Data Mining for Discovering Cognitive Models of Learning

JINJIN ZHAO, JZ, ZHAO

Amazon

CANDACE THILLE, CT, THILLE

Amazon

DAWN ZIMMARO, DZ, ZIMMARO

Amazon

A cognitive model is a descriptive account or computational representation of human thinking about a given concept, skill, or domain. A cognitive model of learning, includes both a way of organizing knowledge within a subject area and an account of how humans develop accurate and complete knowledge of that subject area. Learning designers engage in a variety of practices to unpack knowledge from subject matter experts and novices to develop cognitive models of learning and use those models to guide the design of instruction or instructional technologies. Traditional approaches to eliciting and organizing knowledge, such as conducting a cognitive task analysis (CTA) [14] with experts and novices, are labor-intensive and require specific expertise that many learning designers do not have. However, learning data generated from learners' interaction with courses, can provide insight into how humans think and develop knowledge. As a continued effort, we extend the framework presented in our earlier work [17] to discover and refine cognitive models of learning with learning data. The framework includes 1. a Variational Autoencoder (VAE) and a Gaussian Mixture Model (GMM) that models and clusters cognitive learning patterns; 2. a multidimensional measure that quantifies validity and reliability of the discovered cognitive models of learning; 3. a topic-based solution that interprets the cognitive models from a linguistic perspective; and 4. a simulation-based analysis for both accuracy measures and course refinement insights. We demonstrate the end-to-end solution with two applications and four case studies that are deployed in an openly navigated learning system in a workforce learning environment. We also report the usefulness of the discovered cognitive models of learning with subject matter expert evaluation.

CCS CONCEPTS • Applied computing • Education • E-learning

Additional Keywords and Phrases: cognitive model of learning; human-computer interaction; behavior modeling; natural language processing; knowledge tracing

1 INTRODUCTION

A cognitive model is a descriptive account or computational representation of human thinking about a given concept, skill, or domain. A cognitive model of learning, includes both a way of organizing knowledge within a subject area and an account of how humans develop accurate and complete knowledge of that subject area. Learning designers engage in a variety of practices to unpack knowledge from subject matter experts and novices to develop cognitive models of learning and use those models to guide the design of instruction or instructional technologies. Cognitive models of learning (interchangeable with skill model, knowledge map) provide guidance on the design of instruction or instructional technologies. Learning designers engage in a variety of practices to unpack knowledge from subject matter experts (SMEs) and novices to develop such a knowledge map. Cognitive Task Analysis (CTA) is a standard approach for eliciting knowledge from experts. Learning designers use observational and interview strategies to capture accurate and complete descriptions of expert knowledge, which includes both how experts structure and how humans develop that knowledge. There is compelling evidence that experts are not fully aware of about 70% [3] of their own decisions or how they execute tasks, and are unable to explain tasks fully during interviews. The CTA process is not only labor-intensive but also requires skills that many learning designers do not have to elicit expertise.

Learning behavior data is one kind of observation of the cognitive process of learning. If collected at the right grain size, the sequential behavior data provide step-wise information about the process in which learners develop knowledge and skills. Studies have shown the benefits of mining learning behavior data, especially learner-assessment attempt correctness, to discover the cognitive process of learning. In Learning Factors Assessment (LFA) [2], the skill model is created by fitting a statistical model with human defined learning factors. One shortcoming of LFA is that the learning factors are defined through a massive iterative evidence-based engineering process. Another shortcoming is the pre-defined factor space limits the model from discovering new dimensions that could potentially better explain and represent the cognitive process of learning. eEPIPHANY [11] uses an automated solution to discover the skill model with a Non-negative Matrix Factorization (NMF) technique [8]. Compared to LFA, NMF-based solutions could help reduce learning factor engineering effort; however, NMF-based techniques are sensitive to data sparsity and quality [16]. The major drawback lies in its slow convergence, lack of well-validated methodology for hyperparameter selection, and low reliability of the solution [10]. In cognitive modeling, due to the covert nature of the cognitive process, the field rely on human judgement to evaluate the accuracy of the discovered skill models and incorporate useful skill models into the instructional design. A robust and reliable cognitive model discovery solution can provide accurate insights as well as reduce human evaluation effort.

In previous work [17], the authors first applied Variational Autoencoder (VAE) to construct a task latent representation, hypothesizing that behavior data follow a Gaussian distribution. The authors then applied a Gaussian Mixture Model (GMM) to cluster the latent representation into knowledge components (KC). The VAE models the data distribution and provides the possibility of result interpretation. The authors also demonstrated that the shared data distribution between representation learning and task clustering results better accuracy. We use the same representation learning and task clustering in the current work. We enhanced the clustering step with a best cluster number selection step. We also extend the work by proposing a multidimensional quality measure to quantify the validity of the discovered skill models. The measure solution includes 1. a stability measure with reconstruction loss, loss variance, and item wise robustness statistics; 2. an accuracy measure with knowledge tracing simulation; and 3. a decomposition degree measure with KC orthogonality. In addition to the validity measures, we provide interpretation of the result from a linguistic perspective - leveraging Latent Dirichlet Allocation (LDA) to analyze the key concepts and concepts distribution of the course text. Finally, we conducted a best KC analysis for diagnosed tasks to provide insights for course refinement. The current work extends our prior work in following ways,

1. Extends VAE and GMM with a ‘best cluster number selection’ component in clustering.

2. Proposes a multidimensional measure to quantify the accuracy, robustness, and usefulness of the discovered skill models.

3. Applies NLP topic modeling for assisting designers in interpreting the discovered skill models.

4. Proposes to use knowledge tracing simulation for the diagnosed tasks for course refinement insights.

5. Provides studies for two scenarios and four cases in a workforce learning setting. The two scenarios are a. refining an initial knowledge map with learning data and b. discovering a knowledge map directly from learning data.

2 APPROACH

2.1 Framework Overview

This section presents the overall framework of the proposed cognitive model discovery solution. It contains a set of machine learning techniques from analyzing the learning data with neural networks, identifying cognitive models of learning with clustering algorithms, evaluating the accuracy of cognitive models with knowledge tracing techniques, providing interpretation with natural language processing (NLP) solutions, and using simulation-based analysis to derive insights for refining the knowledge map.

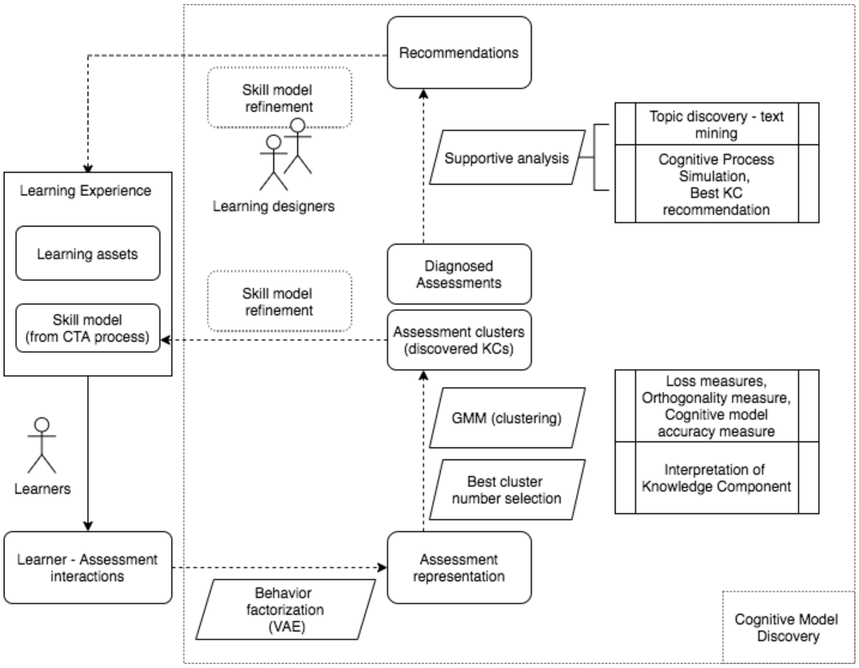

Recommendations
Skill model refinement
Learning designers
Supportive analysis
Topic discovery - text mining
Cognitive Process Simulation, Best KC recommendation
Learning Experience
Learning assets
Skill model (from CTA process)
Diagnosed Assessments
Skill model refinement
Assessment clusters (discovered KCs)
Loss measures, Orthogonality measure, Cognitive model accuracy measure
Interpretation of Knowledge Component
GMM (clustering)
Best cluster number selection
Learners
Learner - Assessment interactions
Assessment representation
Behavior factorization (VAE)
Cognitive Model Discovery

Figure 1. Cognitive model discovery framework.

As shown in Fig 1, learners interact with courses on an online learning platform. The learning data are collected and stored anonymously in a database. We apply behavior factorization with VAE to construct the task representation. We then apply GMM along with Best Cluster Number (see in Approach Section 2.3) selection algorithm to discover the underlying KCs (the task clusters). If the tasks share a same cognitive model, a cluster (a KC) is discovered. If a task does not share any cognitive model with other tasks, the task would be identified as an outlier - a diagnosed problematic task. Both discovered KCs and diagnosed tasks are provided as insights to learning designers for refining the knowledge map and the course. For accuracy and reliability evaluation, we propose a multidimensional measurement solution, including loss-based measures (reconstruction loss, loss variance, and item wise robustness), cognitive model simulation measure, and orthogonality measure. We also propose to provide interpretation for the discovered KCs with topic modeling technique. Lastly, we propose to provide simulation-based method to find insights for diagnosed tasks.

2.2 Task Representation Learning

Learning behavior is a representation of the covert cognitive process of learning. Learning behavior data include browsing activities, clicks, navigation paths, task attempts, attempt correctness, etc. Task attempts and their correctness are critical pieces of evidence that indicate the learner's knowledge state and the cognitive process through which the learner transits from one knowledge state to another. We hypothesize that the tasks designed for the same KC share the same task complexity [6]. Because the difficulty learners have in applying a KC to solve a task indicates the task complexity, we consider the task difficulty factor in discovering the KCs. We define task difficulty as 1–1/d, where d is the total attempt number until the first successful attempt. The reason for this definition is that a learner can make multiple attempts on a single task to demonstrate their proficiency of the underlying skill and the sequential attempt correctness indicates the task difficulty. The more attempts a learner needs to successfully solve a task, the more difficult the task is for that learner. We extract task difficulty at an individual level from the learning data.

After transforming interaction data into a learner-task difficulty matrix, we use the VAE technique to model and unpack the task difficulty. We assume there is a surrogate model that approximates the task difficulty and, within the surrogate model, there are latent factors that contribute to the task difficulty. We use the surrogate model and its latent factors to construct the task representation. We hypothesize the surrogate model is a generative model and the data generated from it follows Gaussian distribution (as a common practice). The reason we apply VAE is that VAE assumes that the data follows certain distribution and is generated by some latent surrogate model. The latent generative model is built by minimizing the Kullback-Leibler divergence between the surrogate generative model and the observation data. The objective of VAE is formulated in Eq. (1), see details in paper [17].

$$L(\phi, \theta, x) = D_{KL}(q_{\phi}(h|x)||p_{\theta}(h)) - E_{q_{\phi}(h|x)}(logp_{\theta}(x|h)) \tag{1}$$

As Eq. (2) shows, the reconstruction loss is added to the final objective function L to construct the latent space,

$$L = L(\phi, \theta, x) + L_{reconstruction}(x - x')^2 \tag{2}$$

where x is the observation and x' is the predicted value through reconstruction. By minimizing KL divergence and reconstructing the observations, a multivariate Gaussian process $p_{\theta}(x|h)$ is derived by learning an approximation $(q_{\phi}(h|x)$ to the posterior distribution of $(p_{\phi}(h|x)$. The derived Gaussian process approximates how the observation data is generated. The intermediate layer generated by the Gaussian process $p_{\theta}(x|h)$ is the task representation.

2.3 Best Cluster Number

During clustering the tasks based on its representations, we need a mechanism to determine the best cluster number. We propose to use two methods, Silhouette and BIC methods, collectively to provide a reliable and accurate clustering result. The Silhouette method [13] provides interpretation and validation of consistency within clusters of data. The technique provides a succinct graphical representation of how well each object has been classified. The Silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The Silhouette score is calculated based on the Silhouette coefficient between nodes,

$$s(i) = \frac{b(i)-a(i)}{\max\,(a(i),b(i))} \tag{3}$$

As Eq. (3) shows, for data point i, $s(i)$ is the silhouette coefficient, $a(i)$ is the average distance between i and all the other data points within the same cluster, $b(i)$ is the minimum average distance from i to all clusters to which i does not belong. The score range is [-1,1], where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters. Bayesian information criterion (BIC) [15] is a criterion for model selection among a finite set of models. It aims to reduce the probability of overfitting by introducing a penalty term that is associated with the parameter number of a model. The BIC score is widely used to choose the number of clusters according to the intrinsic complexity present in a particular dataset. A lower BIC score is preferred in selecting the optimal cluster number.

$$BIC = k \cdot \ln(n) - 2\ln\,(L) \tag{4}$$

As shown in Eq. (4), L is the maximized value of the likelihood function of the model M, i.e. L=$p(x|\theta,M)$ with θ as the parameter value and x as the observed data. n is the number of data points in x or the sample size. k is the number of parameters of θ. Both methods are used to determine the best cluster number. Based on our experiment, the optimal cluster numbers from both methods are usually close to each other. Thus, we propose to take the average best cluster number of both as the final cluster number.

2.4 Task Clustering

We hypothesize that tasks that share similar distribution patterns across latent factors share the same underlying cognitive process. Therefore, we cluster tasks together based on the constructed task representations. In clustering, we use a cosine similarity measure to define the similarity of any given two tasks. As a result of the clustering, similar tasks would form a cluster and different tasks would be distinguishable from other clusters. We hypothesize that the data follows Gaussian distribution. As experiments in the earlier work demonstrated hypothesizing the same data distribution for both task representation construction and clustering achieves a more accurate and reliable result, so we follow the same assumption in clustering. We use a Gaussian Mixture Model that first builds a surrogate model based on the same data distribution and then clusters data based on certain distance measure. Multivariate Gaussian distribution is formulated as Eq. (5),

$$G(X|\mu_d,\varepsilon_d) = \frac{1}{\sqrt{(2\pi)|\varepsilon_d|}}\exp\,(-\tfrac{1}{2}(X-\mu_d)^T{\varepsilon_d}^{-1}(X-\mu_d)) \tag{5}$$

where μ is a d task representation dimension) dimensional vector and ε is the d by d covariance matrix. Gaussian mixture is a mix of several Gaussian distributions as shown in Eq. (6),

$$p(X) = \sum_{k=1}^{K}\pi_k G(X|\mu^k,\varepsilon^k) \tag{6}$$

where π is the mixing coefficient for k-th distribution. A KC is discovered if the cluster contains a minimal number of tasks. The minimal number is pre-defined based on the knowledge type and complexity by the course designer. If there is no KC discovered, the relevant tasks are classified as diagnosed tasks that do not share the same task complexity with other tasks.

2.5 Multidimensional Measure

Understanding the cognitive process of learning is the key for discovering knowledge components. Due to the covert nature of cognitive process, we propose a multidimensional measure to quantify the validity of the discovered cognitive models. The measure includes three dimensions: 1. reconstruction loss, loss variance, and item wise robustness to quantify the model robustness and reliability; 2. cognitive model simulation to measure accuracy; 3. KC orthogonality to measure the knowledge decomposition degree.

2.5.1 Reconstruction loss and loss variance

Reconstruction loss and loss variance measure the reliability and robustness of an algorithm. In other words, it is a measure of the sensitivity of the algorithm towards systematic randomness and hyperparameters. Due to the covert nature of learning, we rely on human judgement to evaluate the results. Thus, we expect the algorithm to be robust and reliable and the result of the algorithm to be reproducible, so that the human evaluation effort can be saved. Reconstruction loss refers to the loss function value during an iterative optimization process. A lower reconstruction loss in both the training and testing phases means a better model fit. Loss variance indicates the robustness of the solution to the systematic randomness and hyperparameter settings. A lower loss variance indicates a more robust solution. However, low loss and loss variance are usually continuous values and do not guarantee a robust discrete clustering result. We define item wise robustness measures to quantify the variability of the discrete clustering result. If tasks are constantly clustered to the same clusters, given the systematic randomness, we claim the solution is robust and reliable.

2.5.2 Cognitive Model Simulation

We hypothesize that if a cognitive model can better simulate the cognitive process in terms of predicting the learning behavior more accurately, the model is a better approximation of the true cognitive model of learning compared to a model that has less accuracy. We leverage the attention-based knowledge tracing technique [1] to simulate the knowledge learning process. We use the 'next attempt correctness' criterion (as a common practice) to guide the simulation. The more accurate the model can predict the next attempt behavior, the closer the model is to the true cognitive status. We use Accuracy (ACC) and Area Under the Curve (AUC) as the accuracy measure. Knowledge tracing simulation is conducted for each KC individually. An empirical threshold is designed based on the knowledge type and the instructional design. The empirical threshold is used to decide whether or not the discovery is valid and statistically significant. If the discovered skill model results in a significant improvement in accuracy, we recommend it to the learning designer for knowledge map refinement.

2.5.3 KC Orthogonality

KC orthogonality is about the degree of the knowledge decomposition. In learning design theory, KC is a description of a mental structure or process the learner uses, either alone or in combination with other KCs to accomplish steps in a task or a problem. A course is better structured if there is less overlap among KCs represented by various learning activities. If the knowledge map is well unpacked and the tasks are clearly associated with the KCs, the knowledge gap of a learner is easier to identify and the feedback can be more targeted for that learner. We propose to use Orthogonality measure ‘average task number per KC’ and ‘average KC number per task’ to measure the KC decomposition.

2.6 KC Interpretation

Linguistic interpretation about the course text can support understanding the intention of the assessment design. We leverage NLP techniques to provide keyword and keyword distribution for the formative assessments, including questions, answers, feedback, and hints. We take a well-adopted approach, topic modeling, to extract the key phrases and the distribution of the key phrases for the given text. We apply Latent Dirichlet Allocation (LDA) [12] as the topic modeling solution. LDA builds a word per topic model and a topic per document model with Dirichlet distributions as the assumption of the word distribution across the document. The linguistic interpretation is provided to designers along with the cognitive models of learning to guide course improvement.

2.7 Simulation-based Best KC Analysis

After clustering, there are tasks diagnosed as individually problematic ones that do not share a cognitive process with others. We propose to use knowledge tracing simulation to provide the best KC analysis for the diagnosed tasks. When a task is diagnosed as an outlier, it is either a practice opportunity for other KCs within the course, a multi-skill task that requires other KCs to solve, or not relevant to the course. We leverage the same knowledge tracing technique from Section 2.5.2 to evaluate the fitness of the task to the KC. For each diagnosed task, we simulate knowledge tracing with and without this task for each KC and observe the accuracy change. If the accuracy of a KC improves or remains similar after adding in this task, we claim that the task shares the same underlying cognitive process and it is a good practice opportunity for that KC. Similarly, if the accuracy of a KC decreases (beyond a thresholding that is defined by designers considering the knowledge type and complexity) after adding in the task, we claim that the task does not share the same KC with the rest of the tasks. If the diagnosed task fits another KC, we report that it is better to associate the task with that KC(s). If the task does not fit any of the KCs, we conclude that the task is either irrelevant to the course or the task is a multi-skill task. In either case, we report that investigation is needed for that task. The report is provided to learning designers as insights for constructing or refining the knowledge map.

3 EXPERIMENTS AND CASE STUDY

There are two probable application scenarios of discovering cognitive models of learning. If designers are able to deliver an initial knowledge map through CTA activities or other design practices, the application is to refine the knowledge map with learning data. If designers are not able to deliver an initial map before learners engage with the course, the application is to cluster learning behaviors into groups and define KCs. We demonstrate how the proposed solution would work in both application scenarios. The courses in the applications are deployed on an openly navigated online learning platform in a workforce learning environment.

3.1 Two Applications and Datasets

We use two courses for the two application scenarios to demonstrate the idea. **Application One** is about fine-tuning an existing knowledge map. Course One is studied in this application. Course One is designed for developing interviewing skills as an interviewer at the company. The course was developed with multiple rounds of experts think-aloud protocols. Each KC is supported with a number of practice opportunities and the associated learning outcomes well defined. One of the KCs (KC1) is dedicated for practicing STAR interviewing skill and it is designed with different difficulty levels of tasks. Another KC (KC2) is designed for a broader learning outcome, including behavior observing and question asking. Two other KCs (KC3, KC4) have similar learning outcomes - collecting evidence for required skills. The goal is to dive into these four KCs and observe if the current knowledge structure can be refined by either 1. splitting a KC into sub KCs if there is more than one cognitive process of learning; 2. merging KCs if the cognitive processes are similar; or 3. diagnosing problematic tasks that do not share the same cognitive process with others within the same KC. We use the following three cases to demonstrate how it works.

1. Case One – resulting in splitting into sub KCs, KC1 has 144k interaction records from 4480 learners on 16 tasks.
2. Case Two – resulting in diagnosing problematic tasks, KC2 has 31k interaction records from 4480 learners on10 tasks.
3. Case Three - resulting in merging into one KC, KC3 has 119k interaction records from 4480 learners on 5 tasks and KC4 has 108k interaction records from 4480 learners on 6 tasks.

Application Two is for demonstrating how to discover cognitive models of learning without human knowledge upfront. Course Two is used in this application. It is designed for developing sales skills related to AWS Services. One of the learning outcomes is to be able to drive effective communication with customers. In order to achieve this learning outcome, there are 72 tasks designed and associated with 6 KCs, initially. The KCs are structured by content type and use case type, such as ‘Cloud value’, ‘AWS benefits’, ‘Consulting’, ‘Responding to questions’. We question whether such KC structure is aligned with cognitive models as learners learn. As we apply the cognitive modeling, we don’t use the initial map in the modeling process. We aim to discover the models purely from the interaction data and observe the real factors that structure the way learners learn this particular subject. The data set we used contains 74k attempts across 1204 learners on 72 tasks.

3.2 Experiment Setting

In order to best demonstrate each module of the framework, we designed experiments as follows. For task representation learning, we demonstrate the robustness and reliability of VAE and the multidimensional measurement with both applications. For task clustering, we compare GMM with K-means to test the data distribution hypothesis and its impact on clustering result with Application One. For best cluster number selection, we use a manual process in Application One to illustrate the general idea. We then apply BIC/Silhouette methods in Application Two to automate the best cluster number selection process. We demonstrate KC interpretation with Application One. We showcase best KC analysis with simulation technique with Application One.

3.3 Task Representation Construction and Clustering

First, we transform learner-task interaction data into task difficulty matrix. If a particular task is not attempted by a learner, we assume the learner has achieved proficiency on the skill and thus skipped the task. Therefore, we fill the missing values with ‘zero’ task difficulty. We construct task representation with both VAE and NMF approaches. During that, we test different hyperparameter settings, such as cluster number and embedding dimension. For cluster number selection, we use BIC and Silhouette as automated solution. For embedding size, we test if the algorithm is insensitive to the embedding size. After that, we conduct KC clustering with both GMM and K-means.

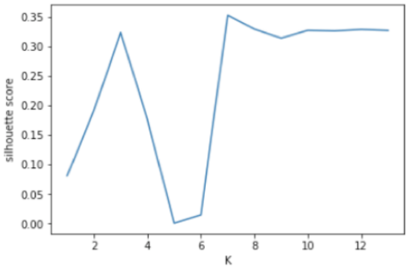


Figure 2. BIC score

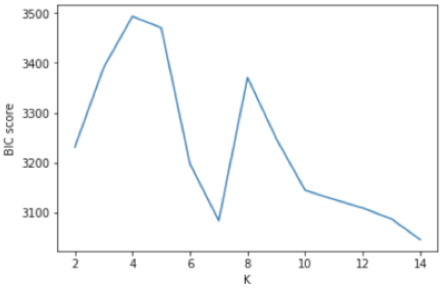


Figure 3. Silhouette score

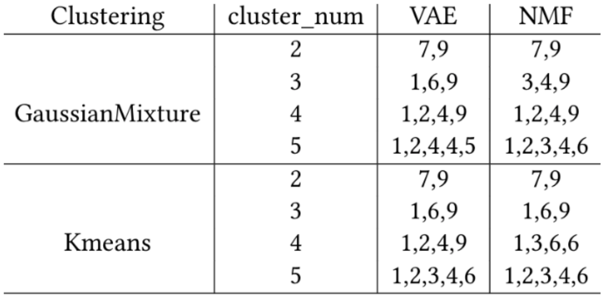
Best Cluster Number. **Application One**. We first demonstrate the impact of best cluster number to clustering result by enumerating the possibilities (manual evaluation) with Application One. The manual process is as follows. We hypothesized that at least three tasks (the number is defined based on the knowledge type and instructional needs by designers) are required to form a KC, the largest experimental cluster number is 16/3, which is 5. Thus, we enumerate the cluster number from 2 to 5 for both VAE and NMF. Experimental result on cluster number is shown in Table 1. As the cluster number is set as 2, there are two skills identified from both approaches (7 tasks in one group and 9 tasks in the other group). When the cluster number increases, individual tasks are identified as clusters from both approaches. Considering the hypothesis ‘three task at least for a KC’, cluster number 2 is selected as the solution. **Application Two**. We automate best KC selection with BIC and Silhouette methods in Application Two. A lower BIC score indicates less overfitting and a higher Silhouette indicates a better grouping (items in the same group are more similar to each other, and items in different groups are more distinguishable from one another). As shown in Fig 2 and 3, cluster number 7 is the best solution that has the lowest BIC score and highest Silhouette score. Thus, the average cluster number of both methods, 7, is selected as the best KC number. **Embedding dimension selection**. Initial experiments show there is no significant difference in the clustering result when the embedding size increases from 30 to 300. However, the clustering result is sensitive to the embedding size when the size is relatively small (less than 30). That said, the task complexity is contributed by only a few orthogonal latent factors. Thus, we test embedding dimension within the range [2,30]. Unlike the best cluster number selection, we don’t have an automated solution for selecting the embedding size. Thus, we aim to find a solution that is insensitive to the dimension selection, so that the effort in selecting the best embedding size can be saved.

3.4 Measure with Loss, Robustness, KC Orthogonality

We demonstrate 1. loss and loss variance, item wise robustness, and 2. KC orthogonality with two applications. **Reconstruction loss and loss variance**. Less reconstruction loss and loss variance indicate a more robust result. Table 2 shows the reconstruction loss for both VAE and NMF for different KCs with different embedding size settings. As it is shown, both VAE and NMF achieve the same clustering results as cluster number is set as 2. And from expert evaluation, ‘Two’ is the best cluster number. Regarding loss, reconstruction loss of VAE varies slightly from 0.02 to 0.06 (with mean 0.0356 and variance 4.799e-05) across embedding sizes and KCs.

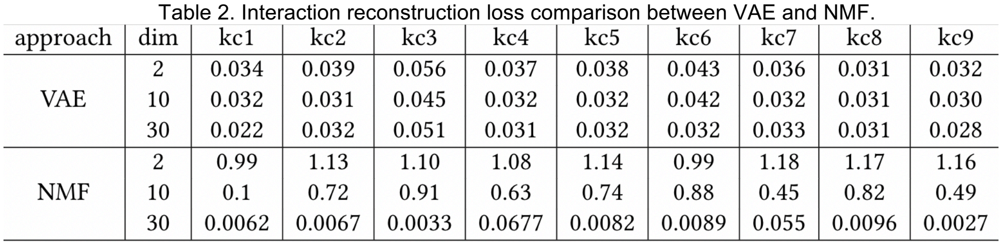
However, for NMF, the reconstruction loss is sensitive to the embedding size. Specifically, with embedding size 2, 10, 30, the average reconstruction loss of NMF is around 1.1, 0.7, 0.018, respectively (with mean 0.5869 and variance 0.2276). In summary, NMF results with a 16 times of reconstruction loss and 4742 times of loss variance in average compared to VAE.

Table 1. Clustering result for VAE and NMF.



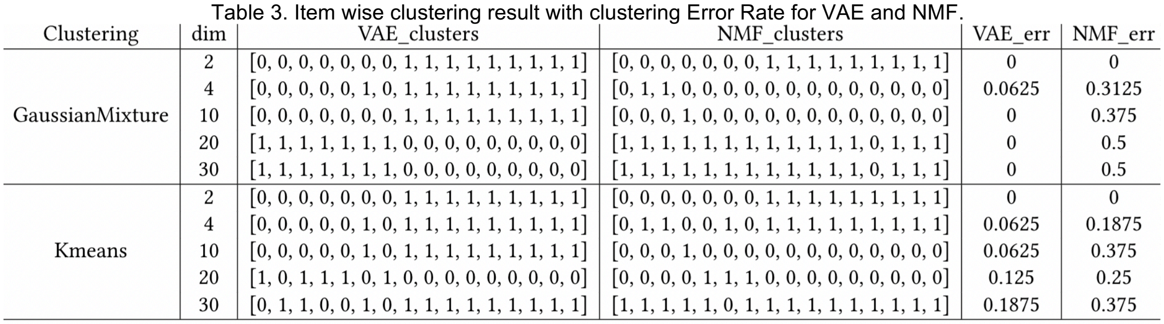
Clustering	cluster_num	VAE	NMF
GaussianMixture	2	7,9	7,9
	3	1,6,9	3,4,9
	4	1,2,4,9	1,2,4,9
	5	1,2,4,4,5	1,2,3,4,6
Kmeans	2	7,9	7,9
	3	1,6,9	1,6,9
	4	1,2,4,9	1,3,6,6
	5	1,2,3,4,6	1,2,3,4,6

Table 2. Interaction reconstruction loss comparison between VAE and NMF.



approach	dim	kc1	kc2	kc3	kc4	kc5	kc6	kc7	kc8	kc9
VAE	2	0.034	0.039	0.056	0.037	0.038	0.043	0.036	0.031	0.032
	10	0.032	0.031	0.045	0.032	0.032	0.042	0.032	0.031	0.030
	30	0.022	0.032	0.051	0.031	0.032	0.032	0.033	0.031	0.028
NMF	2	0.99	1.13	1.10	1.08	1.14	0.99	1.18	1.17	1.16
	10	0.1	0.72	0.91	0.63	0.74	0.88	0.45	0.82	0.49
	30	0.0062	0.0067	0.0033	0.0677	0.0082	0.0089	0.055	0.0096	0.0027

Table 3. Item wise clustering result with clustering Error Rate for VAE and NMF.



Clustering	dim	VAE_clusters	NMF_clusters	VAE_err	NMF_err
GaussianMixture	2	[0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1]	[0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1]	0	0
	4	[0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1]	[0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	0.0625	0.3125
	10	[0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1]	[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	0	0.375
	20	[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0]	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1]	0	0.5
	30	[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0]	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1]	0	0.5
Kmeans	2	[0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1]	[0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1]	0	0
	4	[0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1]	[0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1]	0.0625	0.1875
	10	[0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1]	[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	0.0625	0.375
	20	[1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0]	0.125	0.25
	30	[0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1]	[1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1]	0.1875	0.375

Simply put, VAE provides a more stable reconstruction loss with which human efforts can be saved in evaluating and selecting the best result.

Item wise robustness. Low reconstruction loss and loss variance does not guarantee a stable discrete clustering result. We use item wise robustness as part of the measure. Table 3 shows the clustering results along with clustering error rate with embedding size ranging from 2 to 30. The result from algorithm GMM with embedding size 2 is selected as the benchmark by experts. With that, we compute error rate for other results for comparison. As it is shown, NMF is sensitive to both embedding size and clustering algorithms. Specifically, as embedding size is set as 2, the clustering result achieves the best error rate with both clustering approaches

(0 out 16 is wrong). However, as embedding size increases, error rate increases to 0.5 (8 out 16 is wrong) with GMM and 0.375 with K-means. For VAE, results are much more stable. With GMM as the clustering algorithm, the error rate keeps flat as 0 as embedding dimension increases (a small exception with embedding size with 4). With K-means, the error rate increases slightly from 0 to 0.1875 (3 out 16 is wrong). In average, VAE achieves error rate 0.05 and 0.004 error rate variance. Meanwhile, NMF achieves 0.2875 error rate (12 times of VAE) and 0.025 error rate variance (12.5 times of VAE). To conclude, VAE achieves a stable performance from item wise clustering result while the performance of NMF is heavily dependent on the embedding size and algorithm settings. The results from VAE also show that hypothesizing the same data distribution between task clustering and task representation construction can result in a more robust and reliable outcome. To be more specific, we hypothesize the learning data follows the Multivariate Gaussian distribution in both VAE and GMM. GMM is better than K-means because K-means does not model the data distribution, it uses distance-based measures and it results in a less stable performance.

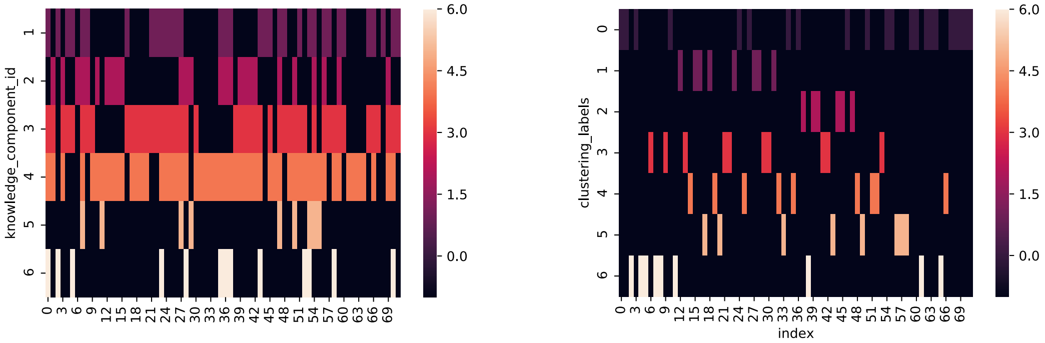


Figure 4. Heatmap by tasks. left: heatmap-initial; right: heatmap-refined

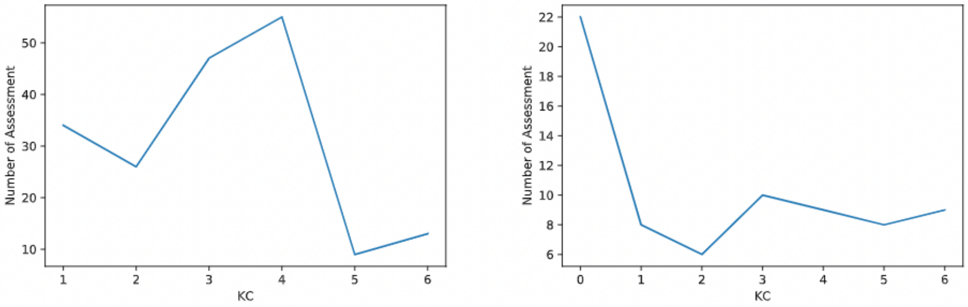


Figure 5. Task count by KC number. left: Orthogonality-initial; right: Orthogonality-refined

KC Orthogonality. KC Orthogonality is a measure of decomposition degree of the knowledge. Fig 4 illustrates the task-KC mapping heatmap for the initial and discovered maps. The x axis is the task index. The y axis is the KC index. Fig 5 shows the task count for each KC for the initial and the discovered maps. The x axis is the KC index. The y axis is the task count for that KC. As shown in the sub fig a in Fig 4 and 5, the task count is unevenly distributed across KCs in the initial map, where some KCs are designed with less than 10 tasks (KC5) while some are designed with more than 50 tasks (KC4). On average, each task is associated with 2.55 KCs. That said, the underlying KCs are intercorrelated with one another from the design. After applying the proposed approach, as shown in the second sub fig in Fig 4 and 5, the discovered knowledge map has a more balanced and orthogonal task-KC relation, where each KC represents one kind of cognitive process that is shared within the associated tasks and is distinguishable from other KCs. If tasks are designed with an orthogonal knowledge map, the knowledge gap can be easily identified as a learner fails on a task. With the knowledge gap identified, targeted feedback can be surfaced to the learner for effective learning.

3.5 Measure with Cognitive Model Simulation

We leverage knowledge tracing simulation to evaluate the skill model accuracy. The accuracy is referred to as the fitness between the observed learning behavior and the true covert cognitive process. Better accuracy in predicting the next behavior indicates a better approximation of the cognitive process. We use Case One and Case Three in Application One to demonstrate how knowledge tracing helps determine whether a better cognitive model is discovered. For Case One, we observe a significant accuracy improvement after splitting the initial KC into sub KCs. For Case Three, we observe a decrease in accuracy after merging two KCs that seem alike. As a recommendation to designers, we suggest splitting KC1 into sub KCs and leaving KC3/KC4 as they are. **Case One**. we conduct knowledge tracing simulation experiments with the settings below,

1. Sub KC-a Scenario 1. Train the KT model with all tasks in the initial KC1. Test with tasks in sub KC-a.
2. Sub KC-a Scenario 2. Train the KT model with tasks in sub KC-a. Test with tasks in sub KC-a.
3. Sub KC-b Scenario 1. Train the KT model with all tasks in the initial KC1. Test with tasks in sub KC-b.
4. Sub KC-b Scenario 2. Train the KT model with tasks in sub KC-b. Test with tasks in sub KC-b.

As shown in Fig 6,the x axis is the iteration number, the y axis is the model fit accuracy score, either ACC or AUC score. The blue line represents the simulation with the initial skill model, the yellow line represents the simulation with the refined skill model. As a result, for both sub KC-a and KC-b, Scenario 2 (yellow line) outperforms Scenario 1 (blueline) significantly on both ACC and AUC measurement (Sub KC-a achieves 2.2% AUC increase, 3.6% ACC increase. Sub KC-b achieves 5.6% AUC increase, 4.8% ACC increase). The accuracy improvement from simulation indicates that the tasks designed in sub KC-a does not share the same cognitive process with the task designed in sub KC-b. That said, a knowledge map with sub KC-a and KC-b is a better structure for the underlying knowledge compared to having tasks in one single KC. **Case Three**. The two KCs are designed for a same learning outcome that is to collect evidences for the required skills. We also use simulation to test how close the cognitive models are from one another. The experiment is set as below,

1. KC3 Scenario 1.Train the KT model with the tasks in KC3. Test with tasks in KC3.
2. KC3 Scenario 2. Train the KT model with tasks in both KCs. Test with tasks in KC3.
3. KC4 Scenario 1.Train the KT model with the tasks in KC4. Test with tasks in KC4.
4. KC4 Scenario 2. Train the KT model with tasks in both KCs. Test with tasks in KC4.

The simulation results are shown in Fig 7, where x axis is the iteration number, y axis is the model fit accuracy score, either ACC or AUC score. Blue line refers to the simulation with the initial separate skill models, the yellow line refers to the simulation with the merged skill model. As a result, for both sub KC3 and KC4, Scenario 1 (blue line) outperforms Scenario 2 (yellow) significantly on both ACC and AUC measurement (the merged skill model results with a 10% ROC and 6% ACC decrease compared to KC3, 10% ROC and 7% ACC decrease compared to KC4). The accuracy decrease from simulation indicates that the tasks designed in the first KC does not share the same cognitive process with the task designed in the second KC. That said, even the two KCs seem alike in terms of their descriptive learning outcomes, the cognitive model of learning is different from one another. Thus, we recommend to keep these two KCs separate as they are initially designed.

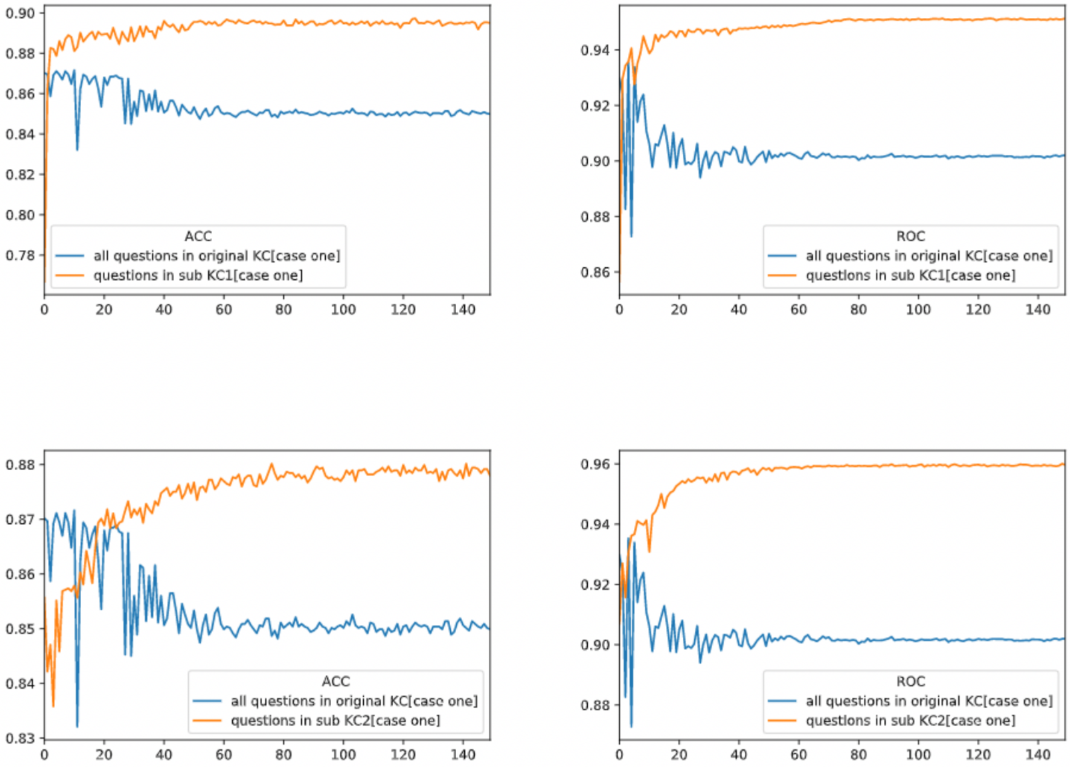


Figure 6. KT simulation for KC-a and KC-b.
upper left: ACC for sub KC-a vs. initial; upper right: AUC for sub KC-a vs. initial;
lower left: ACC for sub KC-b vs. initial; lower right: AUC for sub KC-b vs. initial.

3.6 KC Interpretation

We demonstrate the KC interpretation with Case One in Application One. Keywords and their distribution is as follows,
Keywords for sub KC-a:
[0.034*situation + 0.032*task + 0.031*action + 0.023*result + 0.023*detail + 0.021*understand + 0.019*good+ 0.019*question + 0.018*peel + 0.018*onion]
Keywords for sub KC-b:
[0.025*follow-up + 0.019*dive + 0.019*challenging + 0.018*deep + 0.016*design + 0.016*result + 0.016*manager + 0.015*get + 0.013*provide + 0.013*response]

In sub KC-a, keywords such as ‘situation’, ‘task’, ‘action’, ‘result’, takes up 12% of total text stems. The keywords indicate the tasks are intended for practicing ‘STAR’ technique in interview question asking. In sub KC-b, keywords such as ‘follow-up’, ‘dive’, ‘challenging’, ‘deep’, ‘design’ distribute evenly. The keywords indicate that the tasks are intended for practicing follow-up or deep diving questions. In other words, in sub KC-a, learners are expected to practice asking behavior questions using STAR technique. In sub KC-b, learners are expected to ask follow-up questions to dive deep into the details of the behavior question. In summary, from analyzing learning behavior data, we observe there are two sub KCs in KC1. From text stems, we also observe there are two different skills required - behavior question asking using STAR and follow-up question asking with diving deep - to perform those tasks. The clustering result is 100% aligned with the text stem indicators. Thus, we provide interpretation from linguistic perspective to designers along with the learning data clustering result for knowledge map construction and refinement.

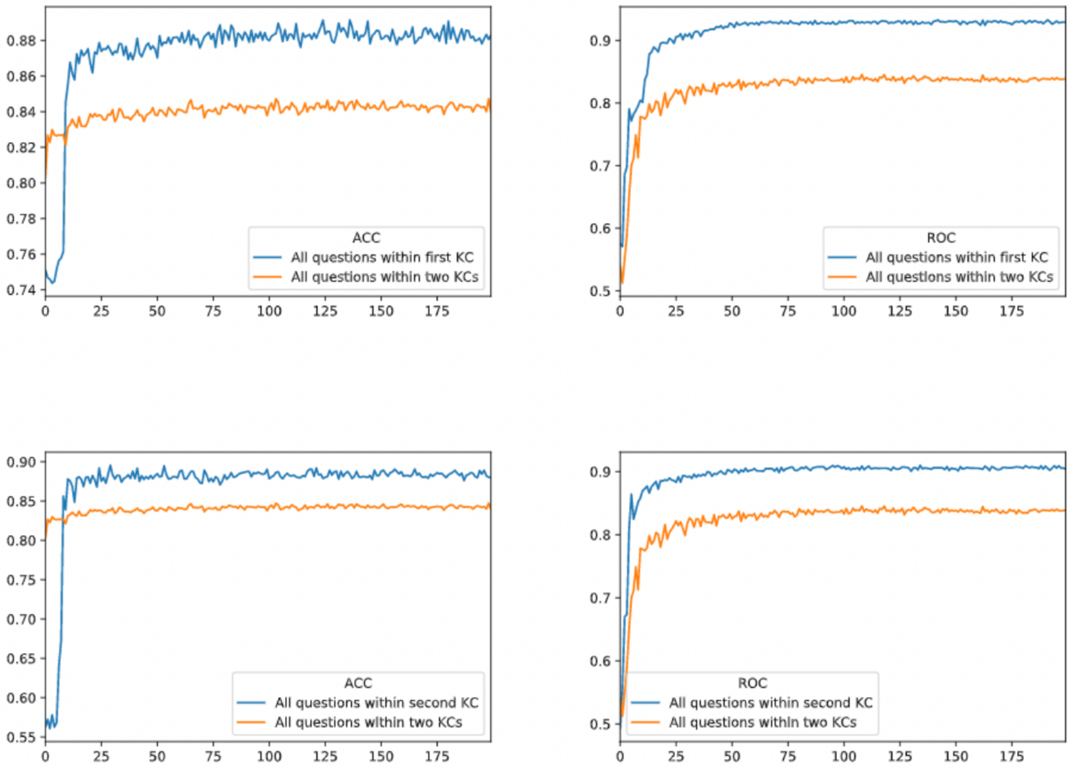


Figure 7. KT simulation for KC3 and KC4.

upper left: ACC for KC3 vs. merged; upper right: AUC for KC3 vs. merged;

lower left: ACC for KC4 vs. merged; lower right: AUC for KC4 vs. merged.

3.7 Simulation Based Best KC Analysis

For the diagnosed tasks, we conduct best KC analysis to identify what is the best KC for the task. We demonstrate the analysis process with Case Two. We run knowledge tracing simulation for initial and refined knowledge map. In Case Two, there are two diagnosed tasks that do not share the same underlying cognitive process. Table 4 shows the best KC analysis for one of the problematic tasks. In the initial knowledge map, the task is associated with five KCs, where the learning designer designed it as a learning opportunity related to or requiring all these five KCs. After conducting cognitive modeling discovery for each KC, the task is diagnosed as a problematic task for KC-h, KC-i, and KC-m (the simulation accuracy improves after removing the diagnosed task). It means the task does not share the same cognitive process with tasks in those three KCs. In KT simulation, ACC and AUC of KC-j and KC-k remains similar (within a 2% pre-defined threshold designed by the designer) with or without the task. It indicates the task shares a closer cognitive process with tasks in KC-j and KC-k. Therefore, we recommend to remove this task from KC-h, KC-i, and KC-m and keep the task in KC-j and KC-k.

Table 4. Best KC Analysis

Setting	metric	KC-h	KC-i	KC-j	KC-k	KC-m
refined	AUC	0.948	0.919	0.940	0.909	0.776
initial	AUC	0.926	0.909	0.934	0.903	0.678
refined	ACC	0.896	0.883	0.870	0.802	0.796
initial	ACC	0.881	0.863	0.866	0.796	0.769
discovery	Best KC (Y/N)	0	0	1	1	0

3.8 Human Evaluation

We conducted human evaluation of the discovered cognitive models for both applications. Experts agree and suggest to implement these changes into the revised cognitive model for each application. **Application One**. We provide human evaluation for the mentioned two cases. In Case One, the KC is initially designed with 16 tasks. All of the tasks are designed to evaluate the learner's ability to practice interviewing skills. Specifically, 7 of the 16 tasks ask learners to identify which question is appropriate to ask in an interview setting. These questions focus more on recognizing the appropriate choice among a set of question options. The other 9 questions are more application focused, in that they require the learner to identify the underlying knowledge concept and apply it to the interviewing scenario to solve the task. In Case Two, the KC contains 10 tasks. These tasks are designed to evaluate the learner's ability to describe the role of the interviewer moderator in the interview process. In 2 out of the 10 tasks, the focus is broad conceptual understanding of this role. The remaining 8 tasks require learners to apply the role of the moderator to specific interview scenarios. These findings indicate the two broad conceptual tasks don't share the same KC as the remaining 8 tasks, and should be removed from the current KC. **Application Two**. A new knowledge map is discovered without referring to the initial map in Application Two. In human evaluation, experts agreed to 80.1% of the knowledge structure and suggested to refine the initial map accordingly. It indicates that learning data provides a significant amount of value in structuring the knowledge and discovering how humans develop the knowledge. As we dive deep, the initial knowledge map is structured mostly based on content attributes and use cases, such as AWS benefits, Cloud value, Consulting questions, and Responses to customer use cases. The new knowledge map is structured mostly from the required skill perspective, such as the ability to Define concepts, Manage conversations with customers, and Introduce strategies. That said, content type and application scenarios may be good indicators in categorizing learning content, but are not accurately reflecting the underlying skills learners need to perform the tasks. For example, two applications (consulting and responding to questions) may require the same skill (conversation skill). Considering the required skills is essential in designing a knowledge map and we found that learning data can help discover the underlying required skills.

4 CONCLUSION AND FUTURE WORK

In this work, we propose a novel framework to discover cognitive models with learning data. The framework includes VAE for task representation construction, BIC and Silhouette methods for best KC number selection, GMM for task clustering, a multidimensional measure including robustness measure with loss related statistics, KC orthogonality measure, and accuracy measure with knowledge tracing techniques. We also provide KC interpretation from linguistic perspective, and best KC analysis for diagnosed tasks. We use two applications implemented on an open navigated learning system to demonstrate how the framework discovers the cognitive models and provides insights for knowledge map construction and refinement. We also evaluate the discovered cognitive models with human experts (80.1% alignment with expert knowledge). In future work, we'd like to continue to refine the proposed framework by observing the learning efficacy of the refined learning experience that were guided by the discovered cognitive models of learning. We also plan to extend the work to other kinds of knowledge learning (e.g., procedural knowledge) and cognitive activities (e.g., application, evaluation) with different learning designs.

REFERENCES

[1] Shreyansh Bhatt, Jinjin Zhao, Candace Thille, Dawn Zimmaro, and Neelesh Gattani. 2020. A novel approach for knowledge state repre sentation and prediction. In Proceedings of the Seventh ACM Conference on Learning@ Scale. 353–356.

[2] Hao Cen, Kenneth Koedinger, and Brian Junker. 2006. Learning factors analysis–a general method for cognitive model evaluation and im- provement. In International Conference on Intelligent Tutoring Systems. Springer, 164–175.

[3] Richard Edward Clark and Jan Elen. 2006. Handling complexity in learning environments: Theory and research. Elsevier.

[4] Michel C Desmarais. 2012. Mapping question items to skills with non- negative matrix factorization. ACM SIGKDD Explorations Newsletter 13, 2 (2012), 30–36.

[5] Carl Doersch. 2016. Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908 (2016).

[6] Kenneth R Koedinger, Albert T Corbett, and Charles Perfetti. 2012. The Knowledge-Learning-Instruction framework: Bridging the science-practice chasm to enhance robust student learning. Cognitive science36, 5 (2012), 757–798.

[7] Mark A Kramer. 1991. Nonlinear principal component analysis using auto associative neural networks. AIChE journal 37, 2 (1991), 233–243.

[8] Daniel D Lee and H Sebastian Seung. 2001. Algorithms for non- negative matrix factorization. In Advances in neural information processing systems. 556–562.

[9] Kart-Leong Lim, Xudong Jiang, and Chenyu Yi. 2020. Deep Clustering With Variational Autoencoder. IEEE Signal Processing Letters 27 (2020), 231–235.

[10] Xihui Lin and Paul C Boutros. 2020. Optimization and expansion of non-negative matrix factorization. BMC bioinformatics 21, 1 (2020), 1–10.

[11] Noboru Matsuda, Tadanobu Furukawa, Norman Bier, and Christos Faloutsos. 2015. Machine Beats Experts: Automatic Discovery of Skill Models for Data-Driven Online Course Refinement. International Educational Data Mining Society (2015).

[12] Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. 2000. Inference of population structure using multilocus genotype data. Genetics 155, 2 (2000), 945–959.

[13] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics 20 (1987), 53–65.

[14] Jan Maarten Schraagen, Susan F Chipman, and Valerie L Shalin.2000. Cognitive task analysis. Psychology Press.

[15] Gideon Schwarzetal. 1978. Estimating the dimension of a model. The annals of statistics 6, 2 (1978), 461–464.

[16] Lijun Zhang, Zhengguang Chen, Miao Zheng, and Xiaofei He. 2011. Robust non-negative matrix factorization. Frontiers of Electrical and Electronic Engineering in China 6, 2 (2011), 192–200.

[17] Zhao, Jinjin, et al. "A Novel Framework for Discovering Cognitive Models of Learning." Proceedings of the Eighth ACM Conference on Learning@ Scale. 2021.