5-2019

PeerLens: Peer-inspired interactive learning path planning in online question pool

Meng XIA

Mingfei SUN

Huan WEI

Qing CHEN

Yong WANG

See next page for additional authors

Author
Meng XIA, Mingfei SUN, Huan WEI, Qing CHEN, Yong WANG, Lei SHI, Huamin QU, and Xiaojuan MA

PeerLens: Peer-inspired Interactive Learning Path Planning in Online Question Pool

Meng Xia[1], Mingfei Sun[1], Huan Wei[1], Qing Chen[1], Yong Wang[1], Lei Shi[2], Huamin Qu[1], Xiaojuan Ma[1]
[1]Department of Computer Science and Engineering, HKUST, Hong Kong, China
[2]School of Computer Science, Beihang University, China
{iris.xia,mingfei.sun,hweiad,qchenah,ywangct}@connect.ust.hk;leishi@buaa.edu.cn;{huamin,mxj}@cse.ust.hk

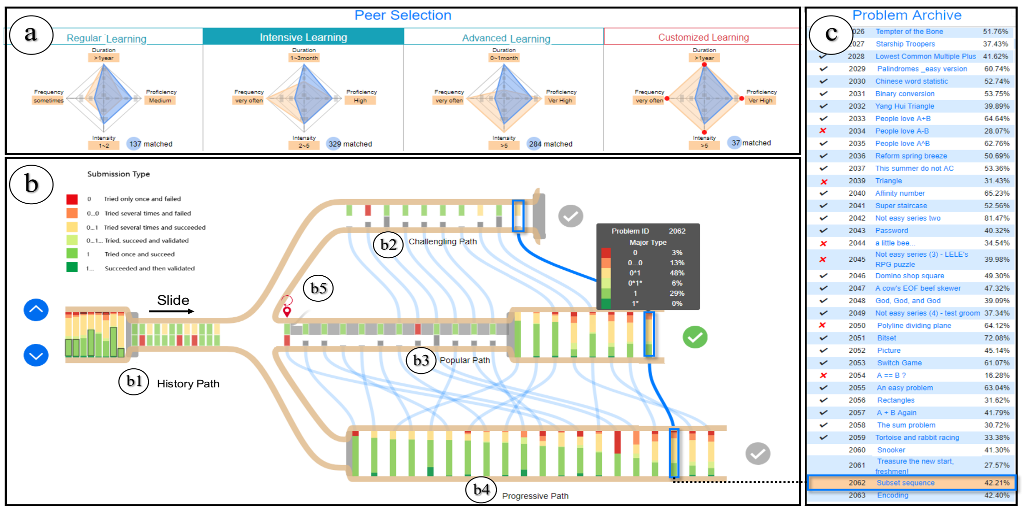


Figure 1: *PeerLens* has views of peer selection (*a*), learning path (*b*), and problem archive (*c*). The history path is compared with a peer group's in *b1*; future paths, challenging (*b2*), popular (*b3*), and progressive (*b4*) are shown with the current problem (*b5*).

ABSTRACT

Online question pools like LeetCode provide hands-on exercises of skills and knowledge. However, due to the large volume of questions and the intent of hiding the tested knowledge behind them, many users find it hard to decide where to start or how to proceed based on their goals and performance. To overcome these limitations, we present PeerLens, an interactive visual analysis system that enables peer-inspired learning path planning. PeerLens can recommend a customized, adaptable sequence of practice questions to individual learners, based on the exercise history of other users in a similar learning scenario. We propose a new way to model the learning path by submission types and a novel visual design to facilitate the understanding and planning of the learning path. We conducted a within-subject experiment to assess the efficacy and usefulness of PeerLens in comparison with two baseline systems. Experiment results show that users are more confident in arranging their learning path via PeerLens and find it more informative and intuitive.

CCS CONCEPTS

• **Information systems** → **Personalization**; • **Human-centered computing** → **Visual analytics**;

KEYWORDS

Question Pool; Learning Path Planning; Visualization

ACM Reference Format:

Meng Xia, Mingfei Sun, Huan Wei, Qing Chen, Yong Wang, Lei Shi, Huamin Qu, Xiaojuan Ma. 2019. PeerLens: Peer-inspired Interactive Learning Path Planning in Online Question Pool. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4-9, 2019, Glasgow, Scotland UK.* ACM, New York, NY, USA. Paper 634, 12 pages. https://doi.org/10.1145/3290605.3300864

CHI 2019, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5970-2/19/05...$15.00
https://doi.org/10.1145/3290605.3300864

1 INTRODUCTION

A question pool is a collection of questions for learners to practice their knowledge online [8]. Question pools can complement online and offline lectures, allowing people to acquire knowledge or hone skills for assignments, exams, interviews and so on [7]. For example, the programming question pool (e.g., LeetCode [19]), a.k.a., online judge, used for coding practice has the largest user base among all types of question pools (more than 30 million according to Wasik et al's ACM Computer Survey paper [37]). Online judge is an important complement to MOOC-style learning, complementary to MOOC with a focus on self-test exercises.

Despite their popularity, the interface of question pools is often less user-friendly. The materials (i.e., questions) in question pool are different from traditional MOOC videos which follow a pre-determined syllabus created by the instructor [5, 43]. As shown in Figure 1(c), the questions are mostly displayed as a lengthy list and indexed by their problem IDs assigned upon creation. Users could be overwhelmed by the large number of problems (up to thousands) in a single question pool [30, 40]. Meanwhile, according to our empirical observations, the real intent of the questions in these online pools is often not revealed literally for examination purposes, though in some cases, a brief description and/or the user feedback of each question is provided. Without such semantic information, it is often difficult for users to determine an appropriate order in taking these online questions for their particular learning scenario. This we call the learning path planning problem.

We surveyed the top-20 online judge platforms listed in Wikipedia [25] and found that only two platforms (CodeWars [22] and LeetCode [19]) support the functionality to recommend similar questions to take next, which still cannot suggest a complete learning path for different users and learning scenarios. To this end, there is a strong demand for the customized learning path planning in the context of existing list-based question pools.

Prior studies have tried to tackle learning path planning from two aspects. One class of works provide the adaptive learning path planning for online courses by extracting a knowledge graph from a pre-defined course syllabus [6, 32]. However, such approaches are often not applicable to online question pools, which do not have structured syllabuses due to the convention to not reveal the associated knowledge taxonomy. Another group of researchers analyzed the action log of existing platform users (e.g., the questions taken, the accuracy, and the online time) to conduct the algorithmic learning path planning [30, 42]. Though certain successes have been achieved, these automatic planning algorithms cannot adapt to the variety of user requirements and their evolving learning scenarios [27, 45]. In addition, the existing learning path planning algorithms work as the black box and it is often hard for users to interpret the recommended path to assist their learning process.

In this paper, we introduce *PeerLens*, a visual analysis system to help learners interactively plan learning paths in online question pools based on the inspiration of their peer learners. The system provides both an informative visual summary of the existing learning path of peer learners and a detailed explanation of the suggested learning paths for planning. In particular, we discover three typical learning scenarios, i.e., regular learning, intensive learning, and advanced learning, by classifying peer learners according to four attributes, i.e., learning duration, frequency, intensity, and proficiency. Users are allowed to compare their existing learning paths with their peer groups to identify a desired learning scenario. By defining the problem submission types, we model both the learning path of the associated peer group and the user's learning path in history. Based on this modeling, we derive three future learning paths: the popular learning path selected by default, the challenging path, and the progressive path. These alternatives meet the specialized requirements of learners at different levels.

The contributions of this work are summarized as follows.

- We propose an integrated zipper-like visual metaphor to represent the historical learning path and the multiple future suggested paths for diversified objectives. An interactive visual analytics system is designed and implemented to facilitate the customized learning path planning through identifying the relevant peer learner group, comparing their learning performance, and finally selecting the promising future learning path under the targeted learning scenario.
- We have introduced a new way to model the learning path by learners' submission behaviors. The new modeled learning path can imply problem difficulty as well as learners' performance, which facilitates visual representation and learning path suggestion.
- We conduct a within-subject experiment to evaluate the performance of *PeerLens*, in comparison with two baseline systems. Experiment results indicate that users are more confident in planning their learning path via *PeerLens*, and they find the system to be more informative and intuitive.

2 RELATED WORK

This section reviews the literature on educational recommendation techniques, and event sequence queries as well as visualizations.

Educational Recommendation Techniques

Many recommendation techniques have been applied in the education domain, which mainly include memory-based techniques and model-based techniques [1, 9].

Memory-based techniques continuously analyze all current data to recommend learning materials and can be classified into three categories [9]. Content-Based (CB) recommends items based on relationships between learning materials (e.g., Chu *et al.* [6]). Collaborative Filtering (CF) recommends items that were used by other similar learners based on the user information such as user ratings (e.g.,Toledo *et al.* [40]).Hybrid techniques consider both learning material and user-related information. For example, Salehi *et al.* [29] recommended learning materials based on materials' sequences and learners' preferences. Concealment of the real intent behind questions for examination purposes and numerous questions under the same learning concept block the way to use CB-based methods. Finding similar learners using CF is not always easy since no record can be accessed for beginners.

Model-based techniques make use of a large amount of data to model the learning process over time as an event sequence and recommend learning materials. For example, Piech *et al.* applied RNN to modeling and predicting learner performance in solving a sequence of questions, which is further refined in [41] to improve the prediction accuracy. However, RNN-based deep learning models are non-transparent and hard to interpret. More human-understandable and interpretable models, such as Markov Chain, have also achieved learning materials recommendation by calculating the transition probability of a group of learners [26, 31, 42]. For example, Huang *et al.* [42] proposed a Markov Chain Model to help learners achieve effective web-based learning transfer based on group-learning paths.

Most previous works, such as [30, 42], are simplified to an order of learning material without considering learner behaviors (e.g., repetition of the same learning material). However, in a question pool, user behaviors towards a specific problem can imply the learner's habits as well as the difficulty level of the problem, which cannot be ignored. Inspired by click stream modeling [35], we present a new method to model the learning path in a question pool utilizing learner behavior (submission times, solving time) and then use existing Markov Chain methods to achieve learning path suggestions.

Event Sequence Queries

To learn from peer sequences, the first step is to help learners to find a group of peer learning sequences that fit his or her learning scenario in terms of learning duration, learning frequency and so on. It is challenging since temporal event sequences consist of multiple attributes and are usually abundant with hundreds or even thousands of steps. Similan *et al.* [39] formulated the sequence query as to finding other event sequences that are similar to a given event sequence. They defined similarity metrics as the editing distance between two sequences, i.e., the number of swaps, missing or extra events to make one sequence exactly the same as another. The output is a ranked list of similar records. Since users do not need to specify the query rules, the similarity scores are hard to interpret. Moreover, it is complex to adjust parameters when using the control panels. Other methods are then proposed to assist users to specify temporal queries, such as required events, temporal relationships between events, and attribute ranges of the events or records [17, 21, 28]. This process requires specific query rules from users to obtain results that exactly match their queries.

Although querying by rules is complex and needs predefined filtering conditions, it could still provide accurate sequences when the user is familiar with features of sequences. To meet this demand, we incorporate this method into our system. To make the filtering process easy, we use a radar chart to help learners set their own learning scenario by four features: learning duration, learning frequency, learning intensity and learning proficiency.

Event Sequence Visualizations

Since a suggested learning path is a sequence of events, one straightforward way is to place events along a horizontal time axis, such as Lifelines [24], CloudLines [18] and TimqueSlice[44]. Episogram [4] draws vertical threads on top of a horizontal timeline to represent events that belong to specific conversations or topics. Sung *et al.* used theme river to visualize the MOOC comments [34]. We take advantage of these designs and use the horizontal axis to show results tried by learners, or aggregated results attempted by a group of learners on each question in chronological order.

While the work mentioned above can show details of one path, sometimes it is necessary to show multiple records at the same time. EventAction [11] used a calendar view to show several time event sequences and placed them in a ranking list to show similarity distance with query sequence. OutFlow [39] summarized multiple event sequences as a network. Despite the three suggested learning paths, we also need to show features of each suggested path to better support the reasoning of recommendation results. Therefore, we propose a novel way to visualize learning paths to balance the detailed information of each sequence with relationships between several sequences.

3 REQUIREMENT ANALYSIS

This paper aims to facilitate learners to plan a learning path in online question pools based on peer learning paths. We worked closely with two experts (E1, E2) in online learning and two online question pool users (S1, S2) to extract the detailed requirements and collect their feedback. E1 is the creator of one famous online judge and has coached International Collegiate Programming Contest in a university for over 10 years. E2 is the operations manager of an online question pool who has collected lots of learner feedback for their platform. S1 and S2 are postgraduate students in the computer science department of our university who have at least four-year experience in using online judge question pools. We conducted detailed discussions with them through emails, Skype meetings and face-to-face discussions. Based on their feedback and a survey on previous studies, we have compiled a list of requirements as follows.

R1: Find peers for a specific learning scenario. Both the two experts and the two students mentioned that different learners have different goals and scenarios when practicing on online question pools. For example, their motivation for using online question pools may be preparing for an IT company interview in two weeks later or systematically improving their coding and algorithm skills within several months. In different scenarios, learners often want to know what the learning paths of other learners are in a similar scenario and how their final performance is. A similar observation is also reported by Janssen *et al.* [15].

R2: Compare with peers' performance. When learners find several peer learning paths that they want to follow, they are often interested in differences between themselves and their peer group in terms of learning diligence and performance (E2, S1, S2). Such a comparison can help learners more accurately evaluate their performance and motivate learners to follow the correct learning path.

R3: Offer flexible learning path suggestions. According to our discussions with the experts and the students, it is also necessary to provide learners with flexible learning path choices that can satisfy their specific needs. Since even when two learners have the same learning goals, their learning status and capabilities may vary a lot. Therefore, it is necessary to offer learners flexible learning path suggestions [9].

R4: Provide convenient interaction and intuitive visual designs for learning path planning. To help learners quickly identify appropriate learning paths for themselves, it is important to enable learners to conveniently interact with the system. Also, considering that the target learners of *PeerLens* are general learners of online question pools, they do not necessarily have a background in data visualization (E2,S1). Therefore, it is critical to provide learners with intuitive visual designs to help them easily understand the encoded information and the suggested learning paths.

4 SYSTEM OVERVIEW

Based on the above requirements, we have designed *PeerLens* to visualize different peer learning paths and facilitate learners to interactively plan their learning paths. Figure 2 illustrates the system architecture, which consists of three major modules: (1) data collection and preprocessing, (2) path planning engine and (3) visualization. The data collection and prepossessing module crawls data from the website and further preprocesses it, e.g., filtering out the empty items.

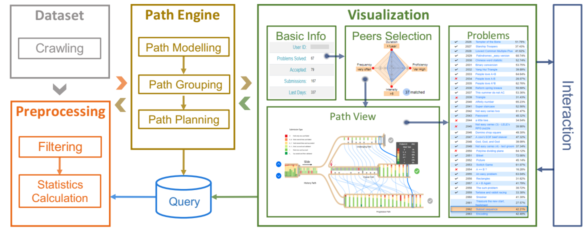


Figure 2: The system architecture of *PeerLens*.

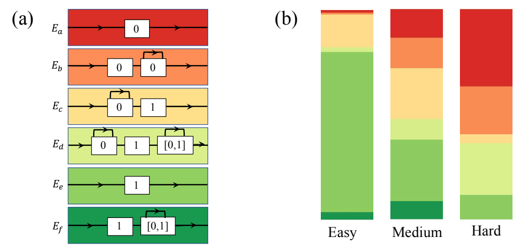


Figure 3: Submission types and their distribution: (a) the six submission types: 1 means *solved*, 0 otherwise; (b) submission type distributions of *Easy*, *Medium* and *Hard* questions.

The path engine module models learning path, groups peer learning paths (R1), and recommends learning paths (R3). The visualization module uses multiple coordinated views to support learning path comparison and planning (R2, R3).

We collected data from one popular online judge, Hangzhou Dianzi University Online Judge, with the owner's consent [16]. We focus on the recent submission records after 2017, which consist of 4625907 submission records from 53617 learners and 5166 programming questions. Each record includes submission time, judge status, problem ID and learner ID.

5 PATH PLANNING ENGINE

The learning path planning engine models the learning path, groups similar learning paths according to four important attributes and further forms learning path suggestions.

Learning Path Modeling

We use *submission type* to describe how a learner solves a specific problem P_i. Figure 3 shows the six submission types $E = \{E_a, E_b, E_c, E_d, E_e, E_f\}$ defined in this paper, where E_a denotes one failed attempt without success, E_b denotes multiple failed attempts without success, E_c denotes multiple failed attempts followed by one success, E_d denotes multiple failed attempts followed by one success and more attempts, E_e denotes one success without further attempts, E_f denotes one success followed by more attempts.

The reasons for introducing *submission type* are as follows. First, the submission record sequence can be encoded in a short way, benefiting further processing. Second, it captures learners' knowledge proficiency. For example, Learner A tried many times until finally solving Problem X, whereas Learner B tried only once and succeeded. It indicates that B is probably more proficient than A in Problem X. Third, it enables the inference of question difficulty level. Figure 3(b) presents three questions with different levels of difficulty based on the distribution of submission types. Finally, it makes the prediction of probability that a learner can solve a question possible, when the learner's history submission records are available. Suppose X denotes the submission type distribution of a question and y is a learner's submission type for that question. Then given all the pairs (X, y) for one learner, we can train a classifier to predict the possible difficulty level of a specific question for this learner, where X is the input feature and y is the output label.

We further define the **submission event** as (X_i, E_i, t_i), where X_i is i-th problem, $E_i \in E$ and t_i is the stay time this learner spent on Problem X_i. Then, a typical learning path can be represented as an ordered set of submission events: $[(X_{i_0}, E_{i_0}, t_{i_0}), (X_{i_1}, E_{i_1}, t_{i_1}), ..., (X_{i_n}, E_{i_n}, t_{i_n})]$, where n is the number of submission events for this learner.

Peer Learning Path Grouping

The grouping of peer learning paths is done in three steps. First, following prior studies [3, 33], we choose four attributes to group learning paths of learners in question pools. They are learning duration (the time span between the first submission event and the last submission event), learning frequency (how often a submission event appears), learning intensity (the number of submission events per day) and learning proficiency ($(E_e + E_f)/\#\{\text{submission events}\}$). Second, we plot histogram overviews to inspect the user distribution along each attribute. Third, domain experts are involved to specify meaningful ranges for each attribute based on the histograms. Two factors are considered in this process: user number within each range and behavior differences between ranges. The grouping results are shown in Table 1. Combining these attributes, we further extract three typical learning scenarios (i.e., regular learning, intensive learning and, advanced learning), as shown in Figure 1(a). This grouping can be reused if the size of the new data is relatively small to that of the existing dataset and when the value distribution is largely unaffected. If the distribution of an attribute has changed considerably or new scenarios are introduced, we rerun the second and third step to update the grouping.

Path Suggestion

We achieve learning path planning based on Markov Chain (MC), as it is more intuitive for human beings to understand. Specifically, we define the state s as a set of problems which have been solved [23, 36], e.g., $s = \{\mathrm{X}_0, \mathrm{X}_1, ..., \mathrm{X}_n\}$.

Table 1: The four performance-based attributes are empirically divided into four ranges.

	Range 1	Range 2	Range 3	Range 4
Duration(months)	0~1	1~3	3~6	>=6
Frequency	0~0.1	0.1~0.2	0.2~0.3	>=0.3
Intensity	0~1	1~2	2~5	>=5
Proficiency	0~0.25	0.25~0.5	0.5~0.75	>=0.75

Note that we do not consider the order in which the problems are solved. Based on this definition, a given peer path $[(X_{i_0}, E_{i_0}, t_{i_0}), ..., (X_{i_n}, E_{i_n}, t_{i_n})]$ corresponds to a state $s = \{X_{i_0}, X_{i_1}, ..., X_{i_n}\}$. State s_i transits to State s_j only when $s_j = s_i \cup X_k$, where X_k is the extra problem in s_j compared with s_i. To generate the component $P_{ss'}$ in the transition matrix P, we count the number of transitions $N_{ss'}$ from s to s' and the number of all transitions N_s from s within the given peer paths, which are included in the peer group selected by the learner. Then $P_{ss'}$ is defined as the ratio of $N_{ss'}$ to N_s. This transition probability matrix P captures the common behaviors of problem solving in the peer group.

Based on this P, we can plan learning paths for a given learner. The most natural path is the most popular path taken by the selected peer group. Given the learner's history path, we first find the corresponding state s_u, and then query P to find the state transition from s_u with the highest possibility. This query is conducted recursively until a path of a certain length is found. This path is then recommended to the learner as the **popular path**.

We also derive two variants from the popular path to meet different learners' needs and characteristics. The first variant is the **challenging path**, which is generated by skipping problems of similar difficulty. To this end, we use cosine similarity to measure the similarity of submission type distributions among the consecutive problems in the popular path and select only one problem from cosecutive and similar problems. Another path variant is the **progressive path**. We reorder the problems from the popular path based on the problem's difficulty level from easiest to hardest, which is also inferred by submission type distributions.

6 VISUAL DESIGN

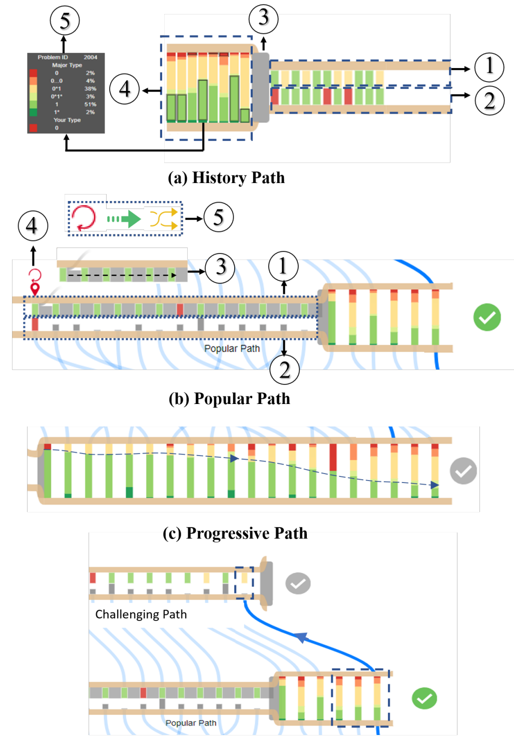
The interface of *PeerLens* is composed of three coordinated views: the peer selection view (Figure 1(a)), the learning path view (Figure 1(b)), and the problem archive view (Figure 1(c)).

Peer Selection View

The peer selection view is designed to facilitate the learners to locate or customize the group of peers whose learning paths are similar to theirs. This corresponds to the requirement R1 in Section 3 of our targeted scenario. As shown in

Figure 1(a), the peer selection view further consists of four radar charts arranged horizontally. The left three charts indicate peer groups in regular, intensive, and advanced learning scenarios, separately. Meanwhile, the rightmost chart allows learners to manually customize their learning scenarios.

For each radar chart in Figure 1(a), there are two star-shaped plots, with the yellow one representing the selected learning group, and the blue one representing the learner himself. The star-shaped plot visualizes key attributes of the corresponding learning group in that the lengths of the four spokes in the plot are proportional to learning duration, frequency, intensity, and proficiency of the learning group respectively. These attribute names and values are labeled on the end of each spoke in detail and the distributions are shown along the axes. The intensive learning group shown in the chart represents the group who use this question pool often for about one to three months, and have solved two to five questions per day with high proficiency. By comparing the two star-shaped plots in the radar chart, learners could understand differences between their own learning history and the profile of the targeted learning group. In the rightmost chart associated with the customized learning group, learners can specify their own learning objectives by dragging the data points of the star-shaped plot in blue. As this interaction proceeds, the number of similar learning paths in the same group will be computed and shown in the blue circle beside the chart.



(a) History Path

(b) Popular Path

(c) Progressive Path

(d) Challenging Path

Figure 4: Learning path view. (a) History path, (b) Popular path, (c) Progressive path, and (d) Challenging path.

Learning Path View

The learning path view compares the learner's learning path with those of the selected peer group (Requirement R2), and offers diverse learning path suggestions (Requirement R3). In this view, a zipper-like visual metaphor is proposed as part of the main design to help learners understand the context of one's learning path by answering "where have I came from?", "where am I?", and "where do I go?". As shown in Figure 1(b), the history path on the left displays which questions have been tried by the learner (Figure 1(b1)), and the future path on the right displays three learning paths suggested by the system (Figure 1(b2, b3, b4)). A location marker in the middle indicates the question currently being worked on (Figure 1(b5)). In all these paths, each question is represented by a tooth on the zipper design.

The history path in Figure 4(a) is made up of three components: the array of upper teeth 4(a-1), the array of lower teeth 4(a-2), and the slider 4(a-3). The upper teeth represent the performance of the selected group. The lower teeth represent current learner's performance on each question. Both upper and lower teeth are arranged in chronological order. To prevent visual clutter, we make use of the slider in the middle to control the amount of information displayed in this view. The slider is placed in the leftmost position by default when the zipper is closed. Each tooth in the lower array represents the submission type of the learner towards a specific question. Each tooth in the upper array shows the major submission type of the selected group in completing that question. In case the learner wants to see the detailed group performance, s/he can drag the slider rightward to open the zipper. The opened part of the teeth will then show the distribution of submission types by the stacked bar chart Figure 4(a-4). The learner's submission type is highlighted on the bar to show his/her position in the group. Submission types are encoded using a sequential color scheme from red to green in Figure 3 to indicate different submission types because we consider that different submission types can differentiate a good performance from a bad performance. The detailed information of the submission types can be assessed when hovering over each bar in the array (Figure 4(a-5)). The problem ID, the distribution of the submission type of each question and the learner's submission type will be given.

The future path displays three suggested learning paths: the challenging path (Figure 1(b2)), the popular path (Figure 1(b3)), and the progressive path (Figure 1(b4)). When the zipper is closed, as shown in Figure 4(b-1), the array of upper teeth encodes the selected learning group's major submission type on each question while the array of lower teeth shows the predicted difficulty of each question for the learner. To avoid overusing colors, we use the height of the grey bar to show the difficulty Figure 4(b-2). The higher the grey bar, the more difficult the question.

We also design visual cues to reason each path, and help learners understand the difference among the three suggested paths. On the popular path, we use the flow inside the path to show the probability of taking the next question in the selected group (Figure 4(b-3)). The branch of the flow indicates the minor number of people going on to do the other questions. Moreover, we use the lines to link the same question on different paths which helps to reason the challenging path and the progressive path which are derived from the popular path. For the progressive path (Figure 4(c)), learners see the detailed distribution of problems on the progressive path showing the growing difficulty from the easiest to the hardest, in accordance with the reordering of problems in Figure 4(b), in terms of difficulty level. The problems on challenging path only link to some of the problems on the popular path. By referring to Figure 4(d), learners can find that only one problem is chosen from three concecutive problems with similar submission distributions (difficulty).

When a learner selects a path, a location marker appears on that path, indicating the problem that the learner is solving. A hint is also shown above it, as depicted in Figure 4(b-4). There are three types of hints: "do this problem again", "move to the next problem", "change a path", as shown in Figure 4(b-5). The suggested path will be updated when the learner changes path or customizes a new learning scenario.

Problem Archive View

The problem archive view in Figure 1(c) is designed to allow learners to quickly map the questions on the learning path with the original question in the pool. When hovering over any bar on the learning path, the corresponding question will be highlighted on the problem list. Learners can click the question on question list to enter original question page. The previous records and hints are shown on the left-hand side of each problem.

Example Use Scenario

As shown in Figure 1, consider Alex, a learner who prepares for a coding interview for a software company. He did not practice much and now only has two months to prepare for the interview. He opened our system and noticed that there is an intensive learning group 1(b1) which has a learning duration of three months and is practiced frequently, solving more than 5 question every day. He selected this group. Then he found his learning history was shown on the history path together with the group's performance, and three suggested learning paths on the right for future study, Figure 1(b2, b3, b4). He compared his performance with the group's performance and found that sometimes he did better than the peer group while at other times he did not. He decided to follow the group's learning path by selecting the popular path. Then he started solving questions. He got wrong several times the first question on the popular path. A hint then appeared at the top of this question, as shown in Figure 1(b5), reminding him to try again because he has not reached average submission times of this group on the same problem.

7 EVALUATION

In the online learning scenario, there is no standardized data set nor process to evaluate a learning path recommendation system [27]. Moreover, different from the online transaction scenario where a recommendation can be judged as correct or not only after a short turn-around time, it usually takes a much longer time and involves more user interactions to evaluate the correctness of a recommendation in the online learning scenario [20]. Thus, in this work we focus on the evaluation of the usefulness and efficacy of the *PeerLens* rather than the accuracy of its recommendation algorithm.

Experiment Design

According to Weibelzahl's work [38], we adopt a four-layer taxonomy to evaluate our system. In particular, we conducted a user study to systematically assess the *informativeness* of the knowledge delivered, the *effectiveness* in facilitating the decision making, the *usability* of the proposed system, as well as the *visual design*.

Participants: We recruited 18 students (7 females, 11 males, age:24±2.85) from a local computer science department to conduct the user study. Each participant received a gift of $25 for their time after the study. All the participants have a basic knowledge of the online judge and question pool, and 9 out of 18 have had long experience using at least one of these online judge systems, e.g., LeetCode, TopCoder. We chose the participants with a computer science background as most of the popular online judges and question pools, as well as this study, are on a programming test topic, for which they could provide us more comprehensible insights.

Experiment setting and procedure: We compared *PeerLens* (namely the *full PeerLens*) with two alternative learning systems. One is the original online judge without an explicit design for the learning path planning (namely the *baseline* system). Questions in the baseline system are sorted by their problem IDs assigned upon creation, which are independent of contents and question difficulties; in a sense, the "recommendation" of the next question is almost random. The other one is a simplified version of *PeerLens* (namely the *primitive PeerLens*). The primitive system uses a truncated design of *PeerLens* with the same recommendation algorithm Figure 5(b). The differences between the primitive

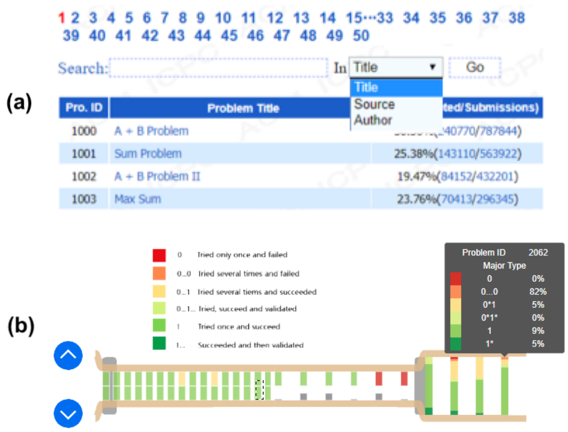


Figure 5: Two systems are compared with *PeerLens* in the evaluation: the baseline learning system (a) and the primitive version of *PeerLens* (b).

and full version lie in: (1) the full version provides multiple learning paths for learners to select by themselves and the primitive version only provides one suggested learning path. (2) the full version makes use of several visual cues and hints to illustrate the semantics and statistics of suggested paths while no such cues are applied in the primitive version to interpret the learning path. According to [30], we designed three learning scenarios in this study: the basic programming practice, the coding qualification test for IT company interviews, and the International Collegiate Programming Contest preparation.To minimize the ordering and learning effect, we counterbalance the three systems in comparison with the three learning scenarios.

The actual experiment is composed of four sessions. In the first session, participants are briefed about the purpose and procedure of the experiment. Each following session lasts approximately 20 minutes and one of the three systems is presented and tested in one different learning scenarios.Each participant is required to conduct two tasks with the provided system. The first task is to determine the starting question under a specific learning scenario; The second task is to find the next question to solve given an existing historical learning path under the same learning scenario. Participants are asked to think aloud about their strategies to pick questions. After finishing all the tasks with a particular system, the participant is required to complete a questionnaire with 7-point Likert scale questions derived from the existing literature [10, 12, 43], which is shown in Table 2.

Hypothesis: We propose the following hypotheses based on the existing literature [12] on peer-based learning.

H1. The proposed visual design of *PeerLens*, regardless of the primitive or full version, performs better than the baseline system in terms of informativeness. Specifically, *PeerLens* systems enjoy their advantages on information accessibility (*H1a*), richness (*H1b*), and sufficiency (*H1c*) compared with the baseline system.

Table 2: Our questionnaire focuses on 4 aspects: informativeness (Q1–Q3), decision making (Q4–Q6), visual design (Q7–Q8) and usability (Q10–Q12).

Q1	The information needed to plan a learning path is easy to access.
Q2	The information needed to plan a learning path is rich.
Q3	The information is sufficient to plan a learning path.
Q4	The system was helpful for me to find a proper learning path for a specific learning scenario.
Q5	I am confident that I find a suitable learning path for the learning scenario.
Q6	The system helps make adjustment according to previous performance.
Q7	The learning path design is intuitive.
Q8	The learning path design helps me understand the suggested path.
Q9	It was easy to learn the system.
Q10	It was easy to use the system.
Q11	I would like to recommend this system to others.

H2. The proposed visual design of *PeerLens*, regardless of the primitive or full version, is better than the baseline system in assisting the decision making. Specifically, *PeerLens* systems provide more confidence (*H2a*), adaptiveness (*H2b*), and assistance (*H2c*) compared with the baseline system.

H3. The full version of *PeerLens* is more informative than the primitive version. In particular, the information accessibility (*H3a*), richness (*H3b*), and sufficiency (*H3c*) of the full version is better than than that of the primitive version.

H4. The full version of *PeerLens* performs better than the primitive version in facilitating the decision making process. Learners will rate the full version better than the primitive version mainly on the confidence (*H4a*), adaptiveness (*H4b*), and assistance (*H4c*).

H5. The primitive version is preferred over the full version. In particular, the primitive version is considered more intuitive (*H5a*), easier to comprehend (*H5b*), learn (*H5c*), and use(*H5d*), and thus is better recommended overall (*H5e*), compared with the full version.

Results and Analysis

We report the participants' quantitative ratings and verbal feedback from the following two aspects, task experience and system design. We run repeated measures ANOVA on each questionnaire item, followed by the Bonferroni post-hoc test on measures with statistically significant differences.

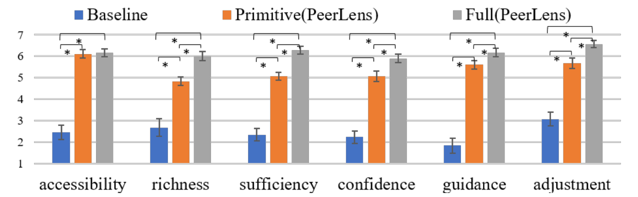


Figure 6: Means and standard errors of *Baseline*, *Primitive*, and *Full* on informativeness and facilitating decision making on a 7-point Likert scale($*: p < .05$).

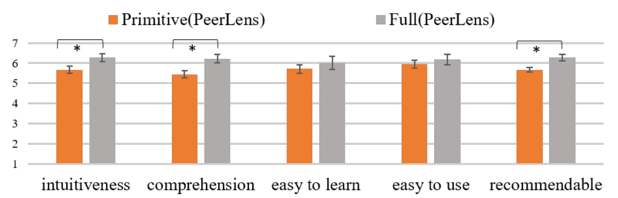


Figure 7: Means and standard errors of *Primitive* and *Full* on visual design and system usability on a 7-point Likert scale($*: p < .05$).

Table 3: Repeated measures ANOVA of *Baseline*, *Primitive*, *Full* on informativeness, decision-making, *Primitive* and *Full* on visual designs and system usability.

		df	*F*	*Sig.*	η^2
Informativeness	accessibility	1	119.05	0.00	0.875
	richness	1	43.59	0.00	0.719
	sufficiency	1	153.86	0.00	0.364
Decision-making	confidence	1	79.12	0.00	0.823
	guidence	1	327.71	0.00	0.951
	adjustment	1	84.24	0.00	0.832
Visual design	intuitiveness	1	6.25	0.23	0.27
	comprehension	1	8.01	0.12	0.32
System usability	easy to learn	1	0.57	0.46	0.03
	easy to use	1	0.60	0.45	0.03
	recommendable	1	11.12	0.00	0.40

Informativeness and decision-making efficacy. Overall, compared with the baseline system, the proposed primitive and full versions of *PeerLens* receive significantly higher scores in all the studied metrics, both in informativeness and decision-making efficacy. In addition, the full version of the *PeerLens* system is shown to be significantly better in informativeness in terms of the information richness and sufficiency. As for the decision-making efficacy, the full version again performs significantly better than the primitive version. Details are featured in Table 3 and Figure 7.

Information accessibility. Participants find assessing information is significantly easier in the full ($Mean = 6.17, SD = 0.185$) and the primitive version ($Mean = 6.11, SD = 0.196$) with ratings significantly higher than the baseline ($Mean = 2.44, SD = 0.336$), (Table 3, **H1a supported**). No significance has been found between the full and primitive version in the Bonferroni post-hoc test ($p = 0.99$, **H3a rejected**).

Information richness. The information provided by the full ($Mean = 6.00, SD = 0.214$) and primitive *PeerLens* ($Mean = 4.83, SD = 0.202$) are reported to be significantly richer than the baseline ($Mean = 2.67, SD = 0.404$), (Table 3, **H1b supported**). We also observe a significant difference between the full and the primitive version in a Bonferroni post-hoc test, $p < 0.05$, **H3b supported**.

Information sufficiency. The information offered by the full version ($Mean = 6.28, SD = 0.195$) and the primitive version of *PeerLens* ($Mean = 5.06, SD = 0.189$) are shown to be sufficient in planning the learning path, compared with the baseline ($Mean = 2.33, SD = 0.291$). The significance is detected in Table 3, **H1c supported**. A Bonferroni post-hoc test indicates that there is also significance difference between the full and primitive versions of *PeerLens*, $p < 0.05$, **H3c supported**. In the verbal feedbacks of the participant: *"The submission type distribution is really useful. I want to know how others perform on these problems. I really hate the basic version for only displaying all the data using a table without statistic information that can be used."*

Confidence in decision-making. Participants reported significantly higher confidence in finding a proper learning path using the full ($Mean = 5.89, SD = 0.196$) and the primitive version of *PeerLens* ($Mean = 5.06, SD = 0.235$), compared with the baseline ($Mean = 2.22, SD = 0.286$), (Table 3, **H2a supported**). The Bonferroni post-hoc test also detects a significant difference between the confidence rating of the full version and the primitive version ($p < 0.05$), **H4a supported**.

Guidance in decision-making. In finding a proper path under a given learning scenario, learners reported that the full version ($Mean = 6.17, SD = 0.202$) and the primitive version of *PeerLens* ($Mean = 5.61, SD = 0.2$) provide significantly more guidance than the baseline ($Mean = 1.83, SD = 0.345$), (Table 3, **H2a supported**). The follow-up Bonferroni post-hoc test also reveals a significant difference between the full and primitive version of *PeerLens*, $p < 0.05$, **H4c supported**.

Adjustment in decision-making: In the second task, learners are asked to determine the next questions to solve according to the historical performance. We evaluate whether the system could help learners to make adjustment accordingly. Results show that both the full version ($Mean = 6.56, SD = 0.166$) and the primitive version ($Mean = 5.67, SD = 0.243$) are significantly better at making adjustments than the baseline ($Mean = 3.06, SD = 0.318$), (Tabale 3, **H2b supported**).

According to the Bonferroni post-hoc test, the full version performs significantly better than the primitive version in providing adjustments, $p < 0.05$, **H4b supported**. Selected verbal feedback: *"The comparison with other is good for me to decide whether to follow this group or not. Because the previous performance is good, thus I want to try the difficult problem on the challenging path."*

In summary, the results on informativeness and decision-making efficacy demonstrate that the proposed visual designs in *PeerLens* are informative as they provide accessible, rich, and sufficient information to learners. The submission distribution of the problem offers a clear visual cue of the difficulty of problems. In addition, the full-version *PeerLens* facilitates the decision-making process by providing more options, which allows the learners to make more adjustments, and offers more guidance to the learners. For example, the comparison of the learner's performance with the peer group's performance help learners choose which path to follow. The visual hints, such as reminding learners to try again, helped them to decide whether to move on. Hence, learners tended to be more confident when planning a path for their own studies.

Visual designs and system usability. Figure 7 summarizes the results of the learner's ratings as well as the Bonferroni post-hoc results on the perspective of the visual designs and system usability between the primitive version of *PeerLens* and the full version of *PeerLens*. Overall, the full-version *PeerLens* system is regarded better in all the metrics than the primitive version, especially in terms of intuitiveness, comprehension, and worthiness of recommendation. The detailed figures are also presented in (Table 3 Figure 6).

Intuitiveness and comprehension. Different from our hypothesis, the primitive version of *PeerLens* ($Mean = 5.67, SD = 0.181$) is less intuitive than the full version ($Mean = 6.28, SD = 0.195$), (Table 3, **H5a rejected**). Meanwhile, the primitive version of *PeerLens* ($Mean = 5.44, SD = 0.185$) is considered less comprehensible than the full version ($Mean = 6.22, SD = 0.207$) . The follow-up Bonferroni post-hoc test further reveals a significant difference between the full and primitive version of *PeerLens*, $p < 0.05$, **H5b rejected**.

Learn, use and worth of recommendation. We do not notice a significant difference in terms of easy to learn and use, between the full version of *PeerLens* and the primitive version in Table 3 . The follow-up Bonferroni post-hoc test also showed there is no significant difference between the full and primitive version of *PeerLens* in terms of being easy to learn and easy to use, $p = 0.46$ and 0.45 respectively, **H5c, H5d rejected**. We conducted a post-hoc analysis on whether participants' rich experience with online judge has an effect on their perceptions of the tools. The difference is insignificant. In addition, learners are more willing to recommend the full version of *PeerLens* ($Mean = 6.28, SD = 0.158$) than the primitive version ($Mean = 5.67, SD = 0.114$) to other users. The Bonferroni post-hoc test $p < 0.05$, **H5e rejected**. Here is one representative verbal feedback, *"This (full-version) is really cool and intuitive. It's very easy to use (full-version), I just need to set a learning scenario and then choose a path."*

We also conducted a post-hoc power analysis. With effect size 0.5, our result has the probability of 0.99 to avoid the Type II error. Overall, the proposed full-version *PeerLens* system is more intuitive and comprehensible for learner to learn and use, and is thus worth recommendation.

8 DISCUSSIONS AND LIMITATIONS

From the experiment results, we derive several design considerations, which potentially enhance the capabilities of online self-learning [2] First, the system needs to be easily extended to other question pools. Our system can be generalized to other question pools owing to the fact that PeerLens only uses the submission records (userID, problemID, submission correctness/score, timestamp) to suggest/visualize a path without the need for problem content information. Online question pools always keep such submission records for users to track their progress, which can be readily employed by PeerLens. Further, for question pools that feature multi-step problem solving such as the math and circuit design discussed in Glassman's work [13, 14], PeerLens can be easily adapted to visualize and suggest path(s) along different steps in solving a problem. Second, to provide a better experience of the learning path planning, the system should provide richer information with more options. From the clickstream data collected during the experiments, all the participants unzipped the paths to check detailed information at least twice. Moreover, in the second task, when learners were asked to do their learning path planning under a particular learning scenario and history record, most participants constructed multiple learning paths. Thus, it can be inferred that learners will require more than one choices for the learning path in a real-world scenario. Third, we discover that the visual design for presenting information is more important than condensing the data to avoid overwhelming learners. Learners prefer richer information shown on demand and step by step. While it is straightforward to stack all the relevant information into the system, how to design the visual representations to avoid information overload is a challenging problem. When designing the student-facing dashboard, the visual representation is a key issue to be considered.

This work still has several limitations. First, our learning path planning algorithm only considers the existing learning paths regardless of any semantic meaning. While it is difficult to automatically extract the semantic information and leverage this information to design more comprehensive algorithms, we plan to apply the crowdsourcing method to add tags, or directly mine these tags from the question pool forums. Second, in the peer group selection, we extract four attributes to specify a learning scenario, but not all the useful learning scenarios have been included. For example, one participant mentioned that she wanted to select a group of peers whose performance in the system is rising in both the accuracy and the difficulty of questions taken. This insight indicates that, when mining the peer group learning sequence, we also need to take into account the dynamic nature of learners. Due to the lack of ground truths, we did not evaluate our learning path planning algorithm. In the future, we will organize a field deployment study to assess the algorithm accuracy and system usability.

9 CONCLUSION AND FUTURE WORK

In this work, we present a novel visual analytics system to help users interactively plan their learning paths in online question pools based on the inspiration of their peer learners. The system provides both an overview of the peer learners' learning attributes to customize the user's learning scenario, and a novel zipper-like learning path view to facilitate the detailed exploration. Three suggested learning paths in future are derived using data mining techniques, which could satisfy the requirements of learners at different levels. Learners can interactively select a learning path and decide their next question to take according to the history performance provided by the system. Our system is evaluated by a within-subject user experiment, which compares the efficacy and usefulness of *PeerLens* with two baseline systems. Experiment results show that learners are more confident in arranging the learning path via our system and they find it more informative and intuitive.

In future work, we plan to deploy *PeerLens* in real-world online judge to collect the action logs and feedback of learners. These data will greatly help us validate the visual analytics process, learning path recommendation algorithm, visualization design, and user interaction of *PeerLens*. We also plan to integrate our system with the component of submission sequence visualization, in order to provide better guidance in learning from peers. Furthermore, through a more systematic study, we will investigate whether participant variability leads to different perceptions when using the tools.

ACKNOWLEDGEMENTS

This work is partially sponsored by Innovation and Technology Fund (ITF) with No. ITS/388/17FP and Lei Shi is supported by NSFC Grant with No. 61772504. We also thank Hangzhou Dianzi University Online Judge for providing the data.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering* 6 (2005), 734–749.

[2] Jim Broadbent and WL Poon. 2015. Self-regulated learning strategies & academic achievement in online higher education learning environments: A systematic review. *The Internet and Higher Education* 27 (2015), 1–13.

[3] Kathleen Burnett, Laurie J Bonnici, Shawne D Miksa, and Joonmin Kim. 2007. Frequency, intensity and topicality in online learning: An exploration of the interaction dimensions that contribute to student satisfaction in online learning. *Journal of Education for Library and Information Science* (2007), 21–35.

[4] Nan Cao, Yu-Ru Lin, Fan Du, and Dashun Wang. 2016. Episogram: Visual summarization of egocentric social interactions. *IEEE Computer Graphics and Applications* 36, 5 (2016), 72–81.

[5] Qing Chen, Xuanwu Yue, Xavier Plantaz, Yuanzhe Chen, Conglei Shi, Ting-Chuen Pong, and Huamin Qu. 2018. ViSeq: Visual Analytics of Learning Sequence in Massive Open Online Courses. *IEEE Transactions on Visualization and Computer Graphics* (2018).

[6] Kuo-Kuang Chu, Chien-I Lee, and Rong-Shi Tsai. 2011. Ontology technology to assist learners' navigation in the concept map learning system. *Expert Systems with Applications* 38, 9 (2011), 11293–11299.

[7] Ricardo Conejo, Eduardo Guzmán, Eva Millán, Mónica Trella, José Luis Pérez-De-La-Cruz, and Antonia Ríos. 2004. SIETTE: A web-based tool for adaptive testing. *International Journal of Artificial Intelligence in Education* 14, 1 (2004), 29–61.

[8] Question Pool Definition. 2018. (2018). https://help.blackboard.com/Learn/Instructor/Tests_Pools_Surveys/Reuse_Questions/Question_Pools_Banks Accessed: 2018-9-21.

[9] Hendrik Drachsler, Hans GK Hummel, and Rob Koper. 2008. Personal recommender systems for learners in lifelong learning networks: the requirements, techniques and model. *International Journal of Learning Technology* 3, 4 (2008), 404–423.

[10] Fan Du, Sana Malik, Georgios Theocharous, and Eunyee Koh. 2018. Personalizable and Interactive Sequence Recommender System. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, LBW002.

[11] Fan Du, Catherine Plaisant, Neil Spring, and Ben Shneiderman. 2016. EventAction: Visual analytics for temporal event sequence recommendation. In *In Proceedings of Visual Analytics Science and Technology (VAST), 2016 IEEE*. IEEE, 61–70.

[12] Fan Du, Catherine Plaisant, Neil Spring, and Ben Shneiderman. 2017. Finding similar people to guide life choices: Challenge, design, and evaluation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 5498–5544.

[13] Elena L Glassman, Ned Gulley, and Robert C Miller. 2013. Toward facilitating assistance to students attempting engineering design problems. In *Proceedings of the ninth annual international ACM conference on International Computing Education Research*. ACM, 41–46.

[14] Elena L Glassman, Aaron Lin, Carrie J Cai, and Robert C Miller. 2016. Learnersourcing personalized hints. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. ACM, 1626–1636.

[15] José Janssen, Colin Tattersall, Wim Waterink, Bert Van den Berg, René Van Es, Catherine Bolman, and Rob Koper. 2007. Self-organising navigational support in lifelong learning: how predecessors can lead the way. *Computers & Education* 49, 3 (2007), 781–793.

[16] Hangzhou Dianzi Online Judge. 2018. (2018). http://acm.hdu.edu.cn/ Accessed: 2018-9-21.

[17] Josua Krause, Adam Perer, and Harry Stavropoulos. 2016. Supporting iterative cohort construction with visual temporal queries. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 91–100.

[18] Milos Krstajic, Enrico Bertini, and Daniel Keim. 2011. Cloudlines: Compact display of event episodes in multiple time-series. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2432–2439.

[19] Leetcode. 2018. (2018). https://leetcode.com/

[20] Gord McCalla. 2004. The ecological approach to the design of e-learning environments: Purpose-based capture and use of information about learners. *Journal of Interactive Media in Education* 2004, 1 (2004).

[21] Megan Monroe, Rongjian Lan, Juan Morales del Olmo, Ben Shneiderman, Catherine Plaisant, and Jeff Millstein. 2013. The challenges of specifying intervals and absences in temporal queries: A graphical language approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2349–2358.

[22] Codefight User Number. 2018. (2018). https://searchsoftwarequality.techtarget.com/news/450424234/CodeFights-offers-a-unique-tool-for-developer-recruiting Accessed: 2018-9-21.

[23] Chris Piech, Mehran Sahami, Jonathan Huang, and Leonidas Guibas. 2015. Autonomously generating hints by inferring problem solving policies. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*. ACM, 195–204.

[24] Catherine Plaisant, Brett Milash, Anne Rose, Seth Widoff, and Ben Shneiderman. 1996. LifeLines: visualizing personal histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 221–227.

[25] Wikipedia Compatitive programming. 2018. (2018). https://en.wikipedia.org/wiki/Competitive_programming Accessed: 2018-9-21.

[26] Jagath C Rajapakse and Loi Sy Ho. 2005. Markov encoding for detecting signals in genomic sequences. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 2, 2 (2005), 131–142.

[27] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender systems: introduction and challenges. In *Recommender systems handbook*. Springer, 1–34.

[28] Alexander Rind, Taowei David Wang, Wolfgang Aigner, Silvia Miksch, Krist Wongsuphasawat, Catherine Plaisant, Ben Shneiderman, et al. 2013. Interactive information visualization to explore and query electronic health records. *Foundations and Trends® in Human–Computer Interaction* 5, 3 (2013), 207–298.

[29] Mojtaba Salehi and Isa Nakhai Kamalabadi. 2013. Hybrid recommendation approach for learning material based on sequential pattern of the accessed material and the learner's preference tree. *Knowledge-Based Systems* 48 (2013), 57–69.

[30] Antonio A Sánchez-Ruiz, Guillermo Jimenez-Diaz, Pedro P Gómez-Martín, and Marco A Gómez-Martín. 2017. Case-Based Recommendation for Online Judges Using Learning Itineraries. In *International Conference on Case-Based Reasoning*. Springer, 315–329.

[31] Ramesh R Sarukkai. 2000. Link prediction and path analysis using Markov chains1. *Computer Networks* 33, 1-6 (2000), 377–386.

[32] Michail Schwab, Hendrik Strobelt, James Tompkin, Colin Fredericks, Connor Huff, Dana Higgins, Anton Strezhnev, Mayya Komisarchik, Gary King, and Hanspeter Pfister. 2017. booc. io: An Education System with Hierarchical Concept Maps. *IEEE Transactions on Visualization and Computer Graphics* 1 (2017), 1–1.

[33] Roy J Shephard. 1968. Intensity, duration and frequency of exercise as determinants of the response to a training regime. *Internationale Zeitschrift fuer Angewandte Physiologie Einschliesslich Arbeitsphysiologie* 26, 3 (1968), 272–278.

[34] Ching-Ying Sung, Xun-Yi Huang, Yicong Shen, Fu-Yin Cherng, Wen-Chieh Lin, and Hao-Chuan Wang. 2017. Exploring Online Learners' Interactive Dynamics by Visually Analyzing Their Time-anchored Comments. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 145–155.

[35] Gang Wang, Xinyi Zhang, Shiliang Tang, Haitao Zheng, and Ben Y Zhao. 2016. Unsupervised clickstream clustering for user behavior analysis. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 225–236.

[36] Shuhan Wang, Fang He, and Erik Andersen. 2017. A unified framework for knowledge assessment and progression analysis and design. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 937–948.

[37] Szymon Wasik, Maciej Antczak, Artur Laskowski, Tomasz Sternal, et al. 2018. A Survey on Online Judge Systems and Their Applications. *ACM Computing Surveys (CSUR)* 51, 1 (2018), 3.

[38] Stephan Weibelzahl. 2001. Evaluation of adaptive systems. In *International Conference on User Modeling*. Springer, 292–294.

[39] Krist Wongsuphasawat, Catherine Plaisant, Meirav Taieb-Maimon, and Ben Shneiderman. 2012. Querying event sequences by exact match or similarity search: Design and empirical evaluation. *Interacting with computers* 24, 2 (2012), 55–68.

[40] Raciel Yera Toledo, Yailé Caballero Mota, and Luis Martínez. 2018. A Recommender System for Programming Online Judges Using Fuzzy Information Modeling. In *Informatics*, Vol. 5. Multidisciplinary Digital Publishing Institute, 17.

[41] Chun-Kit Yeung and Dit-Yan Yeung. 2018. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. *arXiv preprint arXiv:1806.02180* (2018).

[42] Huang Yueh-Min, Huang Tien-Chi, Kun-Te Wang, and Wu-Yuin Hwang. 2009. A Markov-based recommendation model for exploring the transfer of learning on the web. *Journal of Educational Technology & Society* 12, 2 (2009), 144.

[43] Jian Zhao, Chidansh Bhatt, Matthew Cooper, and David A Shamma. 2018. Flexible Learning with Semantic Visual Exploration and Sequence-Based Recommendation of MOOC Videos. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 329.

[44] Jian Zhao, Christopher Collins, Fanny Chevalier, and Ravin Balakrishnan. 2013. Interactive exploration of implicit and explicit relations in faceted datasets. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2080–2089.

[45] Haiping Zhu, Feng Tian, Ke Wu, Nazaraf Shah, Yan Chen, Yifu Ni, Xinhui Zhang, Kuo-Ming Chao, and Qinghua Zheng. 2018. A multi-constraint learning path recommendation algorithm based on knowledge map. *Knowledge-Based Systems* 143 (2018), 102–114.