Adaptive Social Learning Based on Crowdsourcing

Evgeny Karataev and Vladimir Zadorozhny, *Member, IEEE*

Abstract—Many techniques have been developed to enhance learning experience with computer technology. A particularly great influence of technology on learning came with the emergence of the web and adaptive educational hypermedia systems. While the web enables users to interact and collaborate with each other to create, organize, and share knowledge via user-generated content, majority of e-learning systems do not utilize the power of their users to create high quality educational content and provide data for adaptive algorithms. In this paper, we introduce a novel social learning framework that allows anybody to author educational content in a form of mini-lessons, learn lessons by following adaptive learning pathways as well as interact with their peers as in any social network. The proposed approach combines concepts of crowdsourcing, online social networks, and complex adaptive systems to engage users in efficient learning through teaching process. We first describe the main idea behind the framework and how users interact with it, and then we describe SALT system that implements the framework. We also performed evaluation of the SALT system via several classroom studies. Our results show that collective learning experiences can be efficiently utilized in adaptive social learning. We found that students tend to form stable clusters that survive very high similarity threshold. Meanwhile, our learning pathway analysis showed that almost all students have their own unique best pathway. Experiments with various recommendation algorithms showed that most algorithms obtain very small penalty in all classes.

Index Terms—Crowdsourcing, collective intelligence, peer production, social network, complex adaptive system, recommender system

1 Introduction

Personalized learning technologies seek to make the learning process more effective by adapting it to the goals, preferences, and knowledge of each student. Recognizing its potential, personalized learning was named among the 14 Grand Challenges for Engineering (engineeringchallenges.org). Many techniques have been developed over the last several decades to enhance learning experience with computer technology (we will review them in the next section). A particularly important branch of works focused on Adaptive Educational Hypermedia [1] that combined adaptive hypermedia systems with intelligent tutoring systems and rapidly evolved together with emergence of the web. However, traditional knowledge-based personalized guidance does not scale as the number and diversity of students increases [2]. High variability in students' backgrounds and learning preferences makes it extremely difficult to develop a "one-size-fit-all" educational content and a set of efficient personalization strategies.

More recent evolution of the web – Web 2.0 – changed the way information is shared online allowing anyone to consume and more importantly to *produce* informational resources. Web 2.0 blurred the line between content creators and content consumers and as the result empowered large crowd of users to collaborate, organize and share knowledge. Research and development in scalable web technologies showed promising results in combining online platforms and crowdsourcing. A prominent example of a successful crowdsourcing project that relies on the collective intelligence of its contributors – Wikipedia – demonstrated that crowdsourcing can be productive.

Web 2.0 imposed new requirements for web based educational systems [3]. Thus, in this paper we explore a novel way to combine social network and crowdsourcing technologies for the task of personalized learning. Particularly, we address "*extensible personalization and course adaptation*" and "*student active participation in a learning process*" requirements by introducing a novel social educational framework that adapts to individual student needs based on collective learning experiences. This is achieved by allowing students to be actively involved in both learning and teaching processes via social interactions. We utilize social network techniques to enhance the individual student's experience based upon those of her peers. As part of this *learning-through-teaching* approach, students learn from the content created by their peers and are directed to contribute their own content that reflects their preferences and backgrounds.

We developed a *Self-Adaptive Learning through Teaching (SALT)* system that implements this framework as a large-scale Online Social Network (OSN). SALT utilizes dynamic data warehousing, data analysis and data mining techniques to adaptively converge upon productive, personalized *learning pathways*. As more learning-through-teaching experiences are accumulated in the SALT database, SALT users and content items are grouped around similar performance profiles. After that, SALT discovers the most productive learning pathways with respect to a particular group of users and their performance profiles.

- *The authors are with the School of Information Sciences, University of Pittsburgh, Pittsburgh, PA 15260.*
E-mail: Karataev.Evgeny@gmail.com, vladimir@sis.pitt.edu.

Manuscript received 27 Feb. 2015; revised 2 Dec. 2015; accepted 23 Dec. 2015. Date of publication 6 Jan. 2016; date of current version 16 June 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TLT.2016.2515097

We evaluated the proposed technology via several classroom studies. In those studies we explored how SALT can adapt to individual or group of students and perform recommendations. Even though we performed SALT evaluation in classroom settings, the applicability of SALT goes far beyond an academic environment. The SALT framework is domain-neutral and can be efficiently used in different areas, including science, technology, engineering, and math, helping its users to excel in various subjects.

In what follows, we discuss background and related work (Section 2), then we describe the SALT framework and its implementation (Section 3). In Section 4 we cover SALT evaluation and in Section 5 we summarize the results and provide conclusions and remarks on the future work.

2 Background and Related Work

Personalized learning technologies employ comprehensive algorithms to adapt educational material for different groups of learners. For the last several decades much research focused on various aspects of adaptive E-learning. From Learning Management Systems [4], [5] to Intelligent Tutoring Systems [6], [7] to Adaptive Educational Hypermedia (AEH) [1], [8], [9] researches worked on sophisticated educational data mining [10], [11], user (student) modeling [12], [13], [14], [15], and recommendation [16], [17] and navigation algorithms [18], [19] to understand learners and adaptively provide learning resources.

Personalization mechanisms require an advanced domain model of each subject being taught, as well as elaborate indexing of each content item with concepts or skills identified by the domain model. Several frameworks and reference models, such as AHAM [20], XAHM [21], Munich reference model [22], LAOS [23], have been proposed. These works utilize domain, user, teaching, adaptation, and goal and constraints models to simplify the process of building adaptive systems. However, they focus on the closed corpus of documents, where existing resources in the corpus of the system are recommended, and thus often suffer from the tight coupling to the content.

GAF [24] framework introduced ontologies into the domain, resource, connect and group models and addressed the need for the open corpus adaptation where learning resources are not known at design time [25]. Some other examples of attempts to apply AEH techniques to the open corpus of documents are [26], [27], [28], [29], however, all of them assume clear separation between content creators and learners, where content creators are usually domain experts.

Most recently, social dimension has been added to the adaptive E-learning [30]. Frameworks such as SLAOS [31], [32], [33], ALEF [3], [34], Topolor [35] and Topolor 2 [36] integrate the social network functionality with the traditional AEH. In addition to studying personalized learning resources individually, users of social educational adaptive systems get the ability to participate collaboratively in the learning process. Learners are able to communicate to each other via direct messages, comments, and question and answer interfaces. Moreover, the line between authoring learning resources and consuming them is blurred in such systems as learners are able to annotate (e.g., tag, rate, comment, etc.) [37], define terms [38] or create new learning content [31]. The social dimension in adaptive E-learning also influenced works on student modeling and adaptation techniques. A number of works on open social student modeling [39], [40], [41], [42] and social based navigation algorithms [43, 44] showed that the adaptation based on the wisdom extracted from the work of a community of learners combined with traditional knowledge-based guidance supports more optimal content navigation [45]. In addition, the open (social) student models and, more recently, the gamificaiton of E-learning systems [36], [46], [47] showed increase in student engagement, motivation, and knowledge reflection.

As mentioned above, learners are able to collaboratively contribute to the content authoring process in social adaptive e-learning systems, however, very often contributions are only in a form of annotations to the main content created and indexed into domain model by an expert (teacher). A successful example of a system where only students contribute educational content is the PeerWise system [48], [49], however, they only focus on the question and answer format within one subject.

People apply learning-through-teaching in non-systematic ways at different stages of their careers. There have also been some efforts to utilize learning-through-teaching in an academic environment. For example, Herlocker et al. [50] report on college students using what they learn in their college classes to develop curricula and to teach it in an elementary school. To the best of our knowledge, this approach has never been applied as a large-scale systematic learning technique.

In our previous work, we have developed an integrated Exploratorium for database courses [2], [51] to investigate the technical problems and the benefits of different kinds of personalized learning tools. We performed several classroom studies to investigate different aspects of Exploratorium usage. Overall, the Exploratorium had been introduced to 229 students in six graduate and undergraduate database courses. In these studies, we adjusted various settings such as the presence of a particular kind of adaptation for a particular type of interactive learning content. The statistical evaluation showed that there was a significant correlation between students' performance in class and the amount of work they completed within the Exploratorium. While our experience with the Exploratorium was quite successful overall, it also revealed important scalability bottlenecks in the design and implementation of personalized learning tools. Personalized guidance does not scale well as the number and diversity of students increases. Efficiency of learning for different categories of students varies considerably for different topics. Diverse audiences need a much larger variety of content that simply cannot be produced and metadata-indexed by the original developers of personalized systems. In order to address this challenge we started exploring social personalization that expands the range of personalization techniques via broad use of social mechanisms.

In this paper, we propose a novel approach and build a system to explore and evaluate innovative personalization technologies based on social networking and crowdsourcing. Our approach is similar to SLAOS, ALEF, Topolor and Topolor 2 explained above, but we completely relax the requirement for the domain model and instead relay on adaptive algorithms and the wisdom of the crowd for content creation, indexing, sequencing and adaptation. Another difference is the format of the contribution that users make.

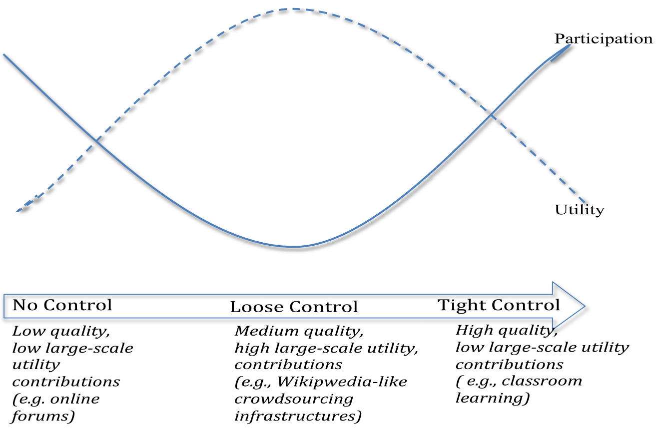

Fig. 1. Impact of degree of control on content contribution.

We require educational content to be contributed in a form of lesslets that extend multiple choice questions format of the PeerWise system and are explained in Section 3.1.

According to [52], crowdsourcing is "the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call". Generally speaking, the content quality and its utility for large user community (large-scale utility) depend on the degree of control that system imposes on content contributors. Fig. 1 explains this concept. Active participants of online forums voluntarily contribute large amounts of low-quality content. In general this content has low utility for large online user communities. For example, it is difficult to rank how useful specific forum postings are for different groups of users. Re-using those postings in different contexts (e.g., within other discussion branches) is also problematic. As another extreme, consider classroom teaching with tightly controlled contributions from teachers and students in form of class material, assignments, solutions and instructional feedback. Class participation is enforced by class policies and therefore is high. At the same time, classroom teaching targets small student groups and its large-scale content utility is low. Crowdsourcing environments, such as Wikipedia, impose a loose control on the content contributors, which reduces the fraction of active participants (less then one percent of Wikipedia users are active contributors) while increasing large-scale utility of the content.

3 SALT Framework

In this section we describe the proposed framework. In order to improve scalability of the personalized guidance we explore a social paradigm to support efficient personalization without expecting that content can be produced and metadata-indexed by the original developers of personalized systems. We propose a novel approach that performs adaptive content sequencing for specific groups of users. This is achieved by allowing users to be actively involved in the process of learning-through-teaching via lightweight social interactions.

3.1 SALT as Online Social Network

SALT is modeled as an Online Social Network similar to popular Facebook social network site. SALT extends the usefulness of OSN by utilizing collective intelligence to gain personalized educational knowledge. SALT users interact with each other by contributing and consuming concise and clear learning objects in a form of mini-lessons (we call them *lesslets*, analogous to applets, portlets, etc.). Thus, any user can seamlessly play the roles of both student and teacher. Each lesslet has four mandatory components: (1) a lesslet name, referring to the concept it is going to teach; (2) a clear and concise explanation of the concept; (3) an example illustrating the explanation in a simple and intuitive way; (4) a test to assess students' success in learning from the lesslet. The lesslets are placed under user created topics.

Fig. 2 (upper part) shows an example where Student 1 contributes a lesslet explaining the concept of a mathematical

Student 1 contributes lesslet L1:

Lesslet Name: Function
Explanation: Function is a relationship between two variables where for each value assumed by one there is a value determined for the other.
Example: Linear function: y=x, quadratic function: y = x^2
Test: Is y=sin(x) a function?
Correct answer: *Yes*

Student 2: I would teach it differently. Lesslet L2:

Lesslet Name: Function
Explanation: Function is a rule of correspondence (mapping) between two sets such that there is a unique element in the second set assigned to each element in the first set.
Example: set1 = {1,2,3}, set2 = {a,b,c,d}.
Mapping_A: 1→a, 2→b, 3→c is a function
Mapping_B: 1→a, 1→b, 3→c is a not a function (elements a and b are both assigned to 1)
Test: Which of the following mappings are functions:
Mapping_C: 1→a, 2→d, 3→d;
Mapping_D: 1→a, 2→d, 3→c, 2→c
Correct answer: *Mapping_C*

Fig. 2. Examples of lesslets L1 and L2.

Student 3 contributes lesslet L3:

Lesslet Name: Functional Dependency

Explanation: A functional dependency (FD) is a constraint between two sets of attributes S1 and S2 in a relation from a database where values of S1 uniquely determines values of S2 (denoted S1→S2)

Example:

Table Emp

	emp_name	salary
t1	Smith	1000
t2	Brown	2000
t3	Smith	4000

Functional dependency: **emp_name → salary** (employee can receive only one salary)
Tuples **t1** and **t3** are violating the above functional dependency,
So **emp_name →salary** does not hold on **Emp**

Test:

Table Emp1

	emp_name	age
t1	Smith	50
t2	Brown	35
t3	Smith	40

Does Functional dependency **emp_name → age** hold on **Emp1**?

Correct answer: *No*

Student 4:	L1 →	L3	Student 6:	L2 →	L3
Test:	success	failure	Test	success	success
Student 5:	L1 →	L3	Student 7:	L2 →	L3
Test:	success	success	Test:	success	success

Fig. 3. Example of learning pathway.

function. Student 2 explores this lesslet and decides that she would teach it in a different way. So, Student 2 then contributes the lesslet L2 (Fig. 2, lower part). SALT lesslets can be used as "stand-alone" lessons or in the larger context of learning pathways. Learning pathways are sequences of lesslets undertaken by a student. We also refer to learning pathways as leaps, which both abbreviates the term (LEArning Pathway), and expresses its intended purpose to advance learning.

From a pedagogical perspective, we believe interactions with SALT will not result in a shallow learning. Instead, we believe that by learning lesslets and taking tests, students engage low-order thinking of Bloom's taxonomy [53] of educational objectives such as remembering, understanding and applying known procedures to new data. And by creating lesslets students get involved into more metacognitive work that results in a deeper and richer learning experience [54], [55]. As one of the students in the system evaluation questionnaire said: "*Having me create a lesson for someone else to look at was a really effective way for me to learn a subject*". Also previous works [55], [56], [57] showed a significant correlation between summative assessment scores and levels of participation in generating self-assessment content.

As in any crowdsourcing system, users motivation to contribute high-quality lesslets is a valid question. In this paper, we don't specifically focus on the factors that would motivate users to participate in our system, however there is a large body of works in crowdsourcing and social computing areas that address users motivation. For example, some of the relevant motivational factors include desire to teach and contribute [58], community identification and social contact [59].

3.2 SALT as Complex Adaptive System (CAS)

SALT is designed as a Complex Adaptive System [60], [61], [62] adjusting its behavior in response to changing learning contexts and environments. Some common examples of CAS are stock markets where buyers, sellers and the market adapt to economic conditions and each other's actions, or harvester-ant colonies, which show the emergence of colonies from relatively simple tasks performed by the ants.

SALT implements self-organized personalization through learning pathways. SALT automatically sequences lesslets, making decisions on the estimated difficulty (currently by estimated difficulty we mean estimated score on the recommended lesslet's test) of each lesslet for a specific group of students based on similarity in performance of other users in the system. Each pathway is annotated with an aggregated assessment of user success in mastering its lesslets. SALT also maintains user profiles based on users' past performance. Any time a user starts exploring a lesslet, SALT recommends pathways that were most successful for similar users and include that lesslet. This approach facilitates the emergence of *most productive pathways* reflecting successful experience of SALT users.

To illustrate this approach, consider Student 3 who contributes the lesslet L3 teaching the concept of Functional Dependency in the context of relational databases (Fig. 3, upper part). Both L1 and L2 could help to better understand L3. However, explaining function as a mapping between two sets (L2) better fits the definition of functional dependency in L3. This is reflected in the learning pathways in Fig. 3 (lower part). We observe that all students who successfully passed L2 were also able to succeed in L3. Meanwhile, only one of the two students successfully passed L3 after successfully completing L1. Thus, SALT would favor the second learning pathway if students aim to learn the concept of Functional Dependency. In the future, we may have another lesslet L4 explaining functional dependency in a different way so that L1→L4 becomes more preferable as compared to L2→L4. The system will eventually adapt to both student skills and corresponding pathways. For example, depending on previous student learning and teaching history, some of the students may receive recommendations to follow the L2→L3 pathway, while others would be advised to take L1→L4, or even L1→L3.

3.3 SALT Adaptive Information Processing

The process of adaptive convergence to the most productive pathways is facilitated through dynamic data analysis techniques supported by SALT. SALT's major adaptive loop is shown in Fig. 4. SALT continuously assesses student performance along the learning pathways, and periodically suggests an optimal pathway for specific topics and groups of students. This information is used to rank lesslets and learning pathways in order to promote and recommend them as examples that can be utilized by other users in creating new lesslets. This adaptive loop is fully automatic and supported

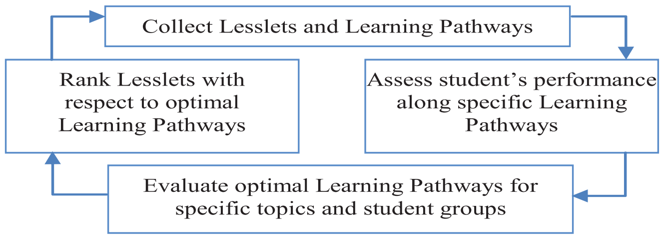

Fig. 4. SALT adaptive loop.

by a SALT data warehouse that performs periodic loading, aggregation and analysis of the collected information.

SALT generates large relationship graphs between lesslets, as well as bipartite graphs involving relationships between lesslets and users. As more learning-through-teaching experiences are accumulated in the SALT data warehouse, students and lesslets are grouped around similar performance profiles. After that, SALT discovers the most productive learning pathways with respect to a particular group of students and their performance profiles. In this way, the adaptive convergence of SALT becomes more efficient. This approach however suffers from typical problem in collaborative filtering (CF) recommendation systems – cold start. We are not addressing the cold start problem in this paper and will work on it in the future work.

3.4 SALT Implementation

We have implemented SALT system as a web application (see Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TLT.2016.2515097 for screen captures). On the back end, SALT is supported by MS SQL 2012 database server, IIS 8.0 web server and .NET 4.5 framework development environment running under Windows Server 2012. The system is written in C# with ASP.NET MVC framework. The full-text search, which allows searching for topics, lesslets, or users, is supported by Lucene.Net1 project. ASP.NET SignalR project supports asynchronous real-time long-running connection for online conversations. On the front end, in addition to ASP.NET views, we utilize D3 library for visualization of relationships between topics and lesslets. SALT is currently available for users to register by invitation only.

When users enter the system and click on a lesslet name, they get access to an explanation, example and test components for that lesslet. While learning a lesslet, users can leave personal notes or public comments, or even add alternative explanations of lesslet's description or example. After users finish learning a lesslet and take a test, the system shows users' score and provides recommended lesslets to take. SALT supports full-text search for lesslets. If there are no lesslets satisfying users' needs, they can either post a wish (this is basically similar to Q&A systems as stackexchange.com or quora.com) for the lesslet or create their own one. By posting a wish with a short description of the question, the user is asking other users to create a lesslet that answers her question. The user will be notified if someone answers her wish and the created lesslet will be available for everyone. Users can also subscribe for any number of topics in which case they will be notified if new lesslets are created under those topics.

SALT as OSN allows users to friend each other. Friends can see each other's activities and participate in online conversations via chat. In addition, when a user is on the learning lesslet page, SALT shows other users who are currently on that page as well. That allows users to collaborate and/or make new friends who have the same interests.

3.5 SALT as Crowdsourcing System

To put SALT into the perspective of crowdsourcing systems on the World Wide Web, we classify it along the dimensions discussed by Doan et al. [63].

Nature of collaboration. The nature of collaboration in SALT is both explicit and implicit. SALT users explicitly create lesslets and evaluate (using likes and comments) lesslets created by others. Additionally, when users take lesslets, they implicitly provide collective learning experiences that are used in the SALT adaptive loop.

Type of target problem. The main target is to build artifacts (lesslets), to evaluate them, and to organize them in personalized learning pathways. Users may also build communities based on common learning and/or teaching interests.

How to recruit and retain users. Recruitment in SALT is on a voluntary basis. There might be several encouragement and retention schemes implemented in SALT. Currently the "ownership" scheme is employed. When a user is accessing a lesslet, she can see who is the creator of the lesslet. The user profile page also includes all lesslets posted by the user with corresponding number of likes. In addition, SALT shows top lesslets in each topic with creator names. Adaptive learning pathways will also retain users, as they provide efficient personalized service.

What can users do. We have described main user activities in Sections 3.1 and 3.4. Most of the activities performed by users are cognitively demanding. However, to alleviate "hard" contribution, we devised the lesslet model as a short and concise lesson. In addition different user may play different roles (see Role of human users below).

How to combine user input. Even though there may be several lesslets in the system teaching the same concept, the lesslets are not merged. Meanwhile, SALT combines lesslets into personalized learning pathways automatically.

How to evaluate user input. Evaluation of created lesslets is twofold: automatic and crowdsourced. First, SALT automatically checks if all required parts of the lesslet are provided. After that SALT utilizes the collective intelligence of its users to continuously improve the quality of its repository (e.g., users can create alternative parts of lesslets, or new lesslets teaching the same concept, write comments, and leave a feedback in the form of likes and aggregate success ratios). The quality and accuracy of lesslets is assessed more efficiently as the scale of SALT grows. In addition, adaptive algorithms help SALT to converge to the best learning and teaching practices.

Role of human users. SALT users are both learners and content providers. By contributing versions of lesslets users help SALT to adapt to needs of other individual users.

4 SALT Evaluation

In this section we describe an initial evaluation of the SALT system that we performed. Particularly, we were interested

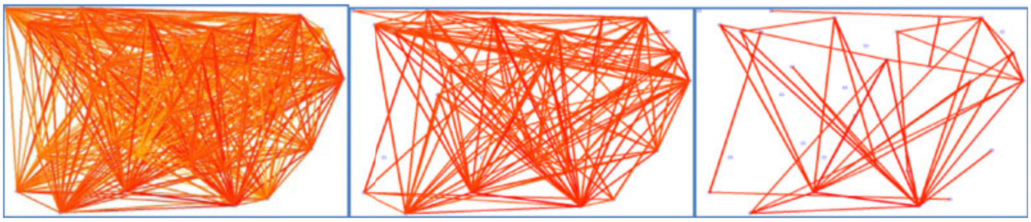

Fig. 5. Examples of relevance networks as we increase the similarity threshold.

in the following research questions: 1) how students group together based on their performance profiles (we report it in Section 4.2); 2) do different groups of users follow different successful learning pathways (we report it in Section 4.3.1); and 3) how collaborative filtering recommendation algorithms perform in SALT settings and whether they depend on the lengths of the learning pathways and the size of the neighborhood (we report it in Sections 4.3.4 and 4.3.5 respectively). We then summarize the results in the Section 5.

4.1 Evaluation Setup

We performed evaluation of the SALT system in both undergraduate and graduate database classes during four terms in the 2011 and 2012 academic years. In total, 260 students contributed about 300 lesslets under 20 different topics and produced over 23,000 activity records by leaving comments and votes, and exchanging messages. Students performed about 11,000 lesslet reviews and took about 8,200 tests. In addition to volunteering contributors, in each class we assigned a number of students to create a lesslet on a selected topic every week. Other students were asked to take each of those lesslets and to express their preferences using the “like” function.

4.2 Relevance Networks (RNs)-Based Analysis of Performance Profiles

To analyze similarities between SALT users for personalization and answer first research question, we applied relevance networks methodology [64]. Following this methodology, first we built a similarity graph reflecting pairwise similarities between students based on their performance along different learning pathways. After that we select subgraphs with similarities higher than a predefined threshold. Such subgraphs are called relevance networks. Fig. 5 shows an example of RNs for SALT users with similarity thresholds 0, 0.88, and 0.926 from left to right respectively. Each node is a student and each edge between two nodes represents students' similarity. As a similarity metric, we used cosine similarity between vectors of students' success rates on lesslet tests. The thickness and color of an edge reflects the strength of similarity. Thicker edges correspond to stronger similarities. More intense red color also indicates a stronger similarity. As we increase the threshold, weak similarity edges disappear resulting in RNs indicating groups of students with strongly similar SALT performance. These groups are used for SALT adaptation. For example, recommendations can be tailored for specific groups of students based on their similarity. Also the similarity among the students may indicate that the users have similar ways of learning.

In order to summarize the observed dynamics, we maintain several RN metrics including the number of associations (edges that surpass the threshold), the number of relevance networks, the number of participating nodes (coverage), and the connectivity (the ratio between the number of edges that surpassed the threshold and the number of all possible edges between participating nodes). Fig. 6 shows these four metrics for undergraduate and graduate students as we increase the threshold from 0 to 1 with 0.1 increments. We observe that undergraduate students (U_Spring_2012) form one stable network, which sustains thresholds lower then 0.9 (# of Relevance Networks is constant until threshold is 0.9). The number of associations and participating nodes are constant until the threshold is equal to 0.6 and 0.8, respectively. We observe different dynamics for graduate students (G_Spring_2012 and G_Fall_2012). As the threshold increases, the number of associations decreases almost linearly; however, the coverage (number of participating nodes) of the RN is stable until threshold is 0.4. The number of participating nodes decreases when threshold passes 0.4. In contrast to undergraduate students, graduate students demonstrate higher dynamics in the number of relevance networks. At 0.7 similarity threshold, the number of relevance networks increases to two in G_Spring_2012, whereas G_Fall_2012 peaked at five relevance networks at the threshold of 0.6. To summarize, students in two classes (U_Spring_2012 and G_Spring_2012) formed stable networks with the majority of students participating in those networks (at least 50 percent of students participated in RNs until similarity threshold reaches 0.9). In G_Fall_2012

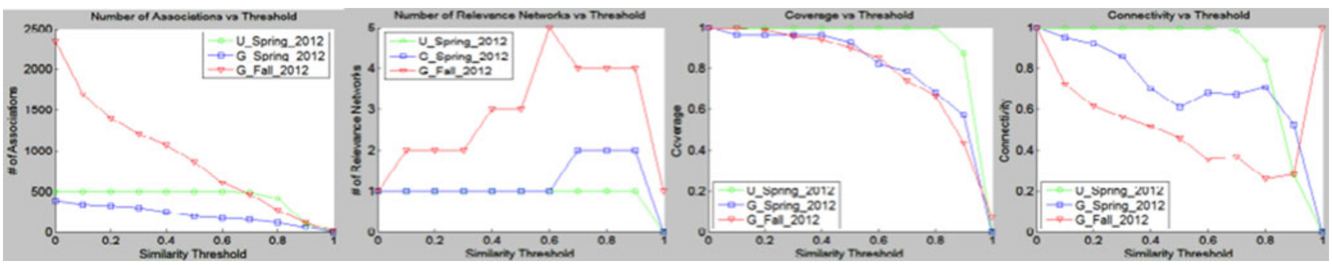

Fig. 6. Combined characteristics of similarity relevance networks as function of similarity threshold.

TABLE 1
SALT Activities Statistics

class	number of students	L – # of lesslets per weekly topic	pathway length	average worst success	average success	average best success
U_Spring_2012	19	4	8	54.82	79.58	97.43
G_Spring_2012	14	4	7	65.14	83.44	97.51
G_Fall_2012	16	6	11	45.49	77.03	98.06

around 50 percent of students also participated in RNs. However they were assigned to several distinct networks. We observe steep decreases in the number of associations and participating nodes in all classes around the threshold values of 0.9. As expected, we did not observe completely similar students (with similarity sustained threshold value of 1). The exception is G_Fall_2012 class, where there is one RN at similarity threshold 1, however the coverage is only about 10 percent. From this dynamic, we conclude that there is a considerable benefit from adaptive personalization based on SALT performance profiles of students.

4.3 Lesslet Recommendation Experiment

The objective of the lesslet recommendation experiments is to explore how efficiently SALT can adapt to user performance profiles using various collaborative filtering algorithms. We are particularly interested to see how SALT recommendations depend on the length of learning pathways and user/neighborhood similarity.

4.3.1 Learning Pathways

We explored learning pathways for all students. We considered alternative learning pathways with one lesslet per assigned topic. Thus, we explored up to L^{LP} learning pathways per student, where LP is the number of weekly topics (LP = pathway length + 1). For this study we ignored transitions of users between lesslets in the same weekly topic. We calculated average scores for each user's best and worst pathway, as well as average among all possible pathways. Then we averaged those numbers among all users for each class. The results are presented in Table 1. We observed significant differences in student performance along learning pathways.

We also calculated the number of different best pathways in each class. We consider two pathways to be different if the number of different lesslets in those pathways is greater than N (where N is between 0 and LP). Therefore, by changing N we can cluster users based on similarity of their best pathways (Fig. 7). From Fig. 7 we can see that almost all students in the three classes have different best pathways. Maximum number of clusters (horizontal blue line) for a class is equal to the number of students in that class. Both U_Spring_2012 (Fig. 7a) and G_Spring_2012 (Fig. 7b) classes have cluster with two completely similar users in terms of best pathways. In G_Fall_2012 (Fig. 7c) each students has their own unique best pathway. However, as we relax pathway's similarity requirements (increase N), number of clusters goes down and therefore users group together.

The results of this analysis show performance similarity groups that can be efficiently used to provide recommendations to users.

A major task of the SALT adaptive personalization engine is to recommend best lesslets that a user should take next. This task requires SALT to provide users with a ranked list of recommended lesslets based on the user prior performance and similarity with performances of other users.

The lesslet recommendation experiment was performed in the following way:

1) Select a user u from the dataset to be the target user who receives recommendations
2) for $w = 1$ to # of weekly topics – 1
   a) Create a temporal dataset which contains u's user history for w weekly topics and history of all other users for $w + 1$ weekly topics
   b) Calculate predicted scores for L lesslets of $w + 1$ weekly topic for the user u
3) Repeat 1-2 for all users
4) Construct best recommended pathways for all users
5) Evaluate recommendations
6) Repeat 1-5 for different similarity algorithms with different thresholds.

4.3.2 Recommendation Algorithms

To provide lesslet recommendations we applied various collaborative filtering algorithms using the Apache Mahout

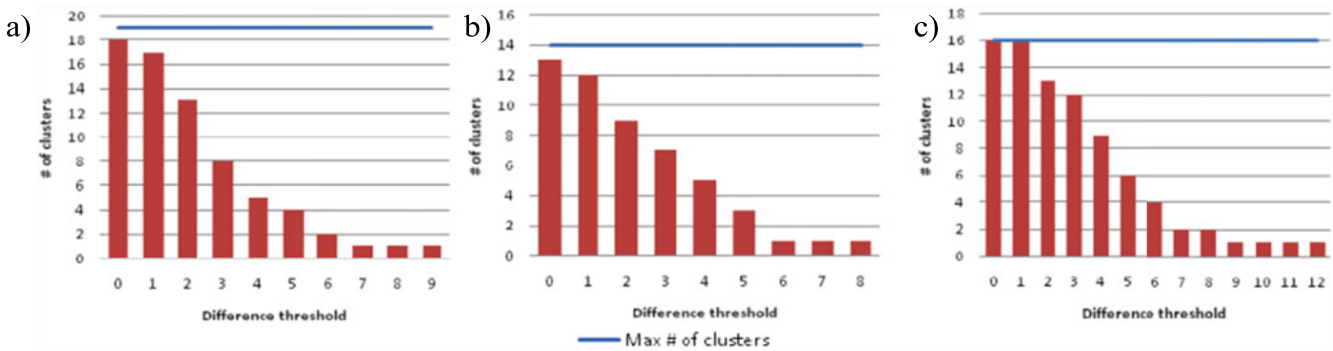

Fig. 7. Clustering of users around best pathways for a) U_Spring_2012, b) G_Spring_2012, and c) G_Fall_2012 classes.

0.7 machine learning library. The generic CF method applies the k nearest neighbors algorithm to predict user ratings. In our case CF algorithms predict lesslets' score. The algorithm computes a prediction for a target user for the lesslets that the user has not taken yet. The algorithm then returns the top lesslets ranked by the calculated prediction. The prediction is based on a number of k of nearest neighbors, which can either be user- or item-based [65]. We also experimented with a number of well-known similarity metrics implemented in Mahout: Pearson correlation [66], Euclidean distance, Spearman correlation [67], and Tanimoto coefficient (also known as Jaccard coefficient) [68].

4.3.3 Evaluation Measures

On each step of a learning pathway we know user's best lesslet based on the score she obtains on it. We also know recommended lesslets. A *penalty* is calculated if user's score on the best recommended lesslet is less than user's score on her best lesslet on a given pathway step. The penalty larger than zero means that the target user would score less by following recommendations. The higher the penalty the worse a recommendation algorithm is.

We start with two matrices $\mathbf{U} \in [0,100]^{|U|\times LP}$ and $\mathbf{R} \in [0,100]^{|U|\times LP}$ which contain lesslet test scores for all users on all steps of the learning pathways. The **U**. matrix contains real scores; the R. matrix contains predicted scores. The (*i*, *j*) th entries in both matrices are arrays of scores for the *i*th user on the *j*th step of the learning pathway.

With the number of users in the class denoted by |U| and the number of weekly topics denoted by LP, we first build two matrices $\mathbf{Ub} \in [0,100]^{|U|\times LP}$, $\mathbf{Urb} \in [0,100]^{|U|\times LP}$ with their respective (i,j)th entries as:

$$\mathrm{Ub}_{i,j} = max_{k=1:L}((\mathrm{U}_{i,j})_k), \tag{1}$$

$$\mathrm{Urb}_{i,j} = \mathrm{U}_{i,j,l}|\mathrm{R}_{i,j,l} = \max_{k=1:L}((R_{i,j})_k), \tag{2}$$

where $\mathrm{Ub}_{i,j}$ – maximum score of user *i* among the L lesslets on *j*th learning pathway's step; $\mathrm{Urb}_{i,j}$ – *i*th user's score on the best recommended lesslet on *j*th learning pathway's step.

Next, we calculate the normalized difference between the two matrices:

$$\mathbf{D} = \frac{1}{100}(\mathbf{Ub} - \mathbf{Urb}). \tag{3}$$

The penalty matrix $P \in [0,1]^{|U|\times LP}$ is calculated as follows:

$$\mathrm{P_{i,j}} = \begin{cases} \mathrm{D_{i,j}} & \text{if } (\mathrm{D_{i,j}} > 0) \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $P_{i,j}$ – penalty for recommender on user *i* on *j*th learning pathway's step.

Mean Cumulative Penalty (MCP). To show if the penalty depends on the length of the learning pathway we calculate the cumulative penalty matrix $CP \in [0,1]^{|U|\times LP}$:

$$\mathrm{CP}_{i,j} = \frac{\sum_{k=1}^{j} \mathrm{P}_{i,k}}{j}, \tag{5}$$

where $CP_{i,j}$ – cumulative penalty of user *i* on step *j*; $\sum_{k=1}^{j} P_{i,k}$– sum of penalties for user *i* on learning pathway from step 1 up to step *j*. If $CP_{i,j}$ decreases when *j* increases, then it means that recommendations for user *i* improve as the system accumulates more history of user's interactions. We then calculate one row matrix of aggregated cumulative penalties among all users on each pathway step $\mathrm{MCP} \in [0,1]^{1\times LP}$:

$$\mathrm{MCP}_{1,j} = \frac{\sum_{k=1}^{|\mathrm{U}|} \mathrm{CP}_{k,j}}{|\mathrm{U}|}, \tag{6}$$

where $MCP_{1,j}$ – mean cumulative penalty over all users on *j*th step of learning pathway.

C-Kendall's correlation coefficient. After a user completes a lesslet, SALT presents a list of recommended lesslets. Those lesslets are ordered (ranked) by predicted score in descending order. We use Kendall's tau ranking coefficient τ to calculate predicted order accuracy [69]. For each pathway step $\mathrm{j} \in 1..\mathrm{LP}$, we build two set S_j (7) and M_j (8) of $|\mathrm{U}| \times \mathrm{L}$ lesslets' ranks by users and recommender respectively

$$\mathrm{S}_j = \{\mathrm{rank}(\mathrm{U}_{1,j}), \mathrm{rank}(\mathrm{U}_{2,j}), \ldots, \mathrm{rank}(\mathrm{U}_{|\mathrm{U}|,j})\}, \tag{7}$$

$$\mathrm{M}_j = \{\mathrm{rank}(\mathrm{R}_{1,j}), \mathrm{rank}(\mathrm{R}_{2,j}), \ldots, \mathrm{rank}(\mathrm{R}_{|\mathrm{U}|,j})\}, \tag{8}$$

where $U_{i,j}$ – list of *i*th user's score on *L* lesslets on *j*th pathway step; $R_{i,j}$ – list of predicted scores for *i*th user's score on *L* lesslets on *j*th pathway step.

After building S_j and M_j, we calculate Kendall's tau correlation coefficient. The coefficient evaluates the degree of similarity between two sets. The value of 1 corresponds to perfect predictions, and the value of -1 corresponds to all possible rank mismatches. We build one row matrix with cumulative Kendall's correlation coefficients (9), which we denote with $\text{C-Kendall} \in [-1,1]^{1\times\mathrm{LP}}$

$$\text{C-Kendall}_{1,j} = \frac{\sum_{k=1}^{j} \mathrm{corr}(S_k, \mathrm{M}_k)}{\mathrm{j}}. \tag{9}$$

Coverage. Not all users might have a representative neighborhood for given properties (e.g., for given similarity threshold). Therefore, especially when similarity threshold is high, a recommender cannot produce any prediction for such users. The *Coverage* considers how many users can receive a recommendation [50], [70]. We build one-row matrix $\mathrm{Coverage} \in [0,100]^{1\times\mathrm{LP}}$. each element of which is a percentage of users who received recommendation (at least one lesslet) on each step of learning pathway:

$$\mathrm{Coverage}_{i,j} = \frac{1}{|\mathbf{U}|}\sum_{i=0}^{|\mathbf{U}|} \begin{cases} 1, & \text{if } \mathrm{R}_{i,j} \neq \emptyset) \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

4.3.4 Learning Pathway Length Analysis

In this section we analyze how users' learning history in form of learning pathway allows system to perform better recommendations. We evaluate a combination of two neighborhood-based recommendation algorithms with four similarity metrics as well as the SlopeOne algorithm. For this experiment we set the similarity threshold equal to 0 to include all users in the target user's neighborhood. Figs. 8, 9, and 10 show algorithms' performance versus learning pathway length for U_Spring_2012, G_Spring_2012 and

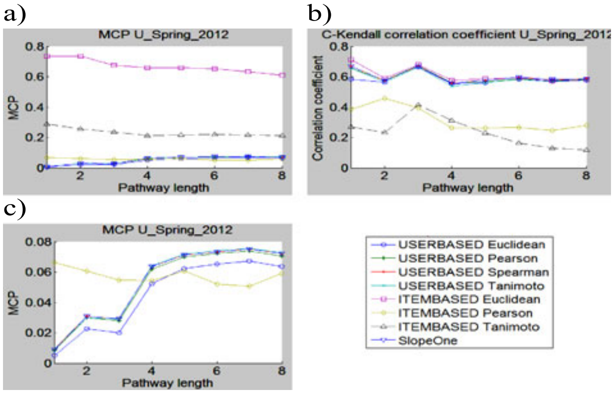

Fig. 8. Evaluation metrics for U_Spring_2012 class versus pathway length. Plot c) hides ITEMBASED euclidean and Tanimoto lines.

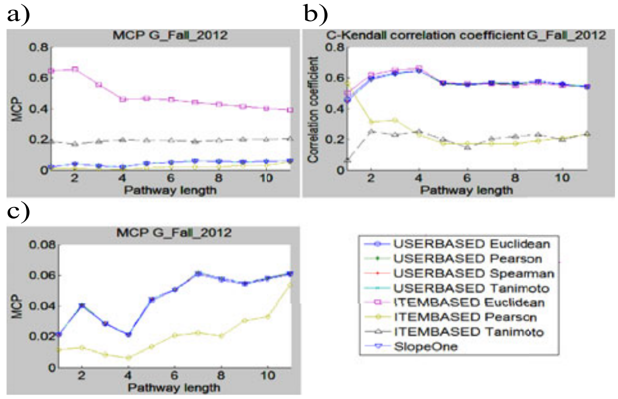

Fig. 10. Evaluation metrics for G_Fall_2012 class versus pathway length. Plot c) hides ITEMBASED euclidean and Tanimoto lines.

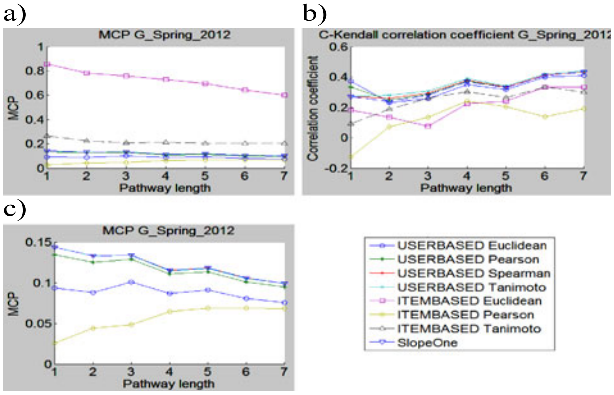

Fig. 9. Evaluation metrics for G_Spring_2012 class versus pathway length. Plot c) hides ITEMBASED euclidean and Tanimoto lines.

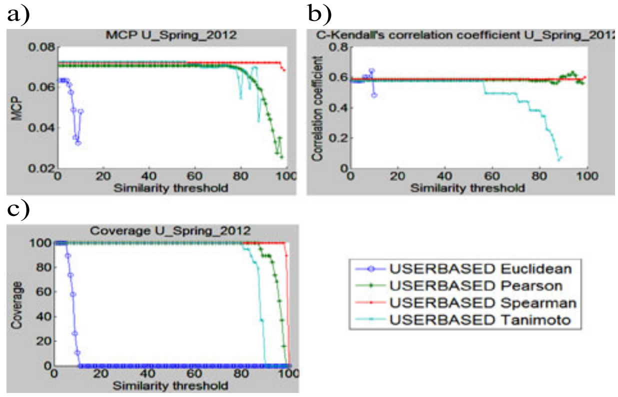

Fig. 11. Evaluation metrics for U_Spring_2012 class versus user similarity.

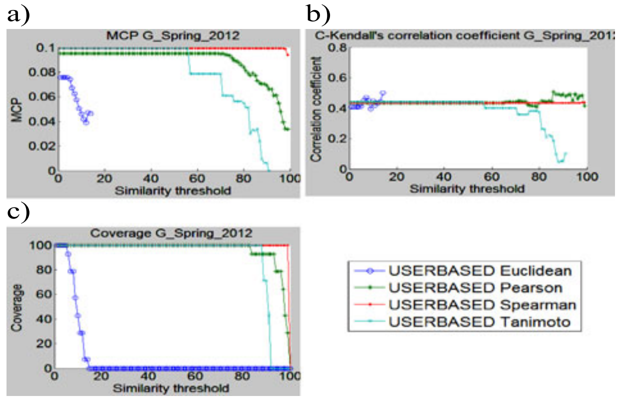

Fig. 12. Evaluation metrics for G_Spring_2012 class versus user similarity.

G_Fall_2012 classes respectively. Two of the item-based algorithms, ITEMBASED Euclidean and ITEMBASED Tanimoto (not shown on plot (c)), consistently performed worse than the other ones in terms of MCP in all classes. However, both of them showed either declining or stable trend. We would need to gather more data and to run ITEMBASED Euclidean on longer pathways to see if it keeps decreasing and converging to a small penalty. Meanwhile, the item-based algorithm with Pearson correlation similarity metric outperformed all other algorithms.

User-based and SlopeOne algorithms have almost identical performance in MCP. USERBASED Euclidean was slightly better than the rest of the algorithms. However, the trends change significantly between classes. In U_Spring_2012 class (Fig. 8c) the MCP value for the algorithms initially increased. Meanwhile, after the fifth step of a learning pathway the trend started to change and all graph reached a plateau, or even started to decline. That may be due to the fact that as users interact more with SALT the algorithms identify more similar users. In the G_Spring_2012 class (Fig. 9c) the performance graphs gradually declined as pathway increased. In the G_Fall_2012 class though (Fig. 10c), all algorithms showed an increasing trend in MCP values.

The predicted order accuracy, which is expressed as the C-Kendall correlation coefficient, demonstrated similar behavior for the U_Spring_2012 (Fig. 8e) and G_Fall_2012 (Fig. 10e) classes. All algorithms showed some fluctuations on short pathways, but leveled off around a pathway of size 4. In case of the G_Spring_2012 (Fig. 9e), all algorithms tended to increase in C-Kendal value.

4.3.5 *Similarity Threshold Analysis*

In this experiment we analyze how recommendation performance depends on the user similarity threshold. By changing the similarity threshold we change the size of the neighborhood that contributes to the predicted score for the target user. We plot MCP, C-Kendall and Coverage versus similarity threshold graphs at the longest pathway for each class in Figs. 11, 12 and 13. As mentioned earlier, Mahout does not support similarity based neighborhood for item-based algorithm, therefore we analyzed only user-based algorithms.

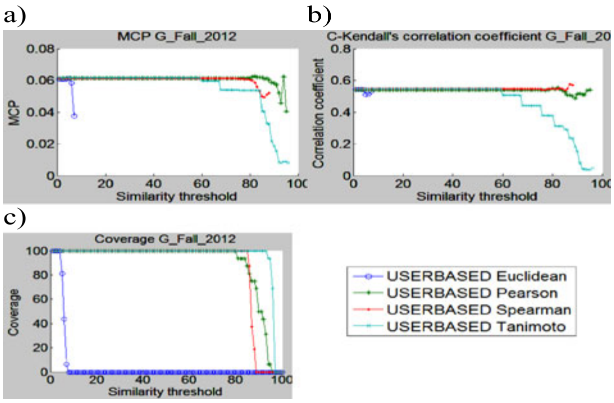

Fig. 13. Evaluation metrics for G_Fall_2012 class versus similarity threshold.

We observe similar performance of the algorithms in all three classes. Most of the algorithms do not change their performance until similarity threshold is high enough (around 60-80 percent). However, as the quality of recommendations starts to increase (in terms of MCP values), number of users who receive recommendations (coverage) decreases.

5 Conclusions

In this paper we present a novel social learning framework (SALT) that integrates social network functionality with traditional adaptive educational hypermedia to engage students into learning through teaching and adapt learning pathways to individual student needs based on collective learning experiences. We report on implementation and evaluation of the SALT system.

As the result of our relevance network analysis, we found that students tend to form stable clusters that survive very high cosine similarity threshold (50 percent of students survive similarity threshold up to 0.9). Meanwhile, our learning pathway analysis showed that almost all students have their own unique best pathway. Students, however, group around best pathways once we start relaxing pathway similarity threshold. Thus, to summarize our findings to answer first and second research questions, the results of the experiments suggest that it is possible to identify group of SALT users with similar learning patterns and use the intra-group similarities to perform personalization.

We further conducted experiments with various recommendation algorithms to answer the third research question. To answer the first part of the question and see how recommendation algorithms depend on the lengths of the learning pathways, we performed learning pathway length analysis (reported in Section 4.3.4). The results of experiments showed that most algorithms obtain very small penalty even on short pathways. However, different similarity measures performed differently in different classes as the length of pathway increased. That phenomenon requires further exploration and might be due to differences in students and class material. To answer the second part of the third research question, we performed similarity threshold analysis (reported in Section 4.3.5) that showed that smaller user neighborhood allowed recommender algorithms to perform better recommendation in terms of penalty, but with sharp decrease in user coverage.

SALT has been used in several courses at the School of Information Sciences, University of Pittsburgh. Over the last two years, we've been collaborating with WISER (www.wiser.pitt.edu) center to employ SALT for some courses at the School of Medicine, University of Pittsburgh. The most recent activities in SALT development include integration of SALT with Mastery Grid system [42] as a content provider.

In future work we plan to explore performance of other methods to facilitate adaptive convergence of SALT. This includes supervised and unsupervised learning, change detection and process mining techniques [71], [72], [73]. It should be noted that educational data mining is a relatively new discipline that, in addition to common machine learning techniques, utilizes psychometrics and other areas of statistics to explore specific types of educational data [74]. We will explore our SALT system as a unique educational data mining testbed that continuously accumulates and aggregates large-scale educational data. In addition we will consider the challenges summarized in [75], including stimulating users' engagement and quality assurance.

References

[1] P. Brusilovsky, "Adaptive and intelligent technologies for web-based education," *Künstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching*, vol. 4, pp. 19–25, 1999.
[2] P. Brusilovsky, S. Sosnovsky, D. H. Lee, M. Yudelson, V. Zadorozhny, and X. Zhou, "An open integrated exploratorium for database courses," *ACM SIGCSE Bull.*, vol. 40, pp. 22–26, 2008.
[3] M. Šimko, M. Barla, and M. Bieliková, "ALEF: A framework for adaptive web-based learning 2.0," in *Proc. IFIP TC 3 Int. Conf. Key Competencies Knowl. Soc.*, 2010, pp. 367–378.
[4] H. Vogten and R. Koper, "Towards a new generation of Learning Management Systems," in *Proc. 6th Int. Conf. Comput. Supported Educ.*, 2014, pp. 514–519.
[5] W. Rice, *Moodle 2.0 E-Learning Course Development*. Birmingham, U.K.: Packt Publishing Ltd, 2011.
[6] D. Sleeman and J. S. Brown, *Intelligent Tutoring Systems*. London, U.K.: Academic, 1982.
[7] M. C. Polson and J. J. Richardson, *Foundations of Intelligent Tutoring Systems*. East Sussex, U.K.: Psychology Press, 2013.
[8] P. Brusilovsky and C. Peylo, "Adaptive and intelligent web-based educational systems," *Int. J. Artif. Intell. Edu.*, vol. 13, pp. 159–172, 2003.
[9] P. Brusilovsky, "Adaptive hypermedia for education and training," in *Adaptive Technologies for Training and Education*, vol. 46. Cambridge, U.K.: Cambridge Univ. Press, 2012.
[10] C. Romero and S. Ventura, "Educational data mining: A survey from 1995 to 2005," *Expert Syst. Appl.*, vol. 33, pp. 135–146, 2007.
[11] R. S. Baker and P. S. Inventado, "Educational data mining and learning analytics," in *Learning Analytics*. New York, NY, USA: Springer, 2014, pp. 61–75.
[12] M. Yudelson, P. Brusilovsky, and V. Zadorozhny, "A user modeling server for contemporary adaptive hypermedia: An evaluation of the push approach to evidence propagation," in *Proc. 11th Int. Conf. User Model.*, 2007, pp. 27–36.
[13] V. Zadorozhny, M. Yudelson, and P. Brusilovsky, "A framework for performance evaluation of user modeling servers for web applications," *Web Intell. Agent Syst.*, vol. 6, pp. 175–191, 2008.
[14] P. Brusilovsky and E. Millán, "User models for adaptive hypermedia and adaptive educational systems," in *The Adaptive Web*. Berlin, Germany: Springer-Verlag, 2007, pp. 3–53.
[15] K. Chrysafiadi and M. Virvou, "Student modeling for personalized education: A review of the literature," in *Advances in Personalized Web-Based Education*. New York, NY, USA: Springer, 2015, pp. 1–24.
[16] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, "Recommender systems in technology enhanced learning," in *Recommender Systems Handbook*. New York, NY, USA: Springer, 2011, pp. 387–415.

[17] M. Holenko Dlab and N. Hoic-Bozic, “Recommender system for web 2.0 supported elearning,” in *Proc. Global Eng. Edu. Conf.*, 2014, pp. 953–956.

[18] H. Drachsler, *Navigation Support for Learners in Informal Learning Networks*. Heerlen, Netherlands: Open Universiteit Nederland, 2009.

[19] L.-H. Wong and C.-K. Looi, “Adaptable learning pathway generation with ant colony optimization,” *Educational Technol. Soc.*, vol. 12, pp. 309–326, 2009.

[20] P. De Bra, G.-J. Houben, and H. Wu, “AHAM: A Dexter-based reference model for adaptive hypermedia,” in *Proc. 10th ACM Conf. Hypertext Hypermedia*, 1999, pp. 147–156.

[21] M. Cannataro and A. Pugliese, “XAHM: An XML-based Adaptive Hypermedia Model and its implementation,” in *Proc. Int. Workshop Hypermedia: Openness, Struct. Awareness, Adaptivity*, 2002, pp. 252–263.

[22] N. Koch and M. Wirsing, “The Munich reference model for adaptive hypermedia applications,” in *Proc. 2nd Int. Conf. Adaptive Hypermedia Adaptive Web-Based Syst.*, 2002, pp. 213–222.

[23] A. I. Cristea and A. De Mooij, “LAOS: Layered WWW AHS authoring model and their corresponding algebraic operators,” presented at the *12th Int. World Wide Web Conf.*, Budapest, Hungary, 2003.

[24] E. Knutov, “GAF: Generic adaptation framework,” in *Proc. 5th Int. Conf. Adaptive Hypermedia Adaptive Web-Based Syst.*, 2008, pp. 400–404.

[25] P. Brusilovsky and N. Henze, “Open corpus adaptive educational hypermedia,” in *The Adaptive Web*, New York, NY, USA: Springer, 2007, pp. 671–696.

[26] Y.-l. Lin and P. Brusilovsky, “Towards open corpus adaptive hypermedia: A study of novelty detection approaches,” in *Proc. 19th Int. Conf. User Model., Adaption Personalization*, 2011, pp. 353–358.

[27] C. H. Muntean and G.-M. Muntean, “Open corpus architecture for personalised ubiquitous e-learning,” *Pers. Ubiquitous Comput.*, vol. 13, pp. 197–205, 2009.

[28] S. Sosnovsky, I. H. Hsiao, and P. Brusilovsky, “Adaptation ‘in the wild ontology-based personalization of open-corpus learning material,” in *21st Century Learning for 21st Century Skills*, New York, NY, USA: Springer, 2012, pp. 425–431.

[29] P. Brusilovsky, G. Chavan, and R. Farzan, “Social adaptive navigation support for open corpus electronic textbooks,” in *Proc. 3rd Int. Conf. Adaptive Hypermedia Adaptive Web-Based Syst.*, 2004, pp. 24–33.

[30] J. Vassileva, “Toward social learning environments,” *IEEE Trans. Learn. Technol.*, vol. 1, no. 4, pp. 199–214, *Oct.–Dec.* 2008.

[31] A. I. Cristea and F. Ghali, “Towards adaptation in e-learning 2.0,” *New Rev. Hypermedia Multimedia*, vol. 17, pp. 199–238, 2011.

[32] A. I. Cristea, F. Ghali, and M. Joy, “Social, personalized lifelong learning,” in *E-Infrastructures and Technologies for Lifelong Learning: Next Generation Environments*. Hershey, PA, USA: IGI Global, 2011, pp. 90–125.

[33] F. Ghali and A. I. Cristea, “Social reference model for adaptive web learning,” in *Proc. 8th Int. Conf. Adv. Web Based Learn.*, 2009, pp. 162–171.

[34] M. Bieliková, M. Šimko, M. Barla, J. Tvarožek, M. Labaj, R. Móro, et al., “ALEF: From application to platform for adaptive collaborative learning,” in *Recommender Systems for Technology Enhanced Learning*. New York, NY, USA: Springer, 2014, pp. 195–225.

[35] L. Shi, G. Gkotsis, K. Stepanyan, D. Al Qudah, and A. I. Cristea, “Social personalized adaptive e-learning environment: Topolor-implementation and evaluation,” in *Proc. 16th Int. Conf. Artif. Intell. Educ.*, 2013, pp. 708–711.

[36] L. Shi and A. I. Cristea, “Making it game-like: Topolor 2 and gamified social e-learning,” presented at the *22nd Conf. User Model., Adaptation Personalization*, Aalborg, Denmark, 2014.

[37] A. Y. S. Su, S. J. H. Yang, W.-Y. Hwang, and J. Zhang, “A Web 2.0-based collaborative annotation system for enhancing knowledge sharing in collaborative learning environments,” *Comput. Educ.*, vol. 55, pp. 752–766, 2010.

[38] M. Svrček and M. Šimko, “Supporting educational content enrichment and learning via student-created definitions,” in *Proc. Int. Workshops New Horizons Web Based Learn.*, 2014, pp. 44–54.

[39] I. H. Hsiao, F. Bakalov, P. Brusilovsky, and B. König-Ries, “Open social student modeling: visualizing student models with parallel introspectiveviews,” in *Proc. 19th Int. Conf. User Model., Adaption Personalization*, 2011, pp. 171–182.

[40] M. Liang, J. Guerra, G. E. Marai, and P. Brusilovsky, “Collaborative e-learning through open social student modeling and Progressive Zoom navigation,” in *Proc. 18th Int. Conf. Collaborative Comput.: Netw., Appl. Worksharing*, 2012, pp. 252–261.

[41] L. Shi, A. I. Cristea, and S. Hadzidedic, “Multifaceted open social learner modelling,” in *Proc. 13th Int. Conf. Adv. Web-Based Learning*, 2014, pp. 32–42.

[42] T. D. Loboda, J. Guerra, R. Hosseini, and P. Brusilovsky, “Mastery grids: An open source social educational progress visualization,” in *Proc. 9th Eur. Conf. Technol. Enhanced Learn.: Open Learn. Teaching Educational Communities*, 2014, pp. 235–248.

[43] I. H. Hsiao, F. Bakalov, P. Brusilovsky, and B. König-Ries, “Progressor: Social navigation support through open social student modeling,” *New Rev. Hypermedia Multimedia*, vol. 19, pp. 112–131, 2013.

[44] P. Brusilovsky, Y. Lin, C. Wongchokprasitti, S. Britell, L. M. L. Delcambre, R. Furuta, et al., “Social navigation support for groups in a community-based educational portal,” in *Proc. Int. Conf. Theory Practice Digital Libraries: Res. Adv. Technol. Digital Libraries*, 2013, pp. 429–433.

[45] R. Hosseini, I. H. Hsiao, J. Guerra, and P. Brusilovsky, “Off the beaten path: The impact of adaptive content sequencing on student navigation in an open social student modeling interface,” in *Proc. 17th Int. Conf. Artif. Intell. Educ.*, 2015, pp. 624–628.

[46] A. C. T. Klock, L. F. da Cunha, M. F. de Carvalho, B. E. Rosa, A. J. Anton, and I. Gasparini, “Gamification in e-learning systems: A conceptual model to engage students and its application in an adaptive e-learning system,” in *Proc. 2nd Int. Conf. Learn. Collaboration Technol.*, 2015, pp. 595–607.

[47] L. Shi, A. I. Cristea, S. Hadzidedic, and N. Dervishalidovic, “Contextual gamification of social interaction—Towards increasing motivation in social E-learning,” in *Proc. 13th Int. Conf. Adv. Web-Based Learn.*, 2014, pp. 116–122.

[48] P. Denny, A. Luxton-Reilly, and J. Hamer, “The PeerWise system of student contributed assessment questions,” in *Proc. 10th Conf. Australasian Comput. Educ.*, 2008, pp. 69–74.

[49] J. Hardy, S. P. Bates, M. M. Casey, K. W. Galloway, R. K. Galloway, A. E. Kay, et al., “Student-generated content: Enhancing learning through sharing multiple-choice questions,” *Int. J. Sci. Educ.*, vol. 36, pp. 2180–2194, 2014.

[50] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Trans. Inf. Syst.*, vol. 22, pp. 5–53, 2004.

[51] P. Brusilovsky, S. Sosnovsky, M. V. Yudelson, D. H. Lee, V. Zadorozhny, and X. Zhou, “Learning SQL programming with interactive tools: From integration to personalization,” *ACM Trans. Comput. Educ.*, vol. 9, p. 19, 2010.

[52] J. Howe, “The rise of crowdsourcing,” *Wired Mag.*, vol. 14, pp. 1–4, 2006.

[53] D. R. Krathwohl, “A revision of Bloom’s taxonomy: An overview,” *Theory Practice*, vol. 41, pp. 212–218, 2002.

[54] J. Sener, “In search of student-generated content in online education,” *E-Mentor*, vol. 4, p. 21, 2007.

[55] M. Sanchez-Elez, I. Pardines, P. Garcia, G. Miñana, S. Roman, M. Sanchez, et al., “Enhancing students’ learning process through self-generated tests,” *J. Sci. Educ. Technol.*, vol. 23, pp. 15–25, 2014.

[56] K. W. Galloway and S. Burns, “Doing it for themselves: students creating a high quality peer-learning environment,” *Chemistry Educ. Res. Practice*, vol. 16, pp. 82–92, 2015.

[57] L. T. Frase and B. J. Schwartz, “Effect of question production and answering on prose recall,” *J. Educational Psychol.*, vol. 67, pp. 628–635, 1975.

[58] M. J. Raddick, G. Bracey, P. L. Gay, C. J. Lintott, P. Murray, K. Schawinski, et al., “Galaxy zoo: Exploring the motivations of citizen science volunteers,” *Astronomy Educ. Rev.*, vol. 9, 2010, http://portico.org/stable?au=pgg3ztfdp8z

[59] N. Kaufmann, T. Schulze, and D. Veit, “More than fun and money. Worker motivation in crowdsourcing—A study on mechanical turk,” in *Proc. 17th Americas Conf. Inf. Syst.*, 2011, pp. 1–11.

[60] B. H. C. Cheng, R. de Lemos, H. Giese, P. Inverardi, J. Magee, J. Andersson, et al., *Software Engineering for Self-Adaptive Systems: A Research Roadmap*, Berlin, Germany: Springer, 2009.

[61] H. He, *Self-Adaptive Systems for Machine Intelligence*. New York, NY, USA: Wiley, 2011.

[62] J. H. Miller and S. E. Page, *Complex Adaptive Systems: An Introduction to Computational Models of Social Life: An Introduction to Computational Models of Social Life*. Princeton, N, USA: Princeton Univ. Press, 2009.

[63] A. Doan, R. Ramakrishnan, and A. Y. Halevy, "Crowdsourcing systems on the world-wide web," *Commun. ACM*, vol. 54, pp. 86–96, 2011.

[64] V. Zadorozhny, L. Raschid, and A. Gal, "Scalable catalog infrastructure for managing access costs and source selection in wide area networks," *Int. J. Cooperative Inf. Syst.*, vol. 17, pp. 77–109, 2008.

[65] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, 2001, pp. 285–295.

[66] C. Desrosiers and G. Karypis, "A comprehensive survey of neighborhood-based recommendation methods," in *Recommender systems handbook*. New York, NY, USA: Springer, 2011, pp. 107–144.

[67] S. Owen, R. Anil, T. Dunning, and E. Friedman, *Mahout in Action*. Greenwich, CT, USA: Manning, 2011.

[68] T. T. Tanimoto, "Elementary mathematical theory of classification and prediction," 1958.

[69] H. Abdi, "The Kendall rank correlation coefficient," in *Encyclopedia of Measurement and Statistics*. Thousand Oaks, CA, USA: Sage, 2007, pp. 508–510.

[70] A. Bellogín, I. Cantador, F. Díez, P. Castells, and E. Chavarriaga, "An empirical comparison of social, collaborative filtering, and hybrid recommenders," *ACM Trans. Intell. Syst. Technol.*, vol. 4, p. 14, 2013.

[71] R. Nisbet, J. Elder Iv, and G. Miner, *Handbook of Statistical Analysis and Data Mining Applications*. New York, NY, USA: Access Online via Elsevier, 2009.

[72] P.-N. Tan, *Introduction to data mining*. Noida, India: Pearson Education, 2007.

[73] W. M. P. van der Aalst, *Discovery, Conformance and Enhancement of Business Processes*. New York, NY, USA: Springer, 2011.

[74] R. Baker and K. Yacef, "The state of educational data mining in 2009: A review and future visions," *J. Educational Data Mining*, vol. 1, pp. 3–17, 2009.

[75] D. S. Weld, E. Adar, L. Chilton, R. Hoffmann, and E. Horvitz, "Personalized online education—A crowdsourcing challenge," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 1–31.

Evgeny Karataev received the Dipl-Ing in computer science in 2011 from the Tomsk State University of Control Systems and Radioelectronics, Russia. He is currently a doctoral student in the Information Science & Technology program at the School of Information Sciences, University of Pittsburgh. His research interests include combination of crowdsourcing techniques with data management, data integration, complex adaptive, and distributed systems.

Vladimir Zadorozhny received the PhD degree in 1993 from the Institute for Problems of Informatics, Russian Academy of Sciences, Moscow. He is an associate professor of information sciences at the University of Pittsburgh. Before coming to the USA, he was a principal research scientist in the Institute of System Programming, Russian Academy of Sciences. Since May 1998, he was a research associate in the University of Maryland Institute for Advanced Computer Studies at College Park until he joined the University of Pittsburgh in September 2001. His research interests include networked information systems, heterogeneous information integration and data fusion, complex adaptive systems and crowdsoursing, query optimization in resource-constrained distributed environments, and scalable architectures for wide-area environments with heterogeneous information servers. He has received several Best Paper Awards and has chaired and served on program committees of multiple Database and Distributed Computing Conferences and Workshops. He is a member of the IEEE.