Data Efficient Educational Assessment via Multi-Dimensional Pairwise Comparisons

Hyunbin Loh[*1], Piljae Chae[*1], Chanyou Hwang[1]
[1]Riiid! AI Research
{hb.loh, piljae.chae, cy.hwang}@riiid.co

ABSTRACT

The assessment of students based on their tasks is important in Education, and many advanced methods are applied to the field to solve this problem. Many recent neural network approaches involve heavy modeling of the contents and students. However, it is shown that using pairwise comparisons without the direct usage of instance features can show better assessments in the aspect of consistency, and speed. These ideas have been examined in various perspectives since Thurstone proposed the idea of Comparative Judgement(CJ). Whereas CJ requires direct comparisons of instances to obtain the final fit of the label, we give a generalization by proposing a label prediction model which uses the multi-dimensional features of pairwise comparisons. By reducing the cost in label inference, an Education service can provide visualizations of multi-dimensional skill levels for better meta-cognition of the users. Experimental results on the open dataset *EdNet KT1* show that our method gives higher accuracy even without using the actual responses for the model input.

Keywords

Educational Assessment, Adaptive Comparative Judgement, Deep Learning, Pairwise Comparison

1. INTRODUCTION

In the development of Intelligent Tutoring Systems (ITS), student assessment from their tasks and interactions is a central problem. It is shown in general education scope, that student assessment is highly correlated with the improvement of motivation, engagement, and achievement of the students. Especially in ITS, the decisions of tutoring strategies in many cases rely on algorithmic assessments of student performances. Instances of tutoring decisions include providing educational feedback or adjusting the provided contents to the students in the system. For interactive education systems, real-time computation of assessment is required, and various methods are implemented to settle the computation time problem.

Methods for student assessment have been studied in various aspects, including well established fields such as Item Response Theory(IRT), Cognitive Diagnosis Model, and Knowledge Tracing. In real-world systems, many assessment methods are based on domain expert knowledge, such as Knowledge Graphs, tagging of contents, and expert designed rule based models (such as the Rasch model in IRT). Recently, data-driven methods with less dependency on domain expert knowledge are also widely applied in ITS. Collaborative Filtering approaches such as Matrix Factorization, or Neural Collaborative Filtering are applied to embed users and items for tasks such as student response prediction, and content recommendation. There are also fully data-driven deep neural networks with no domain expert dependencies that are capable of modeling, prediction, assessment, and recommendation problems in Education such as Deep Knowledge Tracing. These methods not only show high accuracy for the target tasks, but are also easier to apply to new domains since they are domain independent.

However, many existing data-driven methods, including neural network models, require large volumes of training data and also require high costs on inference computations for achieving high performance. Methods based on domain experts can have less complexity, making their operating costs low, but developing such methods often requires a high cost on domain experts. This cost problem can be a barrier on providing e-learning services to people in underdeveloped countries, which also results in digital inequality. [8]

In this paper, we propose two assessment models based on pairwise comparisons to solve the assessment cost problem on data, and inference computation aspect. The key concept of the proposed method is to design a multidimensional generalization of Comparative Judgement(CJ). Instead of using each response data for assessment as in many supervised learning models, we only use pairwise comparisons of user responses. This model can be trained using significantly less label data compared to existing data-driven methods. Also, pairwise comparison data can be gathered within the ITS itself without additional cost. This reduces the cost to gather labeled data, and since the proposed model has less complexity, the computation cost for assessment is also reduced.

We evaluate the proposed methods that use pairwise comparison data by comparing the results with other baseline

Hyunbin Loh, Piljae Chae and Chanyou Hwang "Data Efficient Educational Assessment via Multi-Dimensional Pairwise Comparisons" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 633 - 637

models that directly use response data for predictions.

2. RELATED WORKS

2.1 Student Assessment

Various data-driven student assessment methods have been studied by researchers of ITS, where most are based on three main approaches: knowledge tracing models, collaborative filtering based models, and domain knowledge based models. Knowledge Tracing(KT) is used as a term on various assessment methods to model users by their interactions within a tutoring system [3]. Yudelson, el al. [26] suggested a Bayesian model that estimates the student performance by response correctness data. Piech, et al.[13] applied deep neural networks to KT. Some approaches [1] involve the modeling of contents and students, using pre-trained networks trained for different tasks such as BERT[4], or QuesNet [25]. These approaches directly use the response data of users to estimate the student performance.

Collaborative filtering aims to model users and items to predict potential user-item responses based on user-item interaction data [18]. Using the modeled user and item vectors, one can recommend items to a user that have high predicted labels [17]. Where matrix factorization is widely used due to the simple implementation, neural network models are also suggested to capture more complex features in user-item interactions [7]. The authors of [11] suggested a collaborative filtering based approach to predict the probability of a student answering to a question correctly.

Some methods are based on domain expert knowledge such as Knowledge Graphs, tagging of contents, or expert designed rule based models [5]. Martin, et al.[12] proposed a method to use the Bayesian network that reflects rules designed by domain experts. Item Response Theory(IRT) can be applied by by tagging items with their difficulty, or knowledge requirements [10], [20].

2.2 Pairwise Comparison based Models

Supervised learning (regression and classification) is the process of predicting labels of instances using the features of instances. The features of instances can be structured (nominal, ordinal), or unstructured(image, text, sound). Some models also utilize features that are not from the instance themselves by *pre-training* methods. Pre-training is to train a model on an unsupervised auxiliary task and use the trained model to perform the supervised main task [6].

However, there are also models that predict the labels using pairwise comparisons of instances, without using the features from the labels. An instance of a pairwise comparison based assessment method is Comparative Judgement (CJ). The concept was first introduced by Thurstone [22] in the context of Psychological assessment. CJ takes the order comparison data (high or low) of instance pairs to fit a 1-dimensional ordered label of instances. This method is especially effective in domains where there is no standardized assessment method, such as essay marking, image quality assessment [24], [15]. For instance, the authors of [15] performed a major experiment asking professional markers to give comparisons, instead of direct markings. Performing Adaptive Comparative Judgement shows better reliability, and speed than traditional marking in particular areas such as essay marking [16], and mathematics problem solving [9]. Comparative judgement techniques are also applied in areas such as Psychology, Education [19], [9], [14]. The authors of [23] designed a neural network from pairwise comparison data to solve a regression problem, where the features and labels are uncoupled. They make pairwise comparisons of 1-dimensional values to predict the label. We generalize this idea to build models that use multi-dimensional features of pairwise comparisons to predict the label.

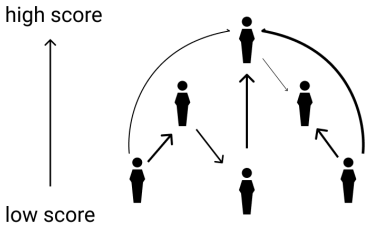

Figure 1: The training step

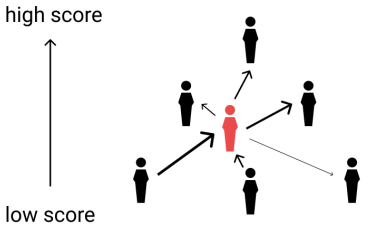

Figure 2: The inference step

3. PROPOSED METHODS

We propose a method to predict the label value of a user. It reflects the responses of the user to items, using pairwise comparisons of responses with other reference users. Possible examples of the labels are preferences on items, expected response values, performance levels on a task, or knowledge levels. The main idea is to model the relative relation of user pairs by the features of pairwise comparisons, as in Figure 1. The arrows between users describe pairwise comparison results, and two users with no arrow in between are incomparable users. For inference, the comparisons of a target user to multiple reference users is used to predict the label of the target user, as in Figure 2.

3.1 Data Description

We use the *EdNet KT1* open dataset [2], which has 95M rows, with columns *userId, questionId, correctness, timestamp*. We do not use any other data source for label data for the experiments to be reproducible from open data. Therefore, note that the following steps to construct labels are not an essential part of the proposed method. If labels of users are available in another experiment setting, then those labels can be used without this additional process. The labels from *EdNet KT1* for this experiment are constructed from the response data by the following procedure:

We sort the items by their total count in the response dataset in decreasing order. Then, we take the first 50 items. Filter the raw data by users who responded to all 50 items, and compute the correctness rate of 50 items and use it as the

label. Filter the raw data by users who responded to all 50 items, and filter out the responses on the chosen 50 items for the experiment.

3.2 Proposed Methods

Before we introduce the details of the proposed models, we describe an example case of the models to illustrate the underlying idea. Consider a case where we have user-item interaction data with columns *userId, itemId, correctness* as in the *EdNet KT1* data case. Fix two users u_1, u_2 and let TT, TF, FT, and FF be the number of items that both u_1, u_2 responded correctly, only u_1 responded correctly, only u_2 responded correctly, both u_1, u_2 responded incorrectly respectively. If $TT = 90, TF = 10, FT = 110, FF = 40$, then u_1 correctly responded to 90% of the items that u_2 correctly responded, where u_2 correctly responded to 45% of the items that u_1 correctly responded. This relation shows an aspect that the knowledge of u_1 includes the knowledge of u_2 more than the other way round. Then, let y_1, y_2 be some label that reflects the educational performances of users u_1, u_2. Then, we can consider a model that takes TT, TF, FT, FF and y_2 as features to predict the label y_1. This model is an example of the first proposed model that we introduce later in this section. The second model that we propose is based on comparisons with multiple users. The main idea is to predict a label from multiple comparisons with other reference users. Now we describe the details of the pre-processing procedure for the proposed models in the general setting.

Consider the general case where we have response data with columns: *user_id, item_id, response*, where the possible responses of users to items are $1, \ldots, r$. We show sample tables for each step of the whole process starting from Table 1.

user_id	item_id	response
1	19	1
1	23	r
2	77	2

Table 1: Raw data example

Fix N items to use as the labels, and filter out responses which have *item_id* in those N items. Group by *user_id* and make r arrays $l_1, \ldots, l_r$ where each l_i is the array of *item_id*s that is responded as i. Then, append the label columns $y_1, \ldots, y_N$ to this table.

user_id	l_1	...	y_1	...	y_N
1	[19,35,63]	...	0.84	...	0.72
2	[4,19,88]	...	0.30	...	0.54
3	[9, 17]	...	0.76	...	0.66

Table 2: User Table Example

This table has columns *user_id*, $l_1, \ldots, l_r, y_1, \ldots, y_N$. Now, we fix *reference users*, which is a subset of the users in the User Table. Then, filter the User Table by the users in the *reference users*.

Join the User Table with the Filtered User Table by the *user_id* column of each table to obtain the table with columns *user_id_1, user_id_2*, and

$$l_{1,1}, \ldots, l_{r,1}, l_{1,2}, \ldots, l_{r,2}, y_{1,1}, \ldots, y_{N,1}, y_{1,2}, \ldots, y_{N,2}.$$

user_id	l_1	...	y_1	...	y_N
1	[19,35,63]	...	0.84	...	0.72
16	[2,64,85]	...	0.89	...	0.78
22	[100,101]	...	0.24	...	0.42

Table 3: Filtered User Table Example

Then, for all $1 \leq i, j \leq r$, append the lengths of the intersections of the array pairs $l_{i,1}, l_{j,2}$ as $x_{i,j}$. Drop the columns *user_id_1, user_id_2*, and $l_{1,1}, \ldots, l_{r,1}, l_{1,2}, \ldots, l_{r,2}$, which finally leaves only the following columns:

$$x_{1,1}, \ldots, x_{r,r}, y_{1,1}, \ldots, y_{N,1}, y_{1,2} \ldots, y_{N,2}.$$

$x_{1,1}$	$x_{1,2}$	...	$x_{r,r}$	...	$y_{1,1}$	...	$y_{N,2}$
25	42	...	34	...	4	...	12
6	22	...	72	...	10	...	28
15	34	...	2	...	1	...	40

Table 4: Pair Table Example

We call this table with $r^2 + 2N$ columns the Pair Table, and we use this table for model training, where the feature columns are $x_{1,1}, \ldots, x_{r,r}, y_{1,1}, \ldots, y_{N,1}$, and the label columns are $y_{1,2}, \ldots, y_{N,2}$.

Now we introduce the proposed models: PC_1, and PC_M. The first model PC_1 is a model that predicts the label of a user by comparison with a single other user. The model uses $x_{i,j}$ and $y_{k,1}$ for features, where $i, j = 1, \ldots, r$, and $k = 1, \ldots, N$. The N-dimensional labels are $y_{k,2}$ for $k = 1, \ldots, N$. When $r = 2$, we call $TT = l_{1,1}, TF = l_{1,2}, FT = l_{2,1}, FF = l_{2,2}$. Note that each row of the input data is a comparison with one other user.

The second model PC_M uses pairwise comparisons $l_{i,j,k}$ for M multiple users $k = 1, \ldots, M$ as features. The labels are the columns $y_1, \ldots, y_N$ of a fixed user. In this model, each row of the input data is the collection of comparisons with multiple users. Then, the loss function is computed by the L1 norm of the N dimensional prediction error. We used a simple fully connected network structure:

- FC($N + r^2$, 64), ReLU
- FC(64, 32), ReLU
- FC(32, N)

3.3 Inference using the proposed model

To predict the labels of a new user u with responses l_1, l_2, ..., l_r, we compute the join of this row with the User Table to make the Pair Table of u, by following the steps in the previous subsection. The created Pair Table is the table of pairwise comparisons of user u, with the other users in the User Table. Feeding the processed features to the proposed models PC_1, PC_M give the predictions of the labels.

3.4 Neural Network baseline models

Our baseline model is a simple neural network model based on *Fully connected feed-forward network*. The network parameters are set to match the network we proposed above

in 3.2. The hyperparameters, which include the model dimension and depth, have been fit to yield the best results. The Neural Network baseline model takes all user responses as input. We name this model *NaiveFC*. In *EdNet KT1* dataset, there are 3 possible labels for each question. 1 for correct response, 2 for incorrect response and 0 if there is no response. Each value is embedded into a latent space, and the embedded values are added as an input to the model. We find the best performance when the latent space dimension is 128.

3.5 Matrix Factorization and Random Forests as Baseline

We also train a Matrix Factorization (MF) model using Alternating Least Squares [21]. In the proposed models, we split users into train/validation, or train/validation/test. However, matrix factorization models cannot be trained to optimize the results on the validation set, since the user-item embedding of only the validation set will be trained. This leaves the data on train users to be ignored. Therefore, we train the MF model on *train* + *validation* users and evaluate on *validation* users. This method of data feeding gives higher accuracy since the validation data is included in the train data. Therefore, Matrix Factorization is not a proper baseline for direct comparison because of train-test data cheating. Still, the results are listed as a comparison of the proposed models. We also train a Random Forest Regression model, with maximum depth 30 and the number of trees 300. Likewise, any other regression model can be used after the pre-processing steps.

4. EXPERIMENTS AND RESULTS

From the *EdNet KT1* dataset, we filter the responses from the users who have solved all the 50 most-responded items. The filtered data consists of 9,539,455 responses from 3692 users. To evaluate the data efficiency of our model, we compare the performance of our model while varying the minimum number of responses of each user in the data. The minimum number is varied by 50, 100, and 200, which is the number of responses after excluding the responses for the 50 items. For each setting, the total dataset is filtered by the users who responded more than the minimum number. Then, the users are split into *train* and *test* users by 9:1.

In the PC_M case, we construct three cases for the size of reference users, which are 8/9, 1/9, and 1/90 randomly sampled users from the *train* users in the filtered dataset. These numbers correspond to 80%, 10%, and 1% of the total users. The PC_M models are named by the portion of response users, and the minimum number of responses excluding the label items. For instance, PC_M50 80% corresponds to the case where the users are filtered by those who answered more than 50 questions excluding the label items, and 80% of the total users are randomly assigned as reference users.

We compare the results with the baseline models *NaiveFC*, Random Forest, and a constant value model. Both *NaiveFC* and Random Forest models take the vector with each element representing the response value of non-label items as input. There are 10782 columns that are used as features, since there are 10832 items in total. The constant value model predicts everything as the average of the labels of the training dataset. The models are trained to predict the label column, which is the correctness rate for 50 label items.

All proposed models based on pairwise comparisons show better performance compared to the baseline models *NaiveFC*, Random Forest, and Matrix Factorization. From the experiments on PC_M, we show that the reduction of 98.75% reference users, also resulting in 98.75% reduction of the feature columns in a different sense, shows similar levels of performance. The 1% reference user case, where there are only 32 reference users, surpasses the performance level of the baseline models, and also shows similar level of performance with more reference users.

The first Matrix Factorization model is trained by the test data included in training, and evaluated by test data. The second model is trained by test data, and evaluated by the same test data. When predicting all the values as the average label values of the training dataset, the MAE for test dataset is 0.1498.

Model	$\text{MAE}_{\text{train}}$	MAE_{test}
$PC_1$100	0.0956	0.0974
$PC_1$200	0.0955	0.0974
PC_M50 80%	0.0956	0.0973
PC_M50 10%	0.0956	0.0969
PC_M200 10%	0.0956	0.0974
PC_M50 1%	0.0958	0.0966
PC_M200 1%	0.0954	0.0968
NaiveFC	0.1560	0.1648
Random Forest	0.0379	0.1011
MF(Test in Train)	0.3391	0.2437
MF(Trained by Test)	0.1872	0.1872
Average	0.1519	0.1498

Table 5: EdNet Results

5. CONCLUSION AND FUTURE WORK

We have presented two assessment models PC_1 and PC_M based on pairwise comparisons. Experiments show that the proposed models give good results in the *EdNet KT1* case. The features TT, TF, FT, FF capture the relative ordering of the educational performance, as described in the beginning of Section 3.2.

Our method can be applied in any domain where multi-dimensional features capture a uniform ordering of labels, as in the education assessment case. To apply the methods to other problems, one can simply exploit the pre-processing method described in the paper for different labels. The experiments of this paper use labels constructed from the response data, but note that this process is made before applying the proposed models. By using external labels, one can skip the label constructing process and simply feed the pairwise comparison features to the proposed models.

Also, this paper only presents the performance of our models using user response data of *EdNet KT1* as features. We presented a baseline of our approach. Further experiments can be made on other open data such as *ASSISTment*. In our expectation, by leveraging richer data into the features, such as time spent for solving a problem and user behaviors

during solving problems, the accuracy of the models would be improved. Also, adjoining pairwise comparison features to existing real-world models can be a way to reduce the inference cost, as well as label data gathering cost. We leave this as our future work.

6. REFERENCES

[1] Y. Choi, Y. Lee, J. Cho, J. Baek, D. Shin, S. Lee, Y. Cha, B. Kim, and J. Heo. Assessment modeling: Fundamental pre-training tasks for interactive educational systems, 2020.

[2] Y. Choi, Y. Lee, D. Shin, J. Cho, S. Park, S. Lee, J. Baek, B. Kim, and Y. Jang. Ednet: A large-scale hierarchical dataset in education, 2019.

[3] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] J.-P. Doignon and J.-C. Falmagne. Spaces for the assessment of knowledge. *International journal of man-machine studies*, 23(2):175–196, 1985.

[6] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.

[7] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[8] J. Hvorecký. Can e-learning break the digital divide? *European Journal of Open, Distance and E-Learning*, 7(2), 2004.

[9] I. Jones, M. Swan, and A. Pollitt. Assessing mathematical problem solving using comparative judgement. *International Journal of Science and Mathematics Education*, 13(1):151–177, 2015.

[10] B. W. Junker and K. Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(3):258–272, 2001.

[11] K. Lee, J. Chung, Y. Cha, and C. Suh. Machine learning approaches for learning analytics: Collaborative filtering or regression with experts? In *NIPS Workshop, Dec*, pages 1–11, 2016.

[12] J. Martin and K. VanLehn. Student assessment using bayesian nets. *International Journal of Human-Computer Studies*, 42(6):575–591, 1995.

[13] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in neural information processing systems*, pages 505–513, 2015.

[14] A. Pollitt. Let's stop marking exams. In *IAEA Conference, Philadelphia*, 2004.

[15] A. Pollitt. The method of adaptive comparative judgement. *Assessment in Education: principles, policy & practice*, 19(3):281–300, 2012.

[16] A. Pollitt and C. Whitehouse. Using adaptive comparative judgement to obtain a highly reliable rank order in summative assessment. 2012.

[17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

[18] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

[19] N. Seery, D. Canty, and P. Phelan. The validity and value of peer assessment using adaptive comparative judgement in design driven practical education. *International Journal of Technology and Design Education*, 22(2):205–226, 2012.

[20] W. F. Strout. A new item response theory modeling approach with applications to unidimensionality assessment and ability estimation. *Psychometrika*, 55(2):293–325, 1990.

[21] Y. Takane, F. W. Young, and J. De Leeuw. Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. *Psychometrika*, 42(1):7–67, 1977.

[22] L. L. Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.

[23] L. Xu, J. Honda, G. Niu, and M. Sugiyama. Uncoupled regression from pairwise comparison data. In *Advances in Neural Information Processing Systems*, pages 3994–4004, 2019.

[24] L. Xu, J. Li, W. Lin, Y. Zhang, Y. Zhang, and Y. Yan. Pairwise comparison and rank learning for image quality assessment. *Displays*, 44:21–26, 2016.

[25] Y. Yin, Q. Liu, Z. Huang, E. Chen, W. Tong, S. Wang, and Y. Su. Quesnet: A unified representation for heterogeneous test questions. *arXiv preprint arXiv:1905.10949*, 2019.

[26] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *International conference on artificial intelligence in education*, pages 171–180. Springer, 2013.