Methods to Refine the Mapping of Items to Skills

Michel C. Desmarais and Peng Xu
Polytechnique Montréal

1 Introduction

A critical part of domain modeling in Intelligent Tutoring Systems is to determine how the domain content relates to skills and to knowledge that we aim the student to learn about. This task can be achieved through data driven techniques. If domain content is associated with a set of questions or exercises, student performance data over theses items can be used to find the latent skills behind the content.

Data driven techniques that map items to skills fall within two main categories: (1) entirely driven from student data performance, or (2) starting with an expert given mapping and refined based on this data. Entirely data driven techniques are appealing because they dispense the tedious efforts required to do the mapping by content experts. However, the mapping obtained from such methods may be hard to interpret and will almost surely contain latent skill factors that do not match the pedagogical structure of the learning content. For this reason, the refinement of expert given mappings has greater utility in most contexts and this chapter focuses on this specific problem.

Another fundamental distinction to be made is whether the data includes a time component. In learning environments, students learn as they interact with the system. A single student's skills mastery profile changes in time and within the same data sample. A number of studies have focused on refining the item to skills mapping with this type of dynamic data, namely Stamper and and Koedinger (2011), Koedinger et al.(2013), and Aleven and Koedinger (2013). Simplifying measures such as taking into account only the first attempt and ignoring hints and

scaffolding can alleviate the complexity of analyzing this type of data (see for eg. González-Brenes, 2015).

The modeling of task to KC/skills for models that include a time dimension, and from data that has a time dimension, is the focus of a previous volume's chapter of the current series (Aleven and Koedinger, 2013). In this chapter, we will focus on static data, where we assume the data is a snapshot in time of the student's skill profile and return in the discussion on the question of how this assumption can be dealt with.

2 Background concepts

Before discussing the item to skills refinement models, let us introduce the background concepts and models.

2.1 Q-matrices

The mapping of items to latent skills is often referred to as a *Q-matrix*. Rows of the matrix are the items and columns are the skills. Q-matrices are generally represented as Boolean matrices. The items can represent question, exercises, or any task that has a clear outcome. Skills are also termed Knowledge Components (KC) (Aleven & Koedinger, 2013).

There are three distinct ways how skills can be considered related to tasks:

(1) Conjunctive: all skills are required to successfully complete the task.
(2) Disjunctive: any skill is sufficient.
(3) Compensatory or additive: all skills contribute to increase the chances of success.

All three versions of Q-matrices obviously entail different student skills models that might be used in a running system. Furthermore, skills can

be continuous or discrete, which also entails different models. And so does the time factor. If learning occurs in the data, the models have to account for it.

However, we should emphasize that although the skills model depend on the type of Q-matrix and on whether the data has a time dimension, the Q-matrix does not. Whether a task involves skills or not is independent of the model, and of the time factor.

2.2 Factorization framework

A valuable framework for conceptualizing the relationship between a Q-matrix $\mathbf{Q}$, the student skill profiles matrix $\mathbf{P}$, and the student test outcome results matrix $\mathbf{R}$, is to define the relationship as a product:

$$\mathbf{R} = \mathbf{P}\mathbf{Q}^{\mathrm{T}} \tag{1}$$

For example, the following matrix product would correspond to a compensatory version of a Q-matrix in which the rows are normalized to sum to 1:

$$\mathbf{R} = \begin{array}{c} \\ r_1 \\ r_2 \\ r_3 \end{array} \begin{array}{c} \begin{array}{ccc} i_1 & i_2 & i_3 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1/2 \\ 1 & 1 & 1/2 \end{bmatrix} \end{array} = \begin{array}{c} \\ r_1 \\ r_2 \\ r_3 \end{array} \begin{array}{c} \begin{array}{cc} s_1 & s_2 \end{array} \\ \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \end{array} \times \begin{array}{c} \\ i_1 \\ i_2 \\ i_3 \end{array} \begin{array}{c} \begin{array}{cc} s_1 & s_2 \end{array} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1/2 & 1/2 \end{bmatrix} \end{array}^{\mathrm{T}}$$

Using the negation operator $\neg$ defined as:

$$\neg x = \begin{cases} 1 & \text{if} \quad x = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

we can define a conjunctive Q-matrix as the product (assuming a normalized Q-matrix):

$$\neg\mathbf{R} = \neg\mathbf{P} \times \mathbf{Q}^{\mathrm{T}} \tag{3}$$

and a disjunctive Q-matrix as:

$$\neg\mathbf{R} = \mathbf{P} \times \mathbf{Q}^{\mathrm{T}} \tag{4}$$

by redefining the first condition of the negation operator ($1 \quad \text{if} \quad x > 0$).

2.3 Cognitive diagnosis models and Q-matrix refinement

Every Q-matrix refinement model has an underlying model, often called a *cognitive diagnosis model*, or CDM. Common models can be described using the matrix factorization framework. For example, the DINA and DINO models described below.

DINA and DINO. DINA and DINO are two well known models for cognitive modeling. The DINA model relies on a conjunctive Q-matrix that corresponds to equation (3), whereas the DINO relies on a disjunctive Q-matrix that corresponds to equation (4). However, these models also include the *guess* and *slip* factors that respectively define the probability of a good response given a predicted 0 in the $\mathbf{R}$ matrix and a probability of an incorrect response given a predicted 1. An adaptation of the factorization framework to accommodate the uncertainty introduced in DINA and DINO is to consider the $\mathbf{R}$ matrix as a probability matrix and to substitute the 0's and the 1's respectively by the $guess$ and $(1-slip)$ factors.

Let us introduce a distinction in the notation and refer from here on to $\hat{\mathbf{R}}$ as the predicted student response outcome (for eg. the product of an estimated student profiles matrix, $\hat{\mathbf{P}}$, and a Q-matrix, $\mathbf{Q}$) and to $\mathbf{R}$ as an observed student response outcome matrix.

Many models consider the estimated student response outcome matrix, $\hat{\mathbf{R}}$, to be a probability matrix. For example, taking the logit of $\hat{\mathbf{R}}$ ($\log(r_{ij}/(1-r_{ij}))$ and the log of $\hat{\mathbf{P}}\mathbf{Q}^{\mathrm{T}}$ leads to flexible log-odds CDM (Hensen, Templin, & Willse, 2009)

But regardless of the specific model used for Q-matrix refinement, the principle of refinement follows a general framework. Given a cognitive diagnostic model and an estimated student profiles matrix $\hat{\mathbf{P}}$, the

refinement process can be considered as searching the space of Q-matrices for a matrix that will minimize the difference between $\mathbf{R}$ and $\hat{\mathbf{R}}$. The difference can be the RSS ($\|\mathbf{R}-\hat{\mathbf{R}}\|^2$), or any other reasonable loss function.

In fact, minimizing the loss function is the same objective whether we are looking at estimating the student profiles or any other parameter of the model. What is unique to the process of Q-matrix refinement is the initial starting point: the expert's matrix. If we were to look for a Q-matrix that solely optimizes prediction, we could instead start from a random Q-matrix or use heuristics to guess a strategic initial Q-matrix. In other words, the process looks for a local minima starting from a point in the Q-matrix space that corresponds to the expert Q-matrix.

3 Refinement Methods

The literature on Q-matrix refinement methods has exploded in recent years, both for static data (de la Torre & Chiu, 2015; Barnes, 2010; Desmarais & Naceur, 2013; Xiang, 2013; Chung, 2014; H. Li & Suen, 2013; Qin et al., 2015; Romero, Ordoñez, Ponsoda, & Revuelta, 2014; Köhn, Chiu, & Brusco, 2015; Nižnan, Pelánek, & Řihák, 2014; Xu & Zhang, 2015) and for dynamic data in which student learning occurs (Stamper & Koedinger, 2011; Koedinger et al. 2013; Aleven & Koedinger, 2013; González-Brenes, 2015; N. Li, Cohen, Koedinger, & Matsuda, 2011). In addition to algorithms that take as input student response outcome data, we find the emergence of methods that integrate text analysis to label and optimize the search for better Q-matrices (Goutte, Léger, & Durand, 2015; N. Li, Cohen, & Koedinger, 2013; Matsuda, Furukawa, Bier, & Faloutsos, 2014).

This is a clear sign of the importance of the problem as well as of the vitality of the research on the topic. Some studies have shown that using data-driven techniques of refinement generally result in Q-matrices that have better predictive power and are a better fit to the data in general

(Aleven & Koedinger, 2013; Durand, Belacel, & Gutte, 2015; Matsuda et al., 2014).

We first review below three examples of Q-matrix refinement algorithms, and show how such algorithms can be combined to obtain substantial gains in the next section. The two first ones are based on the DINA model, whereas the last one is based directly on equation (3).

3.1 minRSS

For a given Q-matrix, there is an ideal response pattern (ideal response vector) for each skill pattern (profile vector). If there are no slip and guess factors, then the response pattern for every category of student profile is fixed. A reasonable assumption is to assume the real response pattern should not differ much from this ideal response pattern. Then the problem is how to measure the difference between the real pattern and ideal pattern. The most common metric for binary data is Hamming distance, that is

$$d_h(r,\eta) = \sum_{j=1}^{J} | r_j - \eta_j |$$

where r is the real response vector while η is the ideal response vector. J is the number of latent skills. Chiu and Douglas (2013) refined this metric based on the idea that if an item has a smaller variance (or entropy), then it should be given higher weight. The formula is

$$d_{\omega h}(r,\eta) = \sum_{j=1}^{J} \frac{1}{\bar{p}_j(1-\bar{p}_j)} | r_j - \eta_j |$$

where $\bar{p}_j$ is the proportion of correct answers of item j. Equipped with this metric, we can find the ideal response matrix that best fits the data, and then find the correspondent profile matrix $\mathbf{P}$. With these results, a powerful method was proposed to update the Q-matrix (Chiu, 2013). First, a squared sum of errors for each item k can be computed by

$$RSS_k = \sum_{i=1}^{N} (r_{ik} - \eta_{ik})^2$$

where N is the number of respondents. Then, the item with the highest RSS is chosen to update its correspondent q-vector. All the other possible q-vectors are tested to calculate their RSS and the q-vector giving the lowest RSS is chosen to replace the original one. We name this method minRSS based on this minimization objective. The Q-matrix is thus updated accordingly, and the whole process will be repeated. The previous changed q-vector is taken out of searching pool for the next iteration. The whole procedure terminates when the RSS for each item no longer changes. This method has a consistency property which was shown by Wang & Douglas (2015). That is, it has good performance under different underlying conjunctive models.

3.2 maxDiff

Under the setting of DINA model, for every item j, there are two model parameters, slip s_j and guess g_j. de la Torre (2008) proposed that a correctly specified q-vector for item j should maximize the difference of probabilities of a correct response between examinees who have all the required attributes and those who do not. For a model invovled with K possible skills, there are 2^K possible q-vectors(i.e skill combination). Denote these possible q-vectors by α_l, $l = 0,1,...,2^K - 1$, then q_j is the correct q-vector if

$$q_j = arg\,max_{\alpha_l}[P(X_j = 1 | \xi_{ll'} = 1) - P(X_j = 1 | \xi_{ll'} = 0)] = arg\,max_{\alpha_l}[\delta_{jl}]$$

for $l, l' = 1,2,...,2^K - 1$ and $\xi_{ll'} = \prod_{k=1}^{K} \alpha_{l'k}^{\alpha_{lk}}$. That is also why we call it maxDiff. An interesting observation is since $P(X_j = 1 | \xi_{ll'} = 1) = 1 - s_j$ and $P(X_j = 1 | \xi_{ll'} = 0) = g_j$, then

$$q_j = arg\,max_{\alpha_l}[1 - (s_j + g_j)]$$

that is, maximizing the difference is equivalent to minimize the sum of the slip and guess parameters. A natural idea is to test all q-vectors to find the maximum δ_{jl} but that is computationally expensive. de la Torre

(2008) proposed a greedy algorithm that adds skills into a q-vector sequentially. First, δ_{jl} is calculated for all q-vectors which contains only one skill and the one with biggest δ_{jl} is chosen. Then, δ_{jl} is calculated for all q-vectors which contains two skills including the previously chosen one. Again the q-vector with the largest δ_{jl} is chosen. This whole process is repeated until no skills increases δ_{jl}. However, this algorithm requires knowing s_j and g_j in advance. For real data, they are calculated by EM (Expectation Maximization) algorithm (de la Torre, 2009).

3.3 ALSC

ALSC (Conjunctive Alternating Least Squares Factorization) is a common matrix factorization (MF) technique. Desmarais and Naceur (2013) proposed to factorize student test results into a Q-matrix and a skills-student matrix with a least squares estimate.

Contrary to the other two methods, it does not rely on the DINA model as it has no slip and guess parameters. ALSC decomposes the results matrix $\mathbf{R}$ based on the least squares estimate.

The factorization consists of alternating between estimates of $\mathbf{P}$ and $\mathbf{Q}$ until convergence. Take conjunctive model for example, starting with the initial expert defined Q-matrix $\mathbf{Q}_0$, an initial least-squares estimate of $\mathbf{P}$ is obtained:

$$\neg\hat{\mathbf{P}}_0 = \neg\mathbf{R}\mathbf{Q}_0(\mathbf{Q}_0^{\mathrm{T}}\mathbf{Q}_0)^{-1} \tag{5}$$

which is the least squares solution of equation (3). Then, a new estimate of the Q-matrix, $\hat{\mathbf{Q}}_1$, is again obtained by the least-squares estimate:

$$\hat{\mathbf{Q}}_1^{\mathrm{T}} = (\neg\hat{\mathbf{P}}_0^{\mathrm{T}}\neg\hat{\mathbf{P}}_0)^{-1}\neg\hat{\mathbf{P}}_0^{\mathrm{T}}\neg\mathbf{R} \tag{6}$$

And so on until convergence. Alternating between equations (5) and (6) yields progressive refinements of the matrices $\hat{\mathbf{Q}}_i$ and $\hat{\mathbf{P}}_i$ that more closely approximate $\mathbf{R}$, the observed student response outcome matrix.

The final $\hat{Q}_i$ is rounded to yield a binary matrix.

4 Combining techniques with ensemble algorithms

The effectiveness of a Q-matrix refinement technique may depend on specific characteristics of the data, or on characteristics of the matrix itself. An algorithm that can learn the conditions under which an approach is more likely to give a reliable answer can, in principle, provide better refinements than any algorithm alone.

A critical factor for the success of the combination approach is the definition of effective factors to allow the learning of how to combine the output of the Q-matrix refinement algorithms. Let us focus on one of these factors, *stickiness*, as an informative example.

Stickiness represents the rate of a given algorithm's false positives for a given cell of a Q-matrix. A false positive is considered a recommended change in the Q-matrix when the ground truth tells us it is wrong: no changes should be recommended. The rate is measured by "perturbating" in turn each and every cell of the Q-matrix, and by counting the number of times the cell is a false positive. The decision tree can use the stickiness factor as an indicator of the reliability of a given Q-matrix refinement algorithm's suggested value for a cell. Obviously, if a cell's stickiness value is high, the reliability of the algorithm's suggestion will be lower.

The question is how to train a decision tree with enough data to use the stickiness and other factors? An original idea introduced in the approach is to use synthetic data for which we know what is the Q-matrix ground truth. Random matrices with a similar ratio of 0/1 are generated and the perturbation process described above is applied to generate tens of thousands of tuples with the following elements:

1. Target value
2. Predicted values from the three algorithms studied

3. Stickiness
4. A few other characteristics of the skill and the item involved

These elements represent the input to a decision tree that essentially learns which of the predicted values (2.) are most likely to be correct given the contextual factors (3. and 4.)

Based on the decision tree combination approach described above, Desmarais et al. (2015) obtained a substantial gain in accuracy over the best of the three refinement algorithms. Considering on an equal basis an error as not recovering the perturbated cell and recommending changes to non perturbated cells, they obtained an error reduction in the range of 50% using real data over Q-matrices defined by experts and around 85% using DINA-based synthetic data for the same matrices.

The combination approach can be considered an *ensemble* technique in the machine learning field. Another well-known ensemble technique is to combine a decision tree with *boosting*. Boosting consists in assigning a weight to each individual observation in the loss function. The weight is increased when the predicted value differs from the observed one, and the classifier, namely the decision tree in our case, is trained with the new weighted loss function. Using the Adaboost boosting algorithm, an additional improvement in error reduction of 18% for real data and 46% for synthetic data is obtained (Xu & Desmarais, 2016).

In terms of correct and incorrect refinements, the ensemble technique, which combines the three algorithms with a boosted decision tree, is able to recover almost all of the perturbated cells to their original value. It improves the rate of recall of perturbated cells from around half to close to all. However, in spite of these improvements, it still introduces a small number of incorrect refinements (proposed changes to cells that were not perturbated). These incorrect refinements can prove disrupting to an expert who uses a Q-matrix refinement tool, as it entails an effort to analyze and assess the proposed refinements, and future efforts should focus on reducing this number.

5 Discussion

An important finding from the work with ensemble techniques is the demonstration of the complementarity of Q-matrix refinement algorithms, at least the three algorithms used in the ensemble studies reported in section 4. The gains to recover perturbed cells to their original value are remarkable and the performance is close to perfect. And whilst the number false refinements remains disrupting, it is not to the extent to undermine the value of the recommended changes.

Can ensemble techniques extend to algorithms to map items to skills with dynamic data? Can we assume the algorithms to refine the mapping of item to skills/KC for dynamic data are also complementary? It is reasonable to believe that similar gains could be obtained if we can assume the complementarity is present. Simplification such as using the first trial (for eg. González-Brenes, 2015) may permit greater variability of approaches, and therefore complementarity in the sources of information that can be combined in an ensemble technique.

6 Recommendations and Future Research

The CTAT tools described in Aleven and Koedinger (2013) for helping an expert to map exercises, items, and tasks to underlying skills are excellent examples of valuable outcomes of models and algorithms that can use data to help an expert refine a Q-matrix (or a KC model in their terminology). These tools allow the comparison of Q-matrix versions, assessment of their fit in terms of predictive power and other measures of fit, and provide different means to refine them.

The approaches and algorithms reviewed in this chapter should lead to such tools, or complement them. They help identify weaknesses in a Q-matrix at the item level, pinpointing missing or irrelevant skills associated with tasks, and also at the skill level: a whole column may

show anomalies that suggest a skill may be ill-defined and unfit to the data.

As emphasized before, the Q-matrix refinement algorithms reviewed in this chapter make the assumption that learning does not occur in the data. This is a serious limitation for data collected from learning environments where learning does occur. Important questions to address are therefore what are the consequences of the violation of this assumption and how can we mitigate the adverse effects and work around them?

An avenue to explore is to transform the dynamic data into a static view. The dynamic cognitive modeling models can generally be conceptualized as a factorization model, akin to the factorization of equation (1), but the $\mathbf{P}$ and the $\mathbf{R}$ matrices have a third dimension which is time. They are then considered tensors models (see Thai-Nghe, Horváth, & Schmidt-Thieme, 2011). In such models, it is possible to predict the student response outcome data at a given time slice, thereby transforming the dynamic data into a static view that the models reviewed in this chapter can handle.

REFERENCES

Aleven, V., & Koedinger, K. R. (2013). Knowledge component (KC) approaches to learner modeling. *Design Recommendations for Intelligent Tutoring Systems*, 189–203.

Barnes, T. (2010). Novel derivation and application of skill matrices: The Q-matrix method. *Handbook on educational data mining*, 159–172.

Chen, Y., Liu, J., Xu, G., & Ying, Z. (2015). Statistical analysis of Q-matrix based diagnostic classification models. *Journal of the American Statistical Association*, 110(510), 850–866.

Chiu, C.-Y. (2013). Statistical refinement of the Q-matrix in cognitive diagnosis. *Applied Psychological Measurement*, 37(8), 598–618.

Chiu, C.-Y., & Douglas, J. (2013). A nonparametric approach to cognitive diagnosis by proximity to ideal response patterns. *Journal of Classification*, 30(2), 225–250.

Chung, M.-t. (2014). *Estimating the Q-matrix for cognitive diagnosis models in a bayesian framework* (Unpublished doctoral dissertation). Columbia University.

de la Torre, J. (2008). An empirically based method of Q-matrix validation for the dina model: Development and applications. *Journal of educational measurement*, 45(4), 343–362.

de la Torre, J. (2009). Dina model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics*, 34(1), 115–130.

de la Torre, J., & Chiu, C.-Y. (2015). A general method of empirical Q-matrix validation. *Psychometrika*, 1–21.

Desmarais, M. C., & Naceur, R. (2013). A matrix factorization method for mapping items to skills and for enhancing expert-based q-matrices. In *Artificial intelligence in education* (pp. 441–450).

Desmarais, M. C., Xu, P., & Beheshti, B. (2015). Combining techniques to refine item to skills q-matrices with a partition tree. In *Educational data mining* 2015.

Durand, G., Belacel, N., & Goutte, C. (2015). Evaluation of expert-based q-matrices predictive quality in matrix factorization models. In *Design for teaching and learning in a networked world* (pp. 56–69). Springer.

Gonz âlez-Brenes, J. P. (2015). Modeling skill acquisition over time with

sequence and topic modeling. In *Aistats*.

Goutte, C., Durand, G., & Léger, S. (2015). Towards automatic description of knowledge components. In *Proceedings of the tenth workshop on innovative use of nlp for building educational applications* (pp.75–80).

Goutte, C., Léger, S., & Durand, G. (2015). A probabilistic model for knowledge component naming. In *Proceedings of the 8th international conference on data mining* (pp. 608–609).

Henson, R. A., Templin, J. L., &Willse, J. T. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika*, 74(2), 191–210.

Koedinger, K. R., Stamper, J. C., McLaughlin, E. A., & Nixon, T. (2013). Using data-driven discovery of better student models to improve student learning. In *Artificial intelligence in education* (pp. 421–430).

Köhn, H.-F., Chiu, C.-Y., & Brusco, M. J. (2015). Heuristic cognitive diagnosis when the Q-matrix is unknown. *British Journal of Mathematical and Statistical Psychology*, 68(2), 268–291.

Li, H., & Suen, H. K. (2013). Constructing and validating a Q-matrix for cognitive diagnostic analyses of a reading test. *Educational Assessment*, 18(1), 1–25.

Li, N., Cohen, W. W., & Koedinger, K. R. (2013). Discovering student models with a clustering algorithm using problem content.

Li, N., Cohen, W.W., Koedinger, K. R., & Matsuda, N. (2011). A machine learning approach for automatic student model discovery. In *Edm* (pp. 31–40).

Matsuda, N., Furukawa, T., Bier, N., & Faloutsos, C. (2014). Machine

beats experts: Automatic discovery of skill models for data-driven online course refinement. In *Educational data mining 2014* (pp. 101–108).

Nižnan, J., Pelánek, R., & Řihák, J. (2014). Mapping problems to skills combining expert opinion and student data. In *Mathematical and engineering methods in computer science* (pp. 113–124). Springer.

Qin, C., Zhang, L., Qiu, D., Huang, L., Geng, T., Jiang, H., . . . Zhou, J. (2015). Model identification and Q-matrix incremental inference in cognitive diagnosis. *Knowledge-Based Systems*, 86, 66–76.

Romero, S. J., Ordoñez, X. G., Ponsoda, V., & Revuelta, J. (2014). Detection of Q-matrix misspecification using two criteria for validation of cognitive structures under the least squares distance model. *Psicologica: International Journal of Methodology and Experimental Psychology*, 35(1), 149–169.

Roussos, L. A., DiBello, L. V., Stout, W., Hartz, S. M., Henson, R. A., & Templin, J. L. (2007). The fusion model skills diagnosis system. *Cognitive diagnostic assessment for education: Theory and applications*, 275–318.

Sinharay, S.,&Almond, R. G. (2007). Assessing fit of cognitive diagnostic models a case study. *Educational and Psychological Measurement*, 67(2), 239–257.

Stamper, J. C., & Koedinger, K. R. (2011). Human-machine student model discovery and improvement using datashop. In *Artificial intelligence in education* (pp. 353–360).

Templin, J. (2015). Diagnostic assessment. *Technology and Testing: Improving Educational and Psychological Measurement*, 285.

Thai-Nghe, N., Horváth, T., & Schmidt-Thieme, L. (2011). Factorization

models for forecasting student performance. In *Edm* (pp. 11–20).

Wang, S., & Douglas, J. (2015). Consistency of nonparametric classification in cognitive diagnosis. *Psychometrika*, 80(1), 85–100.

Xiang, R. (2013). *Nonlinear penalized estimation of true Q-matrix in cognitive diagnostic models* (Unpublished doctoral dissertation). Columbia University.

Xu, P., & Desmarais (2016). Boosted Decision Tree for Q-matrix Refinement. 9th *International Conference on Educational Data Mining* (EDM2016), Raleigh, NC (submitted).

Xu, G., & Zhang, S. (2015). Identifiability of diagnostic classification models. *Psychometrika*, 1–25.