A Novel Framework for Discovering Cognitive Models of Learning

Jinjin Zhao
Amazon.com
Seattle, USA
jinjzhao@amazon.com

Candace Thille
Amazon.com
Seattle, USA
cthille@amazon.com

Neelesh Gattani
Amazon.com
Seattle, USA
neeleshg@amazon.com

Dawn Zimmaro
Amazon.com
Austin, USA
dzimmaro@amazon.com

ABSTRACT
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) [10] 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 the courses, reveal how humans think about and develop knowledge. We propose a novel framework that uses learning data to discover and refine cognitive models of learning. The framework includes a Variational Autoencoder (VAE) module and a Gaussian Mixture Model (GMM) module. We provide one case study in a corporate setting to demonstrate the effectiveness of the proposed framework compared to other approaches.

Author Keywords
Cognitive Modeling; Human-Computer Interaction; Knowledge Tracing Simulation; Human-in-the-loop

CCS Concepts
•Mathematics of computing → Bayesian networks; •Human-centered computing → User models; •Applied computing → Interactive learning environments;

L@S '21, June 22–25, 2021, Virtual Event, Germany.

ACM ISBN 978-1-4503-8215-1/21/06.
http://dx.doi.org/10.1145/3430895.3460156

INTRODUCTION AND RELATED WORK
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. There is compelling evidence that experts are not fully aware of about 70% [2] of their own decisions and how they execute tasks and are unable to explain tasks fully during the interviews. 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. 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 can also provide a representation of the cognitive process of learning. If collected at the right grain size, the sequential behavior data provides step-wise information about the process in which learners develop knowledge and skills. Studies have shown the benefits of mining learning behavior data, especially the learner-assessment attempt correctness, to discover the cognitive process of learning. In Learning Factors Assessment (LFA) [1], the skill model is created by fitting a statistical model with human defined learning factors. One shortcomings of LFA is that 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 [9] uses an automated solution to discover the skill model with a Non-negative Matrix Factorization (NMF) technique [6]. Compared to LFA, NMF-based solutions could help reduce the learning factor engineering effort; however, NMF-based techniques are sensitive to the data sparsity and quality [11]. The major drawback of NMF lies in its slow convergence, lack of well-validated methodology for hyper-parameter selection, and low reliability of the solution [8]. In cognitive modeling, due to the unobservable nature of the

cognitive process, we rely on human knowledge to evaluate the accuracy of the discovered skill models and to 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 selecting interaction data mining techniques, autoencoder [5] and its variants have shown its effectiveness in constructing a dense representation of a high dimensional and sparse dataset. The representation is robust to its systematic randomness and hyperparameter setting. However, the latent space of an autoencoder is constructed without modeling the underlying data distribution and generation process. When it comes to clustering the dense representations, K-means or GMM uses a different measure in quantifying the distance between data points. The misalignment between constructing and clustering the representations results in lower accuracy [7]. Unlike conventional autoencoders that aim to learn a predictor given the observation, Variational Autoencoder (VAE) [3], a probabilistic graphical model, assumes that there is a certain underlying probability distribution (such as Gaussian distribution) from the observations. VAE attempts to find the optimal parameters for a given distribution and use that distribution to generate data that is close to the observations. In short, VAE attempts to simulate how the observation data is generated with a certain data distribution hypothesis. VAE offers an opportunity to align on the goals of representation construction and clustering.

In this work, we propose to applying VAE to construct the assessment (interchangeable with task) representations. We hypothesize that the learner-assessment data follows multivariate Gaussian distribution (as a common practice). We propose to use Gaussian Mixture Model (GMM) over Kmeans [9] for clustering. We also provide a case study to demonstrate the effectiveness of the proposed approach.

APPROACH

Framework Overview

This section presents the overall framework of the proposed cognitive model discovery solution. As shown in Figure 1, learners interact with courses through an online learning platform. The learning data are collected and stored in a database. We apply behavior factorization with VAE to construct the task representation. We then apply GMM to discover the underlying Knowledge components (KCs) or the task clusters. If the tasks share the same cognitive model, a cluster (a KC) will be formed. If a task does not share any cognitive model with other tasks, it will 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.

Task Representation Construction

Learning behavior is a representation of the cognitive process during learning. Task attempts and correctness are critical pieces of evidence that indicate the learners current knowledge state and the cognitive process that transits from one knowledge state to another. We hypothesize that the tasks designed for a same KC share the same complexity [4]. The difficulty

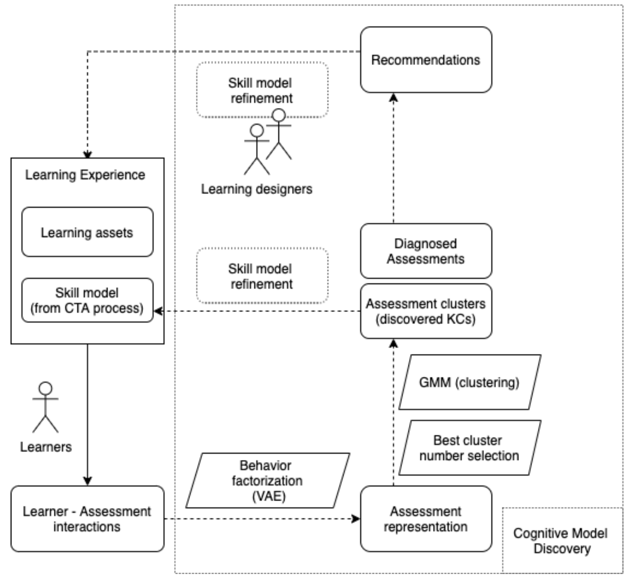

Figure 1. Cognitive model discovery framework. Learning behavior data is collected and anonymously analyzed to discover the underlying knowledge components from the formative tasks. Insights, discovered cognitive models of learning or diagnosed problematic tasks, are provided to designers for refining the knowledge map design.

learners have in applying a KC to solve a task indicates the task complexity. Since a learner can make multiple attempts on a single task to demonstrate one's proficiency of that underlying skill, the sequential learning data indicates the task difficulty. The more attempts a learner needs to successfully solve a task, the more difficult the task is to that learner. We define task difficulty as $1-1/d$, where d is the total attempt number until the first successful attempt [9]. We extract task difficulty at individual level from the learning data.

After transforming interaction data into a learner-task difficulty matrix, we use Variational Autoencoder (VAE) technique to model and unpack the task difficulty by hypothesizing 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 optimized from task difficulty data to construct the task representation. We hypothesize the surrogate model is a generative model and the data generated from it follow a Gaussian distribution (as a common practice). On the other hand, VAE technique assumes that the data follow certain distribution and are generated by some latent surrogate model. Thus, we use VAE as the solution to the task representation construction. The latent generative model is built by minimizing the Kullback-Leibler divergence between the surrogate generative model and the observations.

Task Clustering

We hypothesize 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 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 among clusters. As mentioned in task representation, we hypothesize that the data follow a Gaussian distribution. As experiments demonstrate, hypothesizing the same data distribution for both task representation construction and clustering achieves a more accurate and reliable result, we propose to follow the same assumption in clustering. We propose to use Gaussian Mixture Model that first builds a surrogate model based on the same data distribution and then cluster data based on certain distance measure.

Multivariate Gaussian distribution is formulated as Eq 1,

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

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 2,

$$p(X) = \sum_{k=1}^{K} \pi_k G(X|\mu^k, \varepsilon^k) \tag{2}$$

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 or KCs.

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 1. Clustering result for VAE and NMF with different cluster numbers and clustering algorithms. Two is the best cluster number evaluated by experts. Both VAE and NMF achieve the same clustering results as cluster number is set as 2.

EXPERIMENTS AND CASE STUDY

We demonstrate the proposed cognitive model discovery framework with one application that is implemented on an open navigation online learning platform. The course is designed for developing interviewing skills as an interviewer at a company. One of the KCs is dedicated for practicing STAR interviewing skills with various difficulty-level tasks. The KC (named as KC1) has 144,000 interaction records from 4480 learners on 16 tasks.

Task Representation Construction and Clustering

First, we transform learner-task interaction data into a task difficulty matrix. If a particular task is not attempted by a learner, we assume the learner has 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. After that, we conduct KC clustering with both GMM and Kmeans.

Best cluster number selection

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 cluster 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 size is relatively small (less than 30). It indicates the task complexity is contributed by only a few orthogonal latent factors. Thus, we test embedding dimension with a range [2,30]. In order to save human evaluation effort, we aim to figure out a solution that is insensitive to the embedding dimension so that the effort in selecting the best embedding size can be saved.

Result evaluation

Table 2, shows the reconstruction loss for both VAE and NMF for different KCs with various embedding size settings is presented. The 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. The reconstruction loss of NMF lessens, as the embedding dimension increases. Specifically, with dimension 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). NMF produces 16 times the reconstruction loss and 4,742 times the loss variance, on average, compared to VAE. In short, VAE with a stable reconstruction loss is a preferred solution that could save human efforts in evaluating and selecting the best result.

Table 3, shows the clustering results along with clustering error rate with embedding size varying from 2 to 30 presented. The result from algorithm GMM and embedding size 2 is selected as the benchmark that is evaluated 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 dimension size is set as 2, the clustering result achieves the best error rate with both clustering approaches (0 out 16 is wrong). However, as dimension size increases, error rate increases to 0.5 (8 out 16 is wrong) with GMM and 0.375 with Kmeans. 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 as the clustering approach, the error rate increases slightly from 0 to 0.1875 (3 out 16 is wrong). On

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 2. Interaction reconstruction loss comparison between VAE and NMF with different embedding dimensions for various KCs. VAE achieves stable reconstruction loss and small loss variance when embedding size varies. In the same context, NMF is sensitive to the embedding size selection.

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, 1]	[0, 0, 0, 0, 0, 0, 0, 1, 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, 1]	[0, 1, 1, 0, 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, 1]	[0, 0, 0, 1, 0, 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, 0]	[1, 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, 0]	[1, 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, 1]	[0, 0, 0, 0, 0, 0, 0, 1, 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, 1]	[0, 1, 1, 0, 0, 1, 0, 1, 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, 1]	[0, 0, 0, 1, 0, 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, 0, 1, 1, 1, 0, 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, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]	0.1875	0.375

Table 3. Item wise clustering result with clustering Error Rate for VAE and NMF with different clustering approaches and embedding sizes. VAE with Gaussian Mixture achieves a robust result with error rate flat to zero regardless the embedding size. NMF is sensitive to embedding size regardless the clustering algorithm. For both VAE, Gaussian Mixture is a better clustering algorithm with a same data distribution hypothesized and modeled.

average, VAE achieves an error rate 0.05 and 0.004 error rate variance. However, 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 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 indicates that an aligned data distribution hypothesis between task clustering and task representation construction achieves a more robust and reliable result. In this case, we hypothesize the learning data follows the Multivariate Gaussian distribution in both VAE and GMM. Since Kmeans does not model data distribution, it achieves less robustness.

CONCLUSIONS 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 and GMM for task clustering. We studied one application and observed higher accuracy and more robustness compared to NMF with K-Means approach. In future work, we'd integrate the discovered cognitive models of learning into instructional design in a corporate setting and test its effectiveness in terms of accelerating human learning.

REFERENCES

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

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

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

[4] 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 science* 36, 5 (2012), 757–798.

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

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

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

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

[9] 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).

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

[11] 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.