on Computer Communications*, pp. 1322–1330. IEEE (2018)