Toward Recommendation for Upskilling: Modeling Skill Improvement and Item Difficulty in Action Sequences

Kazutoshi Umemoto
NICT / The University of Tokyo
umemoto@tkl.iis.u-tokyo.ac.jp

Tova Milo
Tel Aviv University
milo@post.tau.ac.il

Masaru Kitsuregawa
The University of Tokyo / NII
kitsure@tkl.iis.u-tokyo.ac.jp

***Abstract*—How can recommender systems help people improve their skills? As a first step toward recommendation for the upskilling of users, this paper addresses the problems of modeling the improvement of user skills and the difficulty of items in action sequences where users select items at different times. We propose a progression model that uses latent variables to learn the monotonically non-decreasing progression of user skills. Once this model is trained with the given sequence data, we leverage it to find a statistical solution to the item difficulty estimation problem, where we assume that users usually select items within their skill capacity. Experiments on five datasets (four from real domains, and one generated synthetically) revealed that (1) our model successfully captured the progression of domain-dependent skills; (2) multi-faceted item features helped to learn better models that aligned well with the ground-truth skill and difficulty levels in the synthetic dataset; (3) the learned models were practically useful to predict items and ratings in action sequences; and (4) exploiting the dependency structure of our skill model for parallel computation made the training process more efficient.**

I. Introduction

Everyone is inexperienced at first. A person's experience or skill in a particular domain improves over time as they take actions such as repeated practices and exposure to many items. For example, a language learner would need to read/write short, simple texts to get familiar with the target language before he/she can read/write longer, more complicated texts. A person learning cooking, as another example, would first cook a simple dish. After mastering basic cooking skills, he/she could try more elaborate dishes requiring more steps, a longer time, and special kitchenware and/or ingredients. The skill improvement in user actions occurs in many other, if not all, real-world domains, including playing musical instruments, tasting sophisticated beers/wines, and appreciating classic/complicated movies.

How can we help people improve their skills? As an answer to this question, we envision *a recommender system for upskilling users*. Such a system would estimate the skill of a target user and recommend to him/her an item with appropriate difficulty for upskilling. There are three main challenges to overcome in achieving this goal. First, we need to estimate the user's current skill level: what may indicate this? Second, we need to know the difficulty of each item: what makes an item difficult? Third, we need to find items suitable for upskilling the user: how can we personalize the recommendation in terms of what, when, and how to present him/her? In this work, we focus on the first two challenges and leave the third for future research. Considerable effort has been devoted to developing sophisticated recommender systems that take into account many concepts (*e.g.*, interest [1], freshness [2], serendipity [3], popularity [4], and explainability [5]). As upskilling is complementary to these concepts, we believe that the knowledge we derive on user skills and item difficulty has a great potential to provide a new perspective on information recommendation.

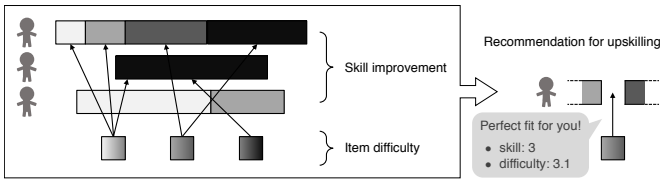


Fig. 1: Two problems we address in achieving recommendation for upskilling: modeling skill improvement and item difficulty.

Given action sequences in which users select items at different times (*e.g.*, user histories of dishes cooked in the past), we address the problems of modeling user skills (and their improvement over time) and item difficulty as illustrated in Figure 1. For the first problem, we use a progression model to learn the monotonically non-decreasing dynamics of user skills. Unlike the existing model [6], which relies solely on item IDs, we utilize multi-faceted features shared by items for making our model robust against rare items. To solve the second problem, we assume that users usually, if not always, select items within their skill capacity (*e.g.*, skilled people cook both easy and difficult dishes while novices typically cook the former). Under this assumption, we use the learned skill model to estimate the difficulty of each item as the degree of skill of typical users who select it.

We conducted experiments using five datasets, of which four were from real domains (language learning, cooking, beer tasting, and movie watching) while one was generated synthetically. The main findings from our experiments are the following: (1) our model successfully captured the progression of domain-dependent skills; (2) multi-faceted item features helped to learn better models that aligned well with the ground-truth skill and difficulty levels in the synthetic dataset; (3) the learned models were practically useful to predict

item selections and user ratings in action sequences; and (4) exploiting the dependency structure of our skill model for parallel computation made the training process more efficient.

In summary, this work makes the following contributions:

- Formalizing skill improvement and item difficulty as core problems to address as a first step toward recommendation for upskilling, which can provide a new perspective on information recommendation;
- Utilizing multi-faceted item features to develop a skill model that is robust against sparse data, and using the learned model to estimate the difficulty of items with the same scale;
- Conducting qualitative and quantitative experiments with five datasets, which demonstrated the interpretability of our skill model, the accuracy of estimated skill and difficulty levels, the usefulness of our models for practical recommendation tasks, and the efficiency of our training algorithm.

II. Related Work

Sequence modeling. Temporal dynamics has been studied as a key factor for modeling sequence data. This line of research can be roughly divided into two main classes: changes in a whole community (*e.g.*, fashion trends [7]) and in individuals (*e.g.*, acquired tastes [8], disease progression [9]), of which the latter is relevant to this work as described below.

Progression modeling [6], [8], [10] aims at inferring the dynamics of invisible states that affect observable outcomes. As the skill improvement and disease progression problems have things in common (*e.g.*, the monotonicity of user skills and disease stages, the sequentiality of item selections and disease treatments), we adopt this approach in this work. McAuley and Leskovec [8] developed a model that consists of multiple recommender systems conditioned on latent variables representing users' experience levels. Their model has a limitation of only being applicable to sequence data with user ratings (*e.g.*, review scores). Yang *et al.* [6] proposed a more general model that can work for sequence data without the rating information. Their model can be viewed as an extension of hidden Markov models in that it requires state transitions to satisfy a monotonicity constraint. We take Yang *et al.*'s model as a basis for learning the improvement of user skill. As their model describes a generative process for each item represented as a distinct categorical value (*i.e.*, ID), its learning does not work well for domains with many rare items. To deal with this issue, we utilize the multi-faceted features shared by items.

Sequential recommendation [11], [12] models the ordering patterns of action sequences by using Markov chains, neural networks, *etc*. Despite its similarity with our goal, sequential recommendation focuses on predicting items that a user will select next. We instead focus on quantifying the time-varying skill of users and the difficulty of items. While beyond the scope of the present work, fusing these complementary studies has a potential to recommend items that both interest users and improve their skills with convincing reasons (Figure 1).

Experience, expertise, and skill. The three terms, experience, expertise, and skill, have been used in the literature to refer to similar concepts on people's ability and knowledge (see Section III for our definition of skill).

In the human computation field, the skill (or ability) estimation [13], [14] is essential in improving the quality of crowdsourcing. Thus, it has been studied intensively with the following assumptions: (1) each worker has a fixed skill level per domain; and (2) a task response (*e.g.*, an answer to a task question) is obtained from each user (or each team [14]). Similarly, knowledge tracing [15], [16], which models the knowledge of students interacting with exercises, has also been studied for years in the education field. This problem takes as input whether students correctly answered exercises in the past. While the problems above seem similar to ours, we consider a complementary situation where (1) the skill of each user can improve over time and (2) each action consists of a triple of the time, user, and item (*i.e.*, no user response).

In the information retrieval field, two types of user expertise have received great attention: search expertise (*i.e.*, familiarity with search systems) and domain expertise (*i.e.*, knowledge on search domains). For both types, the past studies based on log analysis [17], [18] revealed a behavioral difference between experts and non-experts. Building on this finding, we let our skill improvement model have distinct parameters for each skill level, so that it can capture the difference in item selection behavior between skilled and unskilled users.

Difficulty. The difficulty of items is a research subject of particular importance for educational and learning purposes. Much effort has been devoted to this topic, including instructional design [19], text simplification [20], and personalization by reading level [21]. As item difficulty is closely related to user skill, these two concepts have often been jointly modeled [22], [23]. Following those studies, we estimate the difficulty of items by leveraging our skill model learned from action sequences, under the assumption that more skilled users can select more difficult items.

Recently, search as learning [24] has drawn great attention in the information retrieval field [25]. Syed *et al.* [26] developed an algorithm to optimize a ranking of search results for human learning by considering the document difficulty. While their focus is textual documents, a successful model for item difficulty in various action sequences has a potential to provide searchers with a ranking of other types of items (*e.g.*, cooking recipes, movies, and music pieces) such that the difficulty and ordering are optimal for them to improve their ability and/or knowledge with minimum effort.

III. Problem Definition

Let $\mathcal{U}$ be a set of users. In this work, we consider a situation where each user $u \in \mathcal{U}$ takes a sequence of actions, $\mathcal{A}_u$, in a given domain. The length of the action sequence can vary among users. For example, an experienced user may have taken many actions in the past, while a beginner may have a short action sequence. Each action $a \in \mathcal{A}_u$ is denoted by a triplet (t, u, i), where t is the time when u takes a, and i is the item selected or used in a. While other types of actions, such as modification and deletion, may also exist in some domains, we

leave them for future work to simplify our problem settings. Without loss of generality, we assume that the actions in each user's sequence are sorted in chronological order. The items depend on the actions domain. Examples of items include *recipes* that cooks use, *movies* that critics watch and write about, *music pieces* that musicians play, and *educational materials* that students practice with. We denote a set of items as $\mathcal{I}$ and assume that each item $i \in \mathcal{I}$ is represented by a tuple of F multi-faceted features: $i = (i_1, \ldots, i_F)$. As examples of features, a recipe may have ingredients, steps, *etc*., while a movie may have a genre, directors, actors, *etc*. Given a set of user action sequences $\mathcal{A} = \bigcup_{u \in \mathcal{U}} \mathcal{A}_u$, we study two problems in this work, as described below.

A. Skill Improvement

The first problem is to model the development of user skills. Before describing this problem formally, we give a definition of skill similar to the existing definition of experience [8]:

Definition 1. *Skill is some quality of the ability and/or knowledge a user gains over time through his/her actions.*

For example, English learners gain speaking ability by attending an English conversation class, and graduate students gain research knowledge by reading papers relevant to their fields. Following this definition, we represent the level of skill with a positive integer whose range is $\mathcal{S} = \{1, \ldots, S\}$, where a higher skill level indicates a more skilled user. While the integer-based skill representation facilitates the result interpretation and moderates the computational cost, the skill improvement at a high level of granularity can be modeled by increasing the number of skill levels.

Formally speaking, the first problem that we study is to determine the skill level s_{ut} of a user u at each time t when u takes an action $a = (t, u, i)$. As with the existing approach of progression modeling [6], [8], [10], we assume that user skill progresses monotonically: the skill level of each user remains the same or increases as time passes. This places the following constraint on the skill improvement problem:

$$t \leq t' \implies s_{ut} \leq s_{ut'} \quad (\forall u, t, t') \,. \tag{1}$$

In addition, we must address two more requirements, as illustrated in Figure 1. First, users may not experience all skill levels. Some users may have the highest skill level at their first actions observed in given sequence data. It is also possible that some users may not reach the highest skill level no matter how many actions they take. Second, the speed of skill improvement is user dependent. While some users may take many actions to increment their skill by one level, others may progress more quickly. These requirements suggest that a naive solution that segments each user sequence into S groups of equal length to assign the skill level $s \in \mathcal{S}$ to all actions in the s-th group would not work well.

B. Item Difficulty

The second problem is to model the difficulty of each item appearing in action sequences. Following Definition 1, we intuitively define item difficulty as follows:

Definition 2. *Difficulty is the degree of skill required for a user to take satisfactory actions with an item.*

The interpretation of *satisfaction* depends on the domain. In the film domain, casual fans without much experience or knowledge may not be able to fully *appreciate* cinematic masterpieces. In the cooking domain, only those with enough ability may *succeed* in cooking elaborate dishes. According to this definition, we represent the difficulty level with a real number whose range is $[1, S]$, where a higher level indicates a more difficult item. This representation makes it easier to compare users and items by using the same scale. For example, we can say that items with the difficulty level S are suitable only for users with the highest skill level, while items with the lowest difficulty level can be handled by users with any skill level. Representing difficulty levels with higher precision than skill levels enables us to recommend items whose difficulty is moderately challenging for users (*e.g*., $d_i = 3.1$ for $s_{ut} = 3$).

We formalize the second problem that we study as determining the difficulty level d_i of each item $i \in \mathcal{I}$. Note that while some items occur many times in action sequences, others are infrequent. Thus, estimating the difficulty of such rare items is a challenge in this problem.

IV. Skill Improvement

To learn the improvement of user skills, we introduce a statistical model with latent variables. These variables are intended to capture the underlying skill levels of users at different times. Our model describes how action sequences are generated through the latent progression of user skills. As mentioned in Section II, our model builds on Yang *et al*.'s one [6], but we utilize multi-faceted item features to improve robustness against data sparsity.

A. Formulation

Given a skill level $s_{ut} \in \mathcal{S}$ for a user u at a time t, our model uses the following joint probability distribution to describe the generative process of u's action a for selecting an item $i = (i_1, \ldots, i_F)$ at t:

$$a = (t, u, i) \sim P(i \mid s_{ut}) = \prod\nolimits_{f=1}^{F} P_f(i_f \mid \theta_f(s_{ut})) \,, \tag{2}$$

where $P_f(i_f \mid \theta_f(s))$ is a probability distribution with a parameter $\theta_f(s)$. This distribution models the likelihood that the f-th feature i_f of the item i is common among the actions taken by users with the skill level s. As the parameter value can vary across different skill levels, this modeling enables us to capture commonalities within the same skill level and discrepancies between different levels. To make the parameter estimation tractable, we assume in the generation process above that each feature i_f of the item i is independent and identically distributed (iid) given the skill level s_{ut}.

As different domains have different types of items, the selection of probability distributions to model the generative process of item features is domain dependent. Categorical distributions can be used for features that have categorical values (*e.g*., ingredients in a recipe). Features having natural numbers (*e.g*., the number of steps in a recipe) can be modeled

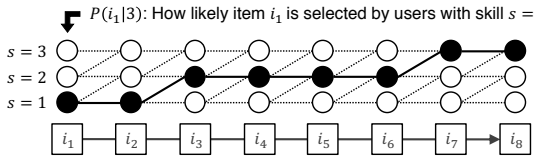


Fig. 2: To find skill assignments, we apply dynamic programming to action-skill graph, subject to skill monotonicity constraint.

with Poisson distributions. For positive, real-valued features, we can use gamma or log-normal distributions. Section VI-A reports the actual distributions that we used in our experiments.

As mentioned in Section III-A, the skill level of each user u is constrained to be monotonically non-decreasing with respect to time. In this work, we consider two possibilities for the skill progression between u's two successive actions a_{ut} and $a_{ut'}$: (1) maintaining the same level (*i.e.*, $s_{ut'} = s_{ut}$), and (2) moving up one level (*i.e.*, $s_{ut'} = s_{ut} + 1$) when $s_{ut} < S$. The assumptions described above follow the base model [6], and we demonstrate in Section VI that our model based on these assumptions yields reasonable experimental results. While we only consider the step-by-step skill improvement here for simplicity, our model is flexible enough to incorporate more complex progressions (*e.g.*, skipping some levels) by introducing a probabilistic distribution for skill transitions [10].

B. Training

Objective function. Given a set of (iid) action sequences, $\mathcal{A}$, and the number of skill levels, S, our objective is to assign a skill level to each user action. (We describe how to determine S for each domain in Section VI-B.) To this end, we fit our model to given sequence data by maximizing the following log likelihood function, subject to the skill monotonicity constraint (Section III-A):

$$\mathcal{L}(\Theta, \Sigma) = \sum_{(t,u,i)\in\mathcal{A}} \log P(i \,|\, s_{ut}) = \sum_{(t,u,i)\in\mathcal{A}} \sum_{f=1}^{F} \log P_f(i_f \,|\, \theta_f(s_{ut}))\,, \tag{3}$$

where $\Theta = \{\theta_f(s) \mid f \in \{1, \ldots, F\} \wedge s \in \mathcal{S}\}$ is a set of model parameters, and $\Sigma = \{s_{ut} \mid (t, u, i) \in \mathcal{A}\}$ is a set of skill assignments.

Jointly optimizing Θ and Σ is impractical because of the complexity of the objective function. A workaround is using the EM algorithm to alternately optimize each variable. EM takes too long to complete, however, for this kind of problems [6], as it computes *soft* assignments for latent variables. To make the model training efficient, we instead follow Yang *et al.*'s *hard* assignment approach, which was reported to run 1,000 times faster than EM with comparable fitting quality [6].

To train our model, we first initialize the model parameters with reasonable values. Then, we alternate the steps of finding the assignments of skill levels and updating the model parameters until convergence. Each step is described below.

Finding skill assignments. This step finds a set of skill assignments, Σ, that maximizes the objective function $\mathcal{L}(\Theta, \Sigma)$ under the skill monotonicity constraint while keeping the current parameter set Θ fixed. We follow Yang *et al.* [6] to efficiently find the best assignments via dynamic programming.

Figure 2 illustrates how to find the best skill assignments for a user u's sequence having $|\mathcal{A}_u| = 8$ actions, where we consider $S = 3$ skill levels. In this figure, each row and column represents a skill level and an action, respectively, and each edge represents a possible path for a skill transition (*i.e.*, to stay or improve). Let $L(u, n, s)$ be the log likelihood of u reaching the skill level s after taking his/her n-th action $a_n = (t_n, u, i_n)$ through the most likely progression path. Then, $L(u, n, s)$ can be calculated recursively as follows:

$$L(u, n, s) = \max_{\delta\in\{0,1\}} L(u, n-1, s-\delta) + \log P(i_n \,|\, s_{ut_n})\,. \tag{4}$$

The most likely skill level at the last action $a_{|\mathcal{A}_u|}$ can be identified with $s_{ut_{|\mathcal{A}_u|}} = \arg\max_{s\in\mathcal{S}} L(u, |\mathcal{A}_u|, s)$ once we calculate $L(u, n, s)$ for all n and s. Then, the best assignments can be obtained by backtracking along the progression path from the node at the $s_{ut_{|\mathcal{A}_u|}}$-th row and the $|\mathcal{A}_u|$-th column.

Updating model parameters. This step finds a model parameter set Θ that maximizes the objective function $\mathcal{L}(\Theta, \Sigma)$ while keeping the current skill assignment set Σ fixed. By exploiting the conditional independence in our model, we can optimize the model parameters separately with respect to each feature f and skill level s. That is, we obtain the optimal value for a parameter $\theta_f(s)$ by maximizing the following likelihood:

$$\mathcal{L}(\theta_f(s)) = \sum_{(t,u,i)\in\mathcal{A}} \mathbb{1}\left[s_{ut} = s\right] \cdot \log P_f(i_f \,|\, \theta_f(s))\,, \tag{5}$$

where $\mathbb{1}\left[p\right]$ is an indicator function that returns 1 if the predicate p is true and 0 otherwise.

As mentioned in Section IV-A, we can use a categorical distribution to model the generative process of a categorical feature $i_f \in \{1, \ldots, C_f\}$ of each item i. The parameter for the categorical distribution associated with a skill level s is expressed as $\theta_f(s) = (\theta_{f1}(s), \ldots, \theta_{fC_f}(s))$, where $\sum_c \theta_{fc}(s) = 1$. The optimal value for this parameter is given in the following closed form:

$$\theta_{fc}(s) = \frac{\lambda + \sum_{(t,u,i)\in\mathcal{A}} \mathbb{1}\left[s_{ut} = s \wedge i_f = c\right]}{\lambda \cdot C_f + \sum_{(t,u,i)\in\mathcal{A}} \mathbb{1}\left[s_{ut} = s\right]}\,, \tag{6}$$

where we apply additive smoothing with a pseudo-count hyperparameter λ to avoid the zero-frequency problem. We set $\lambda = 0.01$ in our experiments, following Shin *et al.* [10].

Similarly, the single parameter of a Poisson distribution can be estimated as follows:

$$\theta_f(s) = \frac{\sum_{(t,u,i)\in\mathcal{A}} \mathbb{1}\left[s_{ut} = s\right] \cdot i_f}{\sum_{(t,u,i)\in\mathcal{A}} \mathbb{1}\left[s_{ut} = s\right]}\,. \tag{7}$$

Some probability distributions (*e.g.*, gamma), however, do not have closed-form expressions for the optimal parameter values. For such distributions, we obtain approximate solutions by applying numerical analysis approaches.

Initializing model parameters. Model initialization is necessary to alternate the above-mentioned steps until convergence. As our objective function is non-convex, a proper initialization is crucial to finding a reasonable local optimum. Thus, we adopt the following initialization approach that has been reported to work well for progression modeling [6], [10].

Assuming that users having longer action sequences are more likely to experience all skill levels, we first select users $\mathcal{U}_{\geq N}$ who have at least N actions. (Section VI-B reports its

actual values we used in our experiments.) We then split each sequence of $\mathcal{U}_{\geq N}$ into S groups uniformly with respect to time and assign the skill level $s \in \mathcal{S}$ to all actions in the s-th group. The resulting assignments for those sequences are used to initialize the model parameters. Once the model is initialized, the sequences of all users are used to repeat the steps of finding skill assignments and updating model parameters.

In some domains, however, several confounders affecting user actions may exist. For example, people tend to watch newly released movies more often than older ones. This *lastness* effect causes the above-mentioned approach to assign higher skill levels to newer movies during initialization. In Section VI-C, we report the resulting unstable learning result in the film domain and demonstrate that simple preprocessing help resolve this issue.

C. Complexity and Speedup

The complexity of our training algorithm is as follows. Initializing skill levels for all actions $\mathcal{A}$ requires $c_{\mathrm{I}} = \mathcal{O}(|\mathcal{A}|)$ operations. Given that the complexity of calculating $P(i \mid s_{ut})$ is linear in the number of features, finding the best skill assignments for a user sequence $\mathcal{A}_u$ via dynamic programming requires $\mathcal{O}(|\mathcal{A}_u|FS)$ operations. Thus, the complexity of the assignment step is $c_{\mathrm{A}} = \mathcal{O}(|\mathcal{A}|FS)$. To update model parameters, we first group actions by the assigned skill levels (in $\mathcal{O}(|\mathcal{A}|)$ steps) and then estimate the parameters for each feature and skill level (in $\mathcal{O}(FS)$ steps). Thus, the complexity of the update step is $c_{\mathrm{U}} = \mathcal{O}(|\mathcal{A}| + FS)$. Let N be the number of iterations required for model convergence. Then, the complexity of the whole training process is $c_{\mathrm{I}} + N(c_{\mathrm{A}} + c_{\mathrm{U}})$, which results in $\mathcal{O}(N|\mathcal{A}|FS)$.

As with the base model [6], we can speed up the training process by exploiting the dependency structure of our skill model. In the assignment step, which would require the longest computation time according to the complexity analysis above, user sequences are independent of each other. Thus, we can perform skill assignments for each user in parallel. Similarly, as any two parameters $\theta_f(s)$ and $\theta_f(s')$ are independent if $s \neq s'$, the update step can be parallelized for each skill level. Note that our skill model has room for further parallelization. Unlike the base model, ours considers multi-faceted item features. Thus, the update step of our model can also be parallelized for each feature. We report the effectiveness of these three parallelization techniques in Section VI-F.

V. Item Difficulty

To address the item difficulty problem, we assume that users, taken as a whole, usually select items whose difficulty level is not greater than their skill level. Under this assumption, we estimate the difficulty of each item in a target domain by leveraging our skill model learned for that domain. In this work, we propose two approaches to this problem.

A. Assignment-based Estimation

The first approach leverages the skill levels that our model assigns to each action in users' sequences. It estimates the difficulty of an item as the *mean* skill level of users who select that item in their actions. Specifically, the difficulty level d_i of an item $i \in \mathcal{I}$ is calculated as follows:

$$d_i = \frac{\sum_{(t,u,i') \in \mathcal{A}} \mathbb{1}\left[i' = i\right] \cdot s_{ut}}{\sum_{(t,u,i') \in \mathcal{A}} \mathbb{1}\left[i' = i\right]}\,. \tag{8}$$

For example, this approach estimates that the item i has a medium difficulty level $d_i = \frac{1+S}{2}$ if half of the users who select i have the lowest skill level while the other half have the highest skill level.

B. Generation-based Estimation

While the assignment-based approach mentioned above is intuitive, it has a practical drawback; the difficulty cannot be estimated for items that have not yet been selected by any user (*e.g.*, new products). Another drawback of this approach is the estimation robustness for rare items that are selected by only a few users. In an extreme case, the estimation could be based on the skill level of a single user. As mentioned earlier, however, skilled users can select both easy and difficult items. If an easy item appears in the last action of a skilled user who has a long sequence, this approach may mistakenly estimate this item as difficult. Therefore, such a simplistic approach is inappropriate for recommending new or rare items for upskilling.

To address this issue, we propose an alternative approach to the item difficulty estimation, which takes item features into consideration. Our basic idea is that the difficulty levels of two items are comparable if they share similar feature values. Instead of relying on the skill assignments, this approach leverages the item generation process of our skill improvement model. Specifically, it estimates the difficulty d_i of an item i as the *expected* skill level that is assigned to the item:

$$d_i = \sum_{s \in \mathcal{S}} s \cdot P(s \mid i)\,, \tag{9}$$

where $P(s \mid i)$ is the posterior probability of assigning a skill level s to the given item i. Using Bayes' theorem, we can decompose this probability into two components:

$$P(s \mid i) = \frac{P(i \mid s) \cdot P(s)}{P(i)} = \frac{P(i \mid s) \cdot P(s)}{\sum_{s' \in \mathcal{S}} P(i \mid s') \cdot P(s')}\,, \tag{10}$$

where $P(i \mid s)$ is the likelihood of observing i in the actions of users with the skill level s, and $P(s)$ is the skill prior.

We can calculate $P(i \mid s)$ with Equation 2 once we learn a skill model. As for $P(s)$, we consider two models.

1) *Uniform:* This model treats the prior as a uniform distribution: $P(s) = \frac{1}{S}$. In other words, it simplifies the posterior as $P(s \mid i) \propto P(i \mid s)$, similarly to the query likelihood model [27] proposed in the information retrieval field.
2) *Empirical:* In reality, the number of skilled users may differ greatly from the number of unskilled users. For such domains where the skill distribution is skewed, the simplification made by the uniform model may make the item difficulty estimation inaccurate. To avoid this, our alternative model empirically estimates the prior from the skill levels assigned to each action in users' sequences: $P(s) = \frac{\sum_{(t,u,i) \in \mathcal{A}} \mathbb{1}[s_{ut} = s]}{|\mathcal{A}|}$.

TABLE I: Dataset statistics after filtering.

Dataset	Source	#Users ($\|\mathcal{U}\|$)	#Items ($\|\mathcal{I}\|$)	#Actions ($\|\mathcal{A}\|$)
Language	Lang-8	51,644	248,009	248,009
Cooking	Rakuten Recipe	6,012	37,092	115,337
Beer	RateBeer	4,540	8,953	1,986,231
Film	MovieLens	85,095	4,589	8,508,819
Synthetic	N/A	10,000	50,000	500,491

C. Complexity

We briefly describe the complexity of our difficulty estimation approaches. The complexity of the assignment-based estimation is $\mathcal{O}(|\mathcal{A}|)$. As for the generation-based approach, the prior is calculated in $\mathcal{O}(1)$ steps for the uniform model and $\mathcal{O}(|\mathcal{A}|)$ steps for the empirical model. Calculating the posterior requires $\mathcal{O}(FS)$ operations. Thus, the uniform and empirical models have complexities of $\mathcal{O}(FS)$ and $\mathcal{O}(|\mathcal{A}| + FS)$, respectively. Note that estimating item difficulty is much faster than learning skill improvement since the former does not require iterative computation.

VI. Experiments

To evaluate our models proposed in Sections IV and V, we conducted experiments using five datasets. Our experiments were designed to answer the following questions:

Q1 Can our skill model capture domain-dependent skills?
Q2 How accurate are our skill and difficulty models?
Q3 How useful are our models for practical tasks?
Q4 How efficiently can we train our models?

A. Datasets

To cover a wide variety of action sequences, we prepared four datasets collected from real domains and one dataset generated synthetically. Table I lists the statistics of these datasets. Note that the numbers in this table were obtained after the filtering process explained in Section VI-B.

Language. As mentioned earlier, language is a typical domain where learners develop their skills. For this domain, we used the NAIST Lang-8 Learner Corpora [28] constructed from Lang-8,[1] a social networking website for language exchange. Lang-8 users can post articles in languages that they are learning. They can also make per-sentence corrections for other users' articles written in their native languages. While Lang-8 supports many languages, for simplicity we focused only on English learners in our experiments. More specifically, we selected English articles written by those users and corrections for their articles. As the corpora consist of textual data (*i.e.*, original sentences and corrected sentences for each article), we calculated the following statistics as item features: the number of sentences (modeled via Poisson distributions), the mean number of corrections per corrector (gamma), and the percentage of corrected sentences (gamma). For categorical features, we extracted correction rules (*e.g.*, from "a" to "the") by finding word alignments that minimized the Levenshtein distance between each pair of original and corrected sentences.

Cooking. Cooking, where recipes correspond to items, also requires skills. For this domain, we used a dataset released for research purposes.[2] This dataset consists of cooking activities posted to Rakuten Recipe.[3] The two types of actions exist on this website: (1) creating new recipes and (2) reporting existing recipes that users have cooked. In our experiments, we focused only on the latter as the target actions for learning the skill improvement because users who can create their own recipes are likely to already have high cooking skills.[4] Each recipe in the dataset includes the following: an ID, an category, a class of cooking time (*e.g.*, about 30 minutes), a class of cooking cost (*e.g.*, about JPY 1,000), a set of ingredients, and a description for each cooking step. We used categorical distributions for the ID, category, time, cost, and ingredient features. In addition, we used Poisson distributions to represent the numbers of ingredients and steps for each recipe.

Beer. We consider selecting beers to drink as actions that may be affected by users' skills of appreciation (or their acquired tastes) [29]. For this domain, we used a dataset [8] collected from RateBeer,[5] a beer review website. This dataset contains all the website's beer reviews from 2000, when it was launched, to 2011. Each review includes the following information for a beer: an ID, a brewer, a style, and an alcohol-by-volume (ABV) value. To model the generation process of beer instances, we used categorical distributions for all but the ABV feature. As the ABV feature has a positive, real-number value, we used gamma distributions instead. We excluded the rating information in each review (*e.g.*, rating scores and review text) from our features, because such subjective information is inappropriate for the properties of items.

Film. As another domain for the appreciation skills, we also consider actions of selecting movies to watch. For this domain, we used a public movie review dataset [30] collected from MovieLens.[6] As the dataset contains limited information about movies (*i.e.*, an ID, a title, and a set of genres for each), we crawled MovieLens to collect directors and actors as additional item information. We then used categorical distributions for the ID, genre, director, and actor features to model the generation process of each movie.

Synthetic. Last, for conducting quantitative experiments, we consider a synthetic dataset with ground-truth skill and difficulty levels.[7] This dataset was generated as follows.

1) We set up three probability distributions (categorical, gamma, and Poisson) with distinct parameters to ensure that the item features for different skill levels tended to have different values. Specifically, the categorical distribution for a skill level s was designed to have higher probabilities of observing the n-th categorical value than other values, where $n \equiv s \pmod{S}$. For the gamma and Poisson distributions, we adjusted the parameters so that samples

[1] http://lang-8.com/

[2] https://www.nii.ac.jp/dsc/idr/en/rakuten/rakuten.html

[3] https://recipe.rakuten.co.jp/

[4] Actually, targeting only the former actions resulted in a model with little difference between skill levels.

[5] https://www.ratebeer.com/

[6] https://movielens.org/

[7] Note that we experimented with multiple synthetic datasets generated with different settings, but we obtained similar trends across these datasets.

drawn for higher skill levels had larger mean values.

2) We generated the same number of items for each skill level. An item i for a skill level s consisted of three features drawn from the aforementioned probability distributions for s. We set the difficulty level of this item to $d_i = s$.
3) We constructed the action sequence $\mathcal{A}_u$ of each user $u \in \mathcal{U}$.
   a) To determine the sequence length $|\mathcal{A}_u|$, we drew a sample from a Poisson distribution with a mean of 50.
   b) We selected the user u's initial skill level s_{ut_1} from $\mathcal{S}$ uniformly at random.
   c) To decide u's current (n-th) action, we randomly sampled an item i_n from $\{i \in \mathcal{I} \mid d_i = s_{ut_n}\}$ with the probability $p = 0.5$, and from the easier item pools otherwise, reflecting our assumption that users usually select items within their skill capacity.
   d) When $d_{i_n} = s_{ut_n}$ in the step (c), we set the next skill level to $s_{ut_{n+1}} = s_{ut_n} + 1$ with $p = 0.1$, and to $s_{ut_{n+1}} = s_{ut_n}$ otherwise. When an easier item was selected instead, we kept the current skill level because the item was less likely to improve the user's skill.
   e) We repeated the steps (c) and (d) to obtain $|\mathcal{A}_u|$ actions.

B. Experimental Setup

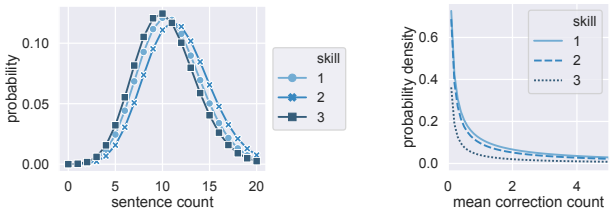
Filtering. As mentioned in Section IV-B, users with few actions are less likely to experience more than one skill level. To focus on action sequences where skill improvement is more likely to occur, we filtered out short sequences from the Beer and Film datasets. Specifically, we excluded users whose action sequences contained less than 50 unique items. For a similar reason, we also excluded items selected by less than 50 unique users. These thresholds were taken from Yang *et al.*'s settings [6]. We used the sequences of all the filtered users when initializing our skill model for these datasets.

For the smaller Language and Cooking datasets, however, this filtering process was too aggressive to keep sufficient data (*e.g.*, less than 500 users were left for the Language dataset). Moreover, each item in Lang-8 (*i.e.*, an article) was always selected by only one user (*i.e.*, the user who wrote that article). For these reasons, we did not apply the aforementioned filtering to these datasets. Instead, we only used action sequences whose length was no less than 50 when initializing the skill improvement model (*i.e.*, $\mathcal{U}_{\geq 50}$ in Section IV-B), following Shin *et al.* [10]. The same setting was applied to the Synthetic dataset.

Skill count. To learn the skill improvement in a given domain, we need to set the number of skill levels S in advance. This is possible for domains where one has prior knowledge. As the beer and film domains have been studied in the literature [6], [8], we followed the same setting (*i.e.*, $S = 5$) for the Beer and Film datasets in our experiments. As for a synthetic dataset, an arbitrary positive integer is acceptable. We thus simply set $S = 5$ for the Synthetic dataset for consistency with the datasets mentioned above.

In contrast, for domains where a proper value of S is unknown a priori, we determined it in a data-driven manner, as with Yang *et al.* [6]. Specifically, we first randomly split

Fig. 3: For domains without prior knowledge, we use the number of skill levels that maximizes log likelihood for held-out test data.



(a) Sentence count distributions

(b) Per-annotator correction count distributions

Fig. 4: Model components learned for language domain.

the whole dataset into a training dataset (90%) and a testing dataset (10%). Then, for each different value of S, we learned the model with the training dataset and measured the log likelihood for the testing dataset. Finally, we selected the value that maximized the likelihood. When measuring the likelihood of observing an action in the testing dataset, we assumed that the user's skill level at the time of this action was the same as that of the chronologically closest action in the training dataset. Figure 3 shows the result of this procedure for the Cooking dataset, from which we set $S = 5$ for this domain. We set $S = 3$ for the Language dataset.

C. Q1: Skill Interpretation

Skill plays different roles in different domains (*e.g.*, the ability to cook dishes, the experience to enjoy the taste of beers). Capturing the improvement of such domain-dependent user skills is essential to building recommender systems for upskilling. To investigate the interpretability of our skill model, we analyzed the model components learned from each dataset.

Language. First, we investigated how English learners improved their foreign language skills. Figure 4 shows the results for two features of the Language dataset. The sentence count feature showed no noticeable trend between different skill levels. The mean of the distribution for each $s \in \{1, 2, 3\}$ was 10.837, 11.633, and 10.320, respectively. On the other hand, the correction count feature tended to decrease as their skill improved (the mean values were 5.062, 4.852, 2.640 for $s = 1$, 2, 3, respectively). Putting these results together, our model was able to learn the tendency that novice language learners are more likely to receive many corrections per sentence.

To dig into the difference between skilled and unskilled users, we analyzed correction rules dominant in each user group. To this end, we measured the difference in the probability of observing a given rule x as the f-th item feature between the highest and lowest skill levels (*i.e.*, $P_f(i_f = x \mid \theta_f(S)) - P_f(i_f = x \mid \theta_f(1))$). This follows McAuley and Leskovec [8], who measured *acquired tastes* as the differences

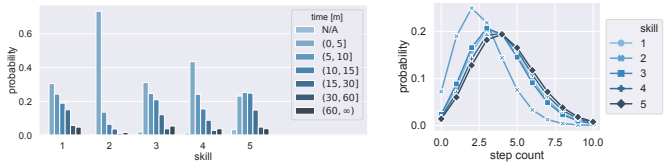
TABLE II: Top 10 corrections dominated by unskilled and skilled language learners, with stronger degrees of dominance indicated by negative and positive scores, respectively.

(a) Users with lowest skill level

Before	After	Score
"i"	**"I"**	−0.0038
ε	"I"	−0.0022
"english"	**"English"**	−0.0013
ε	"a"	−0.0012
e	**"."**	−0.0011
ε	"my"	−0.0007
"."	**e**	−0.0007
ε	"English"	−0.0006
","	**e**	−0.0006
"i"	ε	−0.0006

(b) Users with highest skill level

Before	After	Score
e	**"the"**	0.0014
e	**"("**	0.0009
e	**")"**	0.0009
"the"	**e**	0.0009
ε	"of"	0.0007
"of"	ε	0.0004
e	**"["**	0.0003
e	**"]"**	0.0003
"a"	**"the"**	0.0003
ε	"/"	0.0002



(a) Time distributions (b) Step count distributions

Fig. 5: Model components learned for cooking domain.

in the item bias terms of latent factor models [31] learned for experienced and inexperienced users. Under this measure, it can be said that a rule with a low/high score is dominated by unskilled/skilled users.

Table II lists the top 10 correction rules dominated by unskilled and skilled users. (Note that the symbol ε denotes a missing or deleted word.) Problems with capitalization (*e.g.*, "i" → "I") and punctuation (ε → ".") were typical corrections for unskilled users. For skilled users, on the other hand, we can observe several correction rules that insert parentheses or brackets, which indicate the existence of comments, including "(OK)," given by annotators [28]. Other common rules for the skilled users concerned the use of English articles (*e.g.*, "a" → "the"). These kinds of errors have been reported as common even for advanced Japanese students learning English [32]. Given that Japanese accounted for about one third of the users in this dataset, this finding may suggest that the latent variables in our model were successful in capturing the improvement of the writing skill of English learners.

Cooking. To understand the progression of cooking skill, we analyzed the probability distributions of the following two features learned for the Cooking dataset: the cooking time, and the number of cooking steps. Figure 5 summarizes the learning results for this domain. For skill levels from $s = 2$ to $s = 4$, all the distributions exhibited a common trend. That is, users tended to select more complex recipes (requiring longer time and more steps) as their skill improved. Interestingly, the distributions for the lowest skill level turned out to have shapes similar to those for the medium skill level. This suggests that users without enough skill tended to select too complex recipes in their early actions. A possible explanation for this phenomenon is that they were unable to estimate whether the difficulty of a selected recipe was beyond their cooking skill. The result on this domain violates our assumption on item selection, making the difficulty estimation inaccurate. This calls for modeling user satisfaction in action sequences, which we discuss further in Section VII.

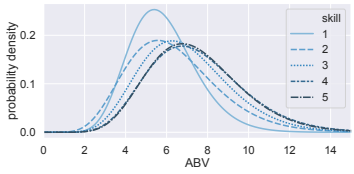


Fig. 6: Alcohol-by-volume (ABV) distributions for each skill level.

TABLE III: Top 10 beer styles dominated by unskilled and skilled users. Scores were calculated similarly to those in Table II.

(a) Users with lowest skill level

Name	Score
Pale Lager	−0.123
Premium Bitter/ESB	−0.020
American Dark Lager	−0.018
Porter	−0.014
German Hefeweizen	−0.014
Amber Ale	−0.014
Premium Lager	−0.013
Malt Liquor	−0.011
Vienna	−0.011
Wheat Ale	−0.010

(b) Users with highest skill level

Name	Score
Imperial/**Double IPA**	0.056
Imperial Stout	0.050
Sour Ale/**Wild Ale**	0.037
India Pale Ale (IPA)	0.035
American Strong Ale	0.032
Saison	0.028
Barley Wine	0.024
Black IPA	0.019
Belgian Strong Ale	0.014
Spice/Herb/Vegetable	0.013

Beer. Next, we present the learning result for the beer domain, which may be involved with users' appreciation skills. Figure 6 shows the ABV distributions learned from the Beer dataset. The distribution was more likely to generate high ABV values as the skill level improved (*e.g.*, the means for $s = 1$ and $s = 5$ were 5.846 and 7.460, respectively). In other words, our model detected that skilled users tended to prefer high-ABV beers, which is in line with past findings [8].

Similarly to our analysis for the Language dataset, we also analyzed the difference in beer style preference between unskilled and skilled users. Table III lists the top 10 beer styles dominated by users with the lowest and highest skill levels. As shown in bold in the table, unskilled users more often selected *lager* beers (*e.g.*, *Pale Lager*) while skilled users tended to prefer *stronger*, more *hoppy* beers (*e.g.*, *India Pale Ale*). These trends are consistent with the findings of McAuley and Leskovec [8]. Note, however, that our model does not require rating scores, unlike their model.

Film. For the film domain, we analyzed the movies watched by users with different skill levels. Table IV lists the top 10 frequent movies (in terms of the learned probability distributions of the movie ID feature) for the lowest and highest skill levels. We can observe that most movies in Tables IVa and IVb were released in the 1980s and 2000s, respectively. While not shown in this paper due to space limitation, typical movies for the medium skill level were mostly released in the 1990s. As mentioned in Section IV-B, users in this domain tend to watch recent movies. Thus, old/new movies are more likely to appear at early/late positions in user sequences. Due to this *lastness* effect, the model mistakenly regarded temporal drifts as user skills in this domain.

To resolve this issue, we preprocessed the Film dataset before learning our skill model. Specifically, we excluded movies that were released after the earliest action in the whole data (*i.e.*, after $\min_{(t,u,i)\in\mathcal{A}} t$), which ensured that every movie could be selected at any time. The result for the preprocessed dataset is summarized in Table V. In contrast to Table IV, the

TABLE IV: Top 10 frequent movies for lowest and highest skill levels (without preprocessing for mitigating lastness effect).

(a) Lowest skill level

Name	Year
Star Wars: Episode IV - A New Hope	1977
Star Wars: Episode V - The Empire Strikes Back	1980
Indiana Jones and the Raiders of the Lost Ark	1981
The Godfather	1972
Back to the Future	1985
Star Wars: Episode VI - Return of the Jedi	1983
Casablanca	1942
The Silence of the Lambs	1991
Fargo	1996
The Princess Bride	1987

(b) Highest skill level

Name	Year
The Dark Knight	2008
Iron Man	2008
Avatar	2009
V for Vendetta	2006
Batman Begins	2005
WALL•E	2008
Juno	2007
Little Miss Sunshine	2006
Inception	2010
Casino Royale	2006

TABLE V: Top 10 frequent movies for lowest and highest skill levels (with preprocessing for mitigating lastness effect).

(a) Lowest skill level

Name	Year
Pulp Fiction	1994
Star Wars: Episode IV - A New Hope	1977
Star Wars: Episode VI - Return of the Jedi	1983
Star Wars: Episode V - The Empire Strikes Back	1980
Batman	1989
Dances with Wolves	1990
Indiana Jones and the Raiders of the Lost Ark	1981
The Shawshank Redemption	1994
True Lies	1994
Jurassic Park	1993

(b) Highest skill level

Name	Year
Rear Window	1954
The Sound of Music	1965
The Graduate	1967
It's a Wonderful Life	1946
The Birds	1963
Gone with the Wind	1939
Psycho	1960
Casablanca	1942
Vertigo	1958
Citizen Kane	1941

list for the lowest skill level includes many *light* movies, such as *Star Wars*, *Indiana Jones*, and *Jurassic Park*. On the other hand, the list for the highest skill level contains movies that are not necessarily widely appealing but are regarded as *classics*. Such examples include *Rear Window*, *Casablanca*, and *Citizen Kane*. A list for the medium skill level (omitted due to space limitation) consists of a mixture of both sides (*e.g.*, *Star Wars* from the lowest and *Casablanca* from the highest).

D. Q2: Accuracy

We also evaluated the objective performance of our models for skill improvement and item difficulty. As our real-world datasets did not contain the ground truth about skill and difficulty levels, we experimented with the Synthetic dataset for this purpose.[8] For comparison with our skill model (hereinafter denoted as Multi-faceted) proposed in Section IV, we prepared two baselines:

- Uniform, which segments each user sequence into S groups of equal length and then assigns the skill level $s \in \mathcal{S}$ to all actions in the s-th group.
- ID, which only uses the ID feature of each item as a component of the skill improvement model. This is equivalent to the existing progression model [6] except for the progression class component, which was excluded from both this model and ours for simplicity and fair comparison.

Following Yang *et al*. [6], we did not compare our model with classic clustering methods that ignore the ordering of latent variables. For the item difficulty evaluation, we compared combinations of the above skill models and our three difficulty models (Assignment, Uniform, and Empirical). Note that the Uniform skill model cannot be combined with the Uniform and Empirical difficulty models, as it does not model the generative process of items. We adopted three correlation measures (Pearson's r, Spearman's ρ, and Kendall's τ) and one error measure (root mean squared error, or RMSE) to evaluate the accuracy of estimating skill and difficulty levels.

[8]While the existing work [6] focused on the stages of chronic kidney disease for a similar experiment, their medical dataset is not the domain of skill learning.

TABLE VI: Accuracy of skill assignment for Synthetic dataset.

Model	Pearson's r	Spearman's ρ	Kendall's τ	RMSE
Uniform	0.345	0.336	0.279	1.767
ID [6]	0.499	0.496	0.417	1.652
ID+categorical	0.651	0.656	0.563	1.571
ID+gamma	0.676	0.680	0.584	1.527
ID+Poisson	0.759	0.775	0.677	1.427
Multi-faceted	**0.819**	**0.842**	**0.754**	**1.316**

TABLE VII: Accuracy of difficulty estimation for Synthetic dataset.

Model		Pearson's r	Spearman's ρ	Kendall's τ	RMSE
Skill	Difficulty				
Uniform	Assignment	0.501	0.516	0.389	1.234
ID [6]	Assignment	0.641	0.653	0.505	1.111
	Uniform	0.637	0.647	0.497	1.127
	Empirical	0.649	0.658	0.507	1.113
Multi-faceted	Assignment	0.858	0.856	0.713	0.777
	Uniform	0.920	0.921	0.800	0.620
	Empirical	**0.921**	**0.925**	**0.805**	**0.614**

Skill improvement. Table VI summarizes the performance of the skill models. As expected, the simplest baseline, Uniform, performed most poorly. The performance improved when we used the other baseline, ID, indicating that progression modeling is promising for this problem. Furthermore, we can observe from the table that adding each item feature did contribute to the performance improvement. When leveraging all item features (*i.e.*, Multi-faceted), we achieved the best performance for all evaluation measures. The 95% CI of Pearson's r for Multi-faceted was $[0.818, 0.820]$, substantially higher than $[0.342, 0.347]$ for Uniform and $[0.497, 0.501]$ for ID. A Wilcoxon signed-rank test with the Bonferroni correction also revealed that Multi-faceted significantly improved the squared error (SE) over both baselines, with $p < 0.01$.

Item difficulty. Table VII lists the performance of the difficulty models. As the item difficulty estimation depends on the performance of the user skill assignment, we can observe a trend similar to that in Table VI: Multi-faceted performed best, followed by ID and then Uniform. As for the difficulty estimation models, Assignment was slightly better than Uniform for ID. In contrast, when focusing on the result of Multi-faceted, Uniform outperformed Assignment. This suggests that the generation-based estimation with the uniform skill prior was not robust when we included a small number of features in our skill improvement model. In contrast, Empirical improved the performance in both cases as it used the empirical skill prior (computed from the assigned skill levels) to estimate the item difficulty. The 95% CI of Pearson's r for our best model (Multi-faceted with Empirical) was substantially higher than those for both baselines ($[0.920, 0.923]$ vs. $[0.494, 0.507]$ and $[0.636, 0.646]$). This best model also achieved significantly smaller SE than the baselines did ($p < 0.01$).

To assess our claim about the advantage of the generation-based models (Section V), we also evaluated the difficulty estimation performance for 291 *rare* items that appeared less than three times in the Synthetic dataset. For these items,

TABLE VIII: Accuracy of skill assignment for $\text{Synthetic}_{\text{dense}}$ dataset.

Model	Pearson's r	Spearman's ρ	Kendall's τ	RMSE
Uniform	0.340	0.334	0.277	1.768
ID [6]	0.925	0.940	0.891	0.954
Multi-faceted	**0.929**	**0.946**	**0.900**	**0.900**

TABLE IX: Accuracy of difficulty estimation for $\text{Synthetic}_{\text{dense}}$ dataset.

Model		Pearson's r	Spearman's ρ	Kendall's τ	RMSE
Skill	**Difficulty**				
Uniform	Assignment	0.794	0.799	0.648	1.162
ID [6]	Assignment	0.948	0.954	0.848	0.660
	Uniform	0.949	0.957	0.853	0.670
	Empirical	0.948	0.954	0.848	0.665
Multi-faceted	Assignment	**0.950**	**0.960**	**0.859**	0.632
	Uniform	0.932	0.927	0.808	0.539
	Empirical	0.932	0.928	0.809	**0.528**

the RMSE scores for Assignment and Empirical were 1.131 (46% increase) and 0.833 (36% increase), respectively. This demonstrates that, for rare items, the generation-based estimation was more robust than the assignment-based estimation.

Data sparsity. To investigate the effect of data sparsity on the modeling performance of user skills and item difficulty, we conducted similar experiments with another synthetic dataset (hereinafter denoted as $\text{Synthetic}_{\text{dense}}$). The only difference between the $\text{Synthetic}_{\text{dense}}$ and Synthetic datasets are the number of items: 10,000 for the former and 50,000 for the latter. In other words, items in $\text{Synthetic}_{\text{dense}}$ is selected, on average, five times more than those in Synthetic. Thus, if a model $\mathcal{M}_1$'s improvement over another model $\mathcal{M}_2$ is greater in Synthetic than in $\text{Synthetic}_{\text{dense}}$, it can be said that $\mathcal{M}_1$ is robust against domains suffering from data sparsity.

Table VIII shows the performance of each skill model for the $\text{Synthetic}_{\text{dense}}$ dataset. While the order of models in terms of their performance was unchanged from Table VI, the difference between the Multi-faceted and ID skill models was smaller than that for the Synthetic dataset. Table IX shows the performance of each difficulty model for the $\text{Synthetic}_{\text{dense}}$ dataset. Similarly, while Multi-faceted outperformed ID, the difference for this dataset was smaller compared with that for the Synthetic dataset (Table VII). It is also worth noting that when combined with the Multi-faceted skill model, the Assignment difficulty model performed better than the Empirical difficulty model for all except one (RMSE) measures, which is contrary to Table VII. This result supports our claim that Assignment does not work well for rare items.

In summary, we found a smaller improvement by our skill and difficulty models when experimented with more dense data. As described earlier, this finding indicates that our approach is particularly beneficial to domains with many items where data sparsity can be a serious issue.

E. Q3: Usefulness

For further quantitative evaluation, we investigated the practical usefulness of the learned skill and difficulty levels in the context of user behavior analysis and recommender systems. Specifically, we focused on two prediction tasks: items and ratings.

Item Prediction. This task aims to predict items selected by users. Following Yang *et al*. [6], we considered two task settings: predicting items at at a *random* position and the *last* position in each sequence. The former is designed for missing data recovery, while the latter measures the ability to forecast the future. For each setting, we used one action from each user sequence for testing and used the rest for model training. As in our evaluation on accuracy (Section VI-D), we used Uniform and ID as baselines. Note that ID has been reported to outperform common prediction models based on machine learning (*i.e*., logistic regression and hidden Markov model) [6]. Among the four real datasets, the Language dataset was excluded in this experiment, because each item in that domain (*i.e*., each article) was selected (written) only once by a particular user.

TABLE X: Performance on predicting items at random positions.

	Cooking		Beer		Film	
Model	Acc@10	RR	Acc@10	RR	Acc@10	RR
Uniform	0.023	0.011	0.019	0.011	0.095	0.044
ID [6]	0.050	0.024	0.025	0.014	0.102	0.046
Multi-faceted	**0.073**	**0.035**	**0.029**	**0.016**	**0.109**	**0.049**

TABLE XI: Performance on predicting items at last positions.

	Cooking		Beer		Film	
Model	Acc@10	RR	Acc@10	RR	Acc@10	RR
Uniform	0.012	0.007	0.008	0.006	0.045	**0.024**
ID [6]	0.043	0.018	0.015	0.008	0.044	0.023
Multi-faceted	**0.060**	**0.026**	**0.018**	**0.009**	**0.047**	0.022

We used each model to make item predictions as follows. (1) We inferred the skill level for each test action from its nearest action in the corresponding training sequence (in the same way that we estimated the skill count in Section VI-B). (2) We selected the probability distribution of item IDs for the inferred skill level. (3) We ranked the items in descending order of probability. The resulting ranking was evaluated with two measures: the top-10 accuracy (Acc@10), which returns 1 if the correct item is ranked in the top 10 and 0 otherwise, and the reciprocal rank (RR), which returns the reciprocal of the rank of the correct item.

Table X lists the mean Acc@10 and RR scores of each model for the random setting. The proposed Multi-faceted model consistently outperformed the two baselines. For all cases, the improvement in RR over both baselines was significant ($p < 0.01$). Note that our model performed much better than random guessing, whose expected Acc@10 and RR scores are $10|\mathcal{I}|^{-1}$ and $\sum_{i=1}^{|\mathcal{I}|}(i|\mathcal{I}|)^{-1}$, respectively. As shown in Table XI, the last setting yielded a similar result except for the Film dataset, where all models performed comparably in terms of RR. A possible explanation for this exception is that the dataset was not very sparse as it had fewer items than other datasets. For both settings, Multi-faceted made the largest improvement for Cooking, which has more items than the other datasets (Table I). An increase in the number of items can accelerate the sparsity of selected items, causing the ID feature alone to be less informative. Exploiting features shared by multiple items enabled our model to achieve the great performance improvement, especially for domains suffering from data sparsity.

Rating Prediction. This task aims to predict rating scores that users provided to items. We considered the same prediction settings as the item prediction task (*i.e*., random and

TABLE XII: Performance (RMSE) on predicting beer ratings.

Position	U+I [31]	U+I+S	U+I+D	U+I+S+D
Random	0.572	0.569	0.569	**0.568**
Last	0.571	0.562	0.568	**0.561**

TABLE XIII: Running time (in hours) of skill model training with different parallelization conditions on Film dataset.

Parallelized?			**Model**	
User	Feature	Skill	ID [6]	Multi-faceted
✗	✗	✗	0.944	9.557
✓	✗	✗	0.425	4.272
✗	✓	✗	N/A	8.305
✗	✗	✓	0.901	8.224
✓	✓ (✗ for ID)	✓	0.374	2.814

last) and used Beer and Film since these are only the datasets at hand that contain rating records. To make the rating scale consistent across these datasets, we normalized all ratings to $[0, 5]$. We learned rating prediction models by using Field-aware Factorization Machine (FFM) [33], an expressive model that can take interactions between features into account. As a baseline, we only used user and item IDs as features to learn an FMM model, which is equivalent to matrix factorization with user and item biases [31]. We investigated how the prediction performance (measured by RMSE) would change by adding skill and difficulty levels as additional features. We followed Juan *et al*. [33] to select the parameters of FFM models.

Due to space limitation, we only show the prediction result for the Beer dataset in Table XII. We can observe a consistent trend for both settings. That is, the FFMs with either skill levels (U+I+S) or difficulty levels (U+I+D) outperformed the baseline (U+I). The best performance was achieved when we added all features (U+I+S+D), suggesting that user skill and item difficulty offer a complementary contribution to the rating prediction. The difference between U+I and U+I+S+D was shown to be significant for both settings ($p = 0.01$).

F. Q4: Efficiency

Training a skill model is the most time-consuming process. As the final experiments, we investigated the efficiency of our parallelization techniques for the model training described in Section IV-C. For this purpose, we used the Film dataset, which is the biggest one at hand.

Table XIII lists the running times of training skill models using five threads with different parallelization conditions (Section IV-C). When trained sequentially, our Multi-faceted model took 8.5 hours more than the ID baseline, since the former consisted of more probabilistic distributions. Among the three techniques, the most effective one was parallelizing assignments for each user, which is in line with our complexity analysis (Section IV-C). The feature-based parallelization, which is only applicable to our model, also contributed to improving the efficiency. Enabling all parallelization techniques reduced the runtime difference between Multi-faceted and ID to less than 2.5 hours. We also changed the number of threads when training skill models with all parallelization techniques enabled. As shown in Figure 7, our Multi-faceted model gained greater benefits from the parallelization with more threads. We believe the reduced running time of Multi-faceted to be acceptable, given its significant improvement

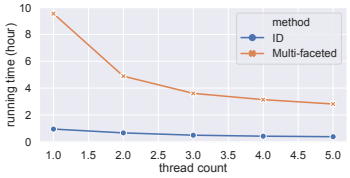


Fig. 7: Running time (in hours) of skill model training with different numbers of threads on Film dataset.

over ID in terms of accuracy and usefulness and the offline nature of the skill improvement problem.

VII. Discussion

Our approach has several limitations. While we assume the independence of item features for simplicity, it may not hold in some situations. The monotonicity of skill improvement is another assumption we made. It is, however, possible that users lose some skills if they have not taken actions for a while. Thus, helping users recover forgotten skills is as important as helping them acquire new skills. According to Ebbinghaus's forgetting curve [34], time and repetition play important roles in memory retention, suggesting that the time gap between consecutive actions and the number of actions selecting the same item [35] could be useful to address this challenge. Other interesting extensions include the consideration of user information in the skill model and the joint optimization of the skill and difficulty models. A user study is also worth conducting to investigate whether the learned skill and difficulty levels align with users' perception.

In what follows, we discuss further steps that need to be taken toward recommendation for upskilling.

How to model user satisfaction for their actions. Users in some domains (*e.g.*, cooking, as reported in Section VI-C) may select too difficult items when they are inexperienced. A model that learns such actions as being typical for unskilled users would repeat the same mistake by recommending difficult items to them. This calls for estimating whether users are satisfied with their actions and incorporating user satisfaction into the skill model. Lessons from the past work on modeling searcher success [36] and satisfaction [37] would be valuable to address this challenge.

How to combine our models with recommender systems. While skill improvement and item difficulty are two major concepts, modeling these alone is still insufficient to make recommendation for upskilling. Recommended items should not only have appropriate difficulty for a target user but also match his/her interest. How can we combine our skill and difficulty models with state-of-the-art recommender systems to personalize the recommendation for upskilling? In Section VI-E, we demonstrated our models' potential to benefit recommendation based on rating prediction. The question above would be more challenging particularly for domains where user ratings are not available.

When and how to recommend items for upskilling. This question still remains unanswered even if the above-mentioned two challenges have been solved. Recommending items for upskilling would be effective only if they are presented at the

right time in the right manner. Otherwise, such recommendation may merely confuse and/or frustrate users. How can we detect appropriate timing? What interfaces, interactions, and explanations are suitable for this recommendation? While the present work considers the difficulty of individual items, finding a ranking of items optimized for skill improvement would be one interesting direction to explore [26].

VIII. Conclusions

This paper has proposed models that, given action sequences, learn the improvement of user skills and estimate the difficulty of each item. To address the data sparsity problem in real domains, our skill model extends the existing progression model by considering the multi-faceted features shared across multiple items. Once trained, the skill model is used to estimate the difficulty level as the mean or expected skill level of users who select each item. Experiments on four real datasets and one synthetic dataset demonstrated the interpretability of the captured domain-dependent skill improvement, the accuracy of the learned skill and difficulty levels, the usefulness of our models for the item/rating prediction tasks, and the efficiency of our training algorithm.

This work has taken a first step toward recommendation for upskilling. Future directions to take further steps include the following: modeling user satisfaction in action sequences to prevent users from repeating the same mistake, combining our models with recommender systems to find items that match both the interests and skills of users, and studying the best time and manner to present recommendations to users.

Acknowledgments

This work has been partially funded by the Japan Society for the Promotion of Science, the Israel Science Foundation, and the Binational US-Israel Science Foundation. It has also been supported by JSPS KAKENHI Grant Number JP17K12787.

References

[1] X. Qian, H. Feng, G. Zhao, and T. Mei, "Personalized recommendation combining user interest and social circle," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1763–1777, 2014.

[2] I. Szpektor, Y. Maarek, and D. Pelleg, "When relevance is not enough: Promoting diversity and freshness in personalized question recommendation," in *WWW '13*, 2013, pp. 1249–1260.

[3] M. Ge, C. Delgado-Battenfeld, and D. Jannach, "Beyond accuracy: Evaluating recommender systems by coverage and serendipity," in *RecSys '10*, 2010, pp. 257–260.

[4] R. Cañamares and P. Castells, "Should i follow the crowd?: A probabilistic analysis of the effectiveness of popularity in recommender systems," in *SIGIR '18*, 2018, pp. 415–424.

[5] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *SIGIR '14*, 2014, pp. 83–92.

[6] J. Yang, J. McAuley, J. Leskovec, P. LePendu, and N. Shah, "Finding progression stages in time-evolving event sequences," in *WWW '14*, 2014, pp. 783–794.

[7] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *WWW '16*, 2016, pp. 507–517.

[8] J. J. McAuley and J. Leskovec, "From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews," in *WWW '13*, 2013, pp. 897–908.

[9] X. Wang, D. Sontag, and F. Wang, "Unsupervised learning of disease progression models," in *KDD '14*, 2014, pp. 85–94.

[10] K. Shin, M. Shafiei, M. Kim, A. Jain, and H. Raghavan, "Discovering progression stages in trillion-scale behavior logs," in *WWW '18*, 2018, pp. 1765–1774.

[11] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *ICDM '18*, 2018, pp. 197–206.

[12] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *WWW '10*, 2010, pp. 811–820.

[13] Y. Baba and H. Kashima, "Statistical quality estimation for general crowdsourcing tasks," in *KDD '13*, 2013, pp. 554–562.

[14] H. Rahman, S. Thirumuruganathan, S. B. Roy, S. Amer-Yahia, and G. Das, "Worker skill estimation in team-based tasks," *Proceedings of the VLDB Endowment*, vol. 8, no. 11, pp. 1142–1153, 2015.

[15] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Modeling and User-Adapted Interaction*, vol. 4, no. 4, pp. 253–278, 1994.

[16] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," in *NIPS '15*, 2015, pp. 505–513.

[17] C. Hölscher and G. Strube, "Web search behavior of internet experts and newbies," *Computer Networks*, vol. 33, no. 1-6, pp. 337–346, 2000.

[18] R. W. White, S. T. Dumais, and J. Teevan, "Characterizing the influence of domain expertise on web search behavior," in *WSDM '09*, 2009, pp. 132–141.

[19] J. Sweller, "Cognitive load theory, learning difficulty, and instructional design," *Learning and Instruction*, vol. 4, no. 4, pp. 295–312, 1994.

[20] T. Kajiwara and A. Fujita, "Semantic features based on word alignments for estimating quality of text simplification," in *IJCNLP '17*, 2017, pp. 109–115.

[21] K. Collins-Thompson, P. N. Bennett, R. W. White, S. de la Chica, and D. Sontag, "Personalizing web search results by reading level," in *CIKM '11*, 2011, pp. 403–412.

[22] J. Liu, Q. Wang, C.-Y. Lin, and H.-W. Hon, "Question difficulty estimation in community question answering services," in *EMNLP '13*, 2013, pp. 85–90.

[23] J. Whitehill, T. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *NIPS '09*, 2009, pp. 2035–2043.

[24] K. Collins-Thompson, P. Hansen, and C. Hauff, "Search as Learning (Dagstuhl Seminar 17092)," *Dagstuhl Reports*, vol. 7, no. 2, pp. 135–162, 2017.

[25] S. Ghosh, M. Rath, and C. Shah, "Searching as learning: Exploring search behavior and learning outcomes in learning-related tasks," in *CHIIR '18*, 2018, pp. 22–31.

[26] R. Syed and K. Collins-Thompson, "Retrieval algorithms optimized for human learning," in *SIGIR '17*, 2017, pp. 555–564.

[27] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *SIGIR '98*, 1998, pp. 275–281.

[28] T. Mizumoto, M. Komachi, M. Nagata, and Y. Matsumoto, "Mining revision log of language learning sns for automated japanese error correction of second language learners," in *IJCNLP '11*, 2011, pp. 147–155.

[29] M. J. Parsons, "The skill of appreciation," *Journal of Aesthetic Education*, vol. 7, no. 1, pp. 75–82, 1973.

[30] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, pp. 19:1–19:19, 2015.

[31] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[32] J. Yamada and N. Matsuura, "The use of the english article among japanese students," *RELC Journal*, vol. 13, no. 1, pp. 50–63, 1982.

[33] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin, "Field-aware factorization machines for ctr prediction," in *RecSys '16*, 2016, pp. 43–50.

[34] H. Ebbinghaus, "Memory: A contribution to experimental psychology," *New York: Teachers College, Columbia University*, 1913.

[35] K. Nagatani, Q. Zhang, M. Sato, Y.-Y. Chen, F. Chen, and T. Ohkuma, "Augmenting knowledge tracing by considering forgetting behavior," in *WWW '19*, 2019, pp. 3101–3107.

[36] M. Ageev, Q. Guo, D. Lagun, and E. Agichtein, "Find it if you can: A game for modeling different types of web search success using interaction data," in *SIGIR '11*, 2011, pp. 345–354.

[37] H. Wang, Y. Song, M.-W. Chang, X. He, A. Hassan, and R. W. White, "Modeling action-level satisfaction for search task satisfaction prediction," in *SIGIR '14*, 2014, pp. 123–132.