Linear Models of Student Skills for Static Data

Michel C. Desmarais, Rhouma Naceur, and Behzad Beheshti

Polytechnique Montreal

Abstract. Current student skills models rely on non linear models such as Bayesian Networks and Bayesian Knowledge Tracing, and on general linear models, such as IRT which can be considered a logistic regression. Only a handful of recent studies have looked at linear models based on matrix factorization techniques. These studies obtained good success over data from dynamic student knowledge states when compared with widely used techniques such as Bayesian Knowledge Tracing. However, there are no reports of linear models applied to static knowledge states data. We introduce different linear models of student skill for small, static student test data that does not contain missing values. We compare their predictive performance the traditional psychometric Item Response Theory approach, and the k-nearest-neighbours approach that is widely used in recommender systems. The results show that that the IRT model is far better than all others. These results are somewhat unexpected given the recent relative success of factorization models for dynamic student test data. They raise the question of whether there is still a large amount of potential performance gain from other non-linear models for dynamic data.

1 Introduction

In the field of recommender systems, linear models have taken a central role. Models based on matrix factorization fared particularly well in recent years. They allow the alignment of users and votes along a few common latent factors, which was proven very efficient for predicting votes. A culminating demonstration of the efficiency of these techniques was given in the Netflix contest [2, 10].

The recommender systems techniques were recently applied in the field of student skills modeling [5, 16, 15]. The 2010 KDD Cup was held over educational data and surely helped in bringing attention of the recommender community over the task of student skills assessment. A comparison with the widely recognized Bayesian Knowledge Tracing approach showed that it compared favourably [15]. Nguyen et al. used a multi-relational matrix and tensor-based factorization to model latent factors (skills) and the time effect to predict student success [15, 14].

The above work was conducted over dynamic student performance data. It consists of logs of success and failures of students on exercises as they interact with a learning system environment. These environments typically exercise the same skills over multiple problem, and often provide hints to help solve the exercises as the student faces difficulties. Obviously, learning will occur during the interaction. In fact, the same question item can be presented many times, failed one or mores time and succeeded in the end. This is in contrast to test data where the items are presented at once without any opportunity to access learning material. We refer to this data as static performance data, or static test data.

A large body of methods have been developed for assessing student skills with static performance data (see [8] for a review). For the vast majority, these methods are either non linear models or general linear models (for eg. logistic regression). The most widely used one is Item Response Theory (IRT) [3], one variant of which is in fact a logistic regression. Although it dates back to almost 50 years, it remains one of the most prominent and an active field of research in psychometrics (for eg. [1]).

Except for a few recent exceptions [18, 6], linear models and matrix factorization techniques have not been investigated for the purpose of skills assessment with static performance data. We define a few linear models and compare their performance with some well known models of student skills assessment, namely the classic 2 parameter IRT approach.

Although the prevailing skills assessment models for static data are simpler to develop, because they do not have to take into account the student learning in time factor, they are designed to take into account the particular characteristics of the skills performance data. For example, in the logistic regression version of IRT, the model uses the sigmoid curve slope parameter to model item discrimination, and the location parameter for item difficulty. It was proven a good fit and highly effective. Other models use the fact that we learn skills in some order and use this information to build graphical models which, in turn, can also be highly effective (see [8]).

The question we address is whether the linear models can match the predictive performance of the traditional models for skills assessment for static test data. Matrix factorization and tensors were shown to match existing state of the art models for dynamic data with a large number of missing values, but that conclusion may not necessarily hold for the other models developed for static data.

2 Results matrix, Q-Matrix, and skills matrix

Student test data can be represented in the form of a results matrix, $\mathbf{R}$, with n row items by m column students. We use the term *item* to represent exercises, questions, or any task where the student has to apply a skilled performance to accomplish it correctly. If a student successfully answers an item, the corresponding value in the results matrix is 1, otherwise it is 0.

A results matrix $\mathbf{R}$ can be decomposed into two smaller matrices:

$$\mathbf{R} \approx \mathbf{W}\mathbf{H} \tag{1}$$

The **W** matrix is generally called the Q-matrix in the cognitive modeling field [12, 13]. We keep the **W** notation here because it is more familiar in matrix factorization. This matrix is an m items by k skills matrix that defines which skills are necessary to correctly answer an item. It allows a "compressed" representation of the data that assumes the item outcome results are determined by the skills involved in each item and the skills mastered by each student. The k skills by n student matrix **H** represents the student skills mastery profiles. The product of **W** and **H** yields the expected results matrix **R**.

Note that the Q-matrix (**W**) can take different interpretations. A *conjunctive* Q-matrix assumes *all* skills in an item row are necessary for success, whereas a *disjunctive* Q-matrix assumes *any* skill is sufficient, and finally a *compensatory* Q-matrix assumes each skill *adds* to item success, which can be interpreted as increasing the chances of success if each item is either succeeded or failed. Equation (1) corresponds to the *compensatory* version of the Q-matrix, but it can be transformed into a *conjunctive* version through negation of the **R** and **H** matrices [9]. In the current work we focus on the compensatory version, which has the greatest similarity to the recommender framework and is what has been used by the previous studies mentioned [15, 5, 16].

3 Similarity with recommender systems and assumptions

Equation (1) is analogous to the decomposition of the (*item* × *user*) votes matrix into two smaller matrices: the (*item* × *preference*) and (*preference* × *user*) matrices. However, the votes matrix is typically sparse and different means have been developed to accommodate matrix factorization techniques with sparse matrices, and to compensate for predictions based on highly uneven number of votes in columns and rows that tend to negatively affect the predictive performance of algorithms.

In the case that we study, we will assume that there are no missing values in the **R** data. This type of data corresponds to the context where we obtain student performance results from a test with a limited number of questions, administered in totality to a number of students. This allows us to explore a number of standard linear models without adaptation for missing values. The context remains valuable when the objective is to adapt and reduce the number of items presented in order to derive the skills matrix **H**, or when the objective is to derive the Q-matrix (**W**) from data without missing values (or where missing values are considered 0 as is often the case, or estimated with techniques such as EM). Data with non missing values is also valuable from a theoretical perspective as it allows us to compare models more easily.

4 Factorization and linear models

We define a number of linear models and compare their predictive performance. This section defines the linear models.

Most of the linear models rely on matrix factorization. Matrix factorization is a general approach for decomposing a matrix into smaller ones based on latent factors. In the case of skills assessment data, the latent factors are the skills and the decomposition follows equation (1) as explained above. Non-Negative Matrix factorization (NMF) is a well known technique for conducting this factorization and it leads itself to a natural interpretation of the Q-matrix, $\mathbf{W}$, since we expect skills to have a null or positive link with item success[1]. Furthermore, NMF allows the column (skills) vectors to be correlated, which is also expected in reality since skills do correlate among themselves. Recent studies showed that NMF is an appropriate technique do derive the Q-matrix from synthetic data [7, 6, 19].

4.1 Outcome prediction

There are multiple models to make predictions as we see later, but they all follow the general principle of using a subset of the observed performance data for each student in $\mathbf{R}$ to predict the remaining subset, and measure the prediction accuracy. In a cross-validation with a matrix factorization model, the $\mathbf{W}$ matrix must be derived with an independent data set for maintaining validity. Therefore we need to separate the data set into four independent subsets. First, we define:

m : number of items
n : number of examinees
k : number of skills

Then, the four independent subsets are defined as:

n_1 : number of examinees in the training set
n_2 : number of examinees in the testing set ($n = n_1 + n_2$)
m_1 : number of items for the observed set
m_2 : number of items for the inferred set ($m = m_1 + m_2$)

We use the notation $\mathbf{A}_{(m,n)}$ to refer to the dimensions (m, n) of a matrix $\mathbf{A}$. Hence, the whole training data set can be referred to as $\mathbf{R}_{(m,n_1)}$, and the test data is the subset $\mathbf{R}_{(m2,n_2)}$

4.2 Factorization and linear regression

Given the factorization of equation (1), we will use regression steps to build different linear models. The regressions follow two general equations that allow estimates of $\mathbf{W}$ and $\mathbf{H}$ from different subsets of data:

$$\hat{\mathbf{H}} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{R} \tag{2}$$

[1] Negative links could be indicative of misconceptions in matrix $\mathbf{W}$, but relaxing the constraint to include negative values in this matrix would make its interpretation much more complex. And unless we add orthogonality constraints like in PCA, it would also widen the solution space and might raise convergence and stability issues.

and

$$\hat{\mathbf{W}} = \mathbf{R}\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1} \tag{3}$$

Equation (2) is a least squares linear regression to estimate $\mathbf{H}$ from $\mathbf{W}$ and $\mathbf{R}$. Equation (3) follows the same principle for estimating $\mathbf{W}$ in equation (1). We refer to the matrices estimated by regression models as $\hat{\mathbf{H}}$ and $\hat{\mathbf{W}}$. Equations (2) and (3) are the basis of regression steps in some of the linear models below.

4.3 Model 1

The first model is based on the estimates of $\hat{\mathbf{H}}$ and $\hat{\mathbf{W}}$, and its predicted values can be computed as:

$$\hat{\mathbf{R}}_{(m_2,n_2)} = \hat{\mathbf{W}}_{(m_2,k)}\ \hat{\mathbf{H}}_{(k,n_2)} \tag{4}$$

This equation states that the predicted results $\hat{\mathbf{R}}_{(m_2,n_2)}$ are obtained from a skills estimate matrix for the examinees of the testing set (n_2), $\hat{\mathbf{H}}_{(k,n_2)}$, and an estimated Q-matrix that matches these skills to the infered items (m_2), $\hat{\mathbf{W}}_{(m_2,k)}$.

The estimate $\hat{\mathbf{W}}_{(m_2,k)}$ is obtained from the subset $\mathbf{R}_{(m_2,n_1)}$: the test items (m_2) from the examinees training set (n_1). Similarly, $\hat{\mathbf{H}}_{(k,n_2)}$ is obtained from the subset $\mathbf{R}_{(m_1,n_2)}$: the observed items of the test set. The regression equations (2) and (3) provide the basis of these estimates which are given by:

$$\hat{\mathbf{W}}_{(m_2,k)} = \mathbf{R}_{(m_2,n_1)}\ \mathbf{H}^T_{(k,n_1)}\ (\mathbf{H}_{(k,n_1)}\ \mathbf{H}^T_{(k,n_1)})^{-1} \tag{5}$$

$$\hat{\mathbf{H}}_{(k,n_2)} = (\mathbf{W}^T_{(m_1,k)}\mathbf{W}_{(m_1,k)})^{-1}\mathbf{W}^T_{(m_1,k)}\ \mathbf{R}_{(m_1,n_2)} \tag{6}$$

Where $\mathbf{W}_{(m_1,k)}$ and $\mathbf{H}_{(k,n_1)}$ are taken from the factorization of $\mathbf{R}_{(m_1,n_1)}$:

$$\mathbf{R}_{(m_1,n_1)} \approx \mathbf{W}_{(m_1,k)}\ \mathbf{H}_{(k,n_1)} \tag{7}$$

4.4 Model 2

The second model consists in computing a regression model to estimate skills involved in the m_2 item set from items in m_1 according to a projection matrix $\mathbf{P}_{(k,m_1)}$:

$$\hat{\mathbf{H}}_{(k,n_2)} = \mathbf{P}_{(k,m_1)}\ \mathbf{R}_{(m_1,n_2)} \tag{8}$$

The projection matrix is defined according to a regression model based on equation (2). We start with an overdetermined system of linear equations:

$$\mathbf{H}_{(k,n_1)} = \mathbf{P}_{(k,m_1)}\ \mathbf{R}_{(m_1,n_1)} \tag{9}$$

we can obtain an estimate of $\mathbf{P}_{(k,m_1)}$ according to a linear regression (akin to equation (2)):

$$\mathbf{P}_{(k,m_1)} = \mathbf{H}_{(k,n_1)}\ \mathbf{R}^T_{(m_1,n_1)}\ (\mathbf{R}_{(m_1,n_1)}\mathbf{R}^T_{(m_1,n_1)})^{-1} \tag{10}$$

where $\mathbf{H}_{(k,n_1)}$ is taken from the factorization of the m_2 items of the training set n_1:

$$\mathbf{R}_{(m_2,n_1)} \approx \mathbf{W}_{(m_2,k)}\ \mathbf{H}_{(k,n_1)} \tag{11}$$

The predicted results are given by:

$$\hat{\mathbf{R}}_{(m_2,n_2)} = \mathbf{W}_{(m_2,k)}\ \hat{\mathbf{H}}_{(k,n_2)} \tag{12}$$

4.5 Model 3

The third model takes a direct route to prediction without reverting to latent skills. Therefore, it does not rely on a factorization but, instead, it uses a regression model to predict non observed items from observed items directly. This is summarized as the projection of $\mathbf{R}_{(m_1,n_2)}$ onto $\mathbf{R}_{(m_2,n_2)}$.

$$\hat{\mathbf{R}}_{(m_2,n_2)} = \mathbf{P}_{(m_2,m_1)}\ \mathbf{R}_{(m_1,n_2)} \tag{13}$$

The projection matrix is computed from the training data set according to:

$$\mathbf{P}_{(m_2,m_1)} = \mathbf{R}_{(m_2,n_1)}\ \mathbf{R}^T_{(m_1,n_1)}\ (\mathbf{R}_{(m_1,n_1)}\mathbf{R}^T_{(m_1,n_1)})^{-1} \tag{14}$$

4.6 Model 4

Model 4 uses two sets of latent factors, one for the observed items and one for the items to infer in the test set. It derives the latent factors with two NMF factorization, and defines a regression model to predict the latent factors of the predicted items from the latent factors of the observed items. The following equations summarizes how the predictions are calculated.

The two factorization are respectively over the observed and predicted items in the training set:

$$\mathbf{R}_{(m_1,n_1)} \approx \mathbf{W}_{(m_1,k)}\ \mathbf{H_1}_{(k,n_1)} \tag{15}$$

$$\mathbf{R}_{(m_2,n_1)} \approx \mathbf{W}_{(m_2,k)}\ \mathbf{H_2}_{(k,n_1)} \tag{16}$$

Then, a projection matrix is obtained to derive $\mathbf{H_2}_{(k,n_1)}$ from $\mathbf{H_1}_{(k,n_1)}$:

$$\mathbf{H_2}_{(k,n_1)} = \mathbf{P}_{(k,k)}\ \mathbf{H_1}_{(k,n_1)} \tag{17}$$

$$\mathbf{P}_{(k,k)} = \mathbf{H_2}_{(k,n_1)}\ \mathbf{H_1}^T_{(k,n_1)}(\mathbf{H_1}_{(k,n_1)}\mathbf{H_1}^T_{(k,n_1)})^{-1} \tag{18}$$

An estimates of the skills for the test students, with regards to the factorization in 15, is obtained as:

$$\hat{\mathbf{H}}\mathbf{_1}_{(k,n_2)} = (\mathbf{W}^T_{(m_1,k)}\mathbf{W}_{(m_1,k)})^{-1}\ \mathbf{W}^T_{(m_1,k)}\ \mathbf{R}_{(m_1,n_2)} \tag{19}$$

Using the projection matrix in equation (18), $\mathbf{P}_{(k,k)}$, the skills for the test student is obtained:

$$\hat{\mathbf{H}}\mathbf{_2}_{(k,n_2)} = \mathbf{P}_{(k,k)}\ \hat{\mathbf{H}}\mathbf{_1}_{(k,n_2)} \tag{20}$$

Finally, the predicted results are:

$$\hat{\mathbf{R}}_{(m_2,n_2)} = \mathbf{W}_{(m_2,k)}\ \hat{\mathbf{H}}_{\mathbf{2}(k,n_2)} \tag{21}$$

Note that this models requires two factorization steps (equations (15) and (16)), which adds to the computational issue we discuss later.

4.7 k-nearest-neighbours, IRT, and expected values models

Three more models are included in the study for comparison purpose.

A k-nearest-neighbours model is also developed. It is based on the standard weighted means predictions as can be found in [4]. It uses the Euclidean distance to find the first 6 neighbours and the cosine to compute the predicted item outcome as a weighted sum. A student and item bias is also added: in fact, it corresponds to the "expected model" explained below and its weight is set to be equivalent to the weighted sum above.

As mentioned in the introduction, the IRT model is a widely used and recognized approach to skills assessment. We used the two parameter logistic implementation in the R framework, `ltm` [11]. This model is a logistic regression model where the location parameter is the item difficulty and the slope is the item discrimination. It is a single skill model. The student skill is derived from the observed item outcomes and it is used to predict the outcome to unobserved items.

The *expected* model is a benchmark that corresponds to the expected outcome based on the item and student average success rates. Item success rate is estimated based on $\mathbf{R}_{m,n_1}$ and student success rate is estimated from $\mathbf{R}_{m_1,n_2}$. The expected percent error of item m_i for student n_j is defined[2] as $\sqrt{\overline{m}_i\, \overline{n}_j}$, where $\overline{m}_i$ and $\overline{n}_j$ are respectively the success rate of item i and student j.

5 Methodology and Results

The validation experiments are conducted over three data sets:

fraction algebra: composed of a 20 question items and 149 students. The data originates from [17]. The domain is elementary fraction algebra. The test was given to high school students. Average score is 61%.

UNIX shell: composed of a 34 question items and 48 students. This data set is characterized by a small number of respondents with an average performance of 54% but a high variance with an almost rectangular distribution of student scores.

College math: composed of a 60 question items and 250 students. Average score is 58%. The students are freshman engineering students.

[2] An alternative definition of the expected success rate would be: $(\frac{\overline{m}_i}{(1-\overline{m}_i)}\frac{\overline{n}_j}{(1-\overline{n}_j)})^{-1}$. Tests with this alternative shows that the results are almost similar, except for the fraction algebra where it performs slightly worst, by 1–2%.

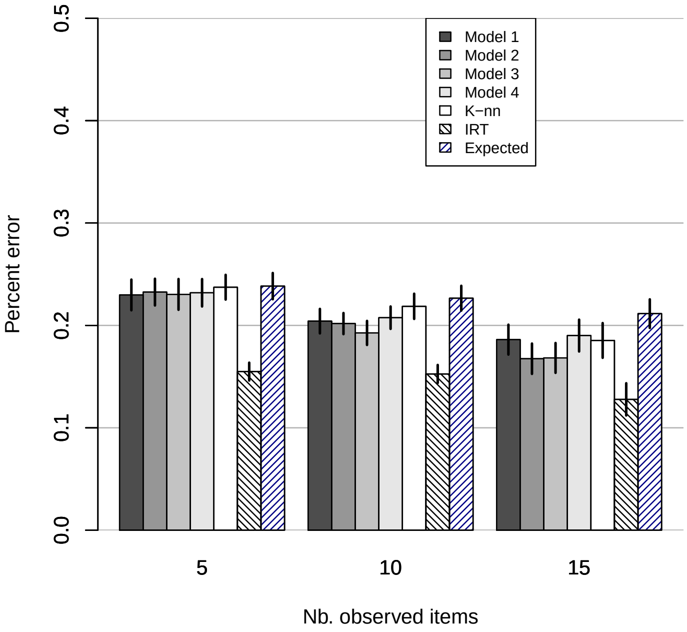

Fig. 1. Performance results of the fraction algebra data set, composed of 20 items and 149 students. Training set is 120. Mean percent error is reported for 24 runs and the vertical bars show one standard error. Factorizations in models 1 to 4 are based on 5 skills.

The question items relate to 10–12 grade fraction arithmetics. This data set was originally reported in [17]. All matrix factorizations are defined for 5 skills.

All models provide predictions for the unobserved items, n_2 based on observed items, n_1. Cross-validation relies on splitting the data into training and testing subsets. All data are reported for a 24-fold cross-validation with repeated measures across models. The details of how predictions are derived from the training and unobserved data depends on the model and is explained in the previous section, but all models end up making predictions for the unobserved items of the testing set, $\hat{\mathbf{R}}_{m_2,n_2}$, and the accuracy is measured according to the percentage of correctly predicted items. Items outcome in all predicted matrices ($\hat{\mathbf{R}}$) are discretized to $\{0, 1\}$.

Figure 1 to 3 report the mean accuracy of each model's predictions for a 24-fold experiment and for three different number of observed items, 5, 10, and 15. The standard error is also reported for each condition.

Of the four linear models, model 1 and 4 are the only ones that are either better (close to 4% better), or at par with the benchmark "expected model". Also

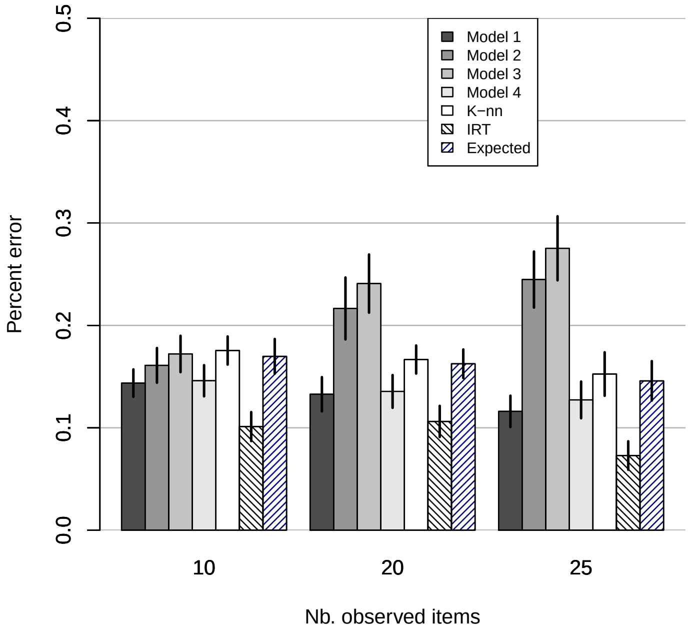

Fig. 2. Performance results of the UNIX data set, composed of 34 items and 48 students. Training set is 40. Mean percent error is reported for 24 runs and the vertical bars show one standard error. Factorizations in model 1 to 4 are based on 5 skills.

noticeable is that the performance of the linear models advantage is inconsistent when compared among them. As expected, the error generally decreases as more items are observed, but models 2 and 3 contrast with this pattern for the UNIX data set, possibly stemming from the few number of students for training in this data set (40).

The IRT model systematically performs better than the "expected" model and substantially better than all others. The error reduction ranges from 2%, for the College math, to 7–8% for UNIX (a reduction by a half of the percent correct for 25 items observed). The linear models sometimes perform better than the benchmark model "expected", but the gain is generally small, and it never is better than IRT.

Finally,the *k*-nearest-neighbours model yields a performance generally comparable to the "expected" model for the fraction algebra and UNIX data sets, but a lower performance for the College math data set.

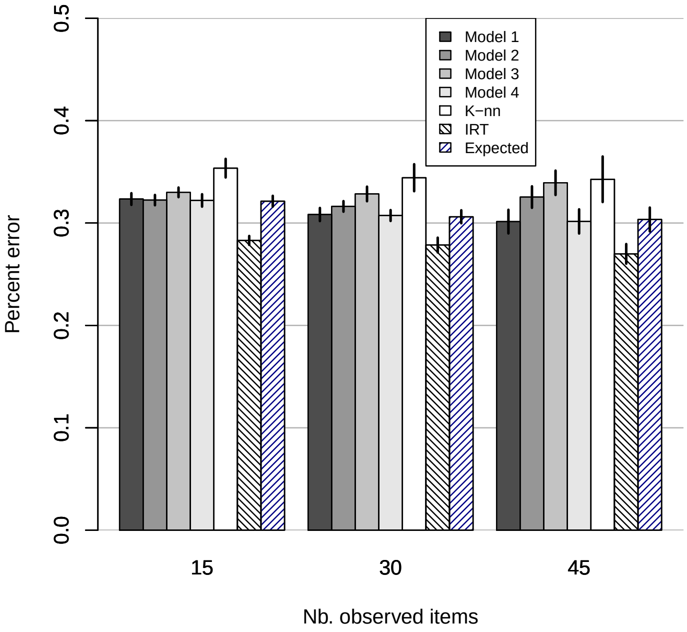

Fig. 3. Performance results of the College math data set, composed of 20 items and 250 students. Training set is 200. Mean percent error is reported for 24 runs and the vertical bars show one standard error. Factorizations in model 1 to 4 are based on 5 skills.

6 Discussion

The most salient results from the experiments is that the classical IRT model is systematically better and more stable than any of the linear models investigated. Only for one condition of the UNIX data set (20 items observed) has it performance been similar to the "expected" condition, and similar to the best linear models.

Of the four linear models, their relative performance varies a little between them, but they do not show a large improvement over the baseline Expected model. Models 2 and 3 show some instability for the smaller UNIX data set. However, from a practical perspective, three linear models require the computation to derive a factorization of some subset of items in **R**, which is generally slow and limits their applicability. Model 3 is the fastest to compute, because it requires no factorization. Its computation is around 0.001 for the fraction algebra data set, whereas Model 1 which has one factorization is around 7 seconds. The k-nearest-neighbours model is also fast to compute: around 0.01 second.

The IRT model computation is around 2 seconds to compute for the training, but it requires a single training phase which can be done offline. For that reason, IRT is a useful model in practice.

Note that the space of skills is not explored within this study. We could expect that the linear models would be improved by a better choice of the number of skills in the factorization, and other possible improvements using biases which is commonplace in recommender systems. Biases would be particularly important if our data contained a large number of missing values in order to offset the weights of predictions based on highly variable number of similar votes due to the missing values.

Note that these models involve a substantial computational burden: models 1, 2, and 4 all require a costly factorization of either the m_1 or m_2 items. Given that the number of combinations is exponential, pre-computation of the factorizations for an operational system becomes impractical for a large number of items. The factorization time also increases considerably with size, which means these models are not adequate for large data sets of more than a few tens of items. Only model 3 allows for relatively quick computations that can be done in real time.

Beyond the computational resources issues, the above results raise questions on whether the linear models can outperform the predominant IRT approach for assessing skills with static test data. Whilst improvements can be brought, there appears to be a strong performance gap to fill and practical issues to tackle before they can be considered viable alternatives to IRT.

The results also raise the question of whether IRT can bring improvements to the existing models for dynamic test data. If IRT is showing strong advantages for static data, we could expect it to show some advantage for dynamic data as well. However, IRT does not model the learning factor that comes into play in this data. But we might expect that a method to integrate it within a dynamic modeling of skills would bring interesting results.

References

1. Baker, F. B., and Kim, S.-H. *Item Response Theory, Parameter Estimation Techniques (2nd ed.).* Marcel Dekker Inc., New York, NY, 2004.
2. Bell, R. M., and Koren, Y. Lessons from the Netflix prize challenge. *SIGKDD Explorations 9*, 2 (2007), 75–79.
3. Birnbaum, A. Some latent trait models and their use in infering an examinee's ability. In *Statistical Theories of Mental Test Scores*, F. M. Lord and M. R. Novick, Eds. Addison-Wesley, Reading, MA, 1968, pp. 397–472.
4. Breese, J. S., Heckerman, D., and Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)* (San Francisco, July 24–26 1998), G. F. Cooper and S. Moral, Eds., Morgan Kaufmann, pp. 43–52.
5. Cetintas, S., Si, L., Xin, Y. P., and Hord, C. Predicting correctness of problem solving in ITS with a temporal collaborative filtering approach. In *Intelligent Tutoring Systems, 10th International Conference, ITS 2010, Pittsburgh, PA, USA,*

June 14-18, 2010, Proceedings, Part I (2010), V. Aleven, J. Kay, and J. Mostow, Eds., vol. 6094 of *Lecture Notes in Computer Science*, Springer, pp. 15–24.

6. Desmarais, M. C. Conditions for effectively deriving a q-matrix from data with non-negative matrix factorization. In *4th International Conference on Educational Data Mining, EDM 2011* (Eindhoven, Netherlands, June 6–8 2011), C. Conati, S. Ventura, T. Calders, and M. Pechenizkiy, Eds., pp. 41–50.
7. Desmarais, M. C. Mapping question items to skills with non-negative matrix factorization. *ACM KDD-Explorations 13*, 2 (2011), 30–36.
8. Desmarais, M. C., and Baker, R. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interactions 22* (2011), 9–38.
9. Desmarais, M. C., Beheshti, B., and Naceur, R. Item to skills mapping: Deriving a conjunctive q-matrix from data. In *11th Conference on Intelligent Tutoring Systems, ITS 2012, Chania, Greece, 14–18 June 2012* (2012), p. (to appear).
10. Jahrer, M., Töscher, A., and Legenstein, R. A. Combining predictions for accurate recommender systems. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010* (2010), B. Rao, B. Krishnapuram, A. Tomkins, and Q. Yang, Eds., ACM, pp. 693–702.
11. Rizopoulos, D. ltm: An R package for latent variable modelling and item response theory analyses. *Journal of Statistical Software 17*, 5 (2006), 1–25.
12. Tatsuoka, K. *Cognitive Assessment: An Introduction to the Rule Space Method*. Routledge Academic, 2009.
13. Tatsuoka, K. K. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement 20* (1983), 345–354.
14. Thai-Nghe, N., Drumond, L., Horváth, T., Nanopoulos, A., and Schmidt-Thieme, L. Matrix and tensor factorization for predicting student performance. In *CSEDU 2011 - Proceedings of the 3rd International Conference on Computer Supported Education, Volume 1, Noordwijkerhout, Netherlands, 6-8 May, 2011* (2011), A. Verbraeck, M. Helfert, J. Cordeiro, and B. Shishkov, Eds., SciTePress, pp. 69–78.
15. Thai-Nghe, N., Horváth, T., and Schmidt-Thieme, L. Factorization models for forecasting student performance. In *Proceedings of EDM 2011, The 4th International Conference on Educational Data Mining* (Eindhoven, Netherlands, July 6–8 2011), C. Conati, S. Ventura, M. Pechenizkiy, and T. Calders, Eds., www.educationaldatamining.org, pp. 11–20.
16. Toscher, A., and Jahrer, M. Collaborative filtering applied to educational data mining. Tech. rep., KDD Cup 2010: Improving Cognitive Models with Educational Data Mining., 2010.
17. Vomlel, J. Bayesian networks in educational testing. *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems 12* (2004), 83–100.
18. Winters, T. *Educational Data Mining: Collection and Analysis of Score Matrices for Outcomes-Based Assessment*. PhD thesis, University of California Riverside, 2006.
19. Winters, T., Shelton, C. R., and Payne, T. Investigating generative factors of score matrices. In *Artificial Intelligence in Education, Building Technology Rich Learning Contexts That Work, Proceedings of the 13th International Conference on Artificial Intelligence in Education, AIED 2007, July 9-13, 2007, Los Angeles, California, USA* (2007), R. Luckin, K. R. Koedinger, and J. E. Greer, Eds., vol. 158 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, pp. 479–486.