Towards discovering problem similarity through deep learning: combining problem features and user behavior.

Dominic Mussack
University of Luxembourg
dominic.mussack@uni.lu

Rory Flemming
University of Minnesota
flemm053@umn.edu

Paul Schrater
University of Minnesota
schrater@umn.edu

Pedro Cardoso-Leite
University of Luxembourg
pedro.cardosoleite@uni.lu

ABSTRACT

Automated learning systems allow educators to scale up their efficacy, while personalized systems retain the ability to customize to the individual student. A core issue in developing such adaptive learning systems is to understand how different items (e.g., math exercises) relate to one another, and to exploit this understanding to predict performance on an item. Data-driven approaches aim to discover latent concepts through embeddings that predict similarity between items, typically using only performance data or item data, but not both. While these embeddings are meant to uncover latent concepts (e.g., associativity in mathematics or chemistry), they are better construed as representing topics that reflect the similarity structure in performance or item features. One major difficulty is that embedded concepts may differ only in presentation and not in substance. For example, when learning about numbers, young children struggle with different representational formats (e.g., finger counts, Hindu-Arabic numeral) despite the underlying concept being the same (e.g., "3"). By incorporating item information that allows structured similarity comparison between an item's content and representational format, we can begin to parse out what aspects lead to behavioral differences. Here we develop a deep learning framework for learning concept embeddings that integrates behavioral and item-features to better factorize embeddings into content and presentation. This allows us to fully represent the complexity of the items space, while still extracting scientifically-useful results from the analysis.

Keywords

education testing, item similarity, deep learning, concept discovery

1. INTRODUCTION

Personalized learning systems use student performance to assess their current knowledge in order to present appropriate questions; understanding item similarity helps in this inference process [6]. When the goal is to predict performance, this similarity measure should most likely correspond to the abilities or skills the items are meant to test (e.g., whether a student knows multiplication or not). Data-driven discovery methods for similarity are needed for larger sets of items [1]. There are many ways to measure similarity [6], however all require that items be represented in the same feature space. This means that one of the most important decisions in computing similarity is the choice of input data, because it determines the meaning of the discovered similarity [7]. Having a rich input feature space which correctly corresponds with the intended application maximizes discovery of relevant relationships. For instance, if a user can struggle with either an item's content or presentation, then features relevant for both are necessary to determine which impacts performance.

Importantly, raw features can be projected into a latent or embedded representational space. This is the common approach for topic modeling in text analysis [3], which finds latent 'topic' representations for documents and words. Typically these embeddings are discovered using either performance or item data, but not both. Using both types of data appropriately is challenging because the method of discovering the embeddings must respect the structural relationship between item features and performance data, and the structure in the item features themselves. To understand the similarity in educational concepts, we want to represent item similarity in terms of how the item features interact with performance data. This is because educational concepts refer to not just intrinsic aspects of the item, but the way that different learners interact with and represent them.

An example that illustrates the difficulty of discovering such embeddings in education is in number representations. Adults can easily switch between various representational formats of numbers (e.g., Hindu-Arabic numerals to mathematical concepts, finger counts to roman numerals, etc) to make assessments [8]. By contrast, young children struggle with some representational formats, which can greatly impact their ability to perform on an otherwise simple problem. This is similar to creating representations that respect the 'style vs. content' distinction [9], in that the latent representation must correspond to the structure that is hypothesized in the data.

Dominic Mussack, Rory Flemming, Paul Schrater and Pedro Cardoso-Leite "Discovering item similarity through deep learning: combining item features and user behavior." In: *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, Collin F. Lynch, Agathe Merceron, Michel Desmarais, & Roger Nkambou (eds.) 2019, pp. 615 - 618

The difficulties in estimating similarity are partly due to the complexity of item features. We demonstrate this using Mathemarmite (mathemarmite.lu), an educational game designed to teach children numeracy. Mathemarmite requires children to appropriately count using different representations (e.g., digits, fingers) (see figure 1). Each item requires selecting the right number of each ingredient to mix together, with the ingredient count represented in various ways. Trials are structured hierarchically, as a single "recipe" can include multiple representations at once with varying count. In other words, each item can have multiple "subitems" that must be completed in any order for successful item completion. This produces an added complexity where any method we develop must deal with both variable number of features and permutation invariance [10]; the order of the items completed is arbitrary.

Figure 1: A standard trial in the educational game Mathemarmite. Children change the visuals of the monster (on the right) by correctly following recipes on the scroll. Ingredients from the shelf can be placed in the cauldron, with correct number of each ingredient shown on the recipe using various numeric representations (here tick marks and a die). Item features include the number of lines, line representation, and number of ingredients per line. Mathemarmite uses an adaptive difficulty system, to keep users engaged at their skill level.

Here we develop a deep learning approach towards discovering similarity in Mathemarmite. Our approach is similar to partial least squares [2], in that we find a directed latent space representation where item similarity is constrained based on similar performance. It is also similar to work by [11], who develop a dynamic key-value memory network (DKVMN) which learns a concept embedding for items, and models an evolving knowledge-state of a learner. A similar memory-augmented neural network approach in educational systems was used by [4]. These approaches use long short-term memory (LSTM) networks to compress a user's history, along with attention-based mechanisms to compute similarity among items. We limit ourselves to a non-dynamic system, given the data size we are working with. Our approach focuses on user-independent concept space (see figure 2), attempting to find a similarity score across users.

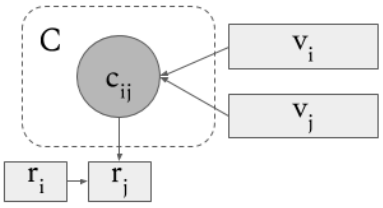

Figure 2: Item features v_j and v_i are used to predict performance on item j by embedding them in a similarity space C. The similarity score c_{ij} is used along with performance on problem i to predict problem j. This constrains the similarity space to finding scores that are similar in terms of their impact on performance.

2. METHODS

2.1 Datasets

We present results using one simulated dataset and one collected educational game dataset (i.e., Mathemarmite). The Mathemarmite dataset is based on 4961 trials from 140 users (after cleaning to remove users with less than 5 trials and users who are not in the target demographic i.e., remove adults). The simulated dataset is constructed to mimic the structure of the Mathemarmite dataset, based on the assumptions of the similarity network (e.g., figure 2). That is, we generate 5 latent "concepts", represented as 5 Gaussian spheres in concept space (each 5d), and sample item concepts from them. Items are then transformed to produce observed features using simple random linear projection matrices, and then these latent concepts determine performance based on a cosine similarity between the concept latent space and the item's concept space score, where the latent spaces are weighted based on user performance. If a user is good at a concept, and there is high overlap between that concept and the item's latent score, then the user has a high performance. We generate 5000 samples from this dataset for 5 users each (25000 items) for training. We also compare training performance on one user versus on all users, referring to them as "Multi user simulated data" and "Single user simulated data".

2.2 Targeted similarity network

2.2.1 Network Architecture

Here we develop a deep learning framework for learning concept embeddings that integrates behavioral and item-features (see figure 3). We construct a targeted similarity network that learns an embedded representation for similarity comparison, based on item performance. We attempt to learn a similarity space for item features v_i by learning a predictive model $p(r_i|v_i, v_j, r_j) = \sigma(c_{ij} \times r_j)$, that is we want to predict the performance of problem i from problem j and the similarity between the two (where σ is a sigmoid function).

Categorical features are embedded into a linear space, using a learned embedding matrix E_1. We then employ a deep set architecture to deal with permutation invariance in the subproblems. Consider an item that involves a series of subproblems to be solved, each of which can be solved in any

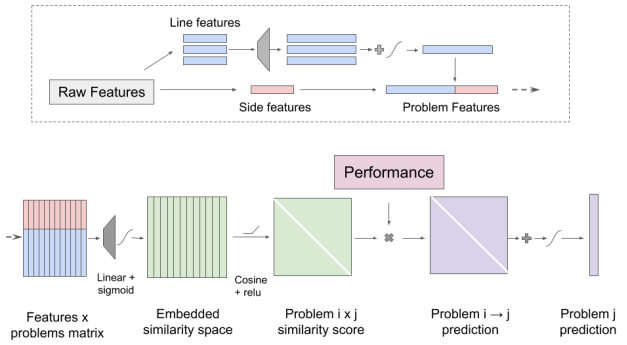

Figure 3: Top: Network feature embedding component, where embedded features are then passed into the main network. Features are processed through these steps, separating line features to deal with permutation invariance, and then concatenated with side features. Bottom: Main network embedding, where features are embedded into a similarity space and similarity scores between all items are computed for prediction.

order. We can then consider the problem P to be composed of subproblems p_k, and side features s. Therefore: $P_i = \{p_1, p_2, \ldots, p_m, s\}$, where the indices k of the subproblems may be swapped, and m is variable across problems (i.e., the number of sub-problems can vary, such that P_i might include only a single subproblem). Each subproblem has an associated feature vector of length l. As mentioned above, each subproblem has one categorical feature labeled a_k, we embed with the same matrix (note $p_k = [a_k, z_k]$ with non-categorical z_k features). To model the permutation invariance, we use sum-decomposition via latent space as in [10], to produce a single vector per problem that represents the combined influence of all subproblems.

Each problem also has a vector of side features, which includes a feature coding the number of subproblems. The decomposed subproblems are then concatenated with the side features to produce an overall item feature set, which we label x_i for problem i. We can describe this initial feature embedding process with the set of functions (see figure 3):

$$\hat{a_k} = E_1^T a_k \text{ for each } a_k \text{ in } P_i \quad (1)$$

$$v_i = \sigma(sum_m(E_2^T \hat{P}_i)) \quad (2)$$

$$x_i = [v_i, s_i] \quad (3)$$

where $\hat{a_k}$ are the embedded categorical features for each subproblem and E_2 is the linear weights for the sum-decomposition (size l by e_2 latent space). The sum is down the same dimension as the stacked feature vectors, such that v_i is length e_2. Then we compute item similarity by projecting the embedded feature matrix X to the similarity space S, and take the cosine similarity of S with itself: Item performance is them predicted off of the similarity score and other item's performance (represented -1 or 1 for incorrect, correct), and passed through a sigmoid function. Items with high similarity are then predicted to have similar performance, while those with low similarity are not. Since our interest is in the similarity space itself, we allow all items in a batch to predict the performance of all other items, by adding their prediction.

$$S = \sigma(W^T X) \quad (4)$$

$$C = sim(S) \quad (5)$$

$$\hat{R} = CR \quad (6)$$

Where R is a batch-size square matrix of the response vector copied down the rows. $sim(S)$ computes the similarity score of all items using positive cosine similarity, $sim(s) = relu(\hat{S}\hat{S}^T)$ where $\hat{S} = \frac{S}{||S||}$. This allows C to take on the interpretation of item by item similarity matrix where C_{ji} (jth row and ith column) being the $i \rightarrow j$ similarity for prediction. The final prediction is then made by summing across $\hat{R}$ column space (ignoring the diagonal $i \rightarrow i$ similarity), that is $\sum_i \hat{R}_{ji}$ to produce a vector of predicted j responses when passed through a final sigmoid output.

Table 1: Final AUC Test Scores

Model	Dataset	AUC Score
Network	Single-user sim	~ 0.999
Network	Multi-user sim	0.998
Network	Mathemarmite	0.518
Subject average	Mathemarmite	0.762
Item average	Mathemarmite	0.541

3. RESULTS

All network experiments are coded using Pytorch. Networks are trained using Binary Cross Entropy loss. Hyperparameter selection is done via cross-validated AUC scores on 100 runs, finding an embedded similarity space of 7 for Mathemarmite and 5 for both simulated datasets. We then train these networks on 500 runs (epochs) with those parameters using a 20% train-validation split (split trials selected at random), showing both train and validation AUC scores in figure 4.

We also compared against two baseline models: predicting performance off of each subject's average performance, and predicting performance off of each items's average performance (leaving out the given trial in both cases). Final AUC test scores are shown in table 1.

Note that AUC of 0.5 is by chance with 1 being perfect, so the Mathemarmite dataset is just above chance performance.

3.1 Visualization

The significant value of this network is in being able to construct a similarity space to compare the items, and then interpret the resulting space. To visualize the space, we use t-SNE [5], which allows us to project the similarity weights of items down to a 2D space for visualizing clusters. As shown in figure 5, we are able to reconstruct our latent concept space in the simulated dataset using the network (single-user). Similar projection exists for multi-user dataset.

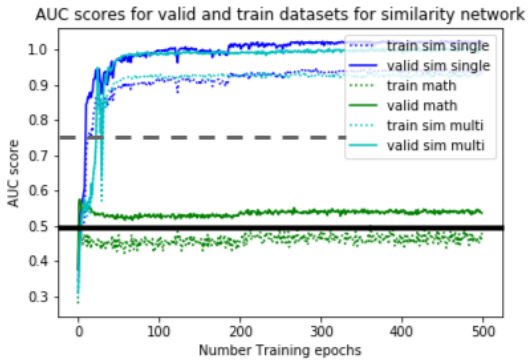

Figure 4: AUC scores across training epochs. Black line indicates chance performance, while dashed line is baseline prediction in the Mathemarmite dataset (average subject performance).

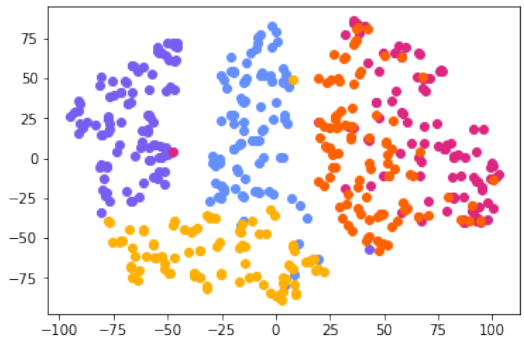

Figure 5: Recovered latent spaces in simulated data (single user). Colors correspond to arbitrary latent concept.

While we do not have ground truth for the Mathemarmite dataset, we can still perform the same visualization. However, given our final performance score, we cannot directly interpret the resulting space, and it is therefore not a meaningful latent concept space.

4. CONCLUSION

In this paper we developed a novel network for learning a similarity space for educational test data, and applied it to simulated and real data. Given that the performance on the simulated dataset is high, the network is able to reconstruct the latent space of items appropriately, given the assumptions of the underlying data generative process. However, the network does not perform well on the collected dataset. Given that we can predict line performance using a standard logistic regression model (AUC at ~ 0.65, where line features predict line performance), item features are informative of line performance, and therefore item performance (recall that items are made of multiple lines). A possible issue with the Mathemarmite dataset is that we intermix different users, with high variability across users. Given the adaptive algorithm that underlies Mathemarmite, only players with higher performances will attempt otherwise harder problems, making it more difficult to separate user and item relations.

An important limitation of our current model is it does not incorporate individual user differences in terms of the found similarity space, therefore it cannot account for changes in a user performance as well (i.e., due to learning); our network cannot learn what it cannot represent. An alternative is to incorporate user-level features, and structure the network in a bilinear fashion, much like standard item response theory. In other words, expert users likely have a different concept space than novice users. Future work involves developing a network that instead allows multiple similarity spaces across users. Such additional complexity should capture the underlying structure of the Mathemarmite dataset and lead to insights on number representational space.

5. ACKNOWLEDGEMENTS

This research was supported by the Luxembourg National Research Fund: ATTRACT/2016/ID/11242114/DIGILEARN and INTER Mobility/2017-2/ID/11765868/ULALA grants

6. REFERENCES

[1] Behdad Bakhshinategh, Osmar R Zaiane, Samira ElAtia, and Donald Ipperciel. Educational data mining applications and tasks: A survey of the last 10 years. *Education and Information Technologies*, 23(1):537–553, 2018.

[2] Paul Geladi and Bruce R Kowalski. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17, 1986.

[3] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[4] Qi Liu, Zai Huang, Zhenya Huang, Chuanren Liu, Enhong Chen, Yu Su, and Guoping Hu. Finding similar exercises in online education systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1821–1830. ACM, 2018.

[5] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[6] Radek Pelanek. Measuring similarity of educational items: An overview. *IEEE Transactions on Learning Technologies*.

[7] Jirí Rihák and Radek Pelánek. Measuring similarity of educational items using data on learners' performance. In *EDM*, 2017.

[8] Roger N Shepard, Dan W Kilpatric, and James P Cunningham. The internal representation of numbers. *Cognitive psychology*, 7(1):82–138, 1975.

[9] Joshua B Tenenbaum and William T Freeman. Separating style and content with bilinear models. *Neural computation*, 12(6):1247–1283, 2000.

[10] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401, 2017.

[11] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, pages 765–774, 2017.