Using Factorization Machines for Student Modeling

Nguyen Thai-Nghe[*,†], Lucas Drumond[†], Tomáš Horváth[†], and Lars Schmidt-Thieme[†]

[†] Information Systems and Machine Learning Lab, University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany
{nguyen, ldrumond, horvath, schmidt-thieme}@ismll.uni-hildesheim.de
[*] Can Tho University, 01 Ly Tu Trong Street, Can Tho City, Vietnam
ntnghe@cit.ctu.edu.vn

Abstract. Predicting student performance (PSP), one of the task in Student Modeling, has been taken into account by educational data mining community recently. Previous works show that good results can be achieved by casting the PSP to rating prediction task in recommender systems, where students, tasks and performance scores are mapped to users, items and ratings respectively, and thus, matrix factorization - one of the most prominent approaches for rating prediction task - is an appropriate choice.
In this work, we propose using Factorization Machines which combine the advantages of Support Vector Machines with factorization models for the problem of PSP. Experiments on two large data sets show that this approach can improve the prediction results over the standard matrix factorization.

1 Introduction

Predicting student performance (PSP), one of the tasks in student modeling, has been taken into account recently [1, 2]. Precisely, PSP is the task where we would like to know how the students learn (e.g. generally or narrowly), how quickly or slowly they adapt to new problems or if it is possible to infer the knowledge requirements to solve the problems directly from student performance data [3–5], and eventually, we would like to know whether the students perform the tasks (exercises) correctly (or with some levels of certainty).

To address the problem of PSP, many methods based on traditional classification/regression techniques have been proposed [6, 7] (see [8] for more details). One of the state-of-the-art approach in student modeling, especially PSP, is Bayesian Knowledge Tracing (BKT) [3] and its variants such as BKT-EM [9] (where the parameters are learned using Expectation Maximization), BKT-BF [10] (which learns the parameters by a Brute-Force approach), or by taking individualization into account [2].

Recently, [1, 11] have proposed using recommendation techniques, e.g., matrix factorization, for PSP. The authors have shown that *predicting student performance can be considered as rating prediction task* since the *student, task*, and *performance* would become *user, item,* and *rating* in recommender systems, respectively. Moreover, [12] have shown that for the problem of PSP, the factorization techniques can produce competitive results to the state-of-the-art BKT models.

In this work, we will show that prediction results can be improved by using Factorization Machines [13] which take the advantages of both Support Vector Machines and Factorization Models.

2 Method

We first summarize the standard Matrix Factorization and the Factorization Machines, then we present an example to see how the data could be represented for the PSP problem.

Matrix Factorization (MF): Matrix factorization is the task of approximating a matrix X by the product of two smaller matrices W and H, i.e. $X \approx WH^T$ [14]. $W \in \mathbb{R}^{U \times K}$ is a matrix where each row is a vector containing the K latent factors describing the student u and $H \in \mathbb{R}^{I \times K}$ is a matrix where each row i is a vector containing the K factors describing the task i. Let w_{uk} and h_{ik} be the elements of W and H, respectively, then the performance p given by student u to task i is predicted by:

$$\hat{p}_{ui} = \langle \mathbf{w}_u, \mathbf{h}_i \rangle = \sum_{k=1}^{K} w_{uk} h_{ik}$$

Factorization Machines (FM): FM is a new model class that combines advantages of Support Vector Machines with factorization model [13]. The FM is a predictor that can work with real valued feature vector and can model the interactions between variables using factorized parameters. Thus, it is appropriate for sparse data environments. Let $\mathbf{x}$ be an input feature vector and y a target variable. Then the prediction of $\mathbf{x}$ is computed by:

$$\hat{y}(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

where $w_0 \in \mathbb{R}$ is a global bias, $w_i \in \mathbb{R}$ models the strength of variable i, and $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$ models the interaction between variable i and j

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \sum_{k=1}^{K} v_{ik} v_{jk}$$

Similar to the MF, for learning the FMs' parameters, stochastic gradient descent is also used (please refer to [13] for details).

Data representation: Similar to an example presented by [13] in recommender systems, here, for Student Modeling, suppose that we have the set of students S, tasks T, and observed data O (student, task, time, performance) as in the following:

$S = \{$ John (J), Mike (M), Bob (B), $\ldots\}$
$I = \{$ Task 1 (I1), Task 2 (I2), Task 3 (I3), $\ldots\}$
$O = \{$ (J, I1, 05/06/2012 14:00, 1), (J, I2, 05/08/2012 15:00, 1),
(M, I2, 05/07/2012 18:00, 0), (B, I2, 05/06/2012 10:00, 1),
(B, I3, 05/08/2012 17:00, 0)$\}$

These data need to be converted to the FM format, as an example in Fig. 1. Each student/task will become a binary attribute to present whether that student/task appears in the transaction. In addition, we can add more features such as the hour (time) when students solve the problem (from a biological point of view, one can guess that when the student solves the problem at 10:00am, he/she has more chance to do it correctly than doing at 18:00pm since he/she may get tired at the latter time). Moreover, we can add the features to present the last task which was solved before the current one.

Feature vector x | Target y

x^1	1	0	0	...	1	0	0	...	14	0	0	0	...	1
x^2	1	0	0	...	0	1	0	...	15	1	0	0	...	1
x^3	0	1	0	...	0	1	0	...	18	0	0	0	...	0
x^4	0	0	1	...	0	1	0	...	10	0	0	0	...	1
x^5	0	0	1	...	0	0	1	...	17	0	1	0	...	0
	J	M	B		I1	I2	I3		Hour	I1	I2	I3		
	Student				Task				Time	Last task				

Fig. 1. Data representation. Each row is a vector including feature $\mathbf{x}$ and target y

3 Experimental Results

3.1 Data sets

In the experiments, we have used two large data sets from the KDD Challenge 2010[1]. These data, namely Algebra and Bridge, represent the log files of interactions between students and the tutoring system. While students solve the problems in the tutoring system, their activities, success and progress indicators are logged as individual rows in the data sets.

The central element of interaction between the students and the tutoring system is the *problem*. Every problem belongs into a hierarchy of *unit* and *section*.

[1] http://pslcdatashop.web.cmu.edu/KDDCup/

Additionally, a different number of *skills* (or knowledge components - KCs), which represent specific skills used for solving the problem (where available), is provided.

Target of the prediction task in these data sets is the *correct first attempt* (CFA) information which encodes whether the student successfully completed the given step on the first attempt (CFA = 1 indicates correct, and CFA = 0 indicates incorrect). The prediction would then encode the certainty that the student will succeed on the first try.

For applying factorization techniques, the information in these data sets can be mapped to *user*, *item*, and *rating* in recommender systems [11], as in the following:

$$\begin{gathered} \textit{student} \mapsto \textit{user}; \\ \textit{task (skill - knowledge component)} \mapsto \textit{item}; \\ \textit{performance (CFA)} \mapsto \textit{rating} \end{gathered}$$

Information of *student*, *task*, and *performance* is summarized in Table 1. All empty values of the skill are considered as a new skill ID. The multiple skills are encoded into a single skill as described in [2].

Table 1. Information of students, tasks, and performances (CFA)

Data set	**#Students**	**#Tasks**	**#Performances**
Algebra 2008-2009 (Algebra)	3,310	2,979	8,918,054
Bridge to Algebra 2008-2009 (Bridge)	6,043	1,458	20,012,498

3.2 Evaluation metric and baselines

Evaluation metric: The root mean squared error (RMSE) is used to evaluate the models as the following:

$$\sqrt{\frac{\sum_{ui \in \mathcal{D}^{test}} (p_{ui} - \hat{p}_{ui})^2}{|\mathcal{D}^{test}|}}$$

Baselines and State-of-the-arts: The proposed approach, which uses the FM[2] for PSP, is compared with some baselines such as *global average*, *student average*, and *biased student task* (this originally is user-item baseline in [14]). The standard *matrix factorization*[3] was also compared since previous works [1, 11] shown that MF can produce competitive results. Furthermore, the proposed approach is also compared with the state-of-the-art BKT-EM[4] and BKT-BF[5] models [10, 9, 3] and Logistic Regression[6] model.

[2] software is available at www.libfm.org
[3] software is available at www.ismll.uni-hildesheim.de/personen/nguyen_en.html
[4] software is available at www.cs.cmu.edu/∼listen/BNT-SM
[5] software is available at users.wpi.edu/∼rsbaker/edmtools.html
[6] software is also available at komarix.org/ac/lr/lrtrirls

3.3 Results

The root mean squared error (RMSE) results are presented in Figure 2. Clearly, we can see that using the FM approach, the prediction results are improved over the other baselines, the standard matrix factorization, and the state-of-the-art BKT models on both Algebra and Bridge data sets[7]. Since the BKT-EM runs rather slow (even intractable) on Bridge data set, we have not reported its result on Bridge.

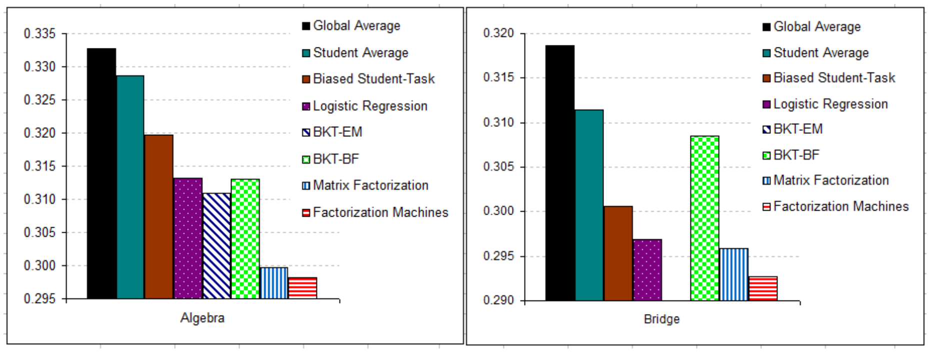

Fig. 2. Root Mean Squared Error (RMSE)

In this experiment, we only use the features of students and tasks (the first two groups of features in Figure 1), however, we believe that adding more features, e.g., the time/hour and the last task (the last two groups in Figure 1), we can reach further improvements.

4 Conclusion

In this work, we have proposed using the Factorization Machines - an open source software - which combined the advantages of SVM and Factorization Model, for predicting student performance.

Another advantage of this approach is that, in the future, we can add more features, e.g., the time/hour, weekday, the number of opportunities that the student has seen the skill, the number of problem views, the tasks were solved by the student previously, the last task, etc, to further improve the prediction results.

Acknowledgments: The first author was funded by the TRIG project of Cantho university, Vietnam. The second author was funded by the CNPq, Brazil. Tomáš Horváth is also supported by the grant VEGA 1/0832/12.

[7] However, significance test should be done in the future

References

1. Töscher, A., Jahrer, M.: Collaborative filtering applied to educational data mining. In: Proceedings of the KDD Cup 2010 Workshop, Washington, DC, USA (2010)
2. Pardos, Z.A., Heffernan, N.T.: Using hmms and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. In: Proceedings of the KDD Cup 2010 Workshop, Washington, DC, USA (2010)
3. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction **4** (1995) 253–278
4. Feng, M., Heffernan, N., Koedinger, K.: Addressing the assessment challenge with an online system that tutors as it assesses. User Modeling and User-Adapted Interaction **19**(3) (2009) 243–266
5. Koedinger, K., Baker, R., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J.: A data repository for the edm community: The pslc datashop. In Romero, C., Ventura, S., Pechenizkiy, M., Baker, R., eds.: Handbook of Educational Data Mining. Lecture Notes in Computer Science. CRC Press (2010)
6. Cen, H., Koedinger, K., Junker, B.: Learning factors analysis a general method for cognitive model evaluation and improvement. In: Proceeding of the Inter. Conf. on ITSs. Volume 4053., Springer Berlin Heidelberg (2006) 164–175
7. Pavlik, P.I., Cen, H., Koedinger, K.R.: Performance factors analysis –a new alternative to knowledge tracing. In: Proceeding of Conference on Artificial Intelligence in Education, IOS Press (2009) 531–538
8. Desmarais, M., Baker, R.: A review of recent advances in learner and skill modeling in intelligent learning environments. User Modeling and User-Adapted Interaction (2011) 1–30
9. Chang, K., Beck, J., Mostow, J., Corbett, A.: A bayes net toolkit for student modeling in intelligent tutoring systems. In: Proceedings of International Conference on Intelligent Tutoring Systems (ITS 2006), Springer (2006) 104–113
10. Baker, R.S., Corbett, A.T., Aleven, V.: More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In: Proceedings of the 9th Inter. Conf. on ITSs, Springer-Verlag (2008) 406–415
11. Thai-Nghe, N., Drumond, L., Horváth, 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 (ERSAT 2011). IGI Global (2011)
12. Thai-Nghe, N., Horváth, T., Schmidt-Thieme, L.: Factorization models for forecasting student performance. In: Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero , C., and Stamper, J. (Eds.). Proceedings of the 4th Inter. Conf. on Educational Data Mining. (2011) 11–20
13. Rendle, S.: Factorization machines. In: Proceedings of the 2010 IEEE International Conference on Data Mining (ICDM 2010), Washington, DC, USA, IEEE Computer Society (2010) 995–1000
14. Koren, Y.: Factor in the neighbors: Scalable and accurate collaborative filtering. ACM Transactions on Knowledge Discovery from Data **4**(1) (2010) 1–24