Item to skills mapping:
Deriving a conjunctive Q-matrix from data

Michel C. Desmarais, Behzad Beheshti, and Rhouma Naceur

Polytechnique Montréal

Abstract. Uncovering which skills are determining the success to questions and exercises is a fundamental task in ITS. This task is notoriously difficult because most exercise and question items involve multiple skills, and because skills modeling may involve subtle concepts and abilities. Means to derive this mapping from test results data are highly desirable. They would provide objective and reproductible evidence of item to skills mapping that can either help validate predefine skills models, or give guidance to define such models. However, the progress towards this end has been relatively elusive, in particular for a conjunctive skills model, where all required skills of an item must be mastered to obtain a success. We extend a technique based on Non-negative Matrix Factorization, that was previously shown successful for single skill items, to construct a conjunctive item to skills mapping from test data with multiple skills per item. Using simulated student test data, the technique is shown to yield reliable mapping for items involving one or two skills from a set of six skills.

Keywords: Student model, Skills modeling, Psychometrics, Q-matrix, matrix factorization, SVD, NMF

1 Introduction

When an ITS personalizes the learning content presented to a student, it has to rely on some classification of this content with regards to skills, and on the student's skills assessment. Therefore, the question items and exercises involved in the assessment must be aligned with these skills. The mapping of items to skills plays a pivotal role in most if not all ITS.

A standard means to model this mapping is the Q-matrix [10, 9]. It defines which skills are necessary to correctly answer an item. Take the Q-matrix in figure 1 (matrix **Q** on the left) composed of 3 skills and 4 items. We find that item i_1 requires two skills, s_2 and s_3, whereas item i_2 requires a single skill, s_3, and so on.

Assuming now that a set of three examinees have mastered skills according to matrix **S** of figure 1 (middle), and that all skills of an item are necessary to correctly answer this item, then we would expect a result that corresponds to matrix **R** in figure 1 (right). This framework corresponds to a *conjunctive* Q-matrix: a line in Figure 1's Q-matrix indicates a conjunction of necessary skills

$$\mathbf{Q} = \text{items}\ \overset{\text{skills}}{\begin{array}{c|ccc|} & s_1 & s_2 & s_3 \\ \hline i_1 & 0 & 1 & 1 \\ i_2 & 0 & 0 & 1 \\ i_3 & 1 & 0 & 0 \\ i_4 & 1 & 0 & 1 \\ \hline \end{array}} \qquad \mathbf{S} = \text{skills}\ \overset{\text{examinee}}{\begin{array}{c|ccc|} & e_1 & e_2 & e_3 \\ \hline s_1 & 1 & 0 & 0 \\ s_2 & 0 & 0 & 1 \\ s_3 & 0 & 1 & 1 \\ \hline \end{array}} \qquad \mathbf{R} = \text{items}\ \overset{\text{examinee}}{\begin{array}{c|ccc|} & e_1 & e_2 & e_3 \\ \hline i_1 & 0 & 0 & 1 \\ i_2 & 0 & 1 & 1 \\ i_3 & 1 & 0 & 0 \\ i_4 & 0 & 0 & 0 \\ \hline \end{array}}$$

Fig. 1. Q-matrix and skills matrix examples.

to succeed the corresponding item. The goal is to bring this framework to a linear system, allowing the application of standard linear algebra techniques.

Barnes [1] gives the following equation for inferring the expected examinee results as the product of the Q-matrix and the skills matrix (adapted from [1] for the transpose of **R**):

$$\mathbf{R} = \neg(\mathbf{Q}(\neg\mathbf{S})) \tag{1}$$

where the operator $\neg$ is the boolean negation, which is defined as a function that maps a value of 0 to 1 and any other value to 0. This equation will yield values of 0 in **R** whenever an examinee is missing one or more skills for a given item, and yield 1 whenever all necessary skills are mastered by an examinee.

Applying the operator $\neg$ on both side of equation (1) and normalizing matrix **Q** to ensure the row sums are 1 yields:

$$\neg\mathbf{R} = \mathbf{Q}(\neg\mathbf{S}) \tag{2}$$

Equation (2) is a standard linear equation where the matrices **R** and **S** are negated. The task of inferring the Q-matrix from $\neg\mathbf{R}$ can therefore be seen as a matrix factorization: the matrix $\neg\mathbf{R}$ is the product of the two matrices, **Q** and $\neg\mathbf{S}$.

2 Comparison with a one skill per item condition

The matrix factorization approach to inferring the Q-matrix from data has been explored by a few researchers [3, 11], but for Q-matrices that involved only a single skill per item. They investigated the Non-negative Matrix Factorization (NMF) [8] technique and showed that it works very well for simulated data, but the technique's performance with real data was degraded. For highly separable skills like mathematics and French, its performance is quite good, assigning correctly the items belonging to each topic. But the technique is very weak at classifying items according to skills such as History and Biology, as measured by Trivia type of questions. These results suggest that expertise necessary to succeed Biology and History questions is not well separated into these two general topics. Presumably, we would find a stronger skill seperation if we studied very specific skills, like the pieces of knowledge behind each question. This is in fact what tutors such as the Cognitive family of tutors and the ASSISTment system do, they rely on fine grain skills mapped to items [7, 5]. For these low-level skills, the conjunctive model, which requires that each skill is mastered for every item that require them, is in general the model used by widely known learning environments such as the Cognitive Tutors family.

The matrix factorization approach of the studies in [3, 11] was based on the additive (compensatory) model of skills, where each skill increases the chances of success to an item. This corresponds to the following equation where the negation operator $\neg$ is omitted:

$$\mathbf{R} = \mathbf{QS} \tag{3}$$

For the one skill per item condition, equations (1) and (3) are equivalent, but they give very different results for two or more skills per item. Following the skill structure example in figure 1, item i_4 would be failed by all examinees according to equation (1) whereas it would be (partly) succeeded according to equation (1), with values above 0 for all examinees on this item.

An obvious followup over the studies by [1, 3, 11] is to apply the NMF technique to equation (2), and to determine if NMF can successfully derive a conjunctive Q-matrix, where skills do not add up to increase the chances of success to an item, but instead are necessary conditions. This is the goal of the current investigation.

3 Non-negative Matrix Factorization

Non-negative matrix factorization (NMF) decomposes a matrix into two smaller matrices. It is used for dimensionality reduction, akin to Principal Component Analysis and Factor analysis. NMF decomposes a matrix of $n \times m$ positive numbers, $\mathbf{V}$, as the product of two matrices:

$$\mathbf{V} \approx \mathbf{WH} \tag{4}$$

Clearly, the matrix $\mathbf{W}$ corresponds to the Q-matrices of equations (2) and (3).

Whereas most other matrix factorization techniques impose constraints of orthogonality among factors, NMF imposes the constraint that the two matrices, $\mathbf{W}$ and $\mathbf{H}$, be non-negative. This constraint makes the interpretation much more intuitive in the context of using this technique for building a Q-matrix. It implies that the skills (latent factors) are additive "causes" that contribute to the success of items, and that they can only increase the probability of success and not decrease it, which makes good sense for skill factors.

It is important to emphasize that there are many solutions to $\mathbf{V} = \mathbf{WH}$. Different algorithms may lead to different solutions. Indeed, many NMF algorithms have been developed in the last decade and they can yield different solutions. We refer the reader to [2] for a more thorough and recent review of this technique which has gained strong adoption in many different fields.

The non-negative constraint and the additive property of the skills bring a specific interpretation of the Q-matrix. For example, if an item requires skills a and b with the same weight each, then each skill will contribute equally to the success of the item. This corresponds to the notion of a *compensatory* or *additive* model of skills as we mentioned earlier. The negation of matrix **R** in equation (2) brings a new interpretation of the Q-matrix where the conjunction of skills are considered necessary conditions to answer the corresponding item. This requires that the matrix **S** be also negated, and it corresponds to **H** in equation (4). However, in applying the negation operator, $\neg$, all values greater than 1 are replaced by 1, and that can be considered as a loss of information.

4 Simulated data

To validate the approach, we rely on simulated data. Although it lacks the external validity of real data, it remains the most reliable means of obtaining test results data for which the underlying, latent skills structure is perfectly known. Any experiment with real data is faced with the issues that the expert-defined Q-matrix may not contain all determinant skills, may not have a perfect mapping, and that all skills may not combine conjunctively and with equal weight, making the interpretation of the results a complex and error prone task. Therefore, assessing the technique over simulated data is a necessary first step to establish the validity the approach under controlled conditions. Further studies with real data will be necessary, assuming the results of the existing study warrants such work.

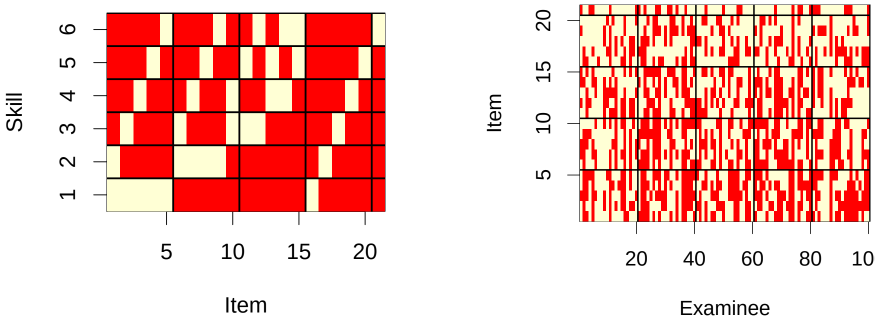
The underlying model and methodology of the simulated data are explained in a previous paper [4] and we briefly review some details this methodology below.

A first step to obtain data of simulated examinee test results is to define a Q-matrix composed of j skills and k items. We chose to define a Q-matrix that spans all possible combinations of 6 skills with a maximum of two skills per item, and at least one skill per item. A total of 21 items spans this space of combinations. This matrix is shown in Figure 2(a). Items 1 to 15 are two-skills and items 16 to 21 are single-skill.

We do not assume that skills all have the same difficulty level, and therefore we assign various difficulty level to each skill. The difficulty is reflected by the probability of mastery. That difficulty will transfer to items that have this skill. The difficulty of the two-skills items will further increase by the fact that they require the conjunction of their skills. An item difficulty is therefore inherited by the difficulty of its underlying skills.

In addition to skills difficulty, examinees need to be assigned ability levels. The ability is reflected by the probability of mastering some skill. Therefore, the probability of mastery of a given skill by a given examinee is a function of examinee ability and skill difficulty levels.

Finally, two more parameters are used in the simulated data, namely the *slip* and *guess* factors. These factors are set as constant values across items. They are essentially noise factors and the greater they are, the more difficult is the task of inducing the Q-matrix from data.



(a) Q-matrix of 6 skills and for which 21 items are spanning the full set of 1 and 2 skill combinations. Items 16 to 21 require a single skill and all others require 2-skills.

(b) Simulated data example of 100 examinees with parameters: slip: 0.1, guess: 0.2, skills difficulties: (0.17, 0.30, 0.43, 0.57, 0.70, 0.83).

Fig. 2. Q-matrix and an example of simulated data with this matrix. Light pixels represent 1's and dark (red) ones represent 0's.

Given the above framework, the process of generating simulated examinee data follows the following steps:

1. Assign a difficulty level to each skill.
2. Generate a random set of hypothetical examinee skills vectors based on the difficulty of each skill and the examinee's ability level. Skill difficulty and examinee ability are each expressed as a random normal variable. The probability density function of their sum provides the probability of mastery of the skill for the corresponding examinee. The skill vector is a sampling in $\{0, 1\}$ based on each skill probability of mastery.
3. Generate simulated data based on equation (2) without taking into account the *slip* and *guess* parameters. This is referred to as the *ideal response pattern*.
4. Randomly change the values of the generated data based on the *slip* and *guess* parameters. For example, with values of 0.1 and 0.2 respectively, this will result in 10% of the succeeded items in the ideal response pattern to become failed, and 20% of the failed items to become succeeded.

The first two steps of this process are based on additive gaussian factors and follow a similar methodology to [3]. For brevity we do not report the full details but refer the reader to the R code available at `www.professeurs.polymtl.ca/michel.desmarais/Papers/ITS2012/its2012.R`.

A sample of the results matrix is given in figure 2(b). Examinee ability shows up as vertical patterns, whereas skills difficulty creates horizontal patterns. As expected, the mean success rate of the 2-skills items 1 to 15 is lower (0.51) than the single skill items 16 to 21 (0.64).

5 Simulation methodology

The assessment of the NMF performance to infer a Q-matrix from simulated test data such as figure 2(b)'s is conducted by comparing the predefined Q-matrix, **Q**, as shown in figure 2(a), with the **W** matrix obtained in the NMF of equation (4).

As mentioned above, the negation operator is applied over the simulated test data and the NMF algorithm is carried over this data. We used the R NMF package [6] and the Brunet NMF algorithm.

We defined a specific method for the quantitative comparison of the matrix **W** with **Q**. First, the **W** matrix contains numerical values on a continuous scale. To simplify the comparison with matrix **Q**, which is composed of $\{0, 1\}$ values, we discretize the numerical values of **W** by applying a clustering algorithm to each item in **W**, forcing two clusters, one for 0's and one for 1's. For example, item 1 in the NMF inferred matrix of figure 4(a) (which we explain later) corresponds to a vector of six numerical values, say $\{1.6, 1.7, 0.0015, 0.0022, 0.0022, 0.0018\}$. This vector clearly cluster into the $\{1, 1, 0, 0, 0, 0\}$ vector of item 1 in figure 4(b). The K-means algorithm is used for the clustering process of each item and we use the `kmeans` routine provided in R (version 2.13.1).

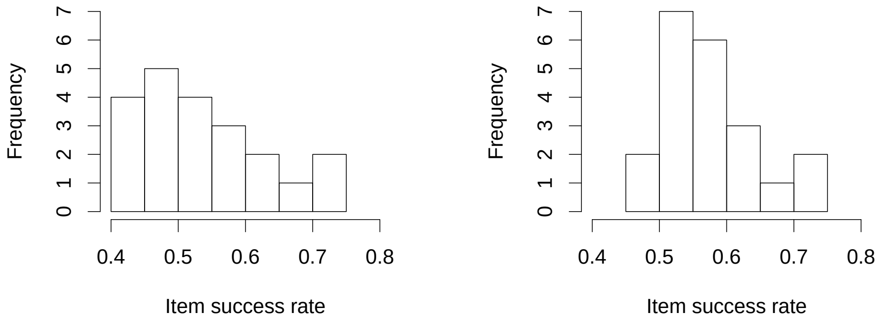
Then, to determine which skill vector (column) of the **W** matrix corresponds to the skill vector of the **Q** matrix, a correlation matrix is computed and the highest correlation of each column vector **W** is in turn matched with the corresponding unmatched column in **Q**.

We will use visual representations of the raw and the "discretized" (clustered) **W** matrix to provide an intuitive view of the results, as well as a quantitative measures of the fit corresponding to the average of the correlations between the matched skills vectors **W** and **Q**.

6 Results

In order for the mean and variance of the simulated data to reflect realistic values of test data, the skill difficulty and examinee ability parameters are adjusted such that the average success rate is close to 60%. Examinee ability is combined with the skill difficulty vectors to create a probability matrix of the same dimensions as **S**, from which **S** is obtained. Figure 3(a) displays a histogram of the 21 items success rate of the *ideal* response patterns for a sample of 2000 examinees, which is generated according to equation (1). Figure 3(b) shows the item success rates after the data is transformed by the application of *slip* and *guess* transformations. This transformation will generally decrease the spread of the distribution.

Figure 4(a) shows a heat map of the matrix **W** inferred from an *ideal* response pattern of 200 simulated examinees. Skill difficulties were set at (0.17, 0.30, 0.43,



(a) Histogram of the item success rates for the *ideal* response pattern.

(b) Histogram of the item success rates after the *slip* and *guess* factors.

Fig. 3. Histogram of item success rates

0.57, 0.70, 0.83) and examinee mean ability and standard deviation respectively at 0 and 0.5. The discetized version of figure 4(a)'s matrix is shown in figure 4(b) and it is identical to the underlying matrix **Q** in figure 2(a).

Figure 4(c) and 4(d) shows the effect of adding *slip* and *guess* parameters of 0.2 for each. The mapping to the underlying matrix **Q** degrades as expected, but remains relatively accurate.

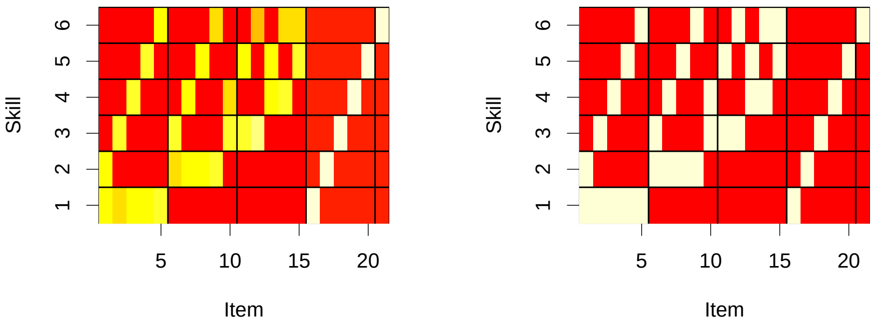
Table 1 reports the results of the quantitative comparison between the **Q** matrix and the **W** matrix inferred as a function of different *slip* and *guess* parameters. These results are based on 10-fold simulations. The mean of the Pearson correlation coeffficient ($\bar{r}$) between **Q** and **W** is reported for the discretized version of **W** obtained with the clustering algorithm described in section 5. In addition, the error rate as computed by this formula is also provided:

$$Err = \frac{\sum_{ij} |w_{ij} - q_{ij}|}{2 \cdot \sum_{ij} |q_{ij}|} \tag{5}$$

Where w_{ij} and q_{ij} are respectively the (i, j) cells of the matrices **W** and **Q**. The error rate will be 0 for a perfectly matched **Q** and 1 when no cells match. A value of 0.5 indicates that half of the non-zero cells are correctly matched. For the matrix **Q**, the error rate of a random assignment of the 36 skills is 69%.

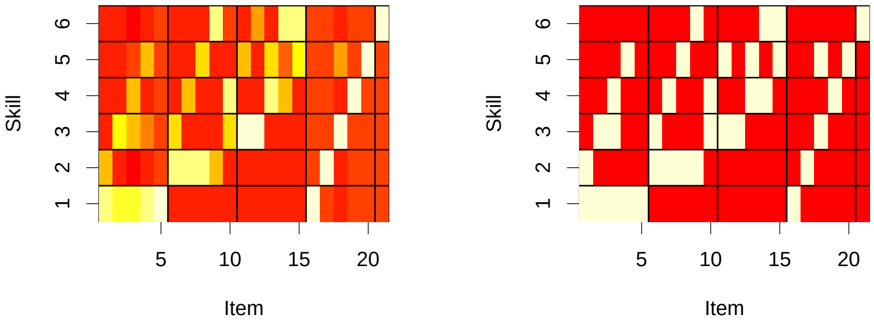
The 0 *slip* and 0 *guess* condition (first line) correspond to figures 4(a) and 4(b), whereas the corresponding 0.2–0.2 condition (line 3) correspond to figures 4(c) and 4(d).

Up to the 0.2–0.2 slip-guess condition, the skill mapping stays relatively close to perfect. On average, approximately only 2 or 3 skills requirements are wrongly assigned out of the 36 skills requirements (7%) at the 0.2–0.2 condition. However,



(a) Matrix **W** without slip and guess factors ($r = 1$).

(b) Discretized **W** without slip and guess factors ($r = 1$).



(c) Matrix **W** with slip and guess factors of 0.2 ($r = 0.91$).

(d) Discretized **W** for slip and guess of 0.2 ($r = 0.93$). Four out of 36 skill requirements are incorrectly mapped in this example.

Fig. 4. Visual representations of the original **Q** matrix and NMF inferred matrices **W**. The correlation reported (r) is computed by a comparison with the theoretical (real) matrix as explained in the text.

Table 1. Quantitative comparison between original **Q** matrix and NMF inferred matrices **W**. Results are based on means and standard deviation over 10 simulation runs.

Slip	**Guess**	$\bar{r}$	**sd(**$\bar{r}$**)**	**Err**	**sd(Err)**
0.00	0.00	1.00	0.00	0.00	0.00
0.20	0.10	0.97	0.03	0.02	0.02
0.20	0.20	0.90	0.06	0.07	0.04
0.20	0.30	0.63	0.08	0.26	0.06
0.20	0.40	0.49	0.07	0.36	0.06

the error rate increases substantially at the 0.3–0.2 slip-guess condition, and at the 0.2–0.4 condition, the quality of the match degrades considerably, with an average of 13/36 wrong assignements (36%).

7 Conclusion

The proposed approach to infer a conjunctive Q-matrix from simulated data with NMF is successful but, as we can expect, it degrades with the amount of *slips* and *guesses*. If the conjunctive Q-matrix contains one or two items per skill and the noise in the data remains below slip and guess factors of 0.2, the approach successfully derives the Q-matrix with very few mismatches of items to skills. However, once the data has slip and guess factors of 0.2 and 0.3, then the performance starts to degrade rapidly.

Of course, with a slip factor of 0.2 and a guess factor 0.3, about 25% of the values in the results become inconsistent with the Q-matrix. A substantial degradation is therefore not surprising. But in this experiment with simulated data, we have a number of advantages that are lost with real data: the number of skills is known in advance, no item has more than two conjunctive skills, skills are independent, and surely other factors will arise to make real data more complex. Therefore, we can expect that even if real data does not have a 50% rate of inconsistent results with the examinees' skills mastery profile, other factors might make the induction of the Q-matrix subject to errors of this scale.

Further studies with real and simulated data are clearly needed. For example, we would like to know what is the mapping accuracy degradation when an incorrect number of skills are modelled. And, naturally, a study with real data is necessary to establish if the approach is reliable in practice.

References

1. T. Barnes. The Q-matrix method: Mining student response data for knowledge. In *AAAI'2005 Workshop on Educational Data Mining*, pages technical report WS–05–02, 2005.
2. Michael W. Berry, Murray Browne, Amy N. Langville, V. Paul Pauca, and Robert J. Plemmons. Algorithms and applications for approximate nonnegative

matrix factorization. *Computational Statistics & Data Analysis*, 52(1):155 – 173, 2007.

3. Michel C. Desmarais. Conditions for effectively deriving a q-matrix from data with non-negative matrix factorization. In Cristina Conati, Sebastian Ventura, Toon Calders, and Mykola Pechenizkiy, editors, *4th International Conference on Educational Data Mining, EDM 2011*, pages 41–50, Eindhoven, Netherlands, June 6–8 2011.
4. Michel C. Desmarais and Ildikó Pelczer. On the faithfulness of simulated student performance data. In Ryan Shaun Joazeiro de Baker, Agathe Merceron, and Philip Jr I. Pavlik, editors, *3rd International Conference on Educational Data Mining EDM2010*, pages 21–30, Pittsburgh, PA, USA, June 11–13 2010.
5. Mingyu Feng, Neil T. Heffernan, Cristina Heffernan, and Murali Mani. Using mixed-effects modeling to analyze different grain-sized skill models in an intelligent tutoring system. *IEEE Transactions on Learning Technologies*, 2:79–92, 2009.
6. Renaud Gaujoux and Cathal Seoighe. A flexible R package for nonnegative matrix factorization. *BMC Bioinformatics*, 11(367):`http://www.biomedcentral.com/1471--2105/11/367`, 2010.
7. K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8:30–43, 1997.
8. Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.
9. K. K. Tatsuoka. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20:345–354, 1983.
10. K.K. Tatsuoka. *Cognitive Assessment: An Introduction to the Rule Space Method*. Routledge Academic, 2009.
11. Titus Winters, Christian R. Shelton, and Tom Payne. Investigating generative factors of score matrices. In Rosemary Luckin, Kenneth R. Koedinger, and Jim E. Greer, editors, *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*, volume 158 of *Frontiers in Artificial Intelligence and Applications*, pages 479–486. IOS Press, 2007.