Sparse Factor Autoencoders for Item Response Theory

Benjamin Paaßen
German Research Center for
Artificial Intelligence
Berlin, Germany
benjamin.paassen@dfki.de

Malwina Dywel
Provadis Partner für Bildung
und Beratung GmbH
Frankfurt a.M., Germany

Melanie Fleckenstein
Provadis Partner für Bildung
und Beratung GmbH
Frankfurt a.M., Germany

Niels Pinkwart
Insitute of Computer Science
Humboldt-University of Berlin
Berlin, Germany

ABSTRACT

Item response theory (IRT) is a popular method to infer student abilities and item difficulties from observed test responses. However, IRT struggles with two challenges: How to map items to skills if multiple skills are present? And how to infer the ability of new students that have not been part of the training data? Inspired by recent advances in variational autoencoders for IRT, we propose a novel method to tackle both challenges: The Sparse Factor Autoencoder (SparFAE). SparFAE maps from test responses to abilities via a linear operator and from abilities to test responses via an IRT model. All parameters of the model offer an interpretation and can be learned in an efficient manner. In experiments on synthetic and real data, we show that SparFAE is similar in accuracy to other autoencoder approaches while being faster to learn and more accurate in recovering item-skill associations.

Keywords

item response theory, logistic models, variational autoencoder, sparse factor analysis

1. INTRODUCTION

A foundational problem in educational data mining is to automatically infer students' ability from their observed responses in a test. Item response theory (IRT) addresses this problem by fitting a logistic model that describes how student ability and item difficulty interact to generate an observed response [5]. However, IRT faces at least two challenges. First, whenever a test involves multiple skills, we need to model the relation between skills and items, which standard IRT does not do [10]. Second, an IRT model contains student-specific parameters which are fitted to a specific population. For any new student, we need to fit at least one new parameter. The former challenge can be addressed via automatic methods for item-skill association learning, such as the $\boldsymbol{Q}$-matrix method of Barnes et al. [2], the alternating least squares method [12], or sparse factor analysis [7]. The second challenge requires a student-independent parametrization of the model, which is offered by variants like performance factor analysis [11] or variational autoencoders [3]. In the present paper, we propose to address both challenges at once by combining sparse factor analysis with autoencoders, yielding a new method which we call sparse factor autoencoder (SparFAE).

In more detail, our contributions are: We introduce SparFAE, a sparse factor autoencoding method for IRT. We provide an interpretation for all parameters in the SparFAE model, as well as an efficient learning scheme. Further, we empirically compare SparFAE to sparse factor analysis [7] as well as variational autoencoders [16] on synthetic and real data and show that SparFAE is similar in accuracy to other encoders but is much faster to learn and more accurate in recovering item-skill associations. Finally, we use SparFAE to analyze an expert-designed math test and verify the identified $\boldsymbol{Q}$-matrix against the expert-designed $\boldsymbol{Q}$-matrix. The source code for all experiments can be found at `https://github.com/bpaassen/sparfae`.

2. BACKGROUND AND RELATED WORK

IRT models the responses of m students on a test with n items. In particular, let $y_{i,j}$ be a random variable, which is 1 if student i answered item j correctly and 0, otherwise. We assume that $y_{i,j}$ is Bernoulli-distributed, where the success probability is given as $p_{i,j} = \sigma(\theta_i - b_j)$, where $\sigma(x) = 1/(1 + \exp[-x])$ is the logistic link function, θ_i is an ability parameter for student i, and b_j is a difficulty parameter for item j [5]. The parameters θ_i and b_j need to be fitted to observed training data, for example, via likelihood maximization or Bayesian parameter estimation [1]. In particular, the negative log likelihood of the data (also known as crossentropy loss) is expressed by the formula

$$\ell = \sum_{i=1}^{m} \sum_{j=1}^{n} -y_{i,j} \cdot \log[p_{i,j}] - (1 - y_{i,j}) \cdot \log[1 - p_{i,j}]. \quad (1)$$

This loss is convex in the parameters θ_i and b_j, meaning that an optimal model can be found efficiently via nonlinear optimization algorithms.

B. Paaßen, M. Dywel, M. Fleckenstein, and N. Pinkwart. Sparse factor autoencoders for item response theory. In A. Mitrovic and N. Bosch, editors, *Proceedings of the 15th International Conference on Educational Data Mining*, pages 17–26, Durham, United Kingdom, July 2022. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.6853067

Over the decades, numerous extensions of this basic scheme have been proposed, such as a discrimination parameter for each item (two-parameter model), a minimum probability of correct answers for each item (three-parameter model), partial credit, and hierarchical models [1, 4, 5]. In this paper, we care particularly about the extension to multiple underlying skills, sometimes called multidimensional IRT [10]. In such a model, we represent a student's ability by a K-dimensional vector $\vec{\theta}_i$, where $\theta_{i,k}$ models the ability of student i for skill k. A consequence of including multiple skills is that we need to model the relationship between skills and items. In this paper, we assume a linear relationship that is captured by an $n \times K$ matrix $\boldsymbol{Q}$, where $q_{j,k}$ models how important skill k is to answer item j correctly. Overall, our model is described by the two equations:

$$p_{i,j} = \sigma(z_{i,j}) \qquad \text{and} \qquad \vec{z}_i = \boldsymbol{Q} \cdot \vec{\theta}_i - \vec{b}, \tag{2}$$

where $\vec{z}_i$ is the vector of response logits for student i, and $\vec{b}$ is the vector of all item difficulties.

Our setup begs the question: how to learn the matrix $\boldsymbol{Q}$? Such coupling matrices between items and skills have been popularized by Tatsuoka [13], who imposed $q_{j,k} = 1$ if skill k is required for item j and $q_{j,k} = 0$, otherwise. Traditionally, such $\boldsymbol{Q}$-matrices have been hand-designed by domain experts [9], but recently, automatic methods to learn $\boldsymbol{Q}$ have emerged, such as the method of Barnes [2] or the alternating recursive method [12]. Crucially, finding an optimal binary $\boldsymbol{Q}$-matrix is challenging due to the discrete search space. To simplify the search, Lan et al. [7] have relaxed the problem by assuming continuous, non-negative entries of $\boldsymbol{Q}$ and applying methods from sparse coding, resulting in a method called Sparse Factor Analysis (SPARFA).

SPARFA applies an alternating optimization scheme. First, we initialize student abilities $\vec{\theta}_i$ randomly, for example with Gaussian noise. Second, for each item j, we adapt the jth row of $\boldsymbol{Q}$ and the difficulty b_j by solving the following optimization problem:

$$\begin{aligned} \min_{\vec{q}_j, b_j} \quad & \ell + \lambda_1 \cdot \|\vec{q}_j\|_1 + \lambda_2 \cdot (\|\vec{q}_j\|_2^2 + b_j^2) \\ \text{s.t.} \quad & q_{j,k} \geq 0 \qquad \forall k \in \{1, \ldots, K\}, \end{aligned} \tag{3}$$

where ℓ is the crossentropy loss (1), $\|\vec{q}_j\|_1 = \sum_{k=1}^K |q_{j,k}|$ is the 1-norm of $\vec{q}_j$, $\|\vec{q}_j\|_2^2 = \sum_{k=1}^K q_{j,k}^2$ is the squared Euclidean norm of $\vec{q}_j$, and λ_1 as well as λ_2 are hyperparameters of the method. The squared Euclidean norm is intended to regularize the model parameters with a Gaussian prior, as usual in IRT [1] (chapter 7). The 1-norm is motivated by sparse coding and encourages sparsity in $\boldsymbol{Q}$, meaning that the optimization process tends to find solutions where many of the entries in $\boldsymbol{Q}$ are zero [17]. In other words, the model is encouraged to connect any item j only to a few skills instead of all skills. This is reminiscent of traditional $\boldsymbol{Q}$-matrices, where $q_{j,k}$ is only nonzero if skill k is required to answer item j correctly [13]. Finally, SPARFA enforces that no entry $q_{j,k}$ can become negative, because a negative $q_{j,k}$ would imply that a higher ability in skill k *reduces* my chance to answer item j correctly, which does not make sense [7]. Note that problem (3) is convex, such that it can be solved efficiently with nonlinear optimizers.

The third step of SPARFA is to optimize the ability parameters $\vec{\theta}_i$ for each student i. This is done by minimizing the crossentropy (1) plus a regularization term $\lambda_2 \cdot \sum_{k=1}^K \theta_{i,k}^2$. We now iterate steps two and three of the SPARFA algorithm until the parameters converge.

Just as in standard IRT, a challenge of SPARFA is that we can not immediately apply a learned model to new students. For every new student i, we need to fit new parameters $\vec{\theta}_i$. Many extensions of IRT have circumvented this problem by removing ability parameters altogether and only using item parameters. For example, performance factor analysis replaces the ability parameter by a weighted count of correct and wrong responses on past items for the same skill [11]. More recently, Converse et al. [3] proposed a variational autoencoder model to simplify the application of IRT models to new students.

A variational autoencoder (VAE) views the student abilities $\vec{\theta}_i$ as a compressed representation of the student's response vector $\vec{y}_i$. More precisely, a VAE tries to learn an encoder function which compresses $\vec{y}_i$ to abilities $\vec{\theta}_i$, and a decoder function which de-compresses $\vec{\theta}_i$ back into estimated responses $\hat{y}_i$, such that $\vec{y}_i$ and $\hat{y}_i$ are close and such that $\vec{\theta}_i$ is standard normal distributed [6]. As decoder, we use a multi-dimensional IRT model (2), whereas the encoder could be a multi-layer artificial neural network [3]. In contrast to traditional IRT models, a VAE model is typically non-convex and multi-layered, and thus needs to be optimized with deep learning methods [3, 6]. Wu et al. [16] have further extended the VAE version of IRT by analyzing the theory more closely and including the difficulty parameters $\vec{b}$ as an additional input to the encoder. Fig. 1 illustrates the approach for a single-layer encoder. The encoder is given as $\vec{\theta}_i = \boldsymbol{A} \cdot \vec{y}_i + \boldsymbol{B} \cdot \vec{b} + \vec{\gamma}$ for some bias $\vec{\gamma}$ (Fig. 1, left, in orange), whereas the decoder is a multi-dimensional IRT model like in (2) (Fig. 1, right, in blue). Note that we obtain all models in this section as special cases of this diagram. If we set the connections $\boldsymbol{B}$ to zero, we obtain the IRT-VAE of [3]. If we, further, remove the connections $\boldsymbol{A}$ and treat $\vec{\theta}_i$ as parameters, we obtain SPARFA. Finally, if we set $K = 1$ and $q_{j,1} = 1$ for all j, we obtain a classic IRT model.

Interestingly, the state-of-the-art VAE approaches do not apply a sparsity penalty to facilitate interpretability of $\boldsymbol{Q}$. Further, deep learning can be quite slow. To address these limitations, we propose an autoencoder model based on the SPARFA loss, which we describe in the next section.

3. METHOD

Our proposed model is a single-layer autoencoder as illustrated in Fig. 1. More formally, our model can be concisely expressed in the following equations:

$$\begin{aligned} \vec{\theta}_i &= \boldsymbol{A} \cdot \vec{y}_i, \\ \vec{z}_i &= \boldsymbol{Q} \cdot \vec{\theta}_i - \vec{b}, \quad \text{and} \\ p_{i,j} &= 1/\big(1 + \exp(-z_{i,j})\big), \end{aligned} \tag{4}$$

where the first equation expresses the encoder and the second and third equation the decoder.

Our interpretation of the parameters is as follows. $\boldsymbol{A}$ maps

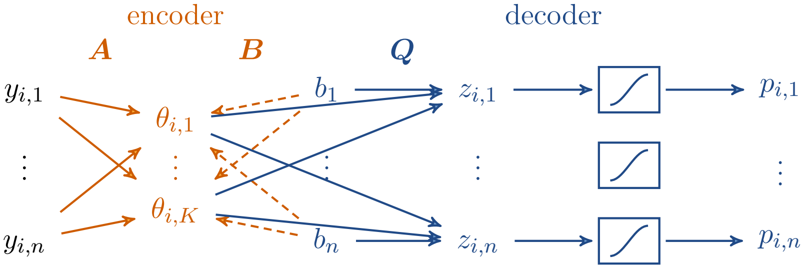

Figure 1: A sketch of a one-layer autoencoder for student responses, following Algorithm 1 of [16]. The Encoder is shown in orange (left), the decoder in blue (right). Encoder bias parameters are not shown for simplicity.

from responses to student ability, with $\alpha_{k,j}$ modeling the amount of ability in skill k that is expressed by answering item j correctly. Conversely, $\boldsymbol{Q}$ maps from abilities to responses, with $q_{j,k}$ modeling how much skill k helps to answer item j correctly. b_j models the difficulty of item j, as before. Note that our model requires no student-specific parameters, such that it can be directly applied to new students.

Note that we do not include "backward" connections $\boldsymbol{B}$ or encoder bias parameters $\vec{\gamma}$ in our model because they do not contribute to the model's expressive power in the single-layer case. Consider a "full" model with $\vec{\theta}_i = \boldsymbol{A} \cdot \vec{y}_i + \boldsymbol{B} \cdot \vec{b} + \vec{\gamma}$. If we plug this expression into our equation for z_i, we obtain

$$\vec{z}_i = \boldsymbol{Q} \cdot \Big(\boldsymbol{A} \cdot \vec{y}_i + \boldsymbol{B} \cdot \vec{b} + \vec{\gamma} \Big) - \vec{b} = \boldsymbol{Q} \cdot \boldsymbol{A} \cdot \vec{y}_i + \boldsymbol{Q} \cdot \boldsymbol{B} \cdot \vec{b} + \boldsymbol{Q} \cdot \vec{\gamma} - \vec{b}.$$

We now absorb $\boldsymbol{B}$ and $\vec{\gamma}$ into $\vec{b}$ by re-defining $\vec{b}$ as $\boldsymbol{Q} \cdot \boldsymbol{B} \cdot \vec{b} + \boldsymbol{Q} \cdot \vec{\gamma} - \vec{b}$, yielding Equations (4). Accordingly, our model requires only $2K + 1$ parameters per item.

We can train the parameters of our model by solving the following minimization problem, inspired by SPARFA.

$$\begin{aligned} \min_{\boldsymbol{A},\boldsymbol{Q},\vec{b}} \sum_{i=1}^{m} \sum_{j=1}^{n} & -y_{i,j} \cdot \log[p_{i,j}] - (1 - y_{i,j}) \cdot \log[1 - p_{i,j}] \quad (5) \\ & + \lambda_1 \cdot (\|\boldsymbol{A}\|_{1,1} + \|\boldsymbol{Q}\|_{1,1}) + \frac{\lambda_2}{2} \cdot (\|\boldsymbol{A}\|_F^2 + \|\boldsymbol{Q}\|_F^2 + \|\vec{b}\|^2) \\ & \text{s.t. } \alpha_{k,j} \geq 0, q_{j,k} \geq 0 \quad \forall k \in \{1, \ldots, K\}, j \in \{1, \ldots, n\} \end{aligned}$$

where $\|\boldsymbol{A}\|_{1,1} = \sum_{j=1}^{n} \sum_{k=1}^{K} |\alpha_{k,j}|$ denotes the entry-wise 1-norm, and where $\|\boldsymbol{A}\|_F^2 = \sum_{j=1}^{n} \sum_{k=1}^{K} \alpha_{k,j}^2$ denotes the squared Frobenius norm. Since the resulting model is an autoencoder-variant of Sparse Factor Analysis, we call it Sparse Factor Autoencoder (SparFAE). We denote the objective function as ℓ_{SparFAE}. As in SPARFA, the Frobenius norm applies a Gaussian prior on the parameters, whereas the 1-norm encourages sparsity. We also apply the same non-negativity constraints as in SPARFA to ensure a meaningful interpretation of $\boldsymbol{A}$ and $\boldsymbol{Q}$. Additionally, the non-negativity constraints are likely to further enhance sparsity, as indicated by non-negative matrix factorization [8].

In contrast to SPARFA, we can *not* decompose this problem into independent problems for each item because there are item-to-item-dependencies: Manipulating $\alpha_{k,j}$ also influences the abilities $\theta_{i,k}$, which in turn influence the probability $p_{i,j'}$ for any item j' with $q_{j',k} \neq 0$. Accordingly, we need to perform a joint optimization of all parameters. However, we do not need to resort to deep learning. Instead, we propose a standard L-BFGS solver, as implemented in the minimize method of scipy [14]. This is facilitated by the surprisingly simple expression for the gradients:

$$\begin{aligned} \nabla_{\boldsymbol{A}} \ell_{\text{SparFAE}} &= \boldsymbol{Q}^T \cdot \boldsymbol{\Delta}^T \cdot \boldsymbol{Y} + \lambda_1 \cdot \boldsymbol{1} + \lambda_2 \cdot \boldsymbol{A}, \\ \nabla_{\boldsymbol{Q}} \ell_{\text{SparFAE}} &= \boldsymbol{\Delta}^T \cdot \boldsymbol{Y} \cdot \boldsymbol{A}^T + \lambda_1 \cdot \boldsymbol{1}^T + \lambda_2 \cdot \boldsymbol{Q}, \text{ and} \\ \nabla_{\vec{b}} \ell_{\text{SparFAE}} &= -\vec{1}^T \cdot \boldsymbol{\Delta} + \lambda_2 \cdot \vec{b}, \end{aligned} \tag{6}$$

where $\boldsymbol{Y}$ is the $m \times n$ matrix of all responses, where $\boldsymbol{\Delta}$ is the $m \times n$ matrix with entries $\delta_{i,j} = p_{i,j} - y_{i,j}$, where $\boldsymbol{1}$ is the $K \times n$ matrix of only ones, and where $\vec{1}$ is an m-dimensional vector of ones. Regarding computational complexity, notice that the matrix products in (6) require $\min\{2 \cdot K \cdot m \cdot n, n^2 \cdot (m + K)\}$ operations, such that each optimization step is in $\mathcal{O}(m \cdot n)$ for constant K. We can simplify our optimization further by inspecting the relationship between $\boldsymbol{A}$ and $\boldsymbol{Q}$.

3.1 Single Matrix Variant

Note that the matrices $\boldsymbol{A}$ and $\boldsymbol{Q}$ have related interpretations. Intuitively, if skill k helps more with item j (high $q_{j,k}$), we would also expect that answering item j correctly is an indicator for skill k (high $\alpha_{k,j}$). Accordingly, it stands to reason that $\boldsymbol{A} = \boldsymbol{Q}^T$.

We can also motivate this setting mathematically. In particular, $\boldsymbol{A} = \boldsymbol{Q}^T$ is optimal if $\boldsymbol{Q}$ is orthogonal, meaning $\boldsymbol{Q}^T \cdot \boldsymbol{Q}$ equals the identity matrix $\boldsymbol{I}$. In that case, $\boldsymbol{Q} \cdot \boldsymbol{Q}^T \cdot \vec{y}_i$ is the orthogonal projection of $\vec{y}_i$ onto the hyperplane spanned by $\boldsymbol{Q}$. In other words, $\boldsymbol{Q} \cdot \boldsymbol{Q}^T \cdot \vec{y}_i$ is the most similar point to $\vec{y}_i$ we can achieve with the decoder $\boldsymbol{Q}$.

However, is it plausible that $\boldsymbol{Q}$ is orthogonal? Indeed, $\boldsymbol{Q}^T \cdot \boldsymbol{Q}$ becomes a diagonal matrix (orthogonal up to scaling) if every item tests exactly one skill. Let J_k be the set of items which test skill k. Then, we obtain: $(\boldsymbol{Q}^T \cdot \boldsymbol{Q})_{k,l} = \sum_{j=1}^{n} q_{j,k} \cdot q_{j,l} = \sum_{j \in J_k}^{n} q_{j,k}^2$ along the diagonal and zero off the diagonal. In other words, the sparser $\boldsymbol{Q}$ becomes, the closer $\boldsymbol{A} = \boldsymbol{Q}^T$ is to optimal.

When we plug $\boldsymbol{A} = \boldsymbol{Q}^T$ into problem (5), we obtain:

$$\begin{aligned}
\min_{\boldsymbol{Q},\vec{b}} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} -y_{i,j} \cdot \log[p_{i,j}] - (1-y_{i,j}) \cdot \log[1-p_{i,j}] \\
& + \lambda_1 \cdot \|\boldsymbol{Q}\|_{1,1} + \frac{\lambda_2}{2} \cdot (\|\boldsymbol{Q}\|_F^2 + \|\vec{b}\|^2) \\
\text{s.t.} \quad & q_{j,k} \geq 0 \quad \forall\, k \in \{1,\ldots,K\}, j \in \{1,\ldots,n\}, \qquad (7)
\end{aligned}$$

where $\vec{z}_i = \boldsymbol{Q} \cdot \boldsymbol{Q}^T \cdot \vec{y}_i - \vec{b}$. The gradient becomes:

$$\nabla_{\boldsymbol{Q}} \ell_{\text{SparFAE}} = \big(\boldsymbol{Y}^T \cdot \boldsymbol{\Delta} + \boldsymbol{\Delta}^T \cdot \boldsymbol{Y} + \lambda_2 \cdot \boldsymbol{I}\big) \cdot \boldsymbol{Q} + \lambda_1 \cdot \boldsymbol{1}^T.$$

This concludes our description of the proposed method.

4. EXPERIMENTS

In our experiments, we evaluate our proposed approach, Sparse Factor Autoencoder (SparFAE), on both synthetic and real-world data. We compare Sparse Factor Analysis (SPARFA) [7], Variational item response theory with a novel lower bound (VIBO) [16], the two-matrix version of SparFAE (SparFAE2), as well as the single-matrix version (SparFAE1). As optimizers, we used L-BFGS for SPARFA and both SparFAE versions, and an Adam optimizer with learning rate 0.005, 100 epochs, and minibatch size 16 for VIBO (these settings are as similar as possible to the original work of [16]). The experimental source code with all details is available at `https://github.com/bpaassen/sparfae`.

4.1 Synthetic Experiments

First, we consider synthetic data, which we sample from a multivariate IRT model with $K = 5$ skills, standard normally distributed abilities $\theta_{i,k}$, and standard normally distributed difficulties b_j. We introduce two different sampling conditions for $\boldsymbol{Q}$: A) We sample a unique skill k for each item j and set $q_{j,k} = 1$, whereas all other entries of $\boldsymbol{Q}$ remain zero. B) We first sample a number of skills $K_j \in \{1,\ldots,5\}$ for each item j with probability $p(K_j) = \frac{6-K_j}{15}$. Then, we draw K_j skills k without replacement and uniform probability for item j and set $q_{j,k}$ to a uniform random number in the range $[0.5, 1]$.

As evaluation measures, we use the area under the receiver-operator-curve in predicting correct responses (AUC), the correlation between the learned difficulties b_j and the actual difficulties (r_b), the correlation between the learned abilities $\theta_{i,k}$ and the actual abilities (r_θ), fraction of agreeing nonzero entries between the learned $\boldsymbol{Q}$ matrix and the true $\boldsymbol{Q}$-matrix ($r_{\boldsymbol{Q}}$), the time needed for training, and the time needed for prediction on new students. Since the ordering of skills is undefined, we allow arbitrary permutations of the skills in the learned $\boldsymbol{Q}$-matrix before computing $r_{\boldsymbol{Q}}$. In practice, we re-order the columns of $\boldsymbol{Q}$ according to the `linear_sum_assignment` function of scipy with the ground-truth $\boldsymbol{Q}$ matrix [14]. We evaluate all measures on a separate sample of $m' = 100$ new students. We repeat all experiments 10 times for each of the hyperparameter settings in Table 1.

Table 1: Hyperparameter settings considered in the experiments.

setting	λ_{VAE}	λ_1	λ_2
1	10^{-5}	10^{-3}	10^{-3}
2	10^{-4}	0.05	10^{-3}
3	10^{-3}	1	10^{-3}
4	0.01	0.05	0.05
5	0.1	1	0.05
6	1	1	1

First, we inspect the effect of hyperparameters for $m = 100$ students and $n = 20$ items. Fig. 2 shows, from left to right, how AUC, r_b, r_θ, and $r_{\boldsymbol{Q}}$ change for higher regularization in conditions A (top) and B (bottom). For AUC, we observe a slight degradation of all methods for higher regularization, with a notable decline for VIBO at the last setting. r_b generally rises for higher regularization, with the exception of SparFAE1, which stays relatively stable around 0.5. r_θ appears stable across regularization and improves only for SPARFA. $r_{\boldsymbol{Q}}$ improves for all methods with higher regularization in condition A (top), and remains roughly stable in condition B (bottom). For the remaining synthetic experiments, we report the results using hyperparameter setting 6 for SparFAE2 and SparFAE1, and hyperparameter setting 5 for SPARFA and VIBO. These settings maximize r_b, r_θ, and $r_{\boldsymbol{Q}}$ while retaining high AUC.

Fig. 3 displays the performance measures for varying numbers of students. We observe that AUC, r_θ, and $r_{\boldsymbol{Q}}$ tend to slightly increase for more students across methods and conditions, with only slight deviances for small numbers of students. The most striking impact is on r_b, which increases for SparFAE1 and VIBO, but *de*creases for SPARFA and SparFAE2.

Fig. 4 displays the performance measures for varying numbers of items. Across methods, AUC decreases, whereas r_b and r_θ increase and $r_{\boldsymbol{Q}}$ remains roughly stable for higher number of items. The decrease in AUC is likely explained by the fact that the models need to compress the information of more items into the same number of skills, which is bound to decrease performance. Conversely, it becomes easier to tease apart the difficulty of each single item for a higher number of items per skill (hence the improvement in r_b). Further, the more items we have in a test, the more accurate we can estimate the underlying ability, which is reflected in better r_θ values.

Finally, Fig. 5 summarizes the effect of hyperparameter setting, number of students, and number of items on training time in logarithmic plots. We observe that stronger regularization reduces the training time for both SparFAE variants, whereas it stays roughly constant for VIBO and SPARFA. This is likely because training time for SPARFA and VIBO is driven by the repeated optimization steps over students, whereas the training time for SparFAE is dominated by a single optimization process. Hence, SparFAE profits more from the simpler loss surface offered by higher regularization. As one would expect, all methods scale roughly linearly with the number of students, SPARFA with roughly 18 ms per student, VIBO with roughly 6 ms per student, and both SparFAE variants with roughly 1.5 ms per student (refer to the gray dashed reference lines). For the num-

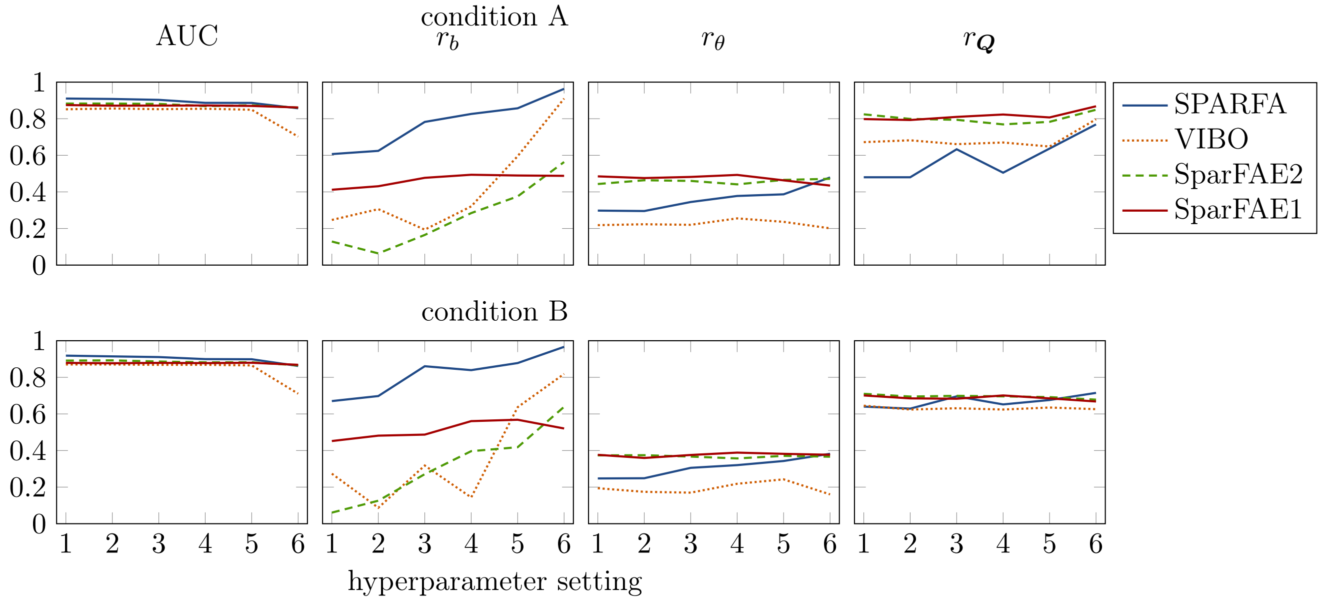

Figure 2: The effect of hyperparameter settings from Table 1 on various performance measures (from left to right) on the synthetic data, either with one skill per item (A, top), or multiple skills per item (B, bottom).

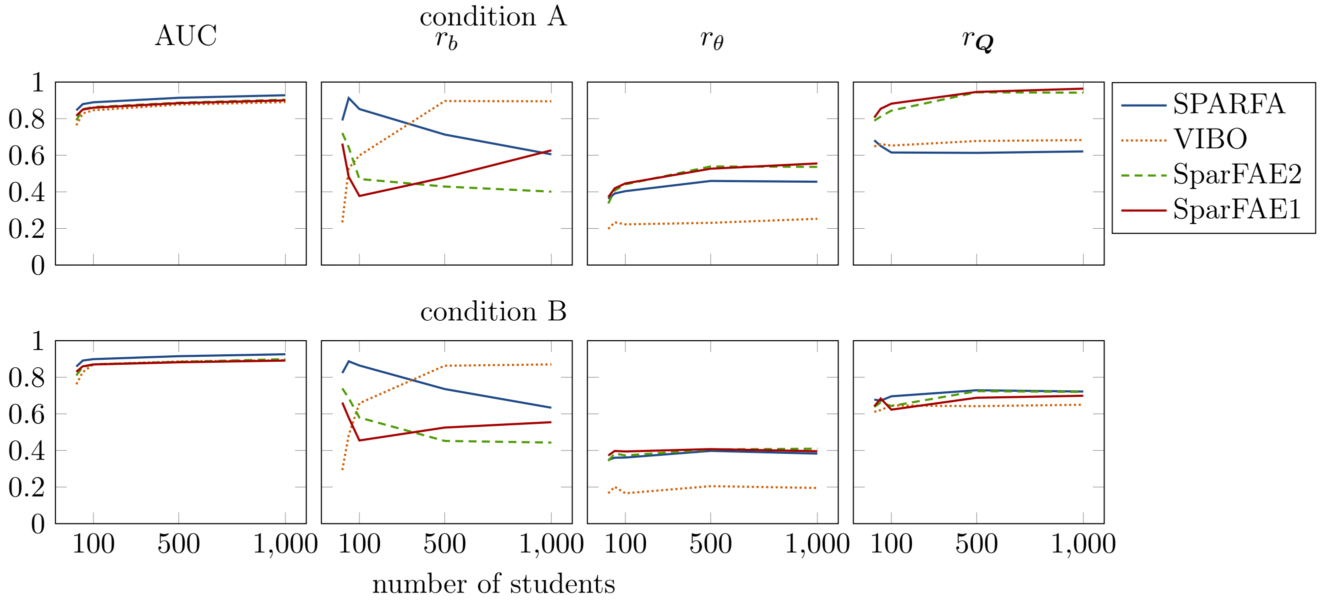

Figure 3: The effect of increasing the number of students on various performance measures (from left to right) on the synthetic data, either with one skill per item (A, top), or multiple skills per item (B, bottom).

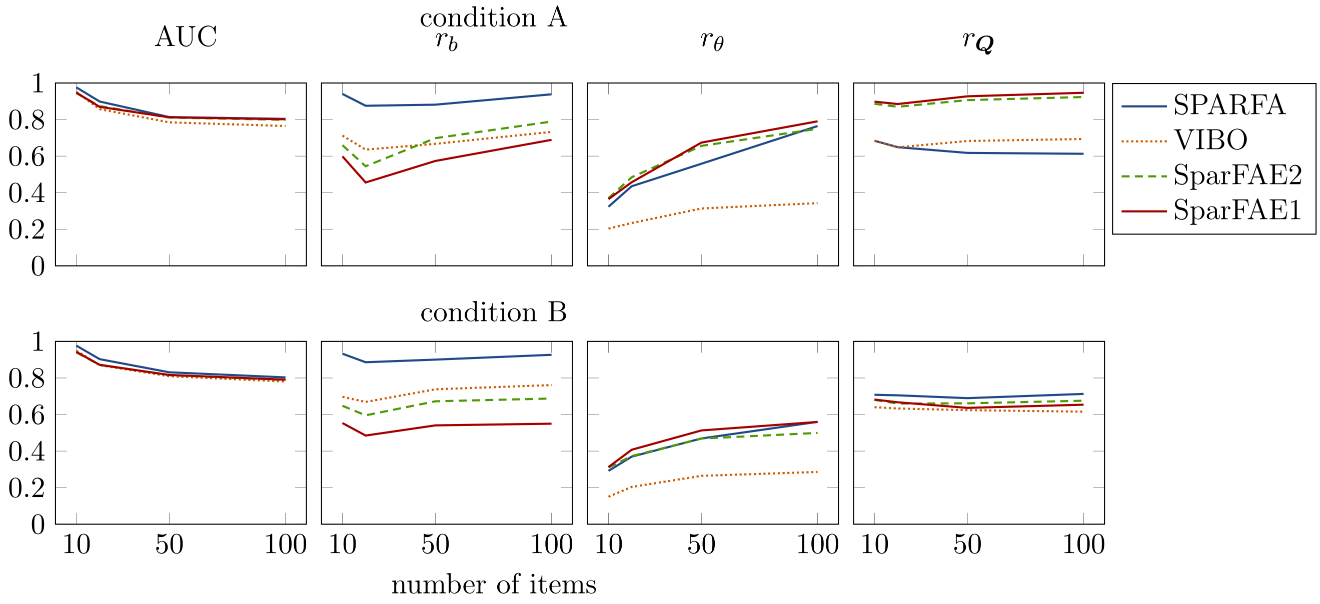

Figure 4: The effect of increasing the number of items on various performance measures (from left to right) on the synthetic data, either with one skill per item (A, top), or multiple skills per item (B, bottom).

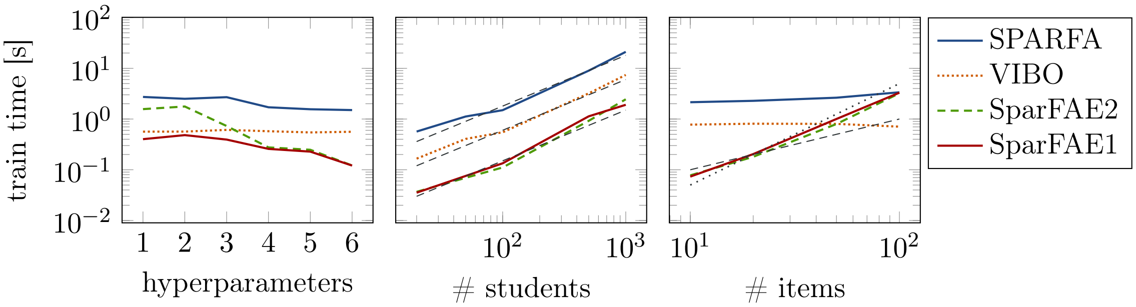

Figure 5: The effect of hyperparameter setting (left), number of students (center), and number of items (right) on training time for condition A. Gray dashed lines are linear references, the gray dotted line is a quadratic reference.

ber of items, the runtime for SPARFA and VIBO remains roughly constant, whereas it increases almost quadratically for both SparFAE variants. This is because the optimization for SPARFA and VIBO is dominated by iterations over students. By contrast, for SparFAE, every single gradient computation already depends linearly on n, and more items may also increase the number of required gradient computations until convergence, thus yielding the super-linear behavior.

4.2 NeurIPS 2020 education data

Next, we consider the NeurIPS 2020 education challenge data by Wang et al. [15]. The data set consists of multiple choice questions to assess mathematics knowledge. Items are grouped into different quizzes. We restricted the data set to the 948 items from task 3 of the challenge and quizzes with at least 50 students[1], which left 65 quizzes. On average, these quizzes contained 14.02 items and had 1675.06 students responding. To estimate the number of skills K, the first author analyzed all 948 items and assigned them to skills. This yielded 14 distinct skills, the most common ones being fractions (190 items), basic algebra (140 items), and algebra with variables (127 items). On average, quizzes involved 3.21 skills. For each quiz, we set K to the first-author estimate, but we upper-bounded K to be at most half the number of items in the quiz.

Based on the pre-defined $\boldsymbol{Q}$-matrix by the first author, we included two more baselines: A VIBO model, where we fixed the decoder matrix to the pre-defined $\boldsymbol{Q}$-matrix, and a SparFAE1 model, where we fixed the $\boldsymbol{Q}$-matrix and only trained the difficulty parameter for each item using logistic regression. We denote these methods as VIBO$_f$ and SparFAE$_f$, respectively.

To perform hyperparameter optimization, we randomly set aside 10 quizzes and evaluated the AUC of all methods for all hyperparameter settings in Table 1 in a 10-fold crossvalidation over students, that is, in each fold we used 90% of students as training data and 10% as test data. The hyperparameter settings which maximized AUC were 2 for SPARFA and SparFAE2, 3 for VIBO, and 4 for SparFAE1.

Next, we performed a 10-fold crossvalidation over students for the remaining 55 quizzes. Note that we can not evaluate r_b, r_θ, or $r_{\boldsymbol{Q}}$, because we have no access to a ground truth for b, θ, and $\boldsymbol{Q}$. However, we can evaluate the sparsity of $\boldsymbol{Q}$, that is, the fraction of zero entries. Sparsity is a rough proxy for the plausibility of a learned $\boldsymbol{Q}$-matrix because high sparsity indicates that $\boldsymbol{Q}$ assigns items to distinct skills.

Table 2 reports the average performance measures across quizzes. Regarding AUC, Wilcoxon signed-rank tests revealed that SPARFA had the highest AUC, followed by SparFAE2, VIBO, SparFAE1, SparFAE$_f$, and finally VIBO$_f$ ($p < 10^{-3}$ for all tests after Bonferroni correction). That being said, the AUC of all methods except SPARFA is very close (at most 2% difference between means). In terms of sparsity, SparFAE1 clearly outperforms SPARFA, VIBO, and SparFAE2 ($p < 10^{-3}$). Note that VIBO does not achieve any sparsity, as it does not encourage sparsity during training. The sparsity of the pre-defined $\boldsymbol{Q}$-matrix was very high (94%) as it assigned each item to only one skill.

With respect to training time, SparFAE1 is considerably faster than SPARFA (ca. 15x), VIBO (ca. 4x), and SparFAE2 (ca. 8x). In terms of prediction time, SparFAE1, SparFAE2, and VIBO perform similarly as their prediction scheme is almost the same (although SparFAE1 is still significantly faster, $p < 10^{-3}$). Only SPARFA is much slower (ca. 300x) because it needs to fit new ability parameters to new students for each prediction.

Finally, we analyzed the relation of AUC to the numbers of students, items, and skills, as well as the amount of missing data in quizzes. Fig. 6 displays scatter plots, where each dot represents one quiz and lines show linear fits. Interestingly, the behavior is very similar for all methods. The linear correlation is $r \approx 0.3$ with number of students ($r = 0.4$ for VIBO; $p < 0.05$), $r \approx -0.4$ with number of items ($p < 0.01$), $r \approx 0.6$ with number of skills ($p < 10^{-3}$), and around zero with the amount of missing data (insignificant). This is in line with our results on synthetic data. The strong correlation with the number of skills is explained by the fact the methods have more parameters to fit the data when we increase K.

4.3 Math assessment data

In a final experiment, we evaluated the ability of SparFAE1 to identify a fitting $\boldsymbol{Q}$-matrix in comparison to an expert-designed $\boldsymbol{Q}$-matrix on real data. To that end, we used data from $m = 30$ students (ages 16-19) on a math assessment test for vocational education in chemistry[2]. The test consisted of $n = 21$ questions, covering $K = 5$ topics, namely basic algebra, fractions, equation solving for a single variable, text tasks with two variables, and (linear) functions. Fig. 7 (top) shows the assignment of items (x-axis) to these five topics (y-axis) as provided by the test designers.

We applied a slightly adapted variant of SparFAE1 with the regularization $\sum_{k=1}^{K} \left(\sum_{j=1}^{n} q_{j,k} - 1 \right)^2$, that is, we punished deviations of the column sums from 1, thereby encouraging orthogonality in $\boldsymbol{Q}$. As regularization strength, we set 1. We performed 30 repeats of SparFAE1 and then selected the $\boldsymbol{Q}$-matrix which maximized accuracy in a leave-one-out crossvalidation over students (the resulting best accuracy was 89%).

The learned $\boldsymbol{Q}$-matrix is shown in Fig. 7 (bottom). We observe that the matrix assigns every item to only one skill, in line with the expert prediction. We further observe that—in line with the experts—the learned $\boldsymbol{Q}$ tends to group items for the basic topics (basic algebra and fractions) together and tends to avoid grouping items for basic topics with items for advanced topics. However, there are also notable differences to the expert $\boldsymbol{Q}$-matrix. In particular, SparFAE1 merges basic algebra and fractions into one skill (except for item 8, which is in skill 4), and includes items 13 and 14. Overall, skill 1 accumulates relatively easy tasks without text- and function components. All other skills contains items which required text comprehension and/or un-

[1] We also excluded one outlier quiz with more than 100 items and a lot of missing data.

[2] `https://projekte.provadis.de/showroom/provadis/Mathematik_Orientierungstest/online`

Table 2: Performance measures on the NeurIPS 2020 education data.

method	AUC	sparsity	training time [s]	prediction time [ms]
VIBO$_f$	0.88 ± 0.05	0.94 ± 0.00	8.01 ± 5.59	1.31 ± 2.76
SparFAE$_f$	0.88 ± 0.04	0.94 ± 0.00	0.05 ± 0.03	0.15 ± 0.13
SPARFA	$\mathbf{0.93 \pm 0.05}$	0.16 ± 0.06	31.0 ± 20.9	633 ± 444
VIBO	0.89 ± 0.05	0.00 ± 0.00	7.83 ± 5.12	0.31 ± 0.18
SparFAE2	0.90 ± 0.05	0.33 ± 0.10	15.7 ± 15.9	0.20 ± 0.13
SparFAE1	0.89 ± 0.04	$\mathbf{0.46 \pm 0.13}$	$\mathbf{1.94 \pm 1.78}$	$\mathbf{0.19 \pm 0.12}$

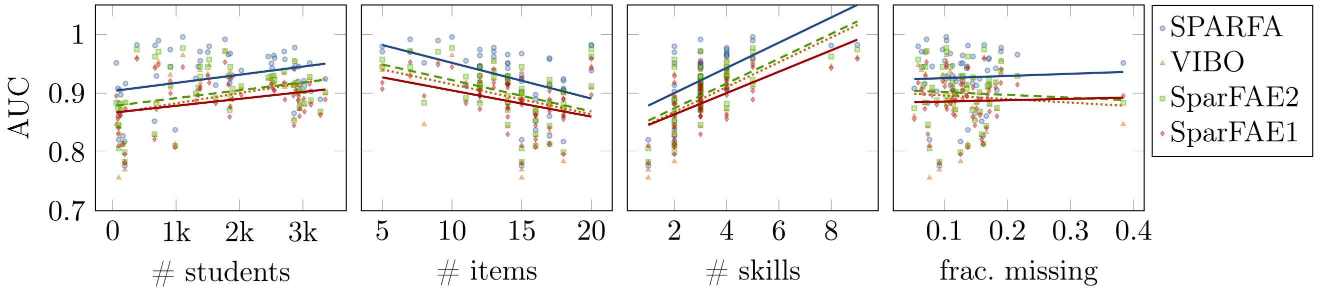

Figure 6: Scatter plots of AUC versus quiz metadata (from left to right: number of students, number of items, number of skills, and fraction of missing data). Lines indicate linear regression fits.

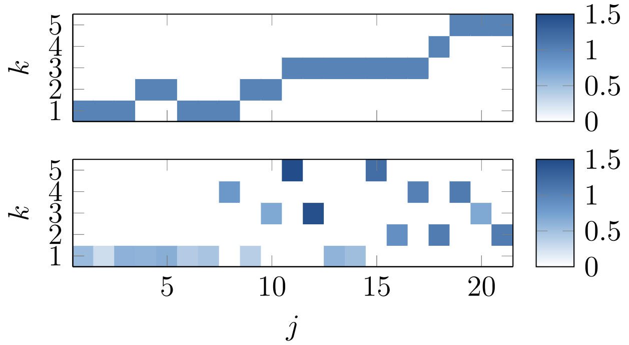

Figure 7: The expert-designed $\boldsymbol{Q}$-matrix (top), and the learned $\boldsymbol{Q}$-matrix via SparFAE1 (bottom) for the math assessment data.

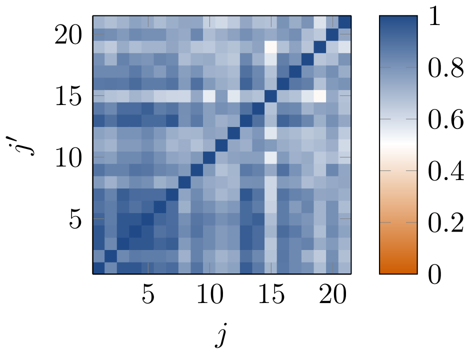

Figure 8: The item-to-item correlations for the math assessment data.

derstanding of functions, but the correspondence to expert-defined skills is less obvious.

To gain deeper insight into the learned $\boldsymbol{Q}$-matrix, we inspected the item-to-item correlations $c_{j,j'} = \sum_{i=1}^{m} x_{i,j} \cdot x_{i,j'} / \sqrt{\left(\sum_{i=1}^{m} x_{i,j}\right) \cdot \left(\sum_{i=1}^{m} x_{i,j'}\right)}$, which are shown in Fig. 8. We observe that items 1-7, 9, and 13-14 exhibit relatively high pairwise correlation, explaining why SparFAE1 grouped them together in skill 1.

Skill 2 groups items 16 and 18, which are both text problems covering variable solution problems, but it also includes item 21, which is a question on functions. Inspecting the correlation matrix, we observe that item 21 generally exhibits low correlation, except for items 16 and 18, which explains the grouping.

Skill 3 groups items without obvious mathematical connection. Item 10 is a fraction problem, item 12 is a variable algebra problem, and item 20 is a function problem. Further, these items exhibit only moderate pairwise correlation. However, the only items with higher correlations are already contained in skill 1 and are, thus, unavailable for skill 3, thus indirectly explaining the grouping.

Skill 4 contains a variable algebra item (8), an equation solving problem (17), and a function problem (19). General variable algebra capacity (8) plausibly enhances equation solving (17) but the function question (19) seems less connected. The correlation matrix reveals that item 19 has generally low correlations, except for items 3, 7, 14, and 17, explaining its grouping with item 17.

Skill 5 contains two equation problems, one symbolic (11) and one text-based (15). Further, item 15 has very low correlations with any other item, except for items 9 and 11, and 20, which explains the grouping with item 11.

Overall, we observe that the learned $\boldsymbol{Q}$-matrix tended to group more basic items together and more advanced items together, in line with expert opinion. Sometimes, the learned $\boldsymbol{Q}$ matrix groups items which do not have an obvious connection, content-wise. In such cases, we could explain the grouping by inspecting the item-to-item correlation matrix.

5. DISCUSSION AND CONCLUSION

We proposed a novel method for factor analysis which extends Sparse Factor Analysis (SPARFA) [7] to an autoencoder approach. Hence, we call our proposed method Sparse Factor Autoencoder (SparFAE). More specifically, our approach encodes student responses to abilities via a linear map $\boldsymbol{A}$ and decodes it again to predicted responses via a multi-dimensional item response theory model with a linear skill-to-item map $\boldsymbol{Q}$. Like SPARFA, our approach encourages sparsity in the $\boldsymbol{Q}$-matrix via non-negativity constraints and L1 regularization. In contrast to SPARFA, we do not need to fit new ability parameters for new students. Instead, we can simply apply $\boldsymbol{A}$, which automatically yields the desired ability parameters. We investigated two versions of SparFAE: One with separate matrices $\boldsymbol{A}$ and $\boldsymbol{Q}$ for encoding and decoding (SparFAE2), and one where we set $\boldsymbol{A} = \boldsymbol{Q}^T$, that is, we use the $\boldsymbol{Q}$-matrix for both encoding and decoding (SparFAE1).

In experiments on synthetic as well as real data, we showed that SparFAE1 is considerably faster than SPARFA, variational autoencoding [16], and SparFAE2. SparFAE1 also achieves higher sparsity, and higher correlation with ground truth $\boldsymbol{Q}$-matrices and student abilities. This comes at the price of slightly lower AUC and less accuracy in recovering ground truth difficulties. We also observed that AUC differences between autoencoder variants were quite small, whereas SPARFA achieved noticeably higher AUC, indicating that student-specific ability parameters allow for a better fit of the data than autoencoding. We also compared the learned $\boldsymbol{Q}$-matrix via SparFAE1 with an expert $\boldsymbol{Q}$ matrix on a math assessment test, revealing some overlap but also meaningful differences which could be explained by item-to-item correlations.

Overall, our results indicate that SparFAE1 is a promising method for fast factor analysis, especially when each item in a test only refers to a single skill. As such, we believe that it can be an interesting tool for test designers who wish to analyze the factor structure of their test on a sample of students. While the learned $\boldsymbol{Q}$-matrix should still be interpreted with care, it can uncover latent item relationships (as we saw on the math assessment data). Our results also motivate the use of $\boldsymbol{Q}$-matrices for both decoding *and* encoding, which can serve as a starting point for future research.

Limitations of SparFAE1 lie in the slightly lower AUC compared to other autoencoders, the ability to recover ground truth difficulty parameters, and the superlinear scaling with respect to the number of items. Future work could address each of these shortcomings. Further, our experimental evaluation is limited to multiple choice m math assessment questions. Future work should include further data sets from other educational domains to ensure that SparFAE1 generalizes. Finally, just as any autoencoders, SparFAE1 makes the assumption that abilities do not change during a test. Future work may consider more dynamic settings, e.g. by incorporating concepts from performance factor analysis or knowledge tracing models.

Acknowledgements

Funding by the German Ministry for Education and Research (BMBF) under grant number 21INVI1403 (project KIPerWeb) is greatfully acknowledged.

6. REFERENCES

[1] F. Baker and S.-H. Kim. *Item Response Theory: Parameter Estimation Techniques.* CRC Press, Boca Raton, FL, USA, 2 edition, 2004.

[2] T. Barnes. The $\boldsymbol{Q}$-matrix method: Mining student response data for knowledge. In *Proceedings of the AAAI 2005 Educational Data Mining Workshop*, pages 1–8, 2005.

[3] G. Converse, M. Curi, and S. Oliveira. Autoencoders for educational assessment. In S. Isotani, E. Millán, A. Ogan, P. Hastings, B. McLaren, and R. Luckin, editors, *Proceedings of the International Conference on Artificial Intelligence in Education (AIED)*, pages 41–45, 2019.

[4] S. Embretson and S. Reise. *Item response theory for psychologists.* Psychology Press, New York, NY, USA, 2000.

[5] R. Hambleton and H. Swaminathan. *Item response theory: Principles and applications.* Springer Science+Business Media, New York, NY, USA, 1985.

[6] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv*, 1312.6114, 2014.

[7] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk. Sparse factor analysis for learning and content analytics. *Journal of Machine Learning Research*, 15(57):1959–2008, 2014.

[8] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[9] R. Liu, A. C. Huggins-Manley, and L. Bradshaw. The impact of $\boldsymbol{Q}$-matrix designs on diagnostic classification accuracy in the presence of attribute hierarchies. *Educational and Psychological Measurement*, 77(2):220–240, 2017.

[10] R. P. McDonald. A basis for multidimensional item response theory. *Applied Psychological Measurement*, 24(2):99–114, 2000.

[11] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis –a new alternative to knowledge tracing. In *Proceedings of the International Conference on Artificial Intelligence in Education (AIED)*, page 531–538, 2009.

[12] Y. Sun, S. Ye, S. Inoue, and Y. Sun. Alternating recursive method for $\boldsymbol{Q}$-matrix learning. In *Proceedings of the 7th International Conference on Educational Data Mining (EDM)*, pages 14–20, 2017.

[13] K. K. Tatsuoka. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20(4):345–354, 1983.

[14] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[15] Z. Wang, A. Lamb, E. Saveliev, P. Cameron, Y. Zaykov, J. M. Hernández-Lobato, R. E. Turner, R. G. Baraniuk, C. Barton, S. P. Jones, S. Woodhead, and C. Zhang. Diagnostic questions: The NeurIPS 2020 education challenge. *arXiv*, 2007.12061, 2020.

[16] M. Wu, R. L. Davis, B. W. Domingue, C. Piech, and N. Goodman. Variational item response theory: Fast, accurate, and expressive. In A. Rafferty, J. Whitehill, V. Cavalli-Sforza, and C. Romero, editors, *Proceedings of the 13th International Conference on Educational Data Mining (EDM)*, pages 257–268, 2020.

[17] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.