Q-matrix Misspecification Detection using Spectral Clustering on TIMSS 2011 Assessment Data

by

Zhihao Guo

An honors thesis submitted in partial fulfillment
of the requirements for the degree of
Bachelor of Science
(Honor Statistics)
at the University of Michigan
2019

Supervisor: Dr. Gongjun Xu
2019.7.9

Acknowledgements

I want to gratefully acknowledge Professor Gongjun Xu for providing instructions, resources, and assistance on this project. I sincerely appreciate his help and advice.

Abstract

Cognitive Diagnostic Models (CDMs) aim to provide information about the degree to which individuals have mastered specific attributes that underlie the success of these individuals on test items. A common component of CDMs for specifying the attributes required for each item is the Q-matrix. Although construction of Q-matrix is typically performed by domain experts, it nonetheless, to a large extent, remains a subjective process, and misspecifications in the Q-matrix, if left unchecked, can have important practical implications. To address this concern, this paper uses an assumption-free model: spectral clustering, as a validation benchmark to detect if the Q-matrix is misspecified, subject to a common CDM frameworks: non-compensatory deterministic input noisy-and-gate (DINA). The paper then proposes an empirical way to determine the amount of misspecifications in Q-matrices. In doing so, the data *TIMSS 2011 Mathematics 4th Grade Austrian Students* from R *CDM* package is used as validation, and the empirical cut-off is found. Results also show that attribute specifications can differ from expert opinions and the underlying model for each item can vary. In addition, the model's performance under different amounts of Q-matrix misspecifications has been studied and it shows that DINA model has certain tolerance to maintain its high-quality performance under small amount of misspecification. Lastly, the paper explores other factors contributing to the model such as slip and guess rate.

Contents

1 Introduction 2

2 Methodology 3

2.1 Cognitive Diagnosis Models . . . 3

2.1.1 Terminology and Notation . . . 3

2.1.2 DINA . . . 4

2.2 Spectral Clustering . . . 5

2.2.1 Similarity Function . . . 5

2.2.2 Unnormalized Spectral Clustering Algorithm . . . 6

2.3 Algorithm to Determine the Misspecification in Q-matrix . . . 8

3 Data Overview 11

3.1 Student Achievement Data . . . 11

3.2 Q-Matrix . . . 12

4 Results 14

4.1 Simulation Study . . . 14

4.2 Real Data Study . . . 19

4.3 Model Stability and Tolerance . . . 23

4.4 Other Effects . . . 26

5 Conclusions and Future Outlook 31

.1 Appendix: R Code for Spectral Clustering . . . 35

Chapter 1

Introduction

In recent years educational research was characterized by an increasing demand of complex information on students' achievement. This may be caused by a growing interest in explaining the results of international comparative studies like the trends in international science study [1], progress in international reading study [2] and the programme for international student assessment [3]. It may also be caused by a strong need to explain the social and ethnic disparities detected in these studies [2]. Generally, international large-scale exams (e.g., TIMSS) have been analyzed with IRT models, which provide a single total score for each examinee. With recent advancements in CDAs, however, there has been a trend toward providing more elaborate results on testing practices. A number of CDMs have been developed to obtain more detailed test results [4]. The shift from single score reporting practices to CDM approaches has also been applied to TIMSS data in several studies [5].

The CDMs assume the relationship between questions and skills strictly follow the corresponding Q-matrix. Thus, the validity of the results depends on the correct specification of the Q-matrix [6]. Incorrect specification of the Q-matrix leads to misclassifications of the examinees in the latent classes [7] and, consequently, to erroneous diagnosis in the attribute mastery. As a result, to detect whether there exists any misspecifications in the Q-matrix is essential to the evaluation of the CDMs performance. When the Q-matrix is known to be misspecified, the CDMs are also known to have a false assumption and thus erroneous. It may also correct educational experts of some subjective misunderstandings they have about the relationships between the questions and the skills reflected, which helps them to generate more accurate Q-matrix next time. In addition to the qualitative detection, I also want to know exactly how many entries of the Q-matrix have been misspecified, especially for those Q-matrices that have been extensively used in the TIMSS tests these years. Knowing how many misspecifications exist in Q-matrix will provide more information about the test to help further study on those data. Furthermore, the effect of the misspecified Q-matrix on the model performance is another important topic, which helps people to determine whether the result of the model is reliable or not. To be specific, I am interested about the sensitivity of the model to the misspecifications and wonder if the model has an ability to maintain its performance when the amount of misspecification in the Q-matrix is small. Lastly, I investigate other factors contributing to the model prediction, such as slip/guess rate. It helps to generate a more comprehensive picture and provides some insightful results to the problem.

Chapter 2

Methodology

2.1 Cognitive Diagnosis Models

As a promising method to model students' responses in an achievement test, CDMs comprise two steps: In the first step, educational experts define several basic abilities for the questions being asked, which are called skills. After that, they construct a so-called Q-matrix which reflects the skills required to answer each question. In the second step, CDMs use Q-matrix and students' responses to those questions to classify students into dichotomous latent skill classes, which describe their mastery of each skill defined. CDMs are able to provide many useful information about the students' skill possession. The main results are threefold: Firstly, the distribution of the skill classes allows for statements how many students in the test population possess certain combinations of skills. Secondly, the skill mastery probabilities include information about the percentage of students in the test population possessing the individual skills. Thirdly, for each individual student a skill class is deduced which is called the student's skill profile and which predicts the possession or nonpossession of the individual skills. Together all three issues provide a solid empirical base for targeted pedagogical interventions both on the level of the test population and on the individual student's level [8].

Non-compensatory deterministic input noisy-and-gate (DINA), is one of the most commonly used and most popular core CDMs. It inherits the idea and method of general CDMs, but is unique in that: the DINA model's noncompensability asserts that students have to possess all skills assigned to an item for successfully mastering it.

2.1.1 Terminology and Notation

The below terms and definitions in this section are directly adapted from *The R Package CDM for Cognitive Diagnosis Models.* [8].

• Response Matrix $\boldsymbol{X}$: In an achievement test in which I students respond to J terms, the response of a student $i, i = 1, ..., I$, to item $j, j = 1, ...J$, will be denoted as X_{ij}. X_{ij} will be binary, with a value of 1 indicating student i answered question j correctly and a value of 0 if incorrect. The $I \times J$ binary matrix $\boldsymbol{X}$ will be the responses of all I students to all J questions. The i-th row $\boldsymbol{X}_i$ of $\boldsymbol{X}$ represents the responses of the student i to all J items.

This is called the i-th student's response pattern. If student i did not attempt question j, then the corresponding X_{ij} will be NA.

• Skill Profile α_k: Educational experts define K skills which students have to possess for mastering all the questions. For a student i a latent dichotomous skill profile $\alpha_i = [\alpha_{i1}, ..., \alpha_{iK}]$ denotes the possession ($\alpha_{ik} = 1$) and non-possession ($\alpha_{ik} = 0$) of the K-th predefined skill. The goal of the CDM is to estimate the individual student's skill profile so that we know exactly what skill each student possesses.

• Q-matrix: Educational experts also define which skills are required to master which item in a $J \times K$ matrix Q, where (j, k)-th element q_{jk} of Q equals 1 if skill k is relevant for the mastery of item j and equals 0 otherwise. Since a question may reflect only one skill or multiple skills, so for j-th row of Q, there can be from only 1 to k 1's if all the skills are needed to master the question.

• Skill Class α_l: Since we do not know the exact skill profile α_k of the students, we define skill classes $\alpha_l, l = 1, ..., L$. The skill classes will be the combination of K skills so that the largest possible number of disjunctive skill classes is $L = 2^K$. For example, if there are 3 skills, then we would have $2^3 = 8$ skill classes. Each student will thus have a skill class distribution $P(\alpha_l), l = 1, ..., L$, and by finding the skill class with the maximum likelihood, we use this skill classes α_l to estimate the student's skill profile α_k.

2.1.2 DINA

In *The R Package CDM for Cognitive Diagnosis Models*, it defines the latent response of students to questions as

$$\eta_{ij} = \prod_{k=1}^{K} \alpha_{ik}^{q_{jk}}$$

where i is the i-th student, j is the j-th question. α is the skill profile $\boldsymbol{\alpha_i} = [\alpha_{i1}, ..., \alpha_{iK}]$ to item j.[8] The vector $[q_{j1}, ..., q_{jK}]$ denotes the q-th row of the Q-matrix which indicates the skills required for the mastery of item j. A student who possesses all or even more than these required skills is expected to master the item and $\eta_{ij} = 1$. Otherwise, for a student who does not possess any one of the skills required, $\eta_{ij} = 0$.

The paper also introduces another parameter called *slip/guess*.[8] If student i is expected to master the item, he nevertheless may slip and fail the item. On the other hand, even if the student is not expected to master the item, he may succeed by a lucky guess. The probability of the occurrence of slip on a question is denoted as g_j while slip is denoted as s_j.

We can get the probability of student i to solve item j:

$$P(X_{ij} = 1 | \alpha_i, g_j, s_j) = (1 - s_j)^{\eta_{ij}} \cdot g_j^{(1-\eta_{ij})} = \begin{cases} 1 - s_j & \text{for } \eta_{ij} = 1, \\ g_j & \text{for } \eta_{ij} = 0. \end{cases}$$

The paper performs the parameter estimation of DINA by means of marginal maximum likelihood (MML) estimation, implemented by expectation-maximization algorithm (EM algorithm)[8].

let

$$P(X_i|\alpha_l,\delta) = \prod_{j=1}^{J} P(X_{ij}=1|\alpha_l;g_j,s_j)^{X_{ij}}[1-P(X_{ij}=1|\alpha_l;g_j,s_j)]^{1-X_{ij}}$$

be the probability of response vector X_i if student i possesses the skills of skill class α_l, $l = 1, ..., L$.

For estimating the DINA model, the marginal log-likelihood

$$\log L(\delta,\gamma) = \sum_{i=1}^{I} \log L(X_i;\delta,\gamma) = \sum_{i=1}^{I} \log\left[\sum_{l=1}^{L} P(X_i|\alpha_l;\delta)\cdot P(\alpha_l|\gamma)\right]$$

is maximized with respect to the item parameters δ and the parameters $\gamma = [\gamma_1, ..., \gamma_{L'}]$ describing the skill class distribution $P(\alpha_l), l = 1, ..., L$.

Prior to the first iteration of the EM algorithm, initial item parameters δ and skill distribution parameters γ have to be chosen. Then, the EM algorithm alternates between the E-step and the M-step until converge. [8]

Once the algorithm converged, the individual student classifications or individual skill profiles can be deduced from the probabilities $P(\alpha_l|X_i)$, according to three methods: maximum a priori (MAP) classification, maximum likelihood estimation (MLE), and expected a posteriori probabilities (EAP). Here I choose MLE as the estimation. Using MLE, an individual classification of student i is obtained by maximizing

$$\hat{\alpha}_{i;MLE} = \underset{\alpha_l}{\operatorname{argmax}}\{P(X_i|\alpha_l)\}$$

2.2 Spectral Clustering

The goal of this paper is to detect the misspecification in the Q-matrix used by DINA model. As a key assumption the model made, some Q-matrices seem dubious or even clearly incorrect, and significantly affect the accuracy of the final results. The Spectral Clustering is used as a benchmark to compare the goodness of DINA model, because Spectral Clustering does not rely on any assumptions, which makes it more reliable in the context of this paper. Meanwhile, It is simple to implement, can be solved efficiently by standard linear algebra software, and very often outperforms traditional clustering algorithms such as the k-means algorithm.

Meanwhile, as the responses matrix is binary, the similarity measure I use needs to work on binary vectors to implement the Spectral Clustering. I also need to define rules to construct the similarity graph and pick the number of clusters.

2.2.1 Similarity Function

The intuitive goal of clustering is to divide the data points into several groups such that points in the same group are similar and points in different groups are dissimilar to each other. To see how far away is one data point to the other, I need to define the distance measure for the data.

Knowing that the matrix X_{ij} is binary, I choose the simple matching coefficient (SMC) since it is sufficient to characterize the vectors and is easy to implement [11]. The algorithm of the method is following:

The comparison of two binary vectors, a and b, leads to four quantities:

$$\begin{aligned}
N_{01} &= \text{the number of positions where a was 0 and b was 1}\\
N_{10} &= \text{the number of positions where a was 1 and b was 0}\\
N_{00} &= \text{the number of positions where a was 0 and b was 0}\\
N_{11} &= \text{the number of positions where a was 1 and b was 1}
\end{aligned}$$

The simple matching coefficient (SMC) is:

$$\text{SMC} = (N_{11} + N_{00})/(N_{01} + N_{10} + N_{11} + N_{00})$$

For example, the following two binary vectors, a and b we get SMC = 0.7:

$$\begin{aligned}
\text{a} &= 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\\
\text{b} &= 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1
\end{aligned}$$

Conceptually, SMC equates similarity with the total number of matches.

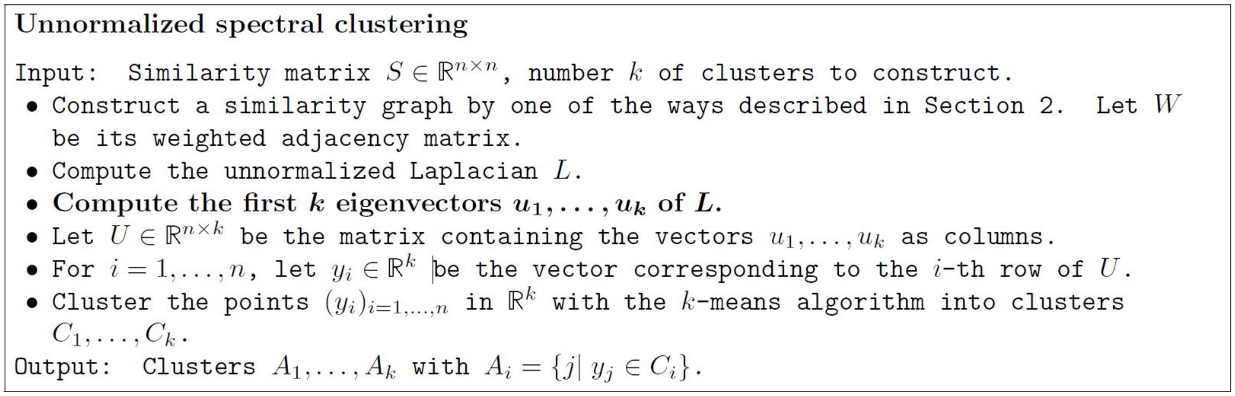
2.2.2 Unnormalized Spectral Clustering Algorithm

In *A Tutorial on Spectral Clustering*, it introduced a method called *Unnormalized Graph Laplacian* to implement the Spectral Clustering algorithm.[10] The idea is to define a similarity matrix S by similarity measure and each data point only keeps k nearest points' distances in the matrix. Then, the *Unnormalized Graph Laplacians matrix* is defined by:

$$L = D - W.$$

where D is the *degree matrix*, which is a diagonal matrix with degrees $d_1, ..., d_n$ on the diagonal. W is the *weighted adjacency matrix* of the similarity matrix where $W = (w_{ij})_{i,j=1,...,n}$, w_{ij} being the similarity between vertices v_i and v_j. If $w_{ij} = 0$ this means that the vertices v_i and v_j are not connected by an edge.[10].

In the paper, it proves that the multiplicity k of the eigenvalue 0 of Laplacian matrix equals the number of connected components in the similarity matrix. Thus, the *Unnormalized Spectral Clustering* algorithm is as following:[10]

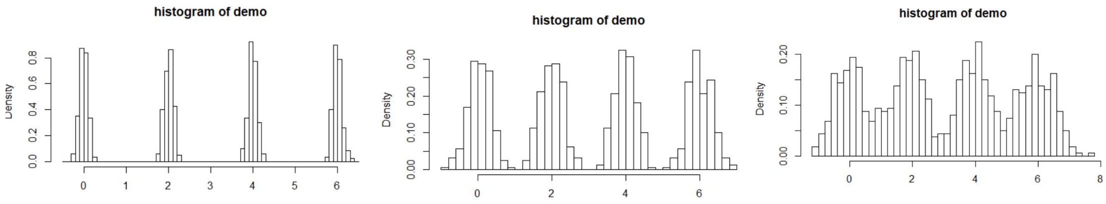


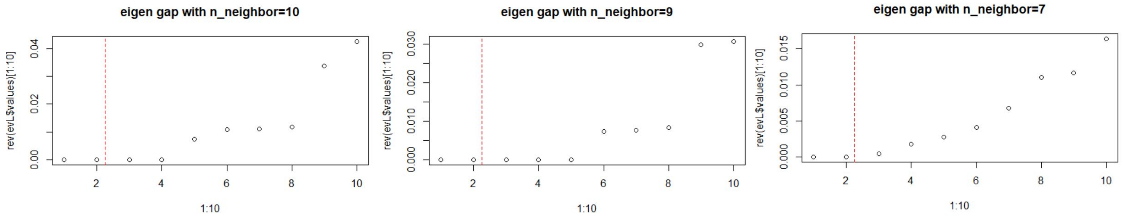
Unnormalized spectral clustering

```
Input: Similarity matrix $S \in \mathbb{R}^{n\times n}$, number $k$ of clusters to construct.
  • Construct a similarity graph by one of the ways described in Section 2. Let $W$
    be its weighted adjacency matrix.
  • Compute the unnormalized Laplacian $L$.
  • Compute the first $k$ eigenvectors $u_1, \dots, u_k$ of $L$.
  • Let $U \in \mathbb{R}^{n\times k}$ be the matrix containing the vectors $u_1, \dots, u_k$ as columns.
  • For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the $i$-th row of $U$.
  • Cluster the points $(y_i)_{i=1,\dots,n}$ in $\mathbb{R}^k$ with the $k$-means algorithm into clusters
    $C_1, \dots, C_k$.
Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j|\ y_j \in C_i\}$.
```

There are two parameters needs to be determined for the method: the number of neighbors for each data to connect when constructing the similarity matrix, and the number of groups to cluster for K-means in the last step.

For the number of clusters, the paper introduces a method called *eigengap heuristic*.[10; 12; 13; 14]. Here the goal is to choose the number k such that all eigenvalues $\lambda_1, ..., \lambda_k$ are very small, but λ_{k+1} is relatively large. If I plot all the eigenvalues from small to large, I am expected to see a "gap" between λ_k and λ_{k+1}, where the eigenvalue suddenly goes up. This is also called the *eigen gap*. In addition, for some clustering, I already known how many cluster it will have. For example, if I cluster the questions based on responses, I know in advance that I need to cluster them into as many groups as the number of skills defined. In this way, I am not making decisions base on *eigengap heuristic*, but the method can still be used as a verification. If the graph does not have a "leap" in the desired number of clusters, then we need to investigate the reason for that.





In the demo plot, I generate data from a mixture of four independent Gaussian distributions with same variance and different means. I change their variance to make the data looked separate or overlapped. From their histograms, samples from distribution with small variance will have fewer overlaps than samples from distribution with large variance. The similarity function I use is Gaussian similarity function $s(x_i, x_j) = exp(-|x_i - x_j|^2/(2\sigma^2))$.

The most similar data points will have a Gaussian similarity measure close to 1 and the least similar measure will be near 0. I see that the first graph has a clear *eigen gap* between 4-th eigenvalue and 5-th eigenvalue. Thus, it is recommended to cluster the data into 4 groups. Note that clustering into 8 groups is also acceptable. As the distributions become overlapped, it is harder for the algorithm to give a clear separation and the number of clusters is not consistent in different trials. When the distributions highly overlapped, there is no *eigen gap* at all, meaning that the algorithm is not able to provide a cut-off in that case. It is hard to tell for the last graph as the plot shows a gradually increasing trend, and the *eigen graph* does not provide any suggestion on the number of clusters.

For the number of neighbors, the way the article recommended is to try different values and compare the *eigen graphs*.[10] In this paper, sometimes I already known how many clusters it should have. I can use this information as a verification to find the proper k value for k-nearest neighbor graph. I loop through couple values of k, and generate their *eigen graph*, and the graph that has a clear *eigen gap* at the desired number of clusters will have my desired value of k. Assume there are 3 skills defined for Q-matrix, then a good choice of k will result in a *eigen graph* look like this:

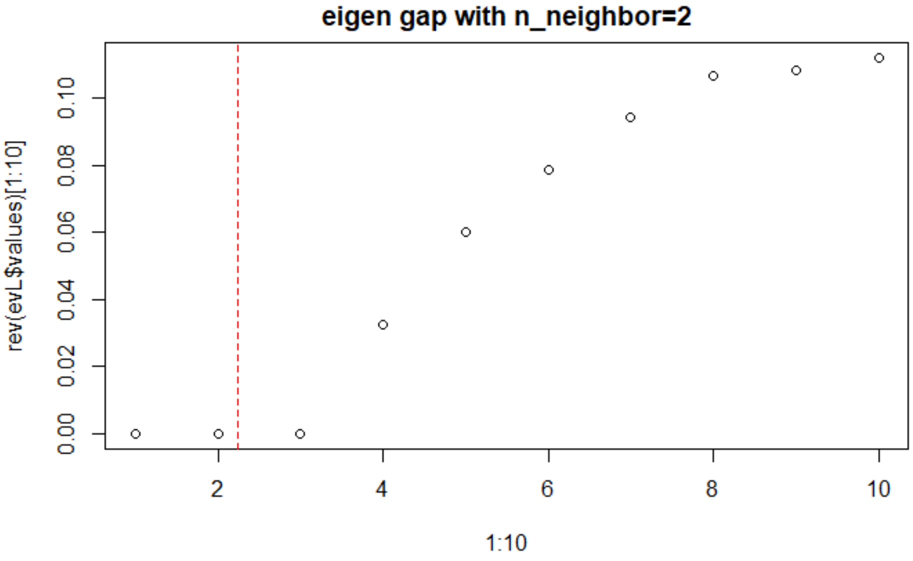


Figure 2.3: A Demo of eigen graph with Good k

2.3 Algorithm to Determine the Misspecification in Q-matrix

To begin with, I define two important terms: *Question Average Group Similarity* and *Gap*

Question Average Group Similarity:

The Q-matrix contains the information about the questions and their corresponding skills.

For each test, educational experts define k skills for the Q-matrix. Assuming the Q-matrix is correct, then according to the Q-matrix, we can easily classify questions into k groups where questions in the same group are believed to represent same skill. The idea is that, for questions in each skill group, the responses of the students to those questions are supposed to be much more similar than responses of students from different groups.

For Spectral Clustering, since it does not depend on the Q-matrix, it will cluster the questions only relying on the students' responses. In other words, questions with similar responses will sure to be clustered into same group by Spectral Clustering, but it might not be the case in DINA model since Q-matrix is defined by people from experience.

Assume m students answered n questions, and we get a $m \times n$ responses matrix X. Under the assumption of Q-matrix, we cluster n questions into k skills. For questions in each skill, we can get the binary students' responses to them from responses matrix X, and we use SMC to obtain the average similarity of those responses. Do this for all k groups and average them together, this is what I call the *Question Average Group Similarity* of DINA model, which will be used as the indicator of the goodness of clustering. For Spectral Clustering, I cluster n questions from responses matrix X into k clusters (matching k skills), and I get the similarity measure of students' responses for each cluster and average them. The result is the *Question Average Group Similarity* of Spectral Clustering. Clearly, the model with higher *Question Average Group Similarity* has a higher quality in its clusters since items in same group are more similar to each other, and thus can be considered as a more accurate method.

Gap:

An assumption-free model,Spectral Clustering,is used as a benchmark, indicating the best clustering quality I am able to get from the data, and I can also get the DINA model *question average group similarity.* It is reasonable to believe that the *question average group similarity* of DINA will be lower than *question average group similarity* of Spectral Clustering since I assume there are misspecifications in the Q-matrix, and I obtain the percentage difference between those two similarity measure and call it the *Gap.* It basically means how far away the DINA model is to the perfect. It is clear that the higher the *Gap* is, the less accurate DINA will be. The *Gap* obtained using the *data.timss11.G4.AUT* and its Q-matrices is called **Real Gap** since those are data collected in real-life, and the one obtained from simulated data will be called **Simulated Gap**.

Determination Algorithm:

To start with, I need to obtain the *Real Gap* and *Simulated Gap.* To simulate data, I generate another responses matrix X' using Q-matrices in the real data, and perform Spectral Clustering and DINA model on the simulated data. Now, for the simulated responses matrix, I am sure that the Q-matrix is 100% correct. However, when I apply DINA model on the data, the Q-matrix I give to the model is deliberately misspecified to some extend. Since the Spectral Clustering does not rely on Q-matrix, its performance will not be affected by the misspecified Q-matrix, but the DINA model performance will be decreased. And the percentage difference in similarity measure between two models is the *Simulated Gap* I want.

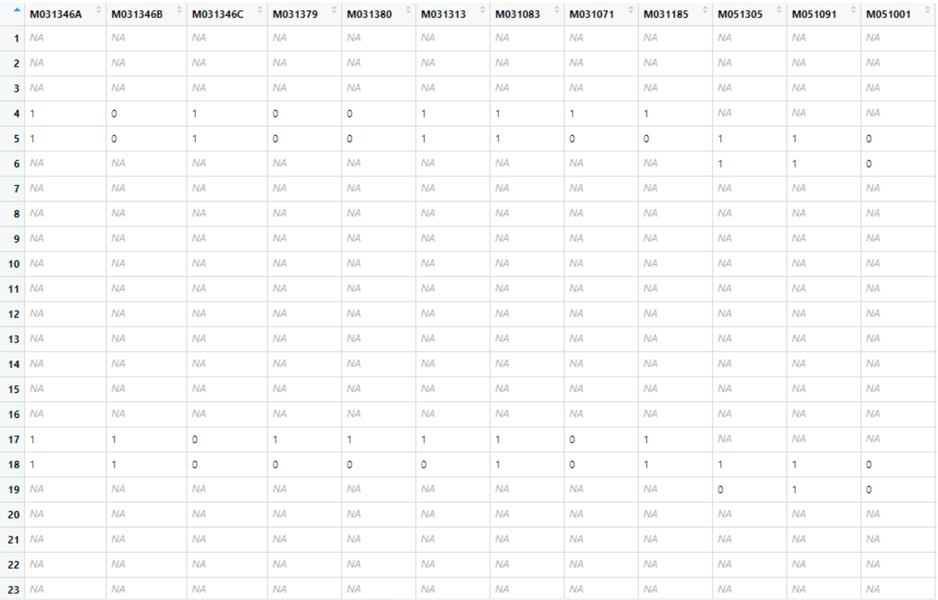
The idea of the empirical cut-off is that: If we keep all other factors in the simulation study the same as the real data study, but only change the Q-matrix, I am able to use *Simulated Gap* to estimate the *Real Gap*. By studying to which portion of Q-matrix misspecification will the *Simulated Gap* to be close enough to the *Real Gap*, I estimate that this portion will be close to the real portion of Q-matrix misspecification in the real data.

Chapter 3

Data Overview

3.1 Student Achievement Data

The real data, named *data.timss11.G4.AUT*, comes from R *CDM* package. It is taken from *TIMSS 2011* dataset of 4668 Austrian fourth-graders. The students who participated in TIMSS 2011 were administered one of 14 assessment booklets, each with a series of mathematics and science items. Some of these items were multiple choice items and some were constructed response items. The student achievement data files contain the actual responses to the multiple choice questions. Besides the students' response to the booklets questions, the data file also includes students' responses to the student, home, teacher, and school background questionnaires. For this paper, I am only interested in the response to the booklets questions that test mathematics and science, so I further subset the data set to only contain students' responses to booklets questions. The subsetted *data.timss11.G4.AUT* has a dimension of 4668 by 174, meaning there are 174 questions, and 4668 test takers. The values are either binary or NAs. Since each student answers only a selection of questions instead of all the questions, the NAs are being marked when the question was not assigned to that student, otherwise, it will be 0 if the student failed to get it, or 1 if the student got it. According to the question booklet they were asked, students can be divided into 14 groups, and each group has roughly around 670 students who answered from 21 to 27 questions. A piece of the matrix X look like this:



	M031346A	M031346B	M031346C	M031379	M031380	M031313	M031083	M031071	M031185	M051305	M051091	M051001
1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
2	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
4	1	0	1	0	0	1	1	1	1	NA	NA	NA
5	1	0	1	0	0	1	1	0	0	1	1	0
6	NA	NA	NA	NA	NA	NA	NA	NA	NA	1	1	0
7	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
8	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
9	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
10	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
11	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
12	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
13	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
14	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
15	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
16	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
17	1	1	0	1	1	1	1	0	1	NA	NA	NA
18	1	1	0	0	0	0	1	0	1	1	1	0
19	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	1	0
20	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
21	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
22	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
23	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

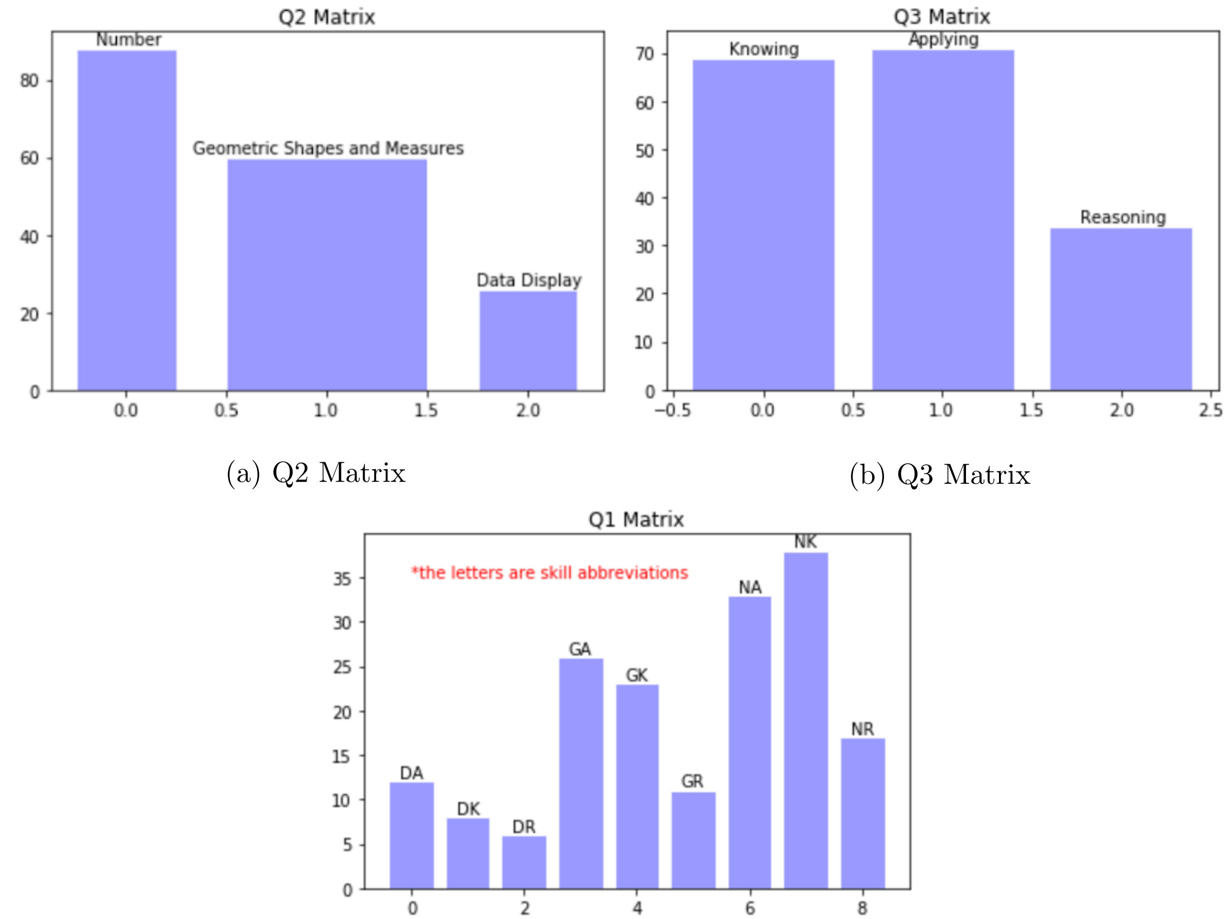
See that there are a lot of NAs in the matrix and the students' answers to the questions they were assigned are marked as either '1' or '0'.

3.2 Q-Matrix

Q-matrix is the mapping of the questions to the skills they reflect. It is a crucial component of CDMs, in that each item is associated with the required attributes to be mastered by examinees for correctly answering the item. Let q_{jk} represents the element in row j and column k of a $J \times K$ Q-matrix, where J and K are the number of items and attributes, respectively. If the k-th attribute is required to answer item j correctly, $q_{jk} = 1$. If it is not required, $q_{jk} = 0$.

The process of constructing the Q-matrix typically involves experts' judgments that could be considered subjective in nature. This can cause serious validation problems as a result of inaccurate parameter estimation and attribute classifications. Moreover, there have been some studies implemented for Q-matrix validation [15].

TIMSS defines two domains for the skills to include all the questions: content and cognitive [16]. In the *data.timss11.G4.AUT*, it uses three Q-matrices, named Q1, Q2, and Q3. Q2 contains the questions and the reflected skills in content domains. The three skills and their abbreviations are: Number(N), Geometric Shapes and Measures(G), and Data Display(D). Q3 matrix has the questions and their cognitive skills. It also has three skills: Knowing(K), Applying(A), and Reasoning(R). Q3 matrix is the mapping to the combination of skills from two domains, so it has $3 * 3 = 9$ skills. In the three Q-matrices, each question only reflects one skill, meaning each row in the Q-matrix will only have one value of 1 and the rest will be 0. Here are the distributions of the skills in each Q-matrix:



(a) Q2 Matrix

(b) Q3 Matrix

(c) Q1 Matrix

Figure 3.1: Q-Matrix Skills Distributions

Chapter 4

Results

The results have four parts: **Simulation Study**, **Real Data Study**, **Model Stability and Tolerance** and **Other Effects**. The first two parts are the primary focus of this study and I derive a method to empirically determine the portion of misspecification in the Q-matrix. In the third part, I am mainly interested in the impact of the Q-matrix misspecification on the DINA model performance, aka its prediction accuracy. The last part is more like an extension to the study done so far where I explore some interesting topics that are worth studying in future.

4.1 Simulation Study

Now, I would like to control the misspecification in the Q-matrix and see what will happen to the *Simulated Gap.* To do this, I mainly used the function *sim.din* from R *CDM* package. The function simulates response matrix X given a pre-defined Q-matrix with either the number of students N, or the students' individual attribute pattern *alpha.* When *alpha* is given, the function will generate response strictly following individual skills given by *alpha.* When N is given, the function will randomly assign individual skills and generate N responses.

In this section, I use *sim.din* to generate responses matrix with random individual skills and three Q-matrices from the real data, then apply Spectral Clustering to get the *question average group similarity.* However, when I apply the DINA model on the simulated data, I change part of the Q-matrix on purpose. There are two factors of the misspecification that I am interested in:
1). The amount of the misspecified entries. For example, the model performance when 10% of the Q-matrix is misspecified or when 60% is misspecified.
2). The way that Q-matrix is misspecified. For example, the model performance when the misspecified question completely represents a wrong skill or it reflects another irrelevant skill besides the correct skill.

The last step is to get the *question average group similarity* of DINA model and compute the *Simulated Gap.*

Note that there is another important factor called *slip/guess rate*, which is the possibility of students guessing a question right or falling the question when possessing the skills required. In the first three sections, I set them both to be 0. This is because I want to simulate the data without any random error. In other words, I want to make sure that any change on the results is due to the change in the Q-matrix.

Simulation Results:

In the real data, there are 14 sets of questions, and each set was attempted by roughly 650 students. Hence, I randomly pick 20 questions each time from the Q-matrix, use *sim.din* to generate a 650×20 responses matrix X_q using Q-matrix subsetted to those questions, and calculate the similarity measure of DINA model with misspecified Q-matrix. For Spectral Clustering, I cluster the questions into the number of skills defined for the Q-matrix. In the three Q-matrices defined for *data.timss11.G4.AUT*, matrix Q2 and Q3 have 3 skills and matrix Q1 has 9 skills. Thus, I need to conduct different Spectral Clustering– one clusters data into 3 groups and the other clusters data into 9 groups. I repeat the whole steps for 20 times to make sure that every question has been selected at least once, then average the results to get the *question average group similarity* for two models. For the misspecification in Q-matrix, I change the amount from 10% of the questions to 90% of the questions, and plot the *question average group similarity* of DINA upon the percentage of the Q-matrix misspecification. In terms of how I misspecify the Q-matrix, I come up with three methods which I think would be the closest three scenarios to the mistakes educational experts could make in reality:

Method 1) Randomly reassign a skill the question reflects. The misspecified question now may represent any skill defined for the Q-matrix. It may reflect a new skill or its original skill still. For example, the skills defined in Q2 matrix are D,G,N and a question reflected skill D, then now it could reflect skill D,G,or N.

Method 2) Completely change the skill the question reflects. The method is very similar to Method 1), and the only difference is that the new skill can not be the same as the old skill. In other words, the misspecified question will always reflect a completely wrong skill. In the previous example, the question now will reflect either G or N.

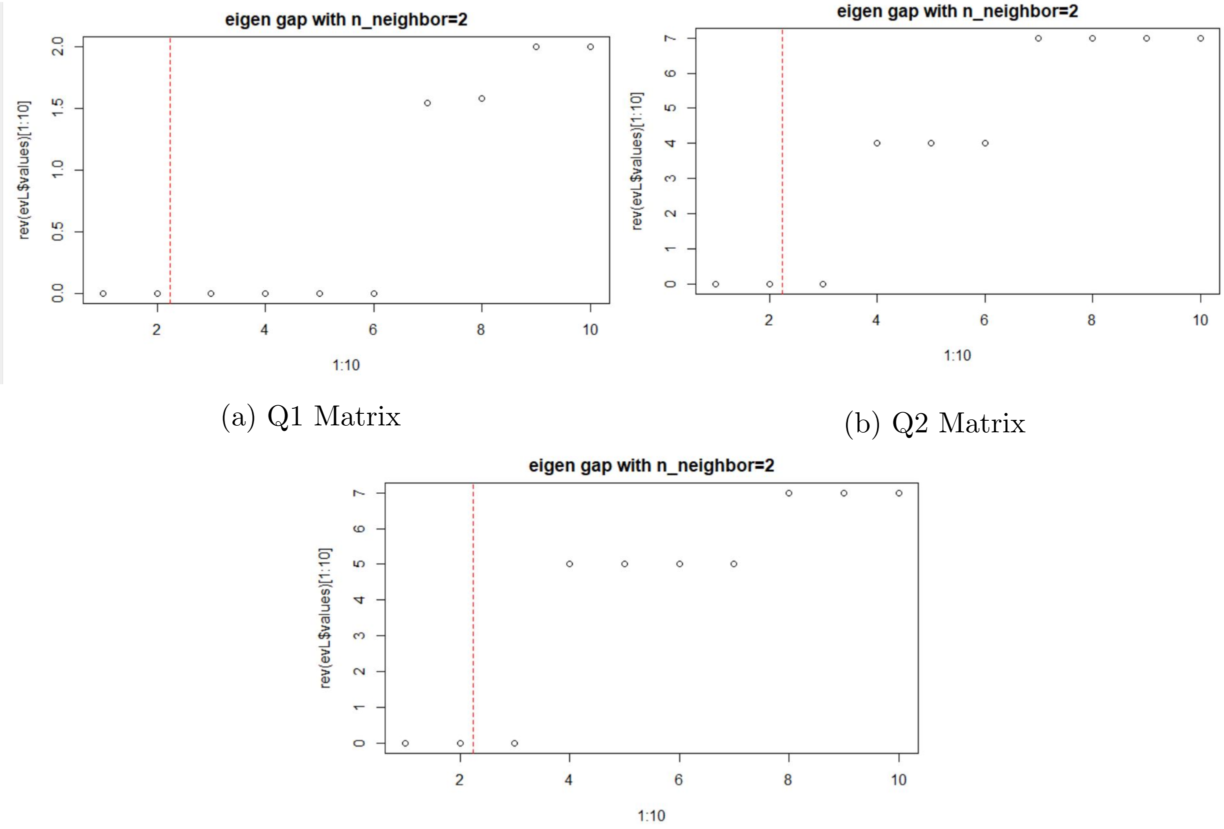
Method 3) Add an irrelevant skill to the question. Keeping the original skill unchanged, I add an extra skill that the question does not reflect. Using the previous question as an example, now it may reflect skills D and G, or skills D and N.

The results of DINA model and Spectral Clustering under three methods are shown below:

First, the *question average group similarity* of Spectral Clustering on three Q-matrices:

	Q1	Q2	Q3
Spectral Clustering	0.9180	0.9779	0.9815

I also include the *eigen graph* plot of Spectral Clustering:



(a) Q1 Matrix

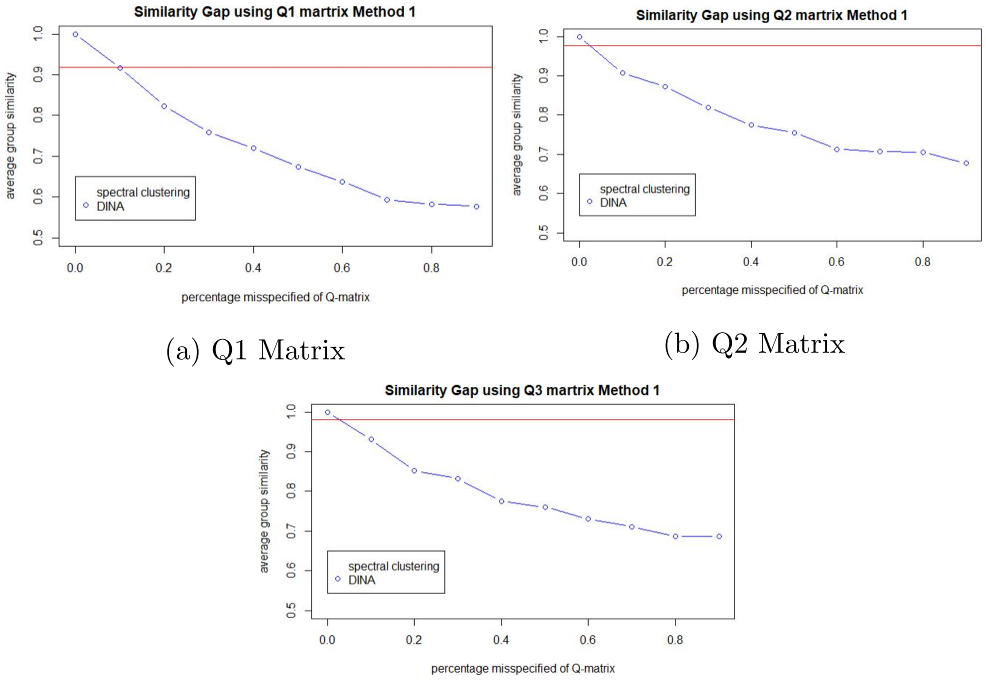
(b) Q2 Matrix

(c) Q3 Matrix

Figure 4.1: eigen graph on Different Q-Matrix

The Spectral Clustering demonstrates high quality in clustering, having *question average group similarity* above 90% for all three Q-matrices. Under Q2 and Q3 cases, the *eigen graph* also suggest to cluster questions into 3 clusters, which matches the number of skills defined for the Q-matrices. For Q1 matrix, the algorithm suggests to have 6 clusters instead of 9, this might be due to the imbalanced data in each cluster, which causes 6 clusters to be dominant with respect of their population. However, the similarity measure is above 0.9, which is still acceptable, so I would not worry about it too much.

Then I work on DINA model with misspecified Q-matrix. Plot the *question average group similarity* of DINA versus the portion of Q-matrix misspecification, and I add the Spectral Clustering result as a red horizontal line on the plot, The plot easily visualizes the effect of Q-matrix misspecifcation on the DINA model performance and the comparison between DINA model and Spectral Clustering.

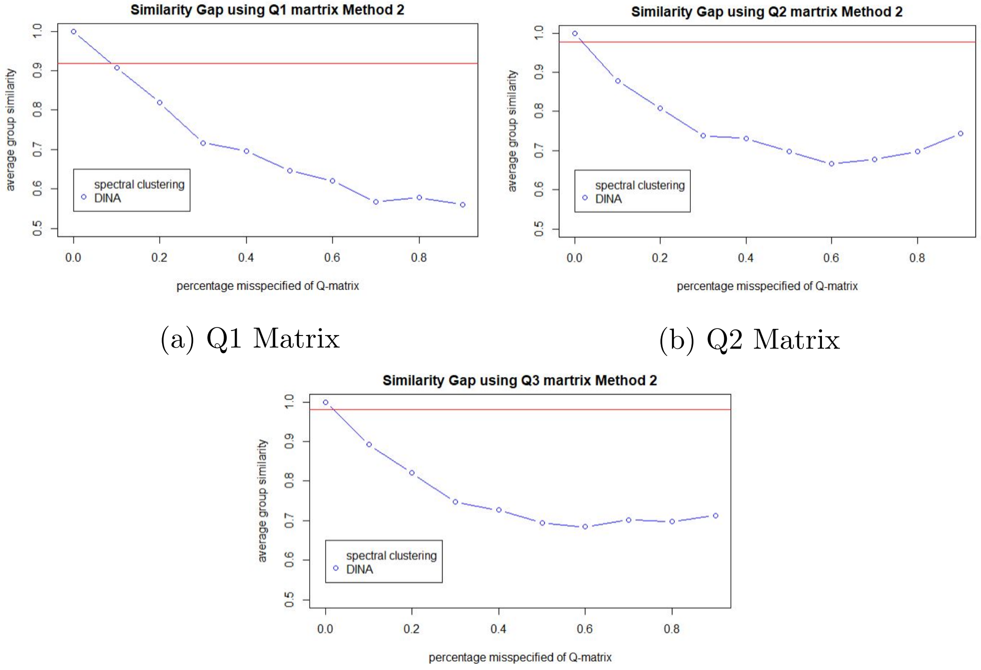


(a) Q1 Matrix

(b) Q2 Matrix

(c) Q3 Matrix

Figure 4.2: Q-matrices Similarity versus portion of Misspecification Method 1

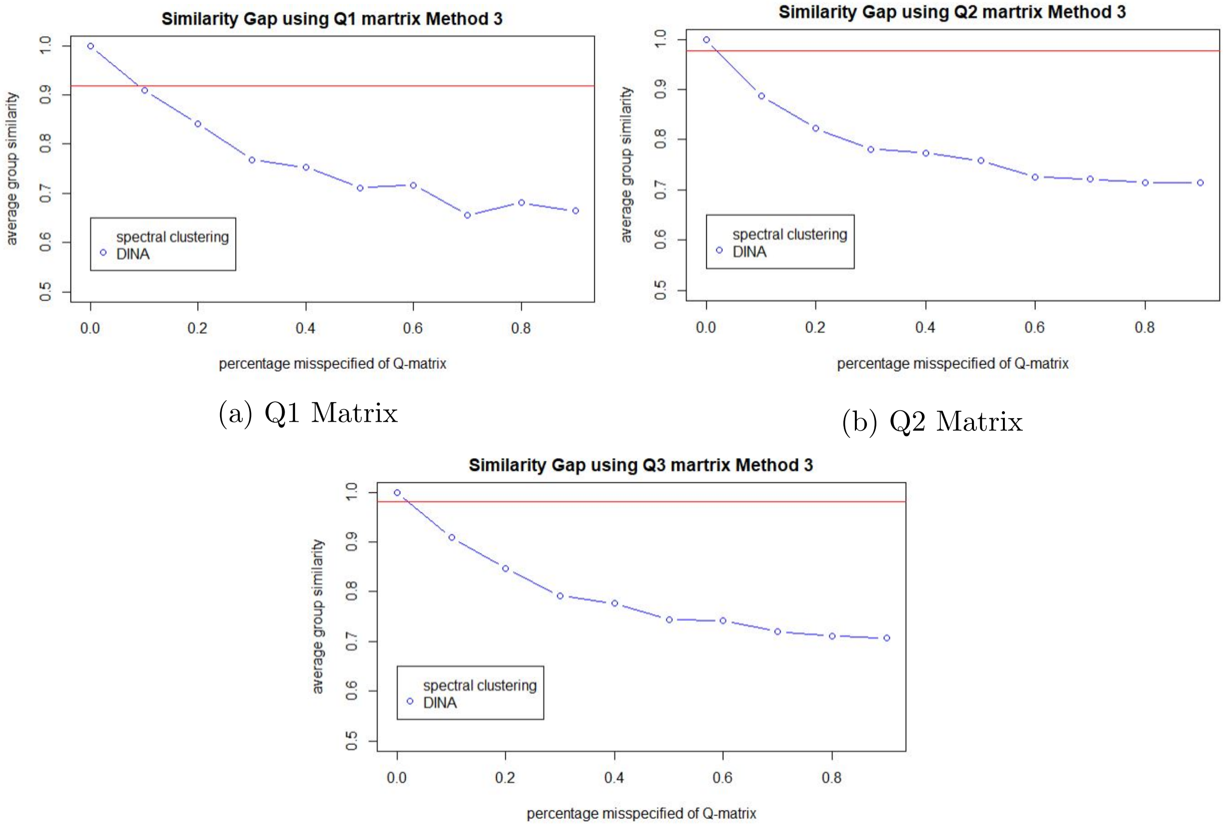


(a) Q1 Matrix

(b) Q2 Matrix

(c) Q3 Matrix

Figure 4.3: Q-matrices Similarity versus portion of Misspecification Method 2



(a) Q1 Matrix

(b) Q2 Matrix

(c) Q3 Matrix

Figure 4.4: Q-matrices Similarity versus portion of Misspecification Method 3

I also compute the *Simulated Gaps* for each Q-matrix:

	10%	20%	30%	40%	50%	60%	70%	80%	90%
Method 1	0.055%	10.29%	17.35%	21.58%	26.57%	30.51%	35.31%	36.54%	37.09%
Method 2	1.18%	10.77%	21.88%	24.13%	29.55%	32.46%	38.11%	36.99%	38.93%
Method 3	0.95%	8.30%	16.28%	17.97%	22.52%	21.91%	28.56%	25.81%	27.65%

Table 4.1: *Simulated Gap* on Q1-Matrix

	10%	20%	30%	40%	50%	60%	70%	80%	90%
Method 1	7.15%	10.73%	16.18%	20.80%	22.76%	27.07%	27.72%	27.90%	30.69%
Method 2	10.16%	17.35%	24.52%	25.21%	28.67%	31.89%	30.67%	28.63%	23.89%
Method 3	9.33%	15.86%	20.13%	20.95%	22.46%	25.83%	26.21%	26.91%	27.01%

Table 4.2: *Simulated Gap* on Q2-Matrix

	10%	20%	30%	40%	50%	60%	70%	80%	90%
Method 1	5.12%	13.11%	15.27%	20.96%	22.50%	25.53%	27.53%	30.00%	30.05%
Method 2	9.12%	16.42%	23.83%	25.97%	29.23%	30.29%	28.49%	28.89%	27.35%
Method 3	7.41%	13.72%	19.36%	20.79%	24.27%	24.44%	26.60%	27.50%	27.96%

Table 4.3: *Simulated Gap* on Q3-Matrix

From the plots, I see that using different methods of misspecification, the estimated amount of misspecification in the Q-matrix will be slightly different. Since Method 1 and Method 3 will still keep part of the original relationship, the misspecification is kind of "lenient", which means that the similarity curve decreases relatively gently with the increase of misspecifications. As a result, under same amount of misspecifications, model performance using Method 1 and Method 3 will be better than performance under Method 2. Meanwhile, in the table, Method 2 tends to give largest gap compare to other methods under small amount of misspecifications, despite which Q-matrix it uses. However, when the portion of misspecification becomes large, such as 80% or 90%, three methods are very close in performance since now most of the entries in the Q-matrix are wrong. For Method 2, the misspecified questions represent completely different skills, so that the similarity curve will experience sharp decrease even when a small portion of the questions are misspecified, resulting large decrease in quality even with a small portion of misspecifications.

When I compare three Q-matrix, I observe that using Q1-matrix will have the best performance (smallest gap) under small amount of misspecifications, such as 10%, among all three Q-matrices. However, it also has the worst performance (largest gap) when the amount of misspecifications becomes relatively large, such as 80%. This is partially due the number of skills in the matrix. When the amount of misspecifications is small, for Q-matrix with more number of skills defined, it will cluster data into more groups, and the misspecifications are further spread out among those groups compared to Q-matrix with fewer number of skills defined. As a result, the amount of misspecifications in each group will be smaller so the gap will be closer too. On the other hand, when the amount of misspecifications are very large, the clustering of data is mostly due to randomness. For Q-matrix will fewer skills, it has a higher chance to "guess" the data correctly since there are fewer options to guess compared to Q-matrix with more skills. Thus, using Q-matrix with smaller amount of skills will surpass the Q-matrix with larger amount of skills in performance when there is high portion of misspecifications.

4.2 Real Data Study

From the subsection 4.1, I studied the performance of the simulated data under different amounts of misspecifications in Q-matrix and obtained their *Simulated Gap*. In this section, I work on the real data in *CDM* package and try to use the results I got so far to empirically determine the amount of misspecifications in the Q-matrices it uses.

There are two primary goals for this section:
1). Confirm that the Q-matrices used in real data have misspecification.
2). Obtain the *Real Gap* of real data and determine the misspecifications.

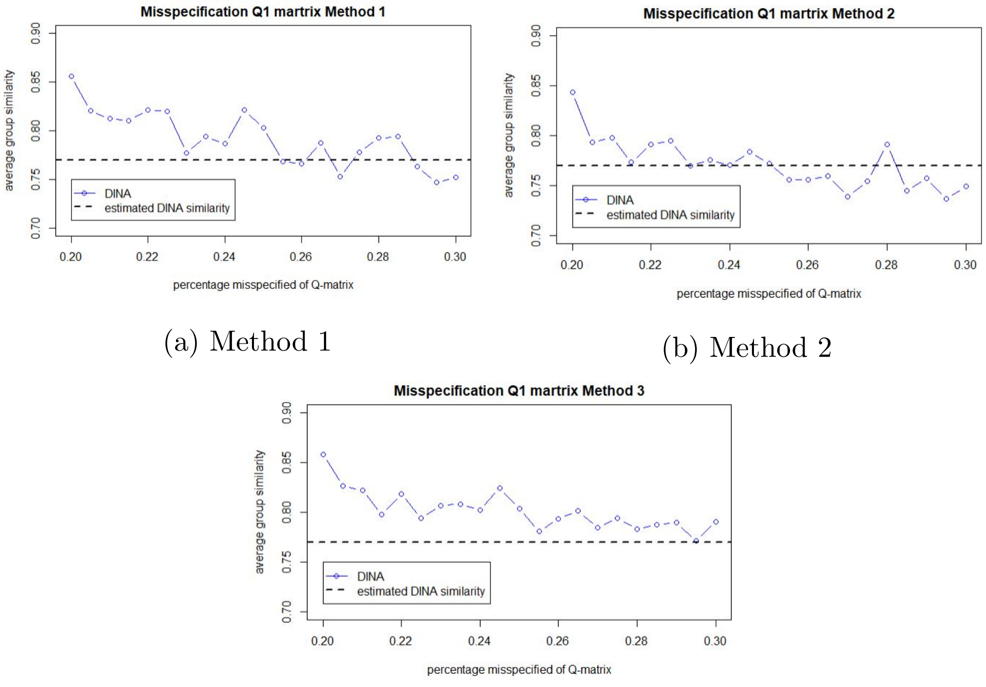
The first step is to determine whether the Q-matrices in the *data.timss11.G4.AUT* have misspecification. Note that according to the questions each student answered, they can be divided into 12 question groups with each group having around 670 students. The *question average group similarity* will be the average similarity measure of those 12 groups. The *real gap* is also calculated by the difference rate between the DINA model similarity and the Spectral Clustering similarity. The results are shown in the below table:

	Q1 matrix	Q2 matrix	Q3 matrix
DINA	0.5641	0.5641	0.5498
Spectral Clustering (3 clusters)	0.6351		
Spectral Clustering (9 clusters)	0.6728		
Real Gap	16.15%	11.18%	13.43%

From the table, no matter what Q-matrix I used, the Spectral Clustering always outperforms the DINA model. If the Q-matrix is 100% correct, then I am supposed to see no difference in two models' performance. The result clearly shows that clustering questions according to Q-matrix does not reflect the true relationship between the questions and the skills.

Now I see that the Q-matrix does not reflect the truth, I am able to conclude that there are some misspecifications exist in all the Q-matrices used in the real data, which jeopardizes the reliability of the clustering of the DINA model. Moreover, I get the *Real Gap* of three Q-matrices.

The next step is to use the *Real Gap* and *Simulated Gap* to determine the amount of misspecifications. For every Q-matrix, I multiple its Spectral Clustering *question average group similarity* from the simulated data with its *Real Gap*, and what I get is the estimated DINA model similarity on simulated data. Then I use *Simulated Gap* to multiple with Spectral Clustering results and find where they cross each other. The amount of misspecifications at the cross is the empirical misspecifications of the real data. To visualize this, I take advantage of Figure 4.2 to Figure 4.4, and narrow down to where the cross happens. The estimated DINA similarity using the *Real Gap* is plotted as a black horizontal line on the plot. The x value of the intersect between this line and the DINA model similarity is the estimated portion of misspecification of the Q-matrix.

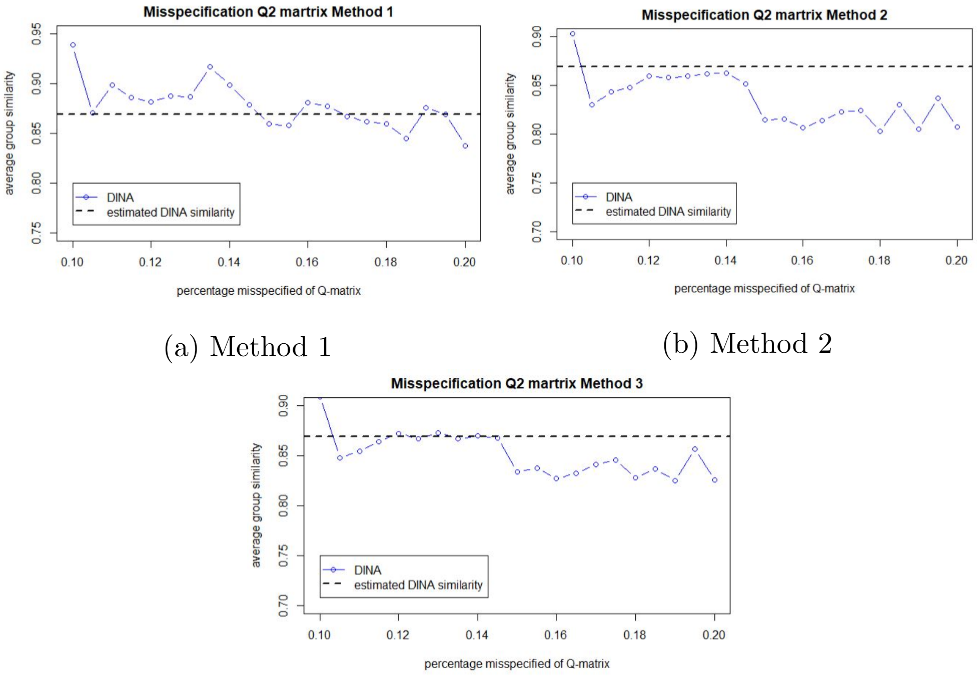


(a) Method 1

(b) Method 2

(c) Method 3

Figure 4.5: Q1-Matrix Misspecification under Different Methods

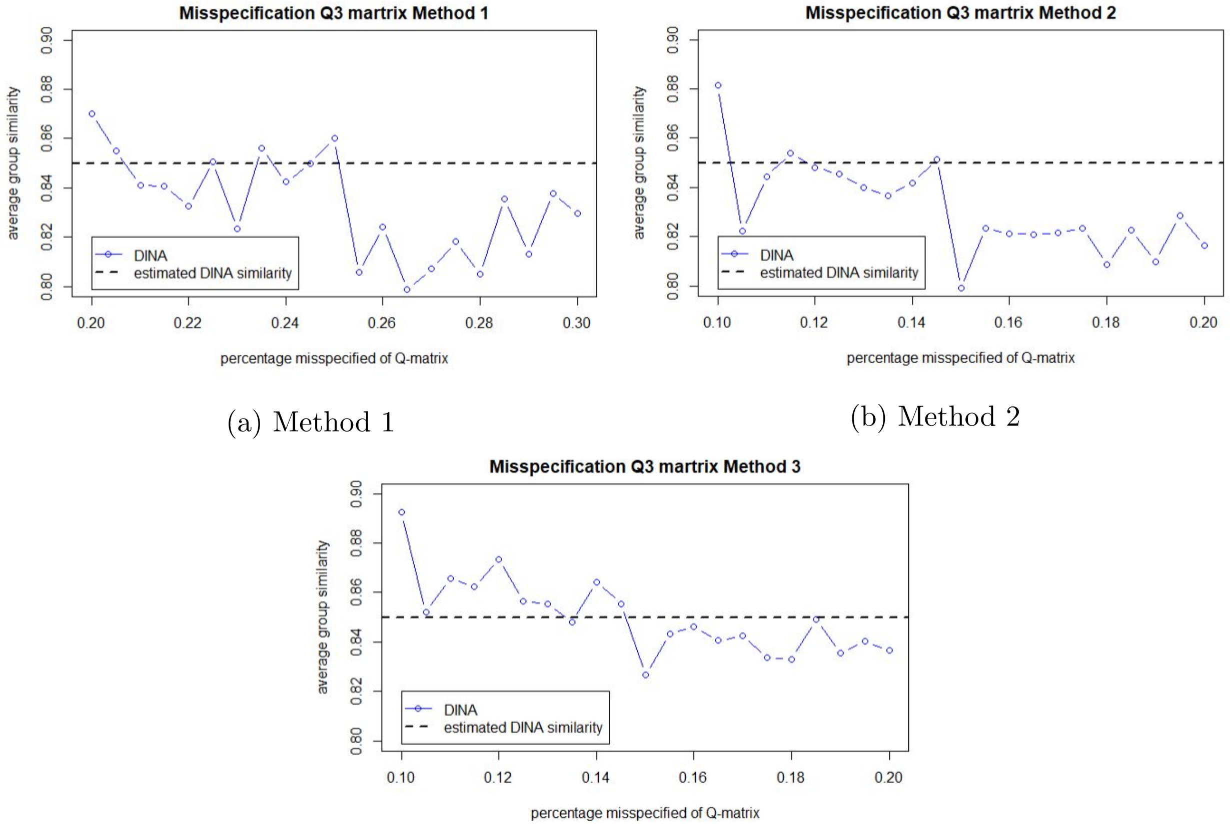


(a) Method 1

(b) Method 2

(c) Method 3

Figure 4.6: Q2-Matrix Misspecification under Different Methods



(a) Method 1

(b) Method 2

(c) Method 3

Figure 4.7: Q3-Matrix Misspecification under Different Methods

In the plots, I set the increment to be 0.5% each time for the portion of misspecifications, and see similarity fluctuates, which makes it hard to determine the exact location of the intersection. However, the plots still offer a lot of information and narrow down the potential range of the misspecification.

For Q1 matrix, all three methods give estimation between 20% to 30% of the whole questions. From the plots, clearly the cross happens in the second half, which is from 25% to 30%. The Method 2 seems to provide smaller estimation of misspecifications (around 25%)compared to other methods (near 30%).

For Q2 matrix, all three methods give the estimations within 10% to 20%. And Method 1 clearly estimates larger misspecifications (about 15%) compared to other two methods(about 10%).

For Q3 matrix, Method 1 estimates the number to fall in the range of 20% to 30% while Method 2 and 3 estimate between 10% to 20%. The misspecification in Method 1 is located in the first half of the interval, around 20%. And Method 2 also has the cross happened at the first half, roughly 12%. Method 3 has estimation in the second half, about 15%.

Studying the three methods respectively, the result echos with what I got from section 4.1. Despite the matrix, Method 2 always needs the lowest amount of misspecifications in order to reach the same error rate compared to other methods.

4.3 Model Stability and Tolerance

Given an assumption of the misspecification method, now I am able to empirically detect the percentage of misspecification in the Q-matrix. For three Q-matrix, none of them has misspecification exceed 30% of the whole data under any methods. However, I still don't know how much effect will this amount of misspecifications have on the DINA model, or when using these misspecified Q-matrices, whether the DINA model's predictions of students individual skill profiles are still reliable. To solve this problem, I need to apply DINA model on misspecified Q-matrices and study the model performance. To start with, I generate misspecified Q-matrix and run DINA using the matrix. Again, I set misspecification in Q-matrix from 10% to 90%, compare the result with Spectral Clustering, and study the difference. But before I work on the model, I need to define a new similarity measure for DINA model, since now it is not only related to the Q-matrix, but also needs to reflect the DINA model. The new *DINA average group similarity* is defined as following:

DINA Average Group Similarity

For each Q-matrix, there are k skills defined by educational experts. For the student skill profile, there are 2^k different skill groups according to the combinations of the skills. In other words, the DINA model clusters students into 2^k groups where students in the same group are believed to possess same skill profile. The idea of this similarity measure is similar to the idea of *Question Average Group Similarity*: For the DINA model, since it will provide individual skill profile, I will cluster students by their profile, and compute the similarity of the answer pattern for students with same skill profile, and average the results over 2^k groups. For Spectral Clustering, I force the method to cluster the students into 2^k groups by their responses, and obtain the average similarity measure of their responses. Again, Spectral Clustering clusters data only depends on the responses data, but DINA model is based on Q-matrix. If DINA model is working well, then the students in the same skill group will be likely to have similar answer patterns to same questions, and their similarity measure should be as good as Spectral Clustering.

In the real data, it can be divided into 12 student groups by common questions answered. Each group has around 340 students who answered around 25 questions. To simulate it, I randomly pick 25 questions from the Q-matrix, and again use *sim.din* to generate a 400×25 responses matrix X_s. To calculate the *DINA average group similarity* of the Spectral Clustering, I cluster 400 students into 2^k skill classes where k is the number of skills defined. For students in each skill class, I obtain the similarity of their responses to the questions. Averaging the 2^k classes' results and repeating the whole procedure 20 times to make sure that every question has been included, I get the *DINA average group similarity* of Spectral Clustering. For DINA model, I apply the model on the simulated responses using misspecified Q-matrix. Since the model predicts individual skill profile, I simply group students by their skill classes. Then, I calculate the average similarity and repeat the whole procedure for 20 times. The result is the *DINA average group similarity* of the model.

Again, I tried the same three methods in section 4.1 to misspecify the Q-matrix. Note that in this section, I did not choose Q1 matrix since it contains 9 skills, and thus $2^9 = 512$ skill classes in total. This is too computational expensive for DINA and Spectral Clustering.

Results:

To start with, I apply Spectral Clustering and DINA on the real data, and compute the *DINA average group similarity* and the *Real Gap*:

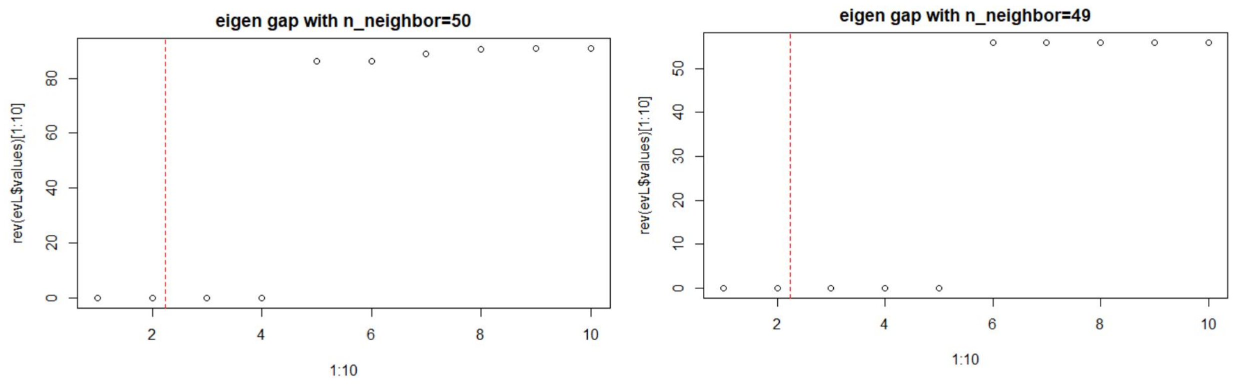
	Q2 matrix	Q3 Matrix
DINA	0.6739	0.6715
Spectral Clustering (3 clusters)	0.7223	
Real Gap	6.70%	7.03%

From the table, using Q2 matrix or Q3 matrix does not have a big difference on the similarity measure, and it is down only by 5% to the Spectral Clustering. The result also supports the assumption that the Q-matrices are not fully accurate. However, compared to the *Real Gap* obtained in section 4.2 without using the DINA model, the *Real Gap* of DINA model is smaller. For Q2 matrix, it shrinks from 11.18% to 6.7%. For Q3 matrix, the gap decreases from 13.43% to 7.03%.

Next, I simulate data and the *DINA average group similarity* of Spectral Clustering I got is:

	Q2	Q3
Spectral Clustering	0.9608	0.9688

And the *eigen graph* of Spectral Clustering to determine the number of neighbors I used to construct the similarity graph.



(a) Q2 Matrix

(b) Q3 Matrix

Figure 4.8: eigen graph on Different Q-Matrix

Similarly, the Spectral Clustering demonstrates high quality in clustering, with scores above 90% for two Q-matrices. The *eigen graph* suggests the gap being different from the theoretical $8(2^3)$ skill classes, but since the similarity is good enough, I would not worry about it.

Then I work on DINA model with misspecified Q-matrix, with Spectral Clustering similarity on the plot as a red horizontal line. Moreover, I add the empirical Q-matrix misspecification range I got from previous section as a vertical line, and get the *DINA average group similarity* under the empirical Q-matrix misspecification:

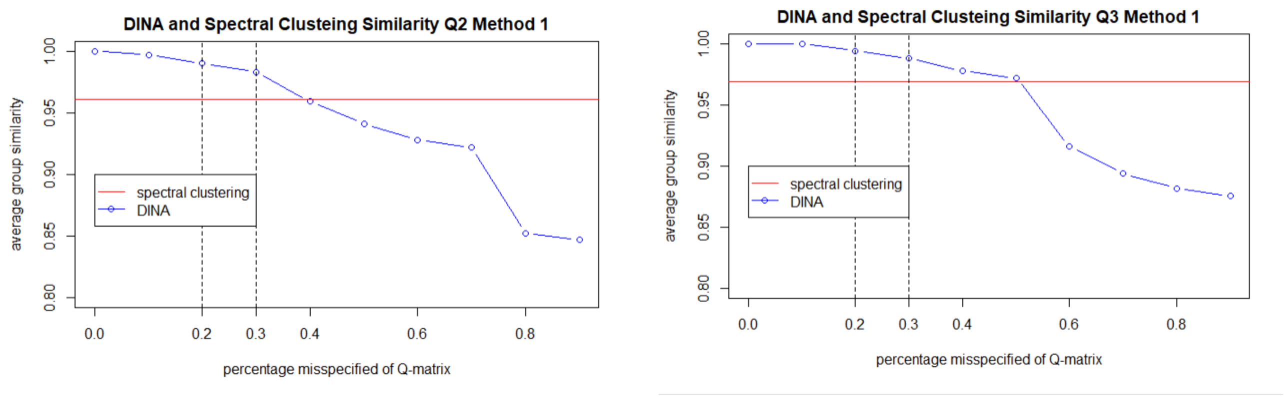


Figure 4.9: Method 1

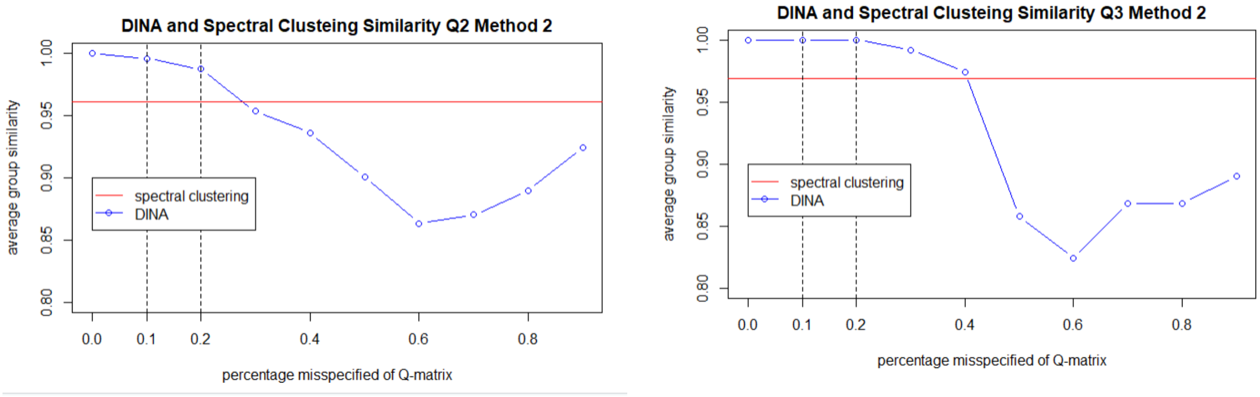


Figure 4.10: Method 2

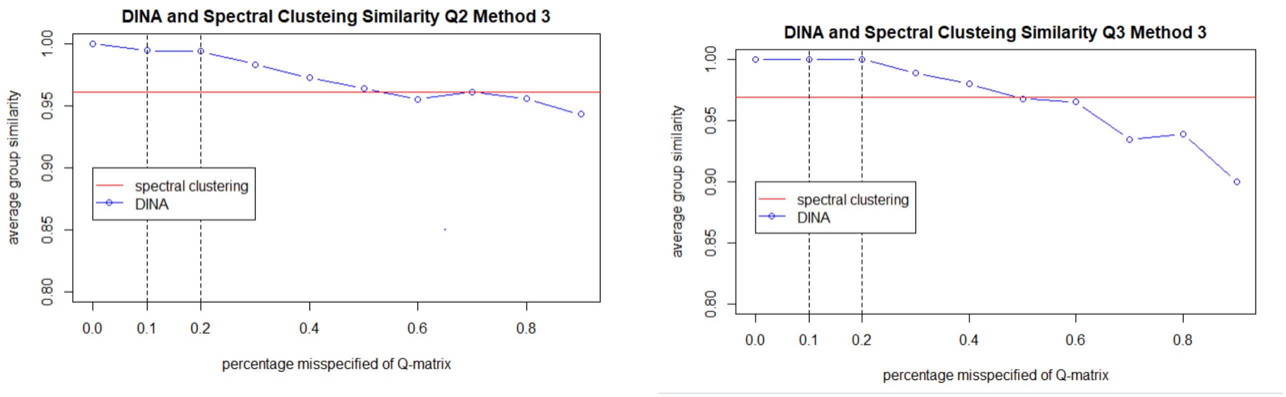


Figure 4.11: Method 3

From the plots, under the range of empirical Q-matrix misspecification, which is below 30%, none of the plots has DINA similarity measure lower than Spectral Clustering similarity measure. Moreover, in most cases, only when a large amount of misspecification occurs, such as more than 50%, will DINA model start to be beaten by Spectral Clustering in performance. The result shows that the DINA model is not very sensitive to small amount of misspecifications in Q-matrix and is able to maintain its high performance. This supports the robustness of the DINA model, and also supports the reliability of the result under empirical Q-matrix misspecification. Meanwhile, it also explains the improvement between the two *Real Gap*s I got. When only using Q-matrix, the *Real Gap*s of Q2 and Q3 are around 12%. However, after using the DINA model, the *Real Gap*s are halved to 6%. This is because the misspecifications are small so that DINA model remains its high-quality in clustering.

4.4 Other Effects

Results from Section 4.3 demonstrate that DINA model is a robust model under small amount of Q-matrix misspecification. However, from real data, I see that there are other factors contributing to the DINA model clustering. In Section 4.3, on real data, DINA model never outperforms Spectral Clustering, but on simulated data, the DINA model achieves better quality even with small portion of Q-matrix misspecification. To address this, I would like to take other factors into account, such as the *slip/guess rate*.

Slip and Guess Rate:

Every question has its unique slip and guess rate, which is determined by the structure of the question itself. *Slip* happens when the student answers the question wrong even the student possesses the skill needed, while *Guess* happens when student who does not have the skill needed but answers the question correctly, simply due to a matter of luck. DINA model is able to calculate the *slip/guess rate*. For example, the real data has an average slip rate of 36.9% and guess rate of 32.4%. Together, they add up to around 70%, which means that 70% of the time, a student's response to the question does not reflect the student's true skill, but is due to some random error. The high *slip/guess* rate is another signal that the DINA model has some issues.

To begin with, I use the average *slip* and *guess* rate from real data to simulate data, and then apply Spectral Clustering and DINA with the same Q-matrices. To study the model and the Q-matrix, I obtain two types of *average group similarity* I defined in the previous sections: *Question Average Group Similarity* and *DINA Average Group Similarity.*

	Q1	Q2	Q3
DINA	0.5490	0.5483	0.5501
Spectral Clustering	0.5489	0.5468	0.5493

Table 4.4: Question Average Group Similarity

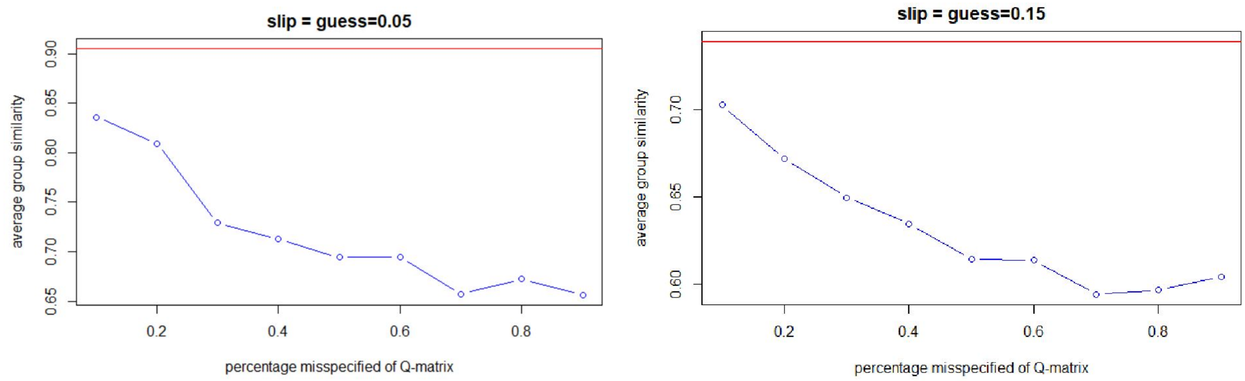
	Q2	Q3
DINA similarity	0.5473	0.5502
Spectral Clustering	0.5413	0.5453

Table 4.5: DINA Average Group Similarity

Since for the DINA model I did not change the Q-matrix on purpose, there is no misspecifications. Thus, the Spectral Clustering and DINA model are expected to give similar results.

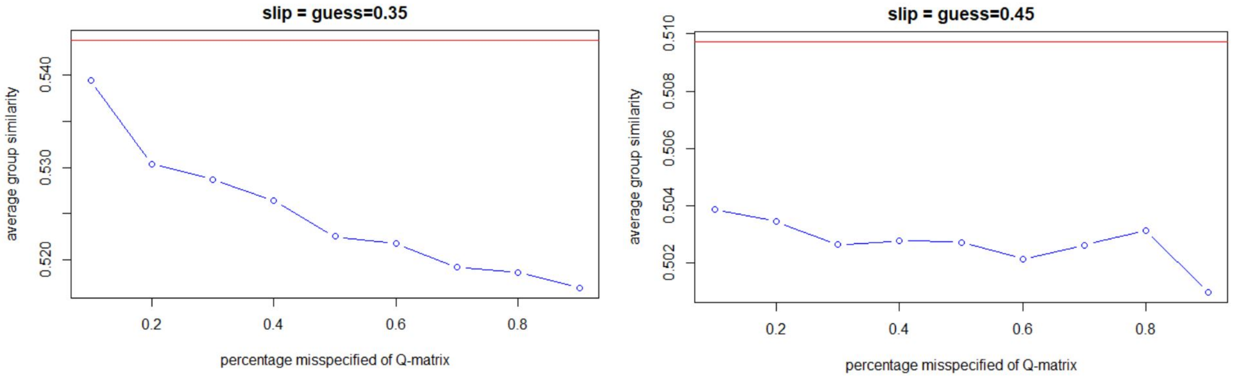
Now, I tried DINA with misspecified Q-matrix. For simplicity, here I only include results on data simulated by Q2 matrix and misspecification Method 1. Using other two Q-matrices or other two misspecification methods will not change the conclusions. When I simulate data, I change the pair of *slip and guess rate* from 10% to 45%, respectively. For each pair of rate, I control the number of misspecified entries from 10% to 90%. The last step is to apply Spectral Clustering and DINA on the simulated data and collect results. I include some representative plots to show the impact of *slip/guess rate* on the model.

For *Question Average Group Similarity*, which only depends on the Q-matrix but not the DINA model, as (*slip,guess*) pair changes, the plot of Spectral Clustering similarity and the Q-matrix similarity looks like this (Again, horizontal red line means the similarity of Spectral Clustering, the blue dot line is the similarity of DINA):



(a) Slip = Guess = 5%

(b) Slip = Guess = 15%



(a) Slip = Guess = 35%

(b) Slip = Guess = 45%

Note that I am using Q2-matrix and misspecification Method 2. The empirical misspecification I got for the Q2 matrix using misspecification method 2 is approximately 11%, so I generate a 11% misspecified Q-matrix and plot the similarity of two models with the *slip/guess* rate changing. I add the black dash vertical line indicating the average *slip, guess rate* of the real data on the plot.

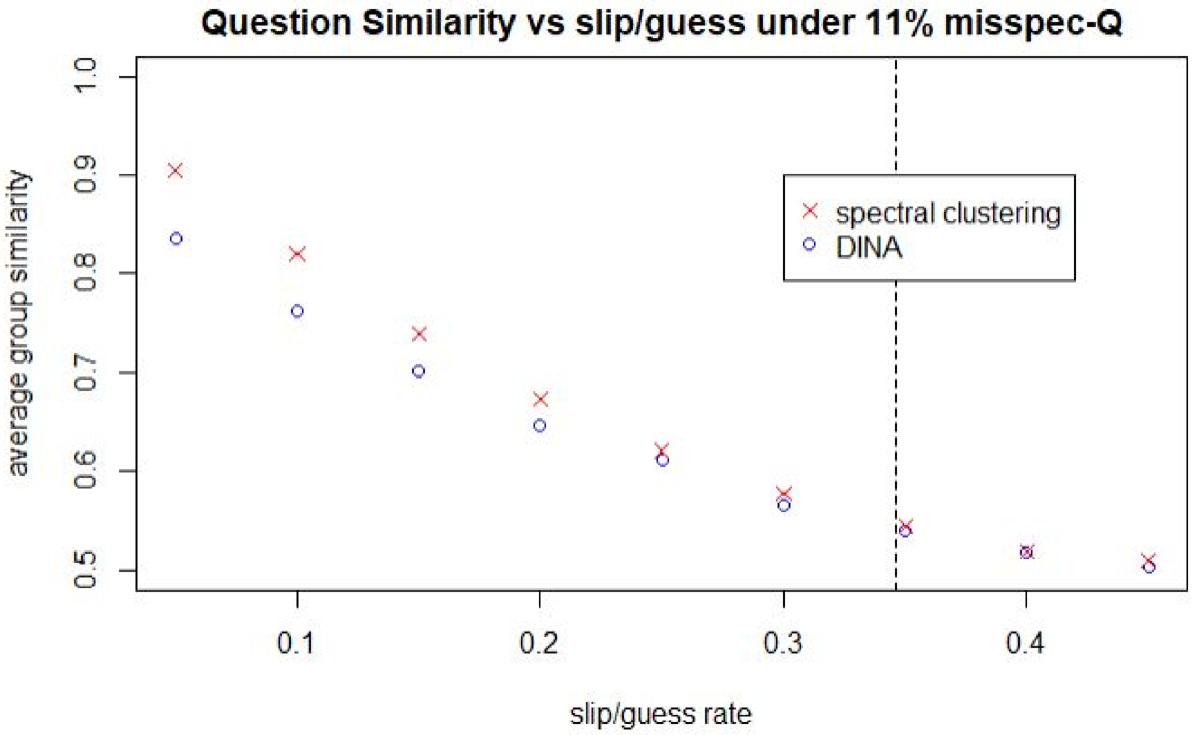
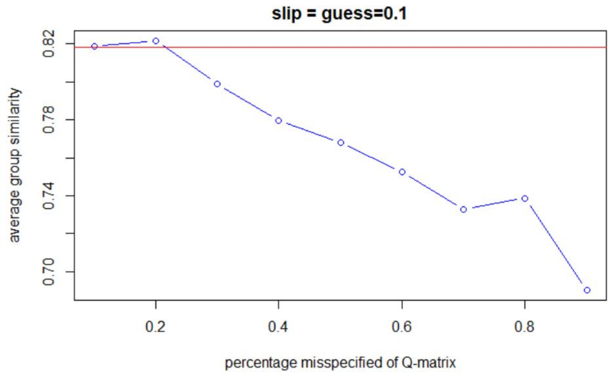


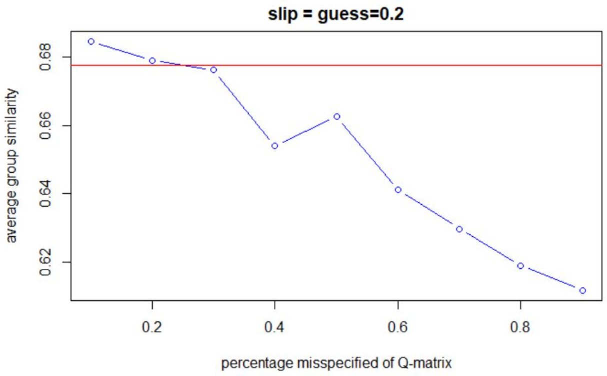
Figure 4.14: Question Similarity plot of slip/guess rate given 11% misspecification

With the increase of the *slip/guess* rate, the random error is increased in the data, and both Spectral Clustering and DINA model performance are becoming worse. Meanwhile, as the *slip/guess* rate goes up, the fluctuation of DINA model's similarity across different misspecification is further decreased, meaning the model is less sensitive to the change in Q-matrix. This is because with a high *slip/guess* rate, the clustering now is mostly due to randomness, but not Q-matrix. For example, when slip = guess = 45%, they add up to 90%, meaning only 10% of the responses will be controlled by Q-matrix. That is why the plot of slip = guess = 45% is almost a horizontal line lying near the bottom. It shows that when the *slip/guess* rate is very high, the result is unreliable. Meanwhile, the plot using empirical misspecifications and real data average *slip, guess* rate shows that the DINA similarity is very close to Spectral Clustering similarity, so that after I take account of the misspecification and *slip/guess* rate, the DINA model is able to achieve similar performance as Spectral Clustering. This adds weight to my empirical cut-off.

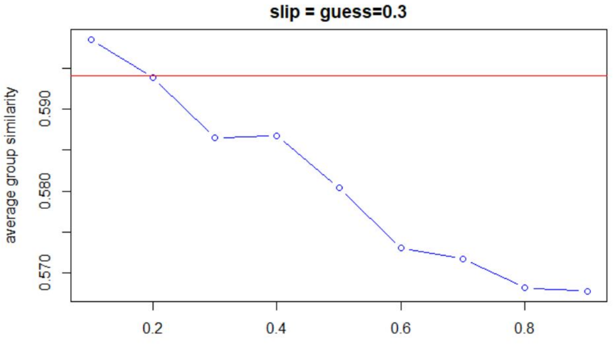
After that, I compute the *DINA Average Group Similarity.* Repeat the same steps, I got:



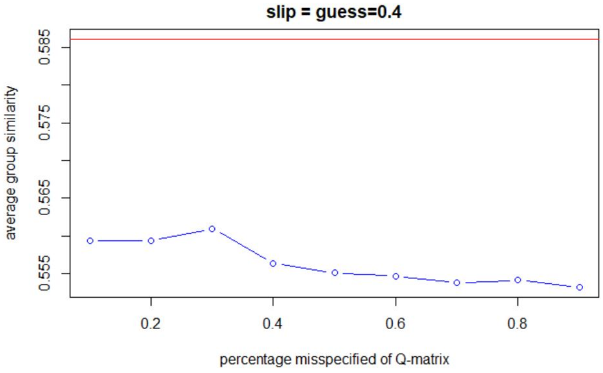
(a) Slip = Guess = 10%



(b) Slip = Guess = 20%



(a) Slip = Guess = 30%



(b) Slip = Guess = 40%

Using the empirical Q-matrix misspecification and real data average *slip/guess* rate, the plot is:

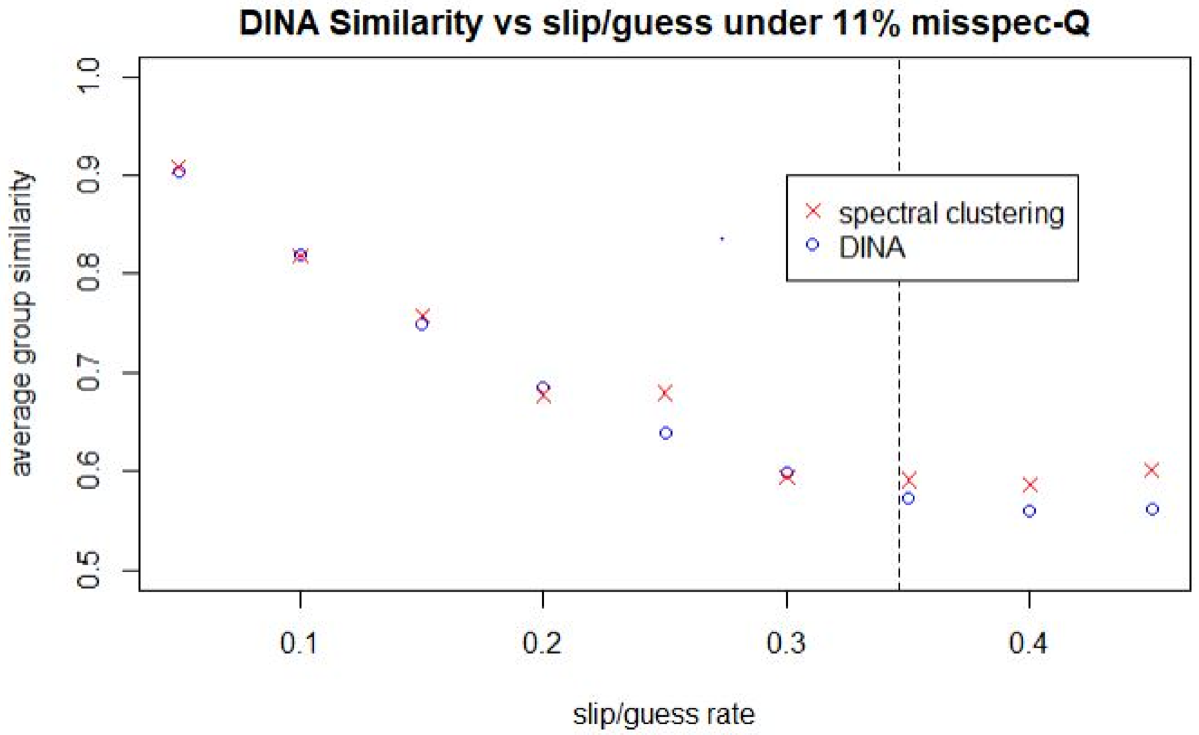


Figure 4.17: DINA Similarity Plot versus slip/guess rate with 11% Q-matrix misspecification

After introducing DINA model, the similarity measure is significantly improved compared to *Question similarity measure*. If the *slip/guess* rate is not extremely high, the model is able to achieve similar quality in clustering as Spectral Clustering. Only when the *slip/rate* is extremely high, such as they add up to 80%, will the DINA model performance be affected. Meanwhile, using the empirical Q-matrix misspecification and average *slip/guess* rate, the DINA model performs really close to Spectral Clustering. This shows that besides tolerance to Q-matrix misspecifications, DINA model also has tolerance to small amount of *slip/guess rate* and is able to perform well without being influenced when the rate is small.

Chapter 5

Conclusions and Future Outlook

Conclusions:

In this paper, I utilize the assumption-free property of Spectral Clustering and use it as a benchmark in the determination of the amount of misspecifications in the Q-matrices used by DINA model. In this paper, Spectral Clustering is able to maintain its high performance even when DINA model gives poor results, which proves its robustness.

On the other hand, DINA model, a model based on the assumption of Q-matrix predefined by educational experts, is outperformed by Spectral Clustering when both were applied on the real data. As a result, I determine that the assumption of DINA model is not fully accurate, and there exists misspecifications in all three Q-matrices used by R *CDM* package.

Moreover, I combine the results from simulation study and real data study, and propose terms *Average Group Similarity* and *Gap* in order to empirically determine the amount of misspecifications in the Q-matrices. I also come up with several misspecification rules with different leniency over the misspecifications to simulate the potential mistakes educational experts could make in reality. Under different misspecification method, I estimate the misspecifications in three Q-matrices. From the results, despite the misspecification rule I use, none of the three Q-matrices has misspecifications exceed 30% of the whole matrix. And in most cases, there are only around 10% of the misspecifications in the matrix, which is not a very high rate.

To further investigate the effect of misspecifications on the DINA model, I study the DINA performance when given misspecified Q-matrices, and I find that the model has certain tolerance to misspecifications and maintains its high performance under small amount of the misspecifications. The result shows the stability and robustness of DINA model.

At last, I study another factor: slip and guess rate. I see that the quality of the clustering will decrease with the increase of the slip/guess rate, and under high rate, the results are unreliable. In addition, I find that DINA model also has certain tolerance to the slip/guess rate as it is able to perform well under relatively small slip/guess rate.

To sum up, under the empirical misspecifications and average slip, guess rate of the real data, despite other assumptions, DINA model is proven to produce high-quality and reliable predictions. In other words, DINA model is a qualified model to analyze the specific data set *data.timss11.G4.AUT* in R *CDM* package, even though there are still misspecifications in the Q-matrices it uses and the randomness still exists.

Future Outlook:

Continuing on this paper, I find a lot of interesting topics to explore further.

To start with, I only come up with an empirical way to determine the misspecifications in a specific data set. Clearly, more explanations and explorations are needed to reveal the logic behind the scene and hopefully, to formulate this method. Also, the determination depends on my assumptions of misspecification method which does not receive any support from models or theories. Thus, new model needs to be made to estimate the way how Q-matrix is misspecified.

After that, for the effect of misspecifications on the DINA model, the "tolerance" to misspecifications needs to be quantified and the reason needs to be figured out. Same need for the "tolerance" to slip/guess rate.

There are other contributing factors other than slip/guess rate, such as the number of questions students were asked, and they have not been covered in this paper due to the interest of time. Furthermore, in the original data set in the R *CDM* package, there are many other types of data other than responses to test questions. For example, the data also includes students' answers to their family, home, teacher and school background. Those factors are also believed to be influential on the student's skill profile and more study needs to be done.

In addition, this paper only chooses a single data set and a few Q-matrices to work on. There are thousands of schools taking the test every year and educational experts have defined many other Q-matrices with different skill sets. It is worth studying multiple responses matrices and Q-matrices, and try to see whether the results in this paper still hold on other data.

Lastly, the DINA model itself can also be replaced by its relatives. CDMs are collections of similar but unique methods with different assumptions. We need to apply different CDMs based on our assumption of the data. For example, if we believe that students only need to possess at least one of the assigned skills for successfully mastering the respective item instead of possessing all the skills, then we should use DINO(non-compensatory deterministic input noisy-OR-gate) and the whole result will be completely different.

Bibliography

[1] Mullis IVS, Martin MO, Ruddock GJ, O'Sullivan CY, Arora A, Erberer E (2005). *TIMSS 2007 Assessment Frameworks.* Boston College, Chestnut Hill.

[2] Mullis IVS, Martin MO, Kennedy AM, Foy P (2007). *PIRLS 2006 International Report: IEA's Progress in International Reading Literacy Study in Primary School in 40 Countries.* Boston College, Chestnut Hill.

[3] OECD (2010). *PISA 2009 Results: What Students Know and Can Do.* Organisation for Economic Co-operation and Development (OECD), Paris, France. doi:10.1787/9789264091450-en.

[4] Rupp, A. A., Templin, J. L., & Henson, R. A. (2010). *Diagnostic assessment: Theory, methods, and applications.* New York,NY: Guilford Press.

[5] Birenbaum, M., Tatsuoka, C., & Yamada, T. (2004). Diagnostic assessment in TIMSS-R: Between-countries and within-country comparisons of eighth graders' mathematics performance. *Studies in Educational Evaluation, 30*, 151-173.

[6] Corter, J. E. (1995). Using clustering methods to explore the structure of diagnostic tests. In P. D. Nichols, S. F. Chipman, & R. L. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 305-326). Hillsdale, NJ: Erlbaum.

[7] Rupp, A. A., & Templin, J. (2008). The effects of Q-matrix misspecification on parameter estimates and classification accuracy in the DINA model. *Educational and Psychological Measurement, 68(1)*, 78-96. doi: 10.1177/0013164407301545

[8] George, A. C., Robitzsch, A., Kiefer, T., Groß, J., & Ünlü, A. (2016). The R Package CDM for Cognitive Diagnosis Models. *Journal of Statistical Software, 74(2).* doi:10.18637/jss.v074.i02

[9] Paek I, Cai L (2014). "A Comparison of Item Parameter Standard Error Estimation Procedures for Unidimensional and Multidimensional Item Response Theory Modeling." *Educational and Psychological Measurement, 74*, 58–76. doi:10.1177/0013164413500277.

[10] Luxburg, U. V. (2007). A tutorial on spectral clustering. *Statistics and Computing, 17*(4), 395-416. doi:10.1007/s11222-007-9033-z

[11] Steinbach M., Ertöz L., Kumar V. (2004) The Challenges of Clustering High Dimensional Data. In: Wille L.T. (eds) *New Directions in Statistical Physics.* Springer, Berlin, Heidelberg

[12] Bolla, M. (1991). *Relations between spectral and classification properties of multigraphs* (Technical Report No. DIMACS-91-27). Center for Discrete Mathematics and Theoretical Computer Science.

[13] Mohar, B. (1997). Some applications of Laplace eigenvalues of graphs. In G. Hahn and G. Sabidussi (Eds.), *Graph Symmetry: Algebraic Methods and Applications* (Vol. NATO ASI Ser. C 497, pp. 225 – 275). Kluwer.

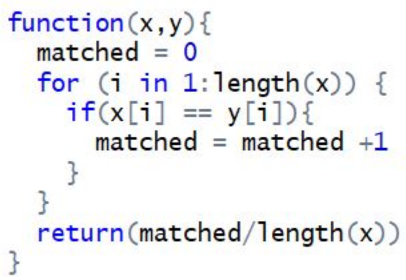
[14] Chung, F. (1997). *Spectral graph theory* (Vol. 92 of the CBMS Regional Conference Series in Mathematics). Conference Board of the Mathematical Sciences, Washington.

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

[16] TIMSS. (2011). *TIMSS 2011 User Guide for the International Database.* Retrieved from https://timssandpirls.bc.edu/timss2011/international-database.html

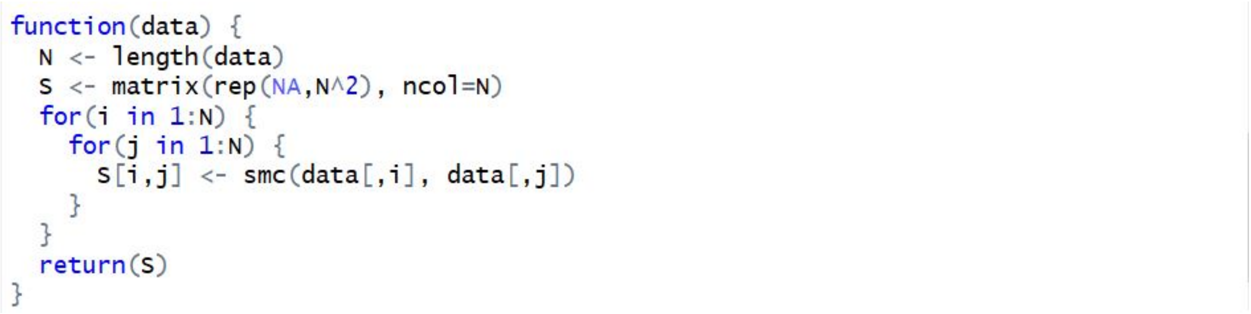
.1 Appendix: R Code for Spectral Clustering

Below are the R codes I wrote for Spectral Clustering implemented by Unnormalized Graph Laplacians using Simple Matching Coefficient as similarity measure.
1. Similarity Function (Simple Matching Coefficient):



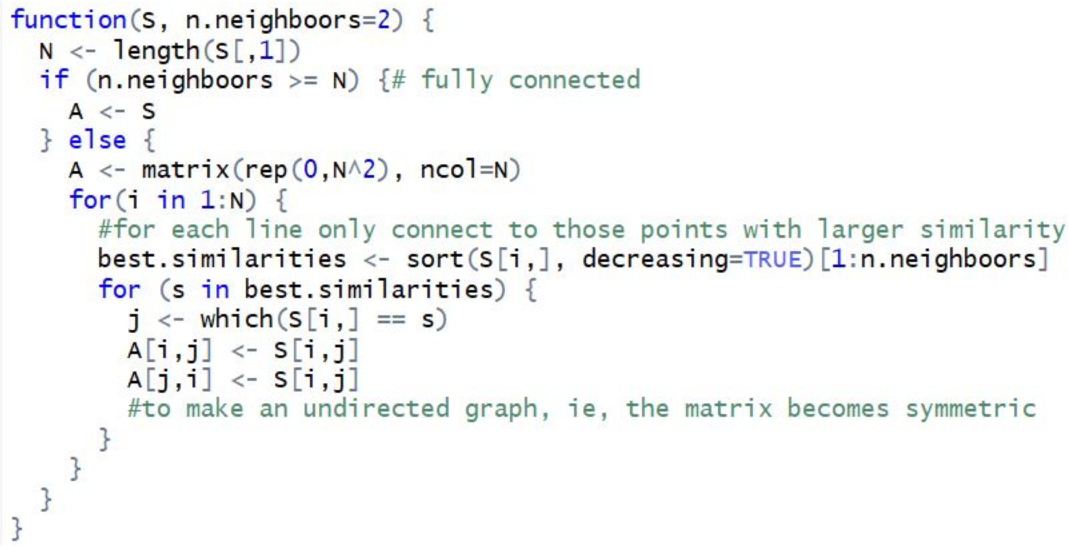
```
function(x,y){
  matched = 0
  for (i in 1:length(x)) {
    if(x[i] == y[i]){
      matched = matched +1
    }
  }
  return(matched/length(x))
}
```

2. make similarity matrix:
Calculate the piece-wise similarity measure of the data points.



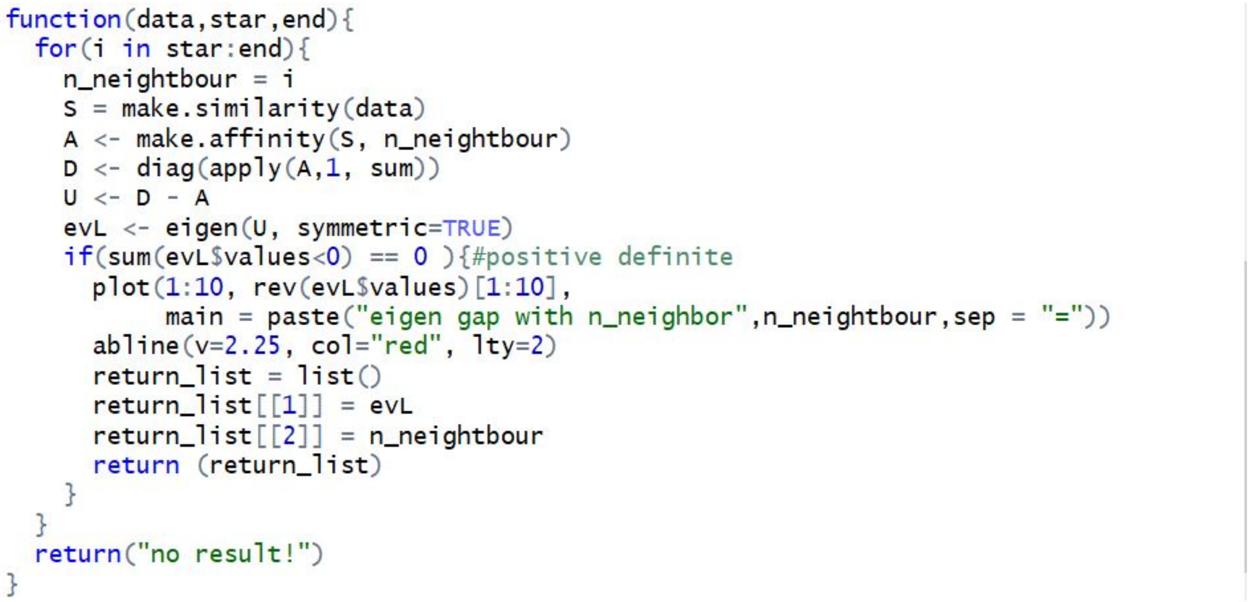
```
function(data) {
  N <- length(data)
  S <- matrix(rep(NA,N^2), ncol=N)
  for(i in 1:N) {
    for(j in 1:N) {
      S[i,j] <- smc(data[,i], data[,j])
    }
  }
  return(S)
}
```

3. make similarity graph:
Use *k-nearest neighbour* method. For each data point, only keeps its k nearest points in the matrix in terms of the similarity measure.



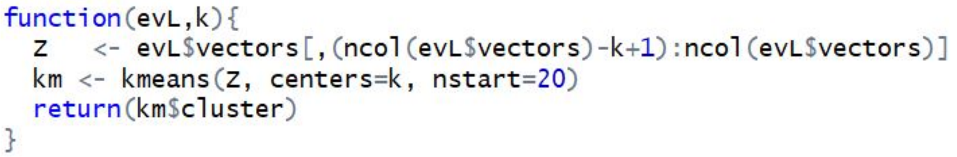
```
function(S, n.neighboors=2) {
  N <- length(S[,1])
  if (n.neighboors >= N) {# fully connected
    A <- S
  } else {
    A <- matrix(rep(0,N^2), ncol=N)
    for(i in 1:N) {
      #for each line only connect to those points with larger similarity
      best.similarities <- sort(S[i,], decreasing=TRUE)[1:n.neighboors]
      for (s in best.similarities) {
        j <- which(S[i,] == s)
        A[i,j] <- S[i,j]
        A[j,i] <- S[i,j]
        #to make an undirected graph, ie, the matrix becomes symmetric
      }
    }
  }
}
```

4. Calculate Laplacian matrix and plot *eigen graph*:
The function accepts a range of integers as the number of neighbors to construct the similarity graph, then calculates the unnormalized graph laplacians, and plots the *eigen graph* if the eigen matrix is positive definite. It will return the first number of neighbor that makes the matrix positive definite.



```
function(data,star,end){
  for(i in star:end){
    n_neightbour = i
    S = make.similarity(data)
    A <- make.affinity(S, n_neightbour)
    D <- diag(apply(A,1, sum))
    U <- D - A
    evL <- eigen(U, symmetric=TRUE)
    if(sum(evL$values<0) == 0 ){#positive definite
      plot(1:10, rev(evL$values)[1:10],
           main = paste("eigen gap with n_neighbor",n_neightbour,sep = "="))
      abline(v=2.25, col="red", lty=2)
      return_list = list()
      return_list[[1]] = evL
      return_list[[2]] = n_neightbour
      return (return_list)
    }
  }
  return("no result!")
}
```

5. K means to cluster data:
Run K-means on the eigen vectors of Laplacian matrix to cluster data and return clusters.



```
function(evL,k){
  Z   <- evL$vectors[,(ncol(evL$vectors)-k+1):ncol(evL$vectors)]
  km <- kmeans(Z, centers=k, nstart=20)
  return(km$cluster)
}
```