

Context-Aware Factorization for Personalized Student's Task Recommendation

Nguyen Thai-Nghe, Tomas Horvath, and Lars Schmidt-Thieme

Information Systems and Machine Learning Lab, University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany
{nguyen, horvath, schmidt-thieme}@ismll.de

Abstract. Collaborative filtering - one of the recommendation techniques - has been applied for e-learning recently. This technique makes an assumption that each user rates for an item once. However, in educational environment, each student may perform a task (problem) several times. Thus, applying original collaborative filtering for student's task recommendation may produce unsatisfied results. We propose using context-aware models to utilize all interactions (performances) of the given student-task pairs. This approach can be applied not only for personalized learning environment (e.g., recommending tasks to students) but also for predicting student performance. Evaluation results show that the proposed approach works better than the none-context method, which only uses one recent performance.

1 Introduction

Recommender systems have been applied for e-learning task recently [1, 2]. One of the techniques, for instance, is collaborative filtering, e.g. k-nearest neighbors (k-NN) or matrix factorization, which takes into account just the last rating of users, i.e. it assumes that a user rates an item once. However, in educational environment, for example, recommending tasks (or problems or exercises) to students, this assumption might not hold since each student can perform the task several times. Furthermore, recommender system for educational purposes is a complex and challenging research direction since the preferred learning activities of students might pedagogically not be the most adequate and recommendations in e-learning should be guided by educational objectives, and not only by the user's preferences [3-5].

On the other hand, recommendation techniques have also been applied for predicting student performance recently [2, 6]. Concretely, [6] proposed a temporal collaborative filtering approach to automatically predict the correctness of students' problem solving in an intelligent math tutoring system. This approach utilized multiple interactions for a student-problem pair by using k-NN method; [2] proposed using matrix and tensor factorization to take into account the "slip" and "guess" latent factors as well as the temporal effect in predicting student performance.

Previous work [2] pointed out that an approach which uses student performance prediction for the recommendation of e-learning tasks could tackle the above

mentioned problems since we can recommend the tasks to the students based on their performance but not on their preferences. Using this approach, one can recommend similar tasks (exercises) to students and can determine which tasks are notoriously difficult for a given student. For example, there is a large bank of exercises where students lose a lot of time solving problems which are too easy or too hard for them. When a system is able to predict students' performance, it could recommend more appropriate exercises for them. Thus, we could filter out the tasks with predicted high performance / confidence since these tasks are too easy, or filter out the tasks with predicted low performance (too hard) or both, depending on the goals of the e-learning system [2].

This work proposes using context-aware models for student's task recommendation which utilize multiple interactions (performances) of a given student-task pair. This approach can be applied not only for predicting student performance as in [2] but also for personalized task recommendation to students. Here, we have not focused on building a real system, but on how to model the student's task recommendation using context-aware approach [7].

2 Data sets and Methods

In this section we first introduce the data sets. We then present the method without taking into account the context (considered as a baseline) and the proposed context-aware methods.

2.1 Data sets

Two data sets are collected from the KDD Challenge 2010 (pslccdatashop.web.cmu.edu/KDDCup), which will be called "Algebra" and "Bridge" for short. We aggregated these data sets to get four attributes: student ID (s), problem ID (i), problem view (v) which tracks how many times the student has interacted with the problem, and performance p ($p \in [0..1]$) which is an average of successful solutions (averaging from "correct first attempt" attribute).

As described in the literature [8, 2], these data sets can be mapped to user-item-rating in recommender systems. In this case, students become users and problems become items which are presented in a matrix (s, i) as in Figure 1a. In this work, the context ("problem view" - v) is taken into account, thus, each data set is presented in a three-mode tensor (s, i, v) as illustrated in Figure 1c.

2.2 Baseline (Without Using Context)

Traditional collaborative filtering has an assumption that each user rates for each item once, which means that only the last rating is used. Similarly, in this work, the last performance p of a student-problem pair (s, i) is used (which ignores the multiple interactions between students and problems) and finally, a matrix factorization model

is applied. The following paragraph briefly summarizes the matrix factorization method (please see the article [2] for more details).

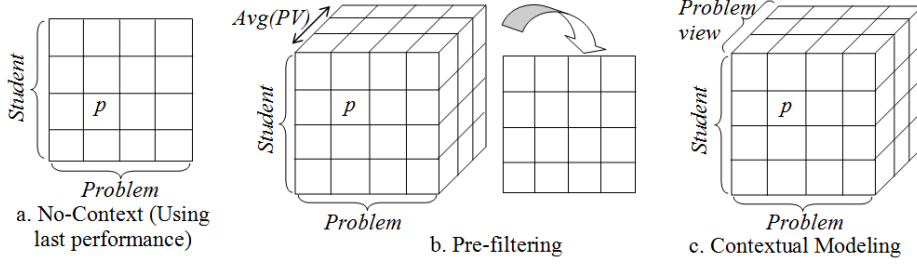


Fig. 1: An illustration of No-Context vs. Context-aware approach

Matrix factorization is the task of approximating a matrix \mathbf{X} by the product of two smaller matrices \mathbf{W} and \mathbf{H} , i.e. $\mathbf{X} \sim \mathbf{WH}^T$ [9]. In the context of recommender systems the matrix \mathbf{X} is the partially observed ratings matrix, $\mathbf{W} \in \mathfrak{R}^{S \times K}$ is a matrix where each row s is a vector containing K latent factors describing the student s and $\mathbf{H} \in \mathfrak{R}^{I \times K}$ is a matrix where each row i is a vector containing K latent factors describing the problem i . Let w_{sk} and h_{ik} be the elements of \mathbf{W} and \mathbf{H} , respectively, then the performance given by a student s to a problem i is predicted by:

$$\hat{p}_{si} = \sum_{k=1}^K w_{sk} h_{ik} = (\mathbf{WH}^T)_{s,i} \quad (1)$$

where \mathbf{W} and \mathbf{H} are model parameters which can be obtained by an optimization process using either stochastic gradient descent or Alternating Least Squares [10] given a criterion such as Root Mean Squared Error (RMSE) or Mean Absolute Error (MAE).

2.3 Context-Aware Methods

We make use of two context-aware methods: “*Pre-filtering*” and “*Contextual Modeling*” [7] (in this work we use matrix and tensor factorization approach instead of heuristic-based and model-based approaches as in [7]).

Pre-filtering (PF): As its name, this method requires pre-processing on the data sets. To do this, the performance p is aggregated (averaged) along the context v . Thus, the three-mode tensor (s, i, v) now becomes the matrix as illustrated in Figure 1b.

After the pre-filtering step, we apply the matrix factorization method to factorize on student-problem pairs (s, i) as described in section 2.2.

Contextual Modeling (CM): In this method, the context v is preserved, thus, we have to deal with the three-mode tensor. Given a tensor \mathbf{Z} of size $S \times I \times V$, where the first and the second mode describe the student and the problem as in previous sections; the third mode describes the context (problem view - v) with size V . Then \mathbf{Z} can be written as a sum of rank-1 tensors, using CANDECOM-PARAFAC [10]:

$$Z = \sum_{k=1}^K \lambda_k w_k \circ h_k \circ q_k \quad (2)$$

where \circ is the outer product, λ_k is a vector of scalar values, and each vector $w_k \in \mathfrak{R}^S$, $h_k \in \mathfrak{R}^I$, and $q_k \in \mathfrak{R}^V$ describes the latent factors of student, problem, and context, respectively. With this approach, the performance of student s for problem i at context v (problem view) is predicted by:

$$\hat{p}_{siv} = \sum_{k=1}^K \lambda_k w_{sk} h_{ik} q_{vk} \quad (3)$$

“Student bias/effect” and “problem bias/effect”: As shown in the literature [11, 8, 2], the prediction result can be improved if one incorporates the biased terms to the model. In educational setting, those biased terms are “student bias/effect” which models how good/clever a student is (i.e. how likely is the student to perform a problem correctly), and “problem bias/effect” which models how difficult/easy the problem is (i.e. how likely is the problem in general to be performed correctly) [2].

With these biases, the performance p in the pre-filtering method becomes

$$\hat{p}_{si} = \mu + b_s + b_i + \sum_{k=1}^K w_{sk} h_{ik} \quad (4)$$

and the performance p in the contextual modeling method (equation 3) becomes

$$\hat{p}_{siv} = \mu + b_s + b_i + \sum_{k=1}^K \lambda_k w_{sk} h_{ik} q_{vk} \quad (5)$$

where μ is global average, b_s is student bias, and b_i is problem bias (how to obtain these values is already described the article [2]).

After the prediction phase, we can filter out the tasks with predicted high performance since these tasks are too easy, or filter out the tasks with predicted low performance (too hard) or both, depending on the goals of the e-learning system. Thus, the appropriate tasks can be delivered to students.

3 Experiments

We describe the experimental setting and then we present the comparison results.

3.1 Experimental setting

We use just the first 5,000 problems in both Algebra and Bridge data sets. We use 3-fold cross-validation and paired t-test with significance level 0.05 for all experiments. We do hyper parameter search to determine the best hyper parameters for all methods. The Matlab Tensor Toolbox is used for experimenting (csmr.ca.sandia.gov/~tgkolda/TensorToolbox).

3.2 Experimental results

Table 1 presents the mean absolute error (MAE) of the context-aware methods (PF and CM) which take into account the multiple interactions of student-problem pairs versus the baseline (without using context).

The PF and CM outperform the baseline method even though we have not used the bias terms. Employing student-problem biases to the models, the context-aware methods have statistically significantly improved to the baseline (on Algebra data set), and the PF method has promising results compared to the others. Without using biased terms, the result of CM is slightly better than PF.

Clearly, from these results we can see that the context-aware methods are suitable for taking into account the multiple interactions between students and problems. Thus, this approach can be a reasonable choice for personalized learning environment, especially recommending tasks (or problems or exercises) to the students.

Table 1. Mean absolute error of the Baseline vs. Context-Aware Methods

Data set	Baseline	Context-Aware Methods			
		PF	PF-Bias	CM	CM-Bias
Algebra	0.247±0.015	0.239±0.017	0.188±0.015	0.239±0.016	0.233±0.012
Bridge	0.193±0.033	0.185±0.030	0.150±0.023	0.183±0.030	0.170±0.024
Average	0.220	0.212	0.169	0.211	0.202

*PF: Pre-Filtering; CM: Contextual Modeling;
Algebra and Bridge-to-Algebra from 2008-2009 data sets*

Moreover, the MAE improvements in the prediction models implicitly mean that the system can recommend the “right” tasks (exercises) to the students, and thus, we can help them reducing their time and effort in solving the tasks by filtering the ones that are too easy or too hard for them. Using these context-aware models, we can generate the performance for a given student-task pair, so the remaining works are wrapping around with an interface to deliver the recommendations. However, this work is out of the scope of this paper, and is leaved for future work.

4 Conclusion

We proposed using context-aware models to utilize all performances (interactions) of the given student-task pairs. We have shown that these methods can improve the prediction results compared to the none-context method, which only uses the last performance. This approach can apply not only for personalized recommending the tasks to students but also for predicting student performance.

It is well-known that factorization methods outperform the k-NNs collaborative filtering [12]. However, the comparison of the context-aware factorization methods with the temporal collaborative filtering (using k-NNs as in [6]) is leaved for future work.

Acknowledgments The first author was funded by the TRIG project of Cantho University, Vietnam. Tomas Horvath is also supported by the grant VEGA 1/0131/09.

References

1. Manouselis, N., Drachler, H., Vuorikari, R., Hummel, H., Koper, R.: Rec. syst. In technology enhanced learning. In: Kantor, P.B., Ricci, F., Rokach, L., Shapira, B. (eds.) 1st Recommender Systems Handbook, Springer-Berlin (2010) 1-29
2. Thai-Nghe, N., Drumond, L., Horvath, T., Krohn-Grimberghe, A., Nanopoulos, A., Schmidt-Thieme, L.: Factorization techniques for predicting student performance. In Santos, O.C., Boticario, J.G., eds.: Educational Recommender Systems and Technologies: Practices and Challenges (In press). IGI Global (2011)
3. Drachler, H., Hummel, H.G.K., Koper, R.: Identifying the goal, user model and conditions of recommender systems for formal and informal learning. *Journal of Digital Information* 10(2) (2009)
4. Santos, O.C., Boticario, J.G.: Modeling recommendations for the educational domain. *Elsevier's Procedia Computer Science* 1(2) (2010) 2793-2800
5. Tang, T., McCalla, G.: Beyond learners interest: Personalized paper recommendation based on their pedagogical features for an e-learning system. In: *PRICAI 2004: Trends in Artificial Intelligence*. (2004) 301-310
6. Cetintas, S., Si, L., Xin, Y., Hord, C.: Predicting correctness of problem solving in ITS with a temporal collaborative filtering approach. In: *International Conference on Intelligent Tutoring Systems*. (2010) 15-24
7. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In Ricci, F., Rokach, L., Shapira, B., Kantor, P.B., eds.: *Recommender Systems Handbook*. Springer (2011) 217-253
8. Thai-Nghe, N., Drumond, L., Krohn-Grimberghe, A., Schmidt-Thieme, L.: Recommender system for predicting student performance. *Elsevier's Procedia Computer Science* 1(2) (2010) 2811-2819
9. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *IEEE Computer Society Press* 42(8) (2009) 30-37
10. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. *SIAM Review* 51(3) (September 2009) 455-500
11. Toscher, A., Jahrer, M.: Collaborative filtering applied to educational data mining. *KDD Cup 2010: Improving Cognitive Models with Educational Data Mining* (2010)
12. Koren, Y.: Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data* 4(1) (2010) 1-24